МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Збірник праць молодих науковців ЦНТУ

Випуск 13

Кропивницький – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Збірник праць молодих науковців ЦНТУ

Випуск 13

Кропивницький – 2023

Збірник праць молодих науковців ЦНТУ. – Вип. 13. – Кропивницький: ЦНТУ, 2023 – 447 с.

Збірник праць молодих науковців складається зі змісту, статей та тез здобувачів вищої освіти по матеріалам дипломних робіт.

УДК 504.75

Ю. Вєлкова, магістр гр. ЕО-21М
Центральноукраїнський національний технічний університет

ЕКОЛОГІЧНА ОЦІНКА ПРОЄКТУ РЕКОНСТРУКЦІЇ ПИЛОГАЗООЧИСНОЇ УСТАНОВКИ ПІДПРИЄМСТВА МЕТАЛУРГІЇ

У статті розглянуто екологічні особливості реконструкції пилогазоочисної установки підприємства металургії. Метою роботи є екологічна оцінка проєкту реконструкції пило-газоочисної установки Побузького феронікелевого комбінату. Об'єктом дослідження є проєкт реконструкції пило-газоочисної установки підприємства металургії. Предмет дослідження: шкідливий вплив виробництва феронікелю на атмосферу та здоров'я людей. Результат роботи: на основі дослідження параметрів апарату на ефективність пиловловлення було проведено відповідні розрахунки впливу підприємства на довкілля після реконструкції, а допомогою програми ЕОЛ ПЛЮС версія 5.3.7 було проведено розрахунок розсіювання забруднюючих речовин в атмосферному повітрі та розрахунки індивідуального ризику для здоров'я населення, які показали, що жодна речовина не буде перевищувати ГДК.
забруднення повітря, пилоочисна система

Постановка проблеми. Металургія – одна з основних галузей промисловості функціональними особливостями якої є одержання металів з перероблювальної сировини у вільному металевому етапі або у вигляді хімічної сполуки. Одним із джерел надходження пилу у повітряний простір є вентиляційні гази дільниць підготовки пиловугільного палива, обпалювальних дільниць, металургійних цехів тощо.

Одні із основних напрямків екологічно безпечної металургійної промисловості є впровадження високоефективних методів очищення газів, пневмотранспортування сировини та пилу, вихрових пиловловлювачів, рукавних і шарових фільтрів, електрофільтрів та 4н..

Стратегічною метою національної екологічної політики є стабілізація і поліпшення стану навколишнього природного середовища України шляхом поетапного досягнення цілей національної екологічної політики, як інтегрованого фактора сталого розвитку.

Актуальність теми дослідження обумовлена тим, що удосконалення технології виробництва високоякісного агломерату, який залишається найбільш масовим компонентом шихти, є пріоритетною задачею, яка спрямована на підвищення ефективності пічної плавки. Особливого значення ця задача набуває в умовах реконструкції пило-газоочисного устаткування.

Основні напрямки державної політики України в галузі охорони довкілля, використання природних ресурсів та забезпечення екологічної безпеки. Програма охорони навколишнього середовища в Кіровоградській області до 2025 року передбачають розроблення та створення високоефективних систем комплексного очищення пилогазових викидів металургійної промисловості з одночасною утилізацією уловлених продуктів і подальшого їх використання у виробництві агломерату. Це обумовлює необхідність розроблення високоефективного пилогазоочисного устаткування.

Метою дослідження є екологічна оцінка проєкту реконструкції пило-газоочисної установки Побузького феронікелевого комбінату.

Для досягнення поставленої мети вирішувались такі задачі:

- теоретичне дослідження процесів пиловловлення та газоочищення;

- визначення максимальних концентрації забруднюючих речовин, що викидаються в процесі виробництва огарку в трубчасто-обертових печах, їх масовий та валовий викид за одиницю часу до та після проведення реконструкції пилоочисного устаткування;

- дослідження конструктивних та режимних параметрів апарату на ефективність пиловловлення;

- дослідження впливу газопилових викидів виробництва на атмосферу та здоров'я людей розрахунковим методом.

Об'єктом дослідження є аналіз дотримання гранично-допустимих концентрацій згідно нормативних документів, зниження техногенного впливу на навколишнє середовище.

Предметом дослідження є аналіз дотримання гранично-допустимих концентрацій згідно нормативних документів, зниження техногенного впливу на навколишнє середовище шляхом застосування високоефективної пилоочисної системи.

Методи дослідження базуються на обстежені існуючої системи пилогазоочисних установок; проведенні інструментально-лабораторного вимірювання параметрів газопилового потоку із визначенням максимальних концентрації забруднюючих речовин, їх масовий і валовий викид за одиницю часу; використання математичного моделювання впливу газопилових викидів виробництва на атмосферу та здоров'я людей до та після реконструкції пилоочисної системи.

У роботі використаний гравіметричний метод визначення масової концентрації речовин у вигляді суспендованих твердих частинок в організованих викидах стаціонарних джерел.

Результати досліджень.

З метою визначення кількісних та якісних характеристик викидів забруднюючих речовин та ефективності роботи існуючих газопилових установок були проведені інструментально-лабораторні вимірювання до впровадження реконструкції даних установок.

Для виробництва огарку (сішання, підігрів та обпал з частковим відновленням) використовують чотири трубчастій обертовій печі, а саме: ТОП-1, ТОП-2, ТОП-3 та ТОП-4.

Для очищення газів трубчастих печей від пилу на кожній печі передбачена двохступенева система очищення відхідних газів (перша – «БУРАН» типу МПУ-100.04 в кількості по 2 шт. для кожної ТОП з ефективністю 85,4% та друга – електрофільтр типу ДП 55-3У з ефективністю 99,6%).

Основним видом палива є: природний газ, природний газ та руднотермічний газ (технологічні гази, що містять СО більше 40%), пиловугільне паливо, пиловугільне паливо та руднотермічний газ.

Речовини у вигляді суспендованих твердих частинок недиференційованих за складомв своєму складі мають вміст інших шкідливих речовин, а саме: алюмінію оксид – 1,6%, залізо та його сполуки (у перерахунку на залізо) – 36,28%, кальцію оксид (негашене вапно) – 3,7%, магнію оксид – 17,5%, нікель та його сполуки (у перерахунку на нікель) – 1,7%, хром та його сполуки (у перерахунку на триоксид хрому) – 0,25%, сірка елементарна – 0,6% та кобальт та його сполуки (у перерахунку на кобальт) – 0,07%.

Дана реконструкція здійснюватиметься шляхом доукомплектації існуючого газоочисного обладнання.

Основним завданням реконструкції установок очистки газу підприємства – відповідність сучасним вимогам до газоочисних систем:

- ефективна робота при попаданні крапельної вологи та дрібнодисперсного пилу;

- високоефективна система імпульсної регенерації рукавних фільтрів;

- відсутність мертвих зон в конструкції фільтра.

Циклофільтр – унікальний фільтр, що поєднує в собі переваги циклону та рукавного фільтра. Робота фільтра ґрунтується на двоетапному очищенні: перший етап – циклон; другий – рукавний фільтр із імпульсною регенерацією. Таке поєднання дозволяє застосовувати циклофільтри для очищення газів з високою вхідною запиленістю.

Після динамічного попереднього очищення основного потоку відбувається очищення залишкового пилу, забезпечуючи ефективне відділення дрібних частинок, високу продуктивність фільтра та мінімальні втрати тиску.

У складі реалізації реконструкції застосовується таке основне технологічне обладнання:

- циклофільтри CF-4270-10R та CF-4270-10L;
- димососи лівого і правого виконання HP23.4/7-9R та HP23.4/7-9L;
- компресори продуктивністю 28 м.куб./хв, за нормальних умов кожен;
- осушувачі;
- масловологовідокремлювачі;
- конвеєри скребкові горизонтальні закритого типу.

Газоочисна установка, до скаду якої входять циклофільтрилівого та правого виконання із системою пиловивантаження та димососів, буде розміщена на відкритому майданчику біля існуючої димової труби підприємства.

Циклофільтри встановлюватимуться на металевій рамі, позначка опорного пояса +12,200 м від рівня землі.

Димосос встановлений на фундамент.

Під установкою циклофільтрів, буде розташована будівля компресорної, в якій розміщуватимуться компресори, осушувачі і масловологовловлювачі та електроприміщення.

Вивантаження пилу здійснюється за допомогою новопроектованої системи скребкових конвеєрів на існуючі конвеєри.

Циклофільтр є пиловловлюючим апаратом циліндричної форми з фільтруючими рукавами і системою регенерації всередині, корпус складається з декількох частин і збирається за допомогою болтових з'єднань.

Після циклофільтрів встановлюються двавідцентрові димососи лівого та правого виконання HP23.4/7-9R та HP23.4/7-9L, продуктивністю 240000 м3/годину кожен. При збільшенні робочої температури газів у системі понад допустиму проводиться розведення холодним повітрям за допомогою клапанів підсмоктування повітря.

Конструктивно димососи мають раму, на якій встановлені: равлик з робочим колесом, проміжний підшипниковий вузол та електродвигун. На фундамент димосос встановлюється без застосування віброопор.

Після завершення реконструкції пилоочисної системи викиди ЗР в атмосфернеповітря, по даному джерелу викидів, знизиться на 91,7%.

Висновок. Сучасні пилоочисні установки на базі циклонів, які найбільше використовуються в різних галузях промисловості, характеризуються недосатаньою ефективністю та значними енергетичними затратами на процес очищення. Розроблення нових конструкцій циклонів, особливо неминуче для вловлювання дрібнодисперсної фракції пилу та збільшення високого ККД очищення газів.

Тенденції останніх років до зменшення енергетичних витрат на процеси пилоочищення проявляються у реконструкції існуючих установок.

ТОВ «Побужський феронікелевий комбінат» – найбільше промислове підприємство Кіровоградської області, що спеціалізується на виробництві кольорових металів, а саме: феронікелю.

Керівництво ТОВ «ПФК» приділяє достатньо уваги безпеці навколишнього середовища в екологічному аспекті. Для цього була розроблена екологічна стратегія розвитку підприємства на впровадження комплексної програми природоохоронних заходів – реконструкції пилоочисної системи.

В ході проведення практичних досліджень визначено:

- проведено теоретичне дослідження процесів пиловловлення та газоочищення;
- визначено максимальні концентрації забруднюючих речовин, що викидаються в процесі виробництва огарку в трубчасто-обертових печах, їх масовий та валовий викид за одиницю часу до та після проведення реконструкції пилоочисного устаткування;

- досліджено конструктивні та режимні параметри апарату на ефективність пиловловлення;

- досліджено вплив газопилових викидів виробництва на атмосферу та здоров'я людей розрахунковим методом.

Аналіз теоретичного узагальнення впровадження третього ступеня очистки газопилового потоку від забруднюючих речовин, що викидаються в атмосферне повітря, а саме циклофільтрів з ефективністю 99,9% позитивно впливає на стан атмосфери та здоров'я людей та зменшує викиди на 91,7%.

В результаті дослідження впливу газопилових викидів виробництва на атмосферу та здоров'я людей визначено, що жодна речовина не перевищує гранично-допустимі концентрації відповідного чинного законодавства, ризик шкідливих ефектів неканцерогенного ризику для здоров'я населення украй малий, індивідуальний ризик канцерогенних ефектів визначений як мінімальний.

Список літератури

1. Баштаннік, М.П., Дворецька, І.В., Онос, Л.М., Савенець, М.В. (2016). Основні засади виділення зон якості атмосферного повітря на території України та їх класифікація згідно з вимогами директив 2004/107/ЄС та 2008/50/ЄС. Наукові праці УкрНДГМІ. Вип. 269 С. 123–137.
2. Баштаннік, М.П., Жемера, Н.С., Кіптенко, Є.М., Козленко, Т.В. Стан забруднення атмосферного повітря над територією України. Наукові праці УкрНДГМІ. 2014. – Вип. № 266. С. 70–93.
3. Барабашова Н.В. Правове забезпечення екологічної безпеки в процесі господарської діяльності. / Барабашова Н.В.//- К., 2008.-58 с.
4. Гігієна атмосферного повітря Гігієна населених місць / Міністерство охорони здоров'я України, Академія медичних наук України, Державна установа "Інститут гігієни та медичної екології ім. О. М. Марзєєва АМН України"; гол. ред. А. М. Сердюк. - К.: Полімед, 1956 Вип. 50. - 2007. - 402,[5] с.: табл. - Бібліогр. в кінці ст. - С. 30-43.
5. Гурець Л.Л. Вибір високоефективного газоочисного обладнання з метою запобігання забруднення атмосфери. Екологічна безпека: наук. журн. 2009. № 2. С. 69–72
6. Данова К. В. Конспект лекцій з дисципліни «Методи оцінки небезпечних та шкідливих виробничих факторів» (для студентів 4 курсу денної форми навчання напряму підготовки 6.050702 «Електромеханіка» спеціальності «Охорона праці на електричному транспорті») / К. В. Данова ; Харк. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. - Х.: ХНУМГ, 2014. – 31 с.
7. ДБН А.2.2-1-2003 Склад і зміст матеріалів оцінки впливів на навколишнє середовище (ОВНС) при проектуванні і будівництві підприємств, будинків і споруд. – К.: Держбуд України, 2004р. [Електронний ресурс]. –URL: https://dbn.co.ua/load/normativy/dbn/1-1-0-242.

УДК 930.25

О. Коломієць, канд. пед. наук, доцент
М. Люлько, магістр гр. ІС-21М (1,4)
Центральноукраїнський національний технічний університет

РОЛЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ОРГАНІЗАЦІЇ ОНЛАЙН-ВИСТАВОК В АРХІВНИХ УСТАНОВАХ

У статті досліджено роль програмного забезпечення в організації онлайн виставок, що має задовольнити потреби в оперативності фіксації, збору, оброблення, зберігання, пошуку та передачі документальної інформації, а також надання цієї інформації в будь-який проміжок часу і в зручній формі для будь-якого структурного відділу архівної установи та її користувачів.
документальні виставки, архівна установа, інформаційні технології, віртуальні фотоальбоми, інформаційні стенди

Актуальність статті. Актуальність проблеми ефективності програмного забезпечення в організації онлайн виставок обумовлюється необхідністю застосування сучасних високотехнологічних і прогресивних підходів до організації діяльності сучасних архівних установ, активного використання ними віртуального середовища, завдяки чому з'являється більш широкий доступ до архівних документів. Одним із напрямів цієї діяльності є організація та проведення документальних виставок онлайн.

Мета статті. Проаналізувати роль програмного забезпечення в організації онлайн виставок, визначити можливості для його удосконалення та підвищення ефективності.

Одним із стратегічних напрямів організації доступу до архівної інформації, що становить історичну та культурну цінність, є оцифрування документів.

У контексті розвитку відкритого суспільства маємо характеризувати й діяльність архівів України в плані репрезентації архівних інформаційних ресурсів в мережі «Інтернет», зокрема й через створення веб-сайтів архівних установ, об’єднаних у єдиний вебпортал «Архіви України», що є невід’ємною складовою інформаційних ресурсів України.

Веб-сайт розглядається як документно-інформаційна система, оскільки йому притаманні всі ознаки такої системи у класичному розумінні: наявність масиву документів, засобів інформаційного пошуку, техніко-технологічних засобів фіксування, зберігання та представлення інформації, що існує у вигляді організованої сукупності електронних документів та їхніх зв’язків.

Офіційний вебпортал «Архіви України» було створено у результаті комплексу організаційно-правових заходів пов’язаних з інформатизацією архівної галузі і на сьогодні є потужним профільним інформаційним ресурсом, який містить величезний масив різнопланових відомостей про документальну спадщину України.

З твердження А. Кисельової, яка визначає «веб-сайт «Архіви України» як «новий вид архівного довідника – комплексний електронний довідник оперативного характеру» [1].

Можливості такого довідника досить широкі: він може включати описову (повідомлення, хроніка, огляд тощо), бібліографічну та археографічну (каталоги, покажчики, списки, переліки), едиційно-текстову (публікація джерел та наукових ресурсів), зображувальну (зокрема й кіно-фото ресурси) інформацію, зокрема, й краєзнавчого змісту, бази даних тощо.

Наявність у кожного державного архіву офіційного веб-сайту є невід'ємною складовою його інформаційної діяльності. Погоджуємося з тим, що можна виокремити наступні основні елементи (форми) представлення інформаційних ресурсів на веб-сайті архіву:

1. Інформативні сторінки веб-сайту – огляди фондів архіву, описи каталогів, картотек і покажчиків, які є в архіві.

2. Електронні довідники різноманітних жанрів і видів – путівники, анотовані реєстри описів тощо.

3. Цифрові зображення документів – у вигляді електронних виставок (онлайн виставок).

4. Власна науково-інформаційна робота працівників архіву – наукові публікації, дослідження.

Характерним для більшості архівних веб-сайтів є те, що практично на кожному з них подано інформацію про фонди. На веб-сайтах обласних архівів України характеристика складу фондів розміщена у рубриках «Фонди», «Склад фондів», «Склад та характеристика фондів», «Огляди фондів», «Науково-довідковий апарат».

Показовими у плані оцифрування фондів став і Державний архів Кіровоградської області, на сайті якого у рубриці «Електронний архів» розміщені цифрові матеріали наявних в установах соціальної пам'яті фондів, описів, путівників, метричних книг ті інших архівних ресурсів [2].

До сучасних методів репрезентації архівної інформації відноситься також організація виставок архівних документів, які функціонують у мережі «Інтернет» в режимі виставок онлайн [3].

Виставки, як важливий напрям ПР-технологій архіву в сучасному соціально-комунікаційному середовищі, розраховані на всі три види взаємодії з оточуючим середовищем: зовнішньосистемним, внутрішньосистемним та міжсистемним.

Середньостатистичний користувач витрачає на знайомство з сайтом приблизно 10-15 хвилин, вирішуючи для себе питання про те, чи залишатися йому на сайті або закривати сторінку. Тому при створенні виставки архів повинен обирати засоби дизайну та оформлення, які привернуть увагу відвідувача експозиції і зацікавлять його. Не менш важливим для успіху виставки є інтерфейс сайту: інтернет-користувач повинен швидко знаходити цікаву для нього інформацію [4].

На вебпорталі ДАКО можна спостерігати постійно діючі документальні виставки онлайн, які класифікуються за такими рубриками:

автограф української історії,
голодомор українського народу,
історія державності,
радянський тоталітаризм, вибори, персоналії,
Друга світова війна,
Чорнобиль, пам'ятні дати, та інші.

Також, окрім створення власного вебпорталу, у роботі ДВО були задіяні певні програми для покращення онлайн виставки. Тому сьогодні на ринку програмного забезпечення можна спостерігати запропоновану низку продуктів, які варто застосувати для створення віртуальних експозицій: Calameo, PhotoPeach, Prezi, Dipity тощо.

Calameo – це сервіс для миттєвого створення інтерактивних публікацій в Інтернеті для читання з комп'ютера. Водночас робота з ними подібна до роботи з паперовими документами: можна перегортати сторінки, відзначати цікаві місця, збільшувати масштаб зображення. До основних переваг Calameo належать такі можливості:

– додавання інтерактивного контенту;
– аналіз поведінки читацької аудиторії;
– популяризація публікації (електронної виставки);
– відображення публікації на конкретному сайті;

– поширення публікації через соціальні мережі.

Зручним інструментом для створення слайд-шоу є онлайн-сервіс PhotoPeach. Використовуючи його засоби, можна сформувати з фотографій презентацію, додавши до неї музичний супровід і коментарі. Перевагою PhotoPeach є відсутність спеціальних вимог до фотографій, оскільки при завантаженні всі файли автоматично масштабуються до потрібних розмірів. Єдиною умовою для застосування сервісу є формат зображень – лише JPG, JPEG, PNG, GIF. Сервіс Prezi призначений для створення медіапрезентацій. Його інструментарій надає можливість наближати й віддаляти слайди, створюючи тривимірний ефект. Prezi доцільно використовувати для підготовки виставки-презентації, невеликих оглядів тощо. Матеріал можна згорнути в одне зображення і, збільшивши певний слайд, зробити акцент на важливому тексті, слові, ілюстрації.

Основним недоліком цього сервісу, як і низки інших, є потреба підключення до Інтернету та англійський інтерфейс. Prezi вирізняють певні переваги, це:

– хмарний сервіс для створення презентацій;

– нелінійна структура;

– велика бібліотека якісних шаблонів для презентацій;

– якісні шрифти;

– динаміка, зумування, візуальні ефекти;

– можливість імпортувати зображення, відео, презентації PowerPoint.

Для відображення хронологічних подій, інформацію про які збережено в архівних фондах, доцільно застосовувати онлайн сервіс Dipity. Цей інструмент дає змогу представити події, явища чи факти в прив'язці до часу. Dipity є одним із найфункціональніших у своїй категорії, адже допомагає створювати хронологічні послідовності, які можна переглядати в кількох варіантах: у тайм-лінії, фотоальбомах, простому текстовому списку подій і навіть на карті, якщо при створенні вказано географічну прив'язку для кожної події.

Такий нюанс вкрай важливий для архівних документів загальноукраїнського значення (матеріали, присвячені незалежності України, її розбудові та утвердженню на політичному, інституційному й міжнародному рівнях). Особливістю Dipity є можливість його інтеграції з найпопулярнішими онлайновими сервісами та джерелами інформації [5].

Сервіс створення інтерактивних хронологічних стрічок дає змогу застосувати сучасний, динамічний і зручний спосіб представлення архівних даних у межах проведення презентації тематичної виставки. Певним недоліком таких сервісів є обмеження на фотографування архівних матеріалів.

Для створення персональних віртуальних тематичних виставок користувачі повинні мати не лише дозвіл на копіювання архівних джерел, а й портативний сканер, призначений для оцифрування текстів і зображень, що дасть змогу швидко опрацювати великий обсяг різноманітних документів (від фотографій до важливих рукописних архівних документів).

Отож, організація документальних онлайн виставок посідає важливе місце в інформаційній діяльності сучасних архівів. Адже завдяки такій роботі архіви відкривають свої інформаційні ресурси для широкого загалу, стають ближчими до суспільства. Інформаційний потенціал є доволі значним, але, разом з тим, не цілком достатнім для подальшого розвитку архівів як сучасних соціально-комунікаційних структур.

Актуальною залишається проблема розробки цільової комплексної програми соціально-наукової та електронної комунікації в мережі архівних установ України, гармонізація та впровадження міжнародних стандартів у цій галузі.

Реалізація перспективних напрямів розвитку веб-сайтів архівів нашої держави, використання сучасних методів репрезентації соціально значущої ретроспективної документної інформації дозволять поглибити процеси їх організаційно-функціональних трансформацій у сучасному інформаційному середовищі, забезпечить надійне підґрунтя переходу до формування єдиного архівного інформаційного простору.

Для удосконалення програмного забезпечення було б добре залучити програму для тривимірної реальності. Завдяки такій програмі, громадяни змогли не лише просто

переглядати виставку, а й мали б можливість побувати в віртуальному архіві. Це допомогло б залучити більше користувачів.

Вважаємо, що документні матеріали, представлені на виставках, можна доповнити також відео та аудіо файлами, це теж могло б допомогти в залученні аудиторії.

Отже, в сучасному інформаційному суспільстві важливими аспектами є відкритість суспільства і вільний доступ до інформації. Відтак, державні архіви України, попри існуючі проблеми, керуючись принципами демократичної і соціально правової держави та в міру наявних можливостей, намагаються забезпечити всім членам українського суспільства умови рівного, нічим не обмеженого доступу до історико-культурної спадщини України.

Список літератури

1. Кисельова А. Використання архівних електронних ресурсів: статистика відвідувань архівного порталу України. Архіви України. 2004. № 1-2. С. 158–167.
2. Офіційний веб-сайт Державного архіву Кіровоградської області. URL: https://dakiro.kr-admin.gov.ua/.
3. Левчук О. Соціалізація архівів у контексті основних тенденцій розвитку сучасного інформаційного простору. Архіви України. Київ, 2021. Вип. № 4. С. 25-41.
4. Тюрменко І.І. Установи соціальної пам'яті у соціокультурному інформаційному простору. монографія Київ, 2020. 50 с.
5. Вовк Н. Електронні виставки архівних документів: сучасний стан та перспективи. Вісн. Книжкова палата. Київ,2018. №5. С. 44-48.

УДК 504.3.06

А. Рижонок, студ. гр. ЕО-21М
Л. Коломієць, доцент кафедри ЕОНСЗСЖ
Центральноукраїнський національний технічний університет

РОЗРОБКА ЗАХОДІВ ДЛЯ ЗМЕНШЕННЯ ЕКОЛОГІЧНИХ ПРОБЛЕМ ОЛІЙНОЕКСТРАКЦІЙНОГО ВИРОБНИЦТВА

У статті розглядається технологія олійного виробництва, визначаються заходи для зменшення атмосферного забруднення.

соняшникова олія, екстракція, рафінація, гдратація, гідрогенізація, санітарно-гігієнічні вимоги, гексановий розчинник, аміак, рекуперація

Актуальність теми. Галузь виробництва олій та твердих жирів займає одне з провідних місць в економіці держави. Водночас є ряд негативних впливів на довкілля технологічного процесу, тому актуальним є впровадження зокрема сучасних систем очищення викидів.

Аналіз останніх досліджень і публікацій. Для того, щоб з'ясувати причини можливого забруднення атмосфери, необхідно охарактеризувати технологічний процес олійно-жирового виробництва, який складається з багатьох послідовних етапів, та врешті зводиться до двох основних методів вилучення ліпідів: пресового та екстракційного.

Основний спосіб переробки насіння соняшнику на підприємствах -екстракційний, за допомогою якого олію добувають методом механічного тиску з подальшою екстракцією розчинником.

Очищення насіння від мінеральних і олійних домішок проводиться з метою підвищення якості олії, збільшення її виходу, а також збереження обладнання. Мінеральні

домішки призводять до зносу обладнання. Вони поглинають олію, збільшують втрати і зменшують її вихід. Для очищення насіння застосовують сепаратори, магнітні апарати.

Обрушування оболонок (шеретування). Відділення оболонок від ядра складається із руйнування покривних оболонок насіння – обрушування і розділення одержаної суміші на ядро і лузгу. Для обрушування застосовують оббивальні машини (рушанки). Добре відрегульована машина обрушує насіння на 95 %.

Подрібнення насіння. Для добування олії з насіння необхідно зруйнувати його клітинну структуру. Кінцевий результат операції подрібнення – перехід олії, що міститься в клітинах насіння, у доступну для подальшого технологічного обробітку форму. Необхідний ступінь подрібнення сировини досягається механізмами, що виконують дроблення, роздавлювання і розтирання ядра. Одержаний після подрібнення матеріал - м'ятка, вона відрізняється великою питомою поверхнею.

Гідротермічний обробіток м'ятки. Олія, адсорбована у вигляді тонких плівок на поверхні частинок подрібненого ядра м'ятки, утримується значними поверхневими силами. Змінюються фізико-механічні властивості м'ятки, вона перетворюється в м'язгу.

Процес приготування м'язги складається із зволоження та підігрівання її до температури 60°С. Вологість м'ятки після зволоження має бути не вища 8–9 %, потім м'ятку нагрівають до температури 105°С. Оброблена таким чином м'ятка називається м'язгою. Кінцева вологість готової м'язги 5–6 %, що забезпечує ефективний попередній віджим олії. Для завершального віджиму параметри м'язги повинні бути такими: температура 110-120°С, вологість 3–4 % [1-3].

М'язгу із ядра соняшника при одноразовому пресуванні на пресах подвійної дії після подрібнення направляють в пропарювально-зволожувальний шнек, там її зволожують насиченою парою до вологості 8–9 % і нагрівають до температури 80-85°С. Вже зволожену м'язгу піддають тепловій обробці в жаровні з доведенням вологи до 2,0–1,5 % і температури 115-120°С. Процес прожарювання триває 40–45 хв. Для приготування м'язги застосовують жаровні, які за конструктивними особливостями поділяють на чанні, шнекові, барабанні

Добування олії із м'язги здійснюється способами пресування або екстракції.

Спосіб пресування. Пресування як спосіб добування олії із насіння здійснюється безпосередньо для одно- або двократного пресування з метою одержання олії, а також може передувати екстракції.

Залежно від тиску пресування і олійності в макусі шнекові преси можна розділити на преси попереднього (неглибокого) добування олії – форпреси, і преси остаточного (глибокого) добування олії – експелери.

Форпреси найбільш широко застосовуються в технологічних схемах екстракційних заводів. Вони відрізняються високою продуктивністю (80 т насіння за добу) при порівняно невисокому добуванні олії (олійність макухи до 15–17%).

Преси глибокого добування олії – експелери – мають значно меншу продуктивність (18–30 т насіння за добу), проте олійність макухи вони доводять до 4–7 %.

Для майже абсолютного добування олії можлива лише екстракція. У відходах, які називають шротом, олії залишається до 1 %. Для збільшення поверхні дотику з розчинником форпресовій макусі надають форми тонких пластинок (пелюстків) товщиною 0,25–0,50 мм, пропускаючи через спарену плющильну вальцівку з гладенькими вальцями. Як розчинник для екстракції застосовують бензин марки А і Б та гексан.

Екстракція олії у своїй фізичній основі є дифузним процесом. Вона може бути виконана трьома способами:

- зануренням підготовлених пелюстків макухи в протипотоковий рух розчинника. Цей спосіб передбачає безперервний процес, при якому і розчинник і пелюстки макухи переміщуються назустріч один одному в шнековому апараті – екстракторі;
- ступінчатим зрошенням розчинником протипотокового переміщення пелюстків – цей спосіб передбачає безперервне переміщення тільки розчинника, а пелюстки макухи залишаються в спокої на стрічці, що рухається в горизонтальному стручковому екстракторі;

- змішаний спосіб – на першій стадії (замочування) передбачає застосування першого способу, а на другій стадії (зрошення) – другого.

При контактуванні пелюстків макухи і розчинника, що відбувається в екстракторі, олія розчиняється у розчиннику, утворюючи так звану місцелу, яка потім виводиться з екстрактора. Місцела містить від 10–15 до 30–35 % розчинника. Шрот, що виходить із екстрактора від 20- 40% розчинника, який видаляється за допомогою нагрівання в шнекових або чанних випаровувачах. У результаті розчинник випаровується, а шрот підсушується і охолоджується. Місцелу після екстрагування фільтрують на спеціальних фільтрах і збирають у місцелозбірники.

Для видалення олії із місцели застосовують відгонку розчинників дистиляцією. Такий процес проводять в установках для двоступінчатої дистиляції, що складається із плівкового, тобто попереднього, та остаточного дистиллятора. Перший працює при атмосферному тиску, а другий - при залишковому до 8 кПа. [18].

Рафінація олії. В сирих оліях завжди містяться найрізноманітніші домішки. Частина їх разом з олією витягується із клітин насіння під впливом підвищеної температури, тиску і органічного розчинника. Тому в товарній олії завжди наявні воски, фосфоліпіди, барвні речовини і продукти розкладу цих речовин (вільні жирні кислоти, моно- і дигліцериди та інші речовини). В олії, одержаній із насіння, також наявні продукти окислення ліпідної природи. Їх вміст залежить від якості насіння й інтенсивності технологічного впливу на насіння при одержанні олії. Крім розчинних речовин, товарна олія містить і механічні домішки.

Рафінація - процес очищення олії від супутніх речовин.

Можна виділити такі методи рафінації:

- фізичні;
- хімічні;
- фізико-хімічні.

Вибір методу рафінації залежить від складу і якості домішок, їх властивостей і призначення олії. Зазвичай очищення олії досягається поєднанням декількох методів.

Фізичні методи. Застосовуються для первинного очищення олії, а також для видалення нерозчинних в олії частинок, які утворюються в процесі рафінації. Видалення з олії твердих частинок м`язги відбувається за допомогою відстоювання, центрифугування і фільтрування.

Для відстоювання олію у тарілчатих відстійниках залишають в спокої протягом тривалого часу. Під дією сили тяжіння важкі частинки осаджуються на дно. Це так звана пасивна рафінація.

Центрифугування – це ефективний метод очищення олії від механічних домішок та води під дією відцентрової сили. Розрізняють розділяючі центрифуги, які застосовують для відділення води від олії, і освітлювальні – для видалення механічних домішок.

Фільтрування дозволяє відділити механічні домішки, густина яких не відрізняється від густини олії. Фільтрують олію через шар спеціальної фільтрувальної тканини на фільтрпресах. Також застосовують спосіб подвійного фільтрування, відповідно до якого крупні частинки відділяються і продукт надходять на першу (гарячу) фільтрацію на рамних фільтрпресах.

Потім олію за допомогою повітряних калориферів охолоджують до температури 20-25°С і повторно спрямовують на фільтрпреси. Відфільтровану й охолоджену олію направляють в склад. Кінцевий продукт при очищенні олії від механічних домішок - нерафінована олія [1,2,4].

Хімічні і фізико-хімічні методи рафінації. За допомогою цих методів ефективно видаляють вільні жирні кислоти, слизі, білки, фосфоліпіди і деякі інші сполуки. До цих методів належать гідратація, адсорбційна рафінація, дезодорація, нейтралізація та виморожування.

Гідратація – вилучення колоїдно-розчинних фосфоліпідів, білкових і інших речовин, за якої кінцевими продуктами є гідратована олія та фосфатидний концентрат. У процесі гідратації олію обробляють водою або розчином повареної солі у реакторах-турбулізаторах або струйних змішувачах. Останній являє собою ежектор, що забезпечує інтенсивне змішування олії і води, якої вводять 0,5–2,0 % до маси олії, температура олії 45-60 °С. Суміш води і олії подають в коагулятор, де проходить формування гідратаційного осаду, що відділяється від олії.

Нейтралізація або лужна рафінація – це процес вилучення вільних жирних кислот, у результаті чого кінцевим продуктом є рафінована олія, а відходами – нерозчинні солі (мильні розчини), які випадають в осад, захоплюють з собою різноманітні домішки: барвні речовини, фосфоліпіди, слизі. Утворений осад отримав назву соапстоки.

Адсорбційна рафінація (відбілювання) – це вилучення барвних речовин за допомогою спеціальних сорбентів. В результаті відбілювання кінцевим продуктом є відбілена рафінована олія, а відходом – відпрацьований сорбент. Повнота видалення пігментів визначається видом олії і її призначенням.

Олії, направлені на гідрогенізацію, які використовуються у виробництві маргарину, повністю знебарвлюються. При виробництві салатних олій вони знебарвлюються частково. Сорбенти (відбілювальні глини, порошки), що застосовують при адсорбційній рафінації, повинні мати велику адсорбційну ємність відносно до пігментів, що видаляють, повинні утримувати мінімальну кількість олії, мати максимально низьку оліємність, не вступати в хімічну взаємодію з олією і легко від неї відділятись. Для виробництва окремих порошків (глин) застосовують природні матеріали: трепел, гумбрин, осканагель і ін. Для підвищення відбілюючої здатності вони повинні піддаватись спеціальній обробці – активації.

Для видалення барвних речовин олію змішують із сорбентом і відділяють із допомогою фільтрації. Кількість сорбенту залежить від виду олії, її призначення, та характеристики сорбенту.

Дезодорація – вилучення речовин, що надають небажаного аромату і частково смаку. Кінцевим продуктом є рафінована дезодорована олія. Відходи – продукти відгонки. Більшість речовин, що визначають смак, аромат олії, відносять до альдегідів, кетонів, низькомолекулярних кислот, вуглеводів. В основі процесу дезодорації лежить відгонка (дистиляція) ароматичних речовин водяною парою.

Для інтенсифікації дезодорації жирів і олій процес здійснюється під глибоким вакуумом і при високій температурі. Олію підігрівають до 60°С і подають в деаератор, там вона розпилюється у вакуумі і підігрівається в тонкій плівці на поверхні змійовиків до 130–180°С. Після цього олію нагрівають до 220-230°С і подають в дезодоратор. Усередині апарата тонка плівка олії, що стікає по вертикальних пластинах, добре контактує з водяною парою, що подається інжекторами знизу. Обігрів дезодоратора здійснюється через зовнішні змійовики і парову сорочку.

Виморожування (вінтеризація) – це процес вилучення з рафінованих дезодорованих олій восків і воскоподібних речовин.

Рафіновану олію зберігають у спеціальних щільно закритих місткостях без доступу повітря, вологи і світла [3,5].

Постановка завдання. Для вирішення питання зниження впливу на довкілля виробництва олії та твердих жирів необхідно виконувати контроль якості сировини та продукції та впроваджувати сучасне очисне обладнання.

Результати досліджень. Соняшникова олія повинна обов`язково відповідати вимогам стандарту ДСТУ 4492:2005, і її виробляють згідно з чинним технологічним регламентом, або технологічною інструкцією, затвердженими у встановленому порядку, з додержуванням вимог ДСП 4.4.4-90.

За визначеннями органолептичних та фізико-хімічних показників олія соняшникова повинна відповідати встановленим санітарно-гігієнічним вимогам. А саме, в першу чергу визначається: прозорість, смак та запах, колірне число, кислотне число, пероксидне число,

масові частки фосфоровмісних речовин, нежирних домішок, вологи та летких речовин, віск та соскоподібні речовини, якісну пробу на мило, температуру спалаху та ступінь прозорості. При визначенні всі перелічені показники знаходилися в межах норми (табл.1).

Таблиця 1- Органолептичні та фізико-хімічні показники олії соняшникової рафінованої та рафінованої дезодорованої

Назва показника	Характеристика показників олії			
	рафінована		рафінована дезодорована	
	невиморожена	виморожена	невиморожена	виморожена
			Д П	Д П
Прозорість	Прозора без осаду			
Смак та запах	Притаманні олії соняшниковій рафінованій без стороннього присмаку, гіркоти та запаху		Смак знеособленої олії, без запаху	
Колірне число, мг йоду, не більше ніж	12		10	
Кислотне число, мг КОН/г, не більше ніж - свіжовиробленої олії - наприкінці терміну зберігання	0,25 0,60		0,25 0,60	
Пероксидне число, 1/2 О ммоль/кг, не більше ніж - під час випуску з підприємства - наприкінці терміну зберігання	6,0 10,0	6,0 10,0	2,0 10,0	2,0 10,0
Масова частка фосфоровмісних речовин, % - у перерахунку на стеароолеолецитин - у перерахунку на P_2O_5	Відсутність Відсутність			
Масова частка нежирових домішок, %	Відсутність			
Масова частка вологи та летких речовин, %, не більше ніж	0,10		0,10	
Віск та воскоподібні речовини	Не визначають	Відсутність	Не визначають	Відсутність
Мило (якісна проба)	Відсутність		Відсутність	
Температура спалаху олії екстракційної, °С, не нижче ніж	225		234	
Ступінь прозорості, фем, не більше ніж	15		15	
Анізидинове число	Не нормують			

При дослідженні масової частки вологи, встановлено, що показник є на рівні 0,10%, що відповідає вимогам.

Вміст токсичних елементів, пестицидів і мікотоксинів в олії соняшниковій не повинен перевищувати встановлені гранично допустимі концентрації. Ці показники зазначені у таблицях 2, 3, 4 [21].

При дослідженні вмісту токсичних речовин, встановлено, що свинець, кадмій, ртуть, мідь, залізо та інші знаходяться в межах норми (табл. 2).

Таблиця 2 - Допустимі рівні вмісту токсичних елементів і мікотоксинів в олії соняшниковій

Назва токсичного елемента	Допустимі рівні, мг/кг, не більше ніж
Свинець	0,1
Миш'як	0,1
Кадмій	0,05
Ртуть	0,03
Мідь	0,5
Залізо	5,0
Цинк	5,0
Афлатоксин B_1	0,005
Зеараленон	1,0

Так, за результатами досліджень, вміст афлатоксину B_1 не перевищує 0,005 мг/кг сировини. Також проводилося дослідження, щодо допустимих рівнів вмісту пестицидів в олії. Усі показники перебували у межах норми (табл. 3).

Таблиця 3 - Допустимі рівні вмісту пестицидів в олії соняшниковій

Назва пестициду	Максимально допустимі рівні, млн$^{-1}$ (мг/кг)		
	Для безпосереднього використовування на харчові цілі	Для перероблення на харчові продукти	На технічні цілі
ГХЦГ гама-ізомер (гексахлоран)	0,05	1,0	більше ніж 1,0
Гептахлор	не допустимо		
ДДТ	0,1	0,25	більше ніж 0,25

Вміст ДДТ не перевищує 0,25 мг/кг олії. Отже, якість продукції задовольняє санітарно-гігієнічні нормативи і вимоги споживачів.

При екстракційному виробництві забруднення повітря відбувається при веденні таких технологічних процесів:

- екстракція олії розчинником;
- грануляція шроту.

На усе пилогазоочисне обладнання олійноекстракційного виробництва заведені паспорти, та заповнюються акти перевірок, які проводяться спеціалізованими організаціями з метою контролю ступеня очищення викидів в атмосферу.

Технологічне джерело шкідливих викидів в атмосферу - це джерело представляє собою викид пароповітряної суміші після установки масляної абсорбції. Повітря, яке викидається, має температуру 30°С, концентрація пари розчинника не більше 20 г/м^3;

Вентиляційні викиди – це джерело представляє собою викид в атмосферу повітря з парою розчинника після вентиляційного устаткування. Розчинник втрачається через нещільності в апаратурі, насосах, арматурі та т.п.

Гранично-допустима концентрація пари розчинника згідно ГОСТ 12.1.005 у приміщенні робочої зони (ГДКрз) - 100 мг/м^3.

Гранично-допустима концентрація пари розчинника при викидах у повітря населених пунктів ГДК м.р., мг/м^3:

- максимально-разова - 5;
- середньодобова -1,5.

Найбільш небезпечні місця, з точки зору вмісту пари розчинника в повітрі, обладнані місцевими відсмоктувачами.

Місця забору та викиду припливної та загально обмінної вентиляції розміщені у відповідності з правилами СНиП 2-04-05.

При виробленні гранульованого шроту технологічними джерелами забруднення повітря є технологічне повітря з циклону. Кількість повітря, яке викидається через циклон - 26000 м^3/год.

Гранично-допустима концентрація у приміщенні робочої зони (ГДКрз):

- шротового пилу - 4 мг/м^3;
- пари розчинника -100 мг/м^3.

Гранично-допустима концентрація шротового пилу при викидах у повітря населених пунктів, мг/м^3:

- максимально-разова - 0,5;
- середньодобова -0,15.

Контроль концентрації повітря з домішками пари розчинника та шротового пилу, що викидається у атмосферу, повинен здійснюватись періодично, але не рідше 1-го разу на рік, у відповідності з графіком, який затверджений керівником підприємства згідно договору з підрядною організацією, що має відповідну акредитацію та право на проведення таких робіт.

В олійноекстракційному виробництві забруднення повітря відбувається при веденні таких технологічних процесів:

- екстракція олії розчинником;
- грануляція шроту.

Завдяки останній реконструкції (заміні рекуперацийного обладнання) підприємство досягло декілька цілей, а саме:

- зменшення викидів в атмосферу гексанового розчинника;
- зменшення плати за викиди в атмосферу;
- максимальне повернення гексанового розчинника в технологічний процес;
- видалення з території підприємства аміачно - холодильної установки.

Гексановий розчинник, який використовується при екстракції олійної сировини є легкозаймистою речовиною, температура самозаймання 262°С.

Дія на організм парів гексанового розчинника наркотична. При диханні може розвинутись гостре та хронічне отруєння. Тому зменшення викидів в атмосферу, в робочу зону парів є не тільки зменшення впливу на навколишнє середовище, а і вимогами безпеки життєдіяльності. Такий результат досягли за допомогою встановлення нового обладнання – масляного абсорбера.

Завдячуючи реконструкції підприємство зменшило викиди гексанового розчинника в атмосферу на 89,6%. Якщо до реконструкції викиди гексанового розчинника з екстракційного цеху складали 138,106 тонн на рік, то після реконструкції викиди зменшилися до 14,423 тон на рік.

А також в зв'язку з видаленням аміачно-холодильної установки викиди аміаку, які складали 0,125 тонни на рік, з території видалена отруйна речовина – аміак. На підприємстві аміак знаходився у газоподібному та рідкому стані в устаткуванні та трубопроводах аміачно-холодильної установки у кількості 0,63 тони.

Висновки. Таким чином, в процесі отримання олії із зернової сировини здійснюється поступання в атмосферу шкідливих речовин. Організаційно-технічні рішення для зниження виділення забруднень передбачають постійну модернізацію згідно новітніх науково-технічних розробок.

Список літератури

1. https://lektsii.org/5-62258.html
2. https://infopedia.su/6x9fa7.html
3. Іваненко Ф. В. Технологія виробництва і переробки продукції тваринництва: Навч.-метод. посібник для самост. вивч. дисц [Електронний ресурс]. — К.: КНЕУ, 2014. — 125 с.
4. https://news.ztu.edu.ua/wp-content/uploads/2019/03/olijnytsya-1.pdf
5. Олійно-жирова промисловість: традиції та інновації. Вітчизняний та світовий досвід : наук.-допом. бібліогр. покажч. / упоряд. Т. П. Фесун ; Наук.-техн. б-ка; Нац. ун-т харч. технологій. – Київ : НУХТ, 2019. – 185 с.

УДК 504.75

М. Синєок, студ. гр. ЕО-21М
Л. Коломієць, доцент кафедри ЕОНСЗСЖ
Центральноукраїнський національний технічний університет

ЕКОЛОГІЧНА ОЦІНКА ЗАХОДІВ З ОХОРОНИ АТМОСФЕРНОГО ПОВІТРЯ НА ЕЛЕВАТОРАХ

У статті висвітлено необхідність використання очисного обладнання в процесах елеваторних комплексів, проаналізовано існуюче обладнання зерноочистки.
елеватори, зернопереробка, зерновий пил, екологічна безпека, фільтри, циклони, аспіраційні системи, газоочисні установки

Актуальність теми. Зернопереробна галузь – це ланка, яка з'єднує виробника сільськогосподарської продукції з ринком хлібопродуктів та виконує функції заготівлі, зберігання, обробки зерна та перетворення сировини в продукти харчування чи корми; включаючи логістику. Україна є аграрною державою, тому даний напрямок народного господарства постійно розвивається, збільшуються потужності, з'являються нові виробничі майданчики з відповідним обладнанням, в процесі роботи якого створюються деякі впливи на довкілля. На фоні росту антропогенного тиску на останнє актуальним зокрема є контроль рівня виділення пилу та впровадження заходів, покликаних знизити забруднення атмосферного повітря.

Аналіз останніх досліджень і публікацій. В процесі зернообробки застосовується різноманітне технологічне обладнання, призначене для очищення, сушіння, подрібнення, провіювання зерна і продуктів його переробки, для змішування і дозування сировини, гранулювання і брикетування комбікормів, а також для транспортування і зберігання. Пиловловлювальне обладнання за призначенням підрозділяється на наступні типи:

– обладнання, яке використовується для очищення повітря, що подається в приміщення системами припливної вентиляції, кондиціонування повітря і повітряного опалення (повітряні фільтри);

– обладнання, що застосовується для очищення від пилу повітря, що викидається в атмосферу системами місцевої витяжної вентиляції (пиловловлювачі).

Залежно від способу відділення пилу від повітряного потоку розрізняють обладнання для уловлювання пилу сухим способом (частки осідають на суху поверхню) і обладнання для уловлювання пилу мокрим способом, при якому відділення частинок від повітряного потоку здійснюється з використанням рідин.

Устаткування, що вловлює пил сухим способом, поділятися на чотири групи: гравітаційне, інерційний, фільтраційну, електричне.

У кожній групі розрізняються види обладнання. Так, група інерційного обладнання для уловлювання пилу сухим способом поділяється на такі види: камерне, жалюзійних, циклон, ротаційне.

Використовується також комбіноване пиловловлювальне обладнання. До нього відноситься обладнання, в якому відділення пилу від повітряного потоку здійснюється послідовно в кілька ступенів, що розрізняються за принципом дії, конструктивним особливостям і способу очищення [1-3].

Слід зазначити, що класифікація обладнання проведена за основним принципом дії. Практично немає газоочисних установок (ГОУ), які працювали б на використанні лише одного фізичного чи хімічного явища. Ступінь очищення повітря від пилу (ефективність) характеризує відношення маси пилу, вловленого в обладнанні, до маси що надійшла в нього пилу (зазвичай в%, іноді в частках одиниці). Ефективність очищення є найважливішою характеристикою пиловідокремлювачів. Отож її приймають до уваги при виборі пиловловлюючого обладнання відповідно до гранично допустимою концентрацією пилу в очищеному повітрі.

Якщо вирішується питання про вибір того чи іншого виду обладнання, то кращим звичайно є той з них, який за інших рівних умов (економічність, капітальні та експлуатаційні витрати та ін.) має більш високу ефективність очищення. Порівнюють при цьому не відсоток (або частку) вловлений пилу, а відсоток (або частку) пропущеної пилу. Наприклад, якщо ступінь очищення одного апарату 99%, а іншого 98%, то вони пропускають відповідно 1 і 2% пилу. Отже, ефективність очищення в першому апараті в два рази вище, ніж у другому.

Ефективність очищення не є повною характеристикою обладнання, так як показує лише масову частку вловленого пилу, не кажучи нічого про фракційному складі вловленого і пропущеної пилу.

Продуктивність характеризується кількістю повітря, яке очищується за 1 год. Пиловідокремлювачі, в яких повітря очищається через фільтруючий шар, характеризуються питомим повітряним навантаженням, тобто кількістю повітря, яке проходить через 1 $м^2$ поверхні, що фільтрує за 1 год.

Гідравлічний опір має важливе значення, оскільки від цієї величини залежить необхідний тиск вентилятора, а отже, і витрата електроенергії. Витрата електричної енергії залежить в значній мірі від гідравлічного опору апарату. У пристроях, заснованих на осадженні пилу під дією електростатичного поля, електроенергія витрачається безпосередньо на створення цього поля. Витрата електроенергії при одноступінчастому очищенні знаходиться в межах від 0,035 до 1,0 кВт-год на 1000 $м^3$ повітря.

При виборі пиловловлюючого обладнання, крім можливостей пиловловлювачів, в уловлюванні частинок відповідної дисперсності враховують також особливості пилу: фізичні і хімічні властивості, в тому числі здатність утворювати вибухонебезпечні суміші, умови займання, схильність до коагуляції і ін., а також цінність пилу з огляду на можливість утилізації.

Серед конструкцій пилоуловлюючих пристроїв розрізняють характеристики (ефективність і ін.), отримані при уловлюванні пилу, на які розраховані дані апарати. Виділяють наступні види:

- Пиловловлювальні камери, принцип роботи яких оснований на дії сили тяжіння (гравітаційної сили);

- Інерційні пиловловлювачі, принцип роботи яких заснований на дії сили інерції;
- Циклони, батарейні циклони, принцип роботи яких заснований на дії відцентрової сили.

Інерційні методи застосовуються в тих випадках, коли газовий потік містить шкідливі домішки у вигляді пилу (розміри частинок 5-50 мкм), туману і диму (розміри частинок 0,1 - 5мкм). Ці методи засновані на осадженні твердих частинок і дрібних крапель туману на поверхні пиловловлювачів і фільтруючих елементів. З цією метою використовують пиловловлювачі і фільтри різної конструкції [4-6].

Інерційні методи очищення газових потоків від шкідливих домішок широко поширені на збагачувальних фабриках, металургійних заводах, теплових електростанціях, що спалюють вугілля і мазут, на підприємствах деревообробки, в шинної промисловості і у виробництві гумових технічних виробів.

При розмірах часток пилу 25-50 мкм і високих їх концентраціях в газовому потоці (більше 50 г/м^3) зазвичай використовують пилоосаджувальні камери і інерційні пиловловлювачі.

Пилоосаджувальна камери в більшості випадків застосовуються для попереднього очищення сильно забруднених газових потоків від великих часток пилу. Запилений газ в пилоосаджувальні камері має швидкість руху 0,2-1,5 м/с. При цьому частинки пилу, що мають розміри більше 50 мкм, осідають на полицях і стінках камери, а очищений газ викидається в атмосферу або подається на наступну стадію очищення - від більш дрібних частинок.

Після утворення шару пилу певної товщини на стінках і полицях даний пристрій вмикається вібраційне пристрій, і пил падає вниз.

Ступінь очищення запиленого газу в пилоосаджувальних камерах не перевищує 40 - 50% .У інерційних пиловловлювача швидкість запиленого газу на вході в апарат становить 5-15 м/с. Принцип дії інерційних пиловловлювачів полягає в наступному.

При збільшенні швидкості руху запиленого газу на частинки пилу одночасно діють сили тяжіння і інерційні сили. Якщо різко змінити напрямок руху газу, то частинки пилу будуть продовжувати свій рух по інерції, що призведе до виділення пилу з газового потоку.

Для запиленого газового потоку з розмірами частинок 25-30 мкм ступінь очищення досягає 65 - 80%. Такі апарати знаходять застосування в металургійній промисловості для первинного очищення газових потоків від пилу.

Перевага при виборі циклону для очищення від пилу надається відцентровим циклонам, які виконують роль пиловловлюючого апарату. Ефективність уловлювання пилу в циклонах підвищується зі зменшенням діаметра корпусу, але при цьому знижується їхня пропускна здатність. Для забезпечення відповідної продуктивності пневмотранспортної установки невеликі циклони групують в батарею, коефіцієнт пиловловлення якої становить 0,76 - 0,85 і дещо підвищується зі збільшенням вхідної швидкості (з 11 до 23 м/с). Використання замість циклонів вихрових пиловловлювачів забезпечує вловлювання частинок пилу розміром 5 - 7 мкм.

Циклони рекомендується використовувати для попереднього очищення газів і встановлювати перед високоефективними апаратами (наприклад, фільтрами або електрофільтрами) очищення.

Основні елементи циклонів - корпус, вихлопна труба, бункер. Газ надходить у верхню частину корпусу через вхідний патрубок, приварений до корпусу тангенційно. Уловлювання пилу відбувається під дією відцентрової сили, що виникає при русі газу між корпусом і вихлопною трубою. Вловлений пил зсипається в бункер, а очищений газ викидається через вихлопну трубу.

Також поширені «сухі» пористі фільтри і електрофільтри.

Для очищення запилених газів все більшого поширення набуває на останніх щаблях суха очистка рукавними фільтрами. Ступінь очищення газів в них при дотриманні правил технічної експлуатації досягає 99,9%. В якості фільтрувальних матеріалів застосовують

тканини з природних волокон (вовняні, рідко бавовняні), з синтетичних (нітронове, лавсанові, поліпропіленові та інші), а також склотканини. Для фільтрувальних тканин найхарактерніше саржеве переплетення. Застосовують також неткані матеріали - фетри.

Фільтри рукавні складаються з корпусу розділеною рукавною плитою, фільтрувальних елементів, клапанних секцій з роздавальними трубами для забезпечення регенерації рукавів імпульсами стисненого повітря.

В процесі фільтрації запилений газ проходить через тканину закритих знизу рукавів всередину, виходить через верхній колектор і видаляється з апарату. Кожен рукав в фільтрі натягнутий на жорсткий каркас і закріплений на верхній решітці (плиті) [3,7-8].

- переваги рукавних фільтрів - висока ступінь очищення газу від тонкодисперсного пилу;
- недоліки - порівняно швидкий знос або закупорка тканини і непридатність для очищення гарячих і вологих газів.

Метод електроосадження. Електрофільтри призначені для уловлювання пилу з розмірами часток до 0,1 мкм з повітря і газів різного хімічного складу, вологості і температури. Принип роботи (уловлювання пилу в електричному полі) полягає в наступному: зміст процесу електростатичного очищення газів ґрунтується на іонізації газу, тобто розщепленні його молекул на позитивно і негативно заряджені іони. Запилені гази пропускають через неоднорідне електричне поле, яке утворюється між осаджувальним 2 і коронувальним 3 електродами. До електрода, розташованого в ізоляторі 4 підводиться випрямлений електричний струм при напрузі 30 - 60 кВ. Осаджувальний електрод звичайно заземлюють і підключають до позитивного полюсу вирівнювача.

Процес ловлення пилу в електричному полі складається з таких підпроцесів: зарядження завислих у газі частинок; руху заряджених частинок до електродів; осадження частинок на електродах і видалення частинок з електродів.

Мокре очищення. При очищенні газів від частинок пилу і переробці газоподібних відходів успішно застосовують мокре пиловловлювання, суху і подальшу мокру очистку. Розвинена поверхня контакту фаз сприяє збільшенню ефективності пиловловлювання. У промисловості використовують мокрі пиловловлювачі крапельного, плівкового і барботажного типів. Конструктивно апарати можуть бути порожніми, тарілчастими, механічного та ударно - інерційного дії, а також швидкісного типу (трубки Вентурі і інші інжектори).

Скрубери (газопромивачі). При об'ємно - рідинному способі потік запиленого газу пропускають через певний обсяг рідини. Для цієї мети використовують пінні пиловловлювачі з провальними тарілками або тарілчасті скрубери, ефективність яких може досягати 90 - 95%.

Для очищення або знешкодження газоподібних відходів або технологічних газів з метою вилучення з них супутніх (корисних) газоподібних компонентів широко використовують метод абсорбції. Абсорбція заснована на безпосередній взаємодії газів з рідинами. Виділяють фізичну абсорбцію, засновану на розчиненні газу в рідині, і хемосорбцію, в основі якої лежить хімічна реакція між газом і рідким поглиначем [8].

Постановка завдання. Для вирішення питання екологічної безпеки підприємств зернопереробної спеціалізації, елеваторних комплексів слід проводити інвентаризацію джерел викидів, моніторинг стану атмосферного повітря, заміри викидів газопилової суміші, та обладнувати виробництва АС (аспіраційними системами) та ГОУ згідно виробничих потреб та потужностей, орієнтуючись на світовий досвід у галузі природоохоронних технологій, використовуючи для модернізації новітня технологічні розробки очисного обладнання.

Виклад основного матеріалу.

Технологічні процеси супроводжуються виділенням різних шкідливих елементів в виробничі приміщення – надлишкових теплоти, вологи, шкідливих газів і пилу. Це несприятливо позначається на мікрокліматі і санітарно-гігієнічному стані цехів підприємств,

сприяючи виникненню небезпечних і шкідливих виробничих факторів, що впливають на працюючих. Підвищені вологість і запиленість погіршують режим експлуатації і скорочують термін служби технологічного обладнання та будівельних конструкцій. Серйозною проблемою на підприємствах галузі є висока вибухо- і пожежонебезпека, причинами якої стають значні неорганізовані пилонадходження органічних горючих речовин і виникнення пожежо-вибухонебезпечних пилоповітряних сумішей.

Крім пилу повітря забруднюється шкідливими газами – оксидом вуглецю, діоксидом сірки, діоксидом вуглецю, що виділяються в приміщеннях зерносушарок. Значна кількість діоксиду вуглецю утворюється в приміщеннях складів зерна. Однак основний шкідливістю, що виділяється при переробці зерна, залишається органічний пил використовуваної сировини, проміжних продуктів і готової продукції.

Процеси навантаження, вивантаження і транспортування сировини, його обробка, складування і зберігання супроводжуються значними пиловиділенням, що призводить при недостатньо ефективної вентиляції, до запиленості повітря у виробничих приміщеннях, набагато перевищує безпечні концентрації.

На зерносушильних комплексах зерно проходить технологічні операції приймання, очищення, сушіння, відпустки, освіження, піддаючись багаторазовому переміщенню транспортними механізмами, самопливом по точках, в системах пневмотранспорту. Тертя зерна об стінки обладнання і трубопроводів призводить до стирання оболонок зерна і виникнення органічного і мінерального пилу, що утворюється через засмічення зерна при збиранні та транспортуванні різними неорганічними домішками. Очищення зерна на сепараторах знижує його початкову запиленість, водночас, оскільки частина зернового пилу знаходиться в зв'язаному стані, залягаючи в борозенках і оболонках зерен, пиловиділення має місце на кожному етапі технологічного процесу. Значні пиловиділення спостерігаються під час продування повітрям шару зерна при активному вентилюванні і сушінні.

Збільшенню пилонадходження в атмосферу сприяють як недостатня герметизація устаткування, так і неефективна робота аспіраційних систем (АС), газоочисних установок (ГОУ) і вентиляції в цілому. Навіть при задовільній роботі АС та ГОУ в повітрі присутній пил, що походить від процесу отримання кінцевого продукту. Гранично допустима концентрація (ГДК) зернового пилу - 4 мг/м³, борошняного пилу - 6 мг/м³. У працівників, особливо за недотримання вимог охорони праці, можуть виникати професійні захворювання верхніх та нижніх дихальних шляхів через перебування в зоні запилення протягом робочого часу та багаторічної трудової діяльності.

В окремих зонах виробничих приміщень і при аварійних ситуаціях концентрація пилу в повітрі може перевищувати нормативні значення і підвищуватися до вибухонебезпечних концентрацій. Пил, зважений в повітрі, поступово осідає на будівельних конструкціях та технологічному обладнанні, утворюючи нещільний, легко зрушуваний з місця рухом повітря шар осілого пилу. Вторинне запилення, викликане здійманням осілого пилу при підвищеній рухливості повітря різко збільшує кількість пилу в повітрі і може призвести до вибуху.

Кількісний та якісний склад пилових часточок залежить від сировини, що переробляється, його вологості, типу технологічного устаткування і його технічного стану, а також від ефективності роботи вентиляційних систем.

Пил зернопереробних підприємств та елеваторів являє пожежо- та вибухонебезпечну суміш; витаючий в повітрі – вибухонебезпечний, осілий на будівельні конструкції та обладнання – пожежонебезпечний. Вибухонебезпечні концентрації можуть утворюватися в технологічному і транспортному устаткуванні, в силосах та бункерах, в трактах АС та ГОУ і пневмотранспорту, в пиловловлюючому обладнанні.

Пиловий вибух є найстрашнішим наслідком присутності зернового пилу, крім пожеж, займань та задимлень. Це практично миттєве загорання дрібних часточок зернового пилу, котрий через органічне походження має високий вміст вуглецевих сполук, що і сприяє миттєвій реакції горіння, яка у випадку різкого підвищення температури та тиску виражається вибухом. [2,7].

Висновки.

Таким чином, в сільському господарстві галузь зернопереробки має значну роль та впливає на довкілля, оскільки використовує машини, механізми та технологічне обладнання, в процесі використання яких для сушіння, вентилювання, очищення зерна спричиняється утворення великих об'ємів органічного пилу.

Організаційно-технічні рішення для зниження виділення пилу передбачають обладнання ГОУ та АС, важливим елементом таких є батарейні циклони, які дозволяють уловлювати та накопичувати зерновий пил.

Список літератури

1. https://esu.com.ua/article-15961
2. Джирма С.О., Яцун В.В., Дарієнко В.В., Горпинченко О.В. Технологія зведення збірних і монолітних залізобетонних елеваторів: навчальний посібник. Кропивницький: Видавець Лисенко В.Ф.. – 2022. – 108 с.
3. http://agro-business.com.ua/agro/ekonomichnyi-hektar/item/24134-ahrosektor-1991-2021-u-tsyfrakh-i-faktakh.html
4. Екологічна безпека [Текст]: підручник / В. М. Шмандій, М. О. Клименко, Ю. С. Голік та ін. – Херсон: Олді-плюс, 2013. – 366 с.
5. ДБН В.2.6-221:2021 https://dnaop.com/html/60659/doc94.2.2-20_2012
6. http://elib.tsatu.edu.ua/dep/mtf/ophv_21/page5.html
7. https://agroelita.info/scho-take-suchasnyj-zernovyj-elevator/
8. https://elevator.com.ua/ru/blog/dvizhenie-zerna-na-elevatore-i-avtomatizaciya-ego-ucheta-s-pomoschyu-programmnykh-resheniy-ges

УДК 504.54

М. Колеснік, студ. гр. ЕО-21М
Центральноукраїнський національний технічний університет

ЗАХОДИ ЕКОЛОГІЧНОЇ БЕЗПЕКИ ПРИ ПОВОДЖЕННІ З ЕКСПЛУАТАЦІЙНИМИ ВІДХОДАМИ АВТОТРАНСПОРТУ

У статті висвітлено необхідність повторного використання відходів автотранспорту та розвиток технологій для їх безпечної утилізації. Швидке зростання кількості автотранспорту сприяє збільшенню рівня забруднення довкілля внаслідок виділення шкідливих речовин у навколишнє середовище.
автотранспорт, утилізація, екологістика, відходи, ресурозбереження, охорона нпс

Актуальність теми. Актуальним є впровадження новітніх ефективних технологій утилізації відходів автотранспорту, які мають мінімальний негативний вплив на навколишнє середовище та економічно доступних. Щороку кількість автомобілів як у світі так і в нашій країні збільшується, тому проблема утилізації відходів автотранспорту, стає все більш актуальною. Відходи, які утворюються при експлуатації автотранспортної техніки, характеризуються великою неоднорідністю за складом і динамікою утворення, та при неправильному поводженні з ними можуть завдавати значну шкоду атмосфері, земельним і водним ресурсам.

Аналіз останніх досліджень і публікацій. Щороку експлуатується величезна кількість автомобілів близько 1 млрд легкових та більше 335млн одиниць комерційного автотранспорту(а згідно з прогнозами міжнародної асоціації автовиробників (OICA) вона може збільшитись до 3 млрд одиниць лише за 10 років, то ж під час їх використання виникає відповідна кількість відпрацьованих матеріалів серед яких є такі небезпечні речовини як

фільтри для очищення масла, відпрацьоване масло,автомобільні шини,різні механічні домішки та ін.

Постановка завдання. Однією з важливих екологічних проблем на території України залишається проблема поводження з відпрацьованими автомобільними масляними фільтрами через недосконаліть існуючої системи їх збору та відсутність ефективних методів утилізації. У зв'язку з відсутністю в більшості міст пунктів збору і підприємств з утилізації відпрацьованих фільтрів, їх, як правило, складають разом з відходами металів або твердими побутовими відходами, а частіше їх просто викидають. Зміни в законодавстві полегшують роботу з відходами виробництва і споживання, а якщо бути точніше вдосконалюють її.

Виклад основного матеріалу. Враховуючи утворення відходів АТЗ масштаби яких вражають необхідно створювати та вживати заходи для організації системи поводження з відпрацьованими матеріалами,з метою вилучення ресурсоцінних складових.[1].

Перше що потрібно робити з відходами АТЗ це їх збір,в багатьох країнах запроваджують санкції та стимули для громадян та підприємств, що б заохотити їх до правильного поводження та утилізації таких відходів.

Зважаючи на подану вище інформацію потрібно запобігти потраплянню мільйонам тон відходів автотранспорту на звалище,а удостовіритись їх повній утилізації.Адже задля їх виготовлення було використано чималу кількість цінних матеріалів(чорні та кольорові метали, пластмаси та гумові вироби, скло та кераміка, дерево та картон, текстильні матеріали то що.)Тому відходи автотранспорту повинні ставати джерелами вторинних матеріальних ресурсів.[2].

Зважаючи на те що є данні чисельності населення та рівня автомобілізації різних країн світу, можна визначити кількість автомобілів, а потім і кількість відпрацьованих автомобільних фільтрів, що утворюються кожного року.Таблиця 1.1

Таблиця 1.1 – Масштаби утворення відпрацьованих фільтрів

п	Масштаб утворення	Кількість жителів,чол.	Рівень автомібілізації, шт/1000 чол.	Кількість машин,шт.	Фільтрів за рік шт/рік
1	Світ	7 432 663 275	135	1 003 409 542	2 006 819 084
2	Країни західної Європи:	190 827 090	572	109 153 095	218 306 190
	Австрія	8 219 743	529	4 348 244	8 696 488
	Бельгія	10 438 353	489	5 104 355	10 208 710
	Ліхтенштейн	36 713	750	27 534	55 068
	Люксембург	509 074	665	338 534	677 068
	Монако	30 510	732	22 333	44 466
	Нідерланди	17 630 632	466	7 796 475	15 592 950
	Німеччина	81 305 856	517	42 035 128	84 070 256
	Франція	65 630 092	481	31 568 363	63 136 726
	Швейцарія	7 925 517	521	4 129 194	8 258 388
3	Сша	325 197 000	423	137 558 331	275 116 662
4	Україна	2 414 900	202	8 567 810	17 135 620

Одне з найголовніших завдань утилізації відходів АТЗ це зменшення ресурсів для виготовлення нових деталей, на сьогодні коефіцієнт утилізації яких становить 95%[3-5].

Найбільш розповсюдженими відходами автотранспорту є відпрацьовані мастильні матеріали (ВММ), вони забруднюють атмосферу, воду та ґрунт.

“Розробка інструментів, апаратів, контейнерів, способів, систем та пристроїв які підтвердженні патентами на винахід належать наступним світовим науковцям: Б.В. Елвард, Д.Х. Лютц.”

“Б. Кнол, Д. Крепс – запропонували пристрої, інструмент та воронку для зливу відпрацьованого масла з автомобільного фільтра” [6-9].

“Винахід Г.Д. Макрае забезпечує вилучення та регенерацію відпрацьованих масел, який являю собою систему обробки масляних фільтрів, а також матеріалів, забруднених нафтопродуктами” [10].

“Т. Морі, С. Моримото та інші запропонували цілий комплекс способів та обладнання з рециклінгу непридатних складових АТЗ” [11].

Найдосконаліший процес рециклінгу масляних фільтрів, розроблений Г.А Колтуновим.Він містить в собі розрізання корпусу масляного фільтра та відділення кришки від корпусу,та відпрацьованих деталей масляного фільтра,що далі ідуть на сортування за придатністю для їх подальшого використання, потім передаються на обробку з механічною та/або гідравлічною дією направленою на них, після вище описаного,вони поступають на контроль технічних параметрів з подальшою передачею придатних деталей на збирання нових масляних фільтрів, а непридатних – на подальше сортування та утилізацію[12].

Згідно з проведеного огляду стану розв'язання проблем впливу відходів АТЗ, подальших досліджень найбільш потребує питання утилізації найнебезпечнішої складової відпрацьованих автомобільних масляних фільтрів – забрудненого паперу, який не тільки містить в собі цінні компоненти для повторного використання, а й також загрожує довкіллю.

Найчастіше відходи з пластмасовмістних матеріалів переробляють у вторинну полімерну сировину для повторного використання при виробництві різноманітних виробів, піроліз з одержанням вуглеводневої сировини для енергетичного та хімічного застосування, спалювання разом з твердими побутовими відходами з отриманням теплової та електричної енергії або ж звичайне захоронення на полігонах.Найрозумніший спосіб поводження з утилізованими пластмасовими деталями автомобілів є їх застосування в подрібненому вигляді в складі полімерних композицій того ж складу для виробництва аналогічних деталей[13].

Зі зростанням автопарку у світі, прямо пропорційно зростає і накопичення зношених автомобільних шин.Дані Європейської Асоціації по вторинній переробці шин вказують на те що загальна вага непереробленних шин становить: у Європі-2,5 млн тонн; у США-2,8 млн тонн; у Японії-1,0 млн тонн.

Важливість перероблення зношених автомобільних шин і гумотехнічних виробів, має велике екологічне та економічне значення для всіх країн світу, зношені шини, складується в місцях їх експлуатації (в автогосподарствах, на аеродромах, промислових і сільськогосподарських підприємствах, гірничозбагачувальних комбінатах і т.д.).Шини, що залишаються на звалищах, тривалий час забруднюють навколишнє середовище унаслідок високої стійкості до дії зовнішніх чинників (сонячного світла, кисню, озону, мікробіологічних дій).

Місця, які слугують скупченням,є сприятливим середовищем для розмноження різних тварин, які є носіями різних захворювань, крім того, вони є пожежонебезпечними та при горінні негативно спливають на навколишнє середовище

У табл. 1.2 наводяться дані про кількість утилізованих шин і способи їх вторинного використання у ряді країн Європи, США і Японії [14]

Таблиця 1.2 - Кількість утилізованих шин в Європі, США і Японії й способи їх перероблювання.

Країна	Об'єм утворення тис. т	Вивезено на звалище %	Отримання енергії %	Відтворення протектору %	Отримання гумової крихти %	Експорт %	інше %
Німеччина	550	2	38	18	15	18	9
Великобританія	450	67	9	18	6	-	-
Франція	425	52	10	13	6	19	-
Італія	330	53	14	27	-	6	-
Сша	2800	59	22	9	9	3	1
Японія	840	8	43	9	12	25	3

Висновки. В провідних країнах світу створена спеціальна галузь промисловості з перероблення та утилізації автомобільних відходів, яка займається проблемами вторинної переробки матеріалів, а також повторного залучення до виробництва деталей бувших у використанні, реалізуючи їх за зниженими цінами, хоча ці автокомпоненнти часто не поступаються новим за якістю та ресурсомісткістю. Варто зазначити, що навіть у найсучасніших повних ланцюжках утилізації відходів автотранспорту економічно вигідні далеко не всі їх ланки. На сьогодні в Німеччині працюють близько 40 шредерів, вони переробляють на рік 1,5 млн. т. матеріалів,а з законодавчого боку у Німеччині працює Федеральний Закон «Про економічний рециклінг Німеччини», що спрямований на створення умов, при яких автовиробники, вже на стадії проектування і виготовлення автомобілів мусять домагатися зменшення кількості майбутніх відходів.

Список літератури

1. Електронний архів (Розділ 3 Поводження з небезпечними відходами) [Електронний ресурс]. – Режим доступу: http://golos.kievcity.gov.ua/files/2014/6/19/rozdil-3.pdf/. – Назва з екрану. Авторециклинг в странах Балтии // Рециклинг отходов. – № 2. – 2007. –С. 19-21.
2. Утилизация авто: опыт Евросоюза [Электронный ресурс]. / Автоцентр. – 2013, 18 августа. —Режим доступа: https://www.autocentre.ua/avtopravo/avtobiznes/utilizatsiya-avto-opytevrosoyuza-265694.html
3. Авторециклинг в странах Балтии // Рециклинг отходов. – № 2. – 2007. –С. 19-21. 13. Виллер С. Опыт США. ЦЕЛЬ -100 % переработка автомобиля + прибыль //Авто-грин. – № 1. – 2005. – С. 25-26.
4. Car Recycling Business in Japan. JETRO Japan Economic Report. – June-July 2006. –13 pp.
5. Рециклинг отходов — прорывной проект XXI века отходов [Электронный ресурс]. – М., 2008. – Режим доступа: http://www.innosfera.org/node/727
6. Патент US 78499668 B1, США, МПК: F 16 N 33/00. Воронка для зливу масла з фільтра / Девід Крепс. 2010.
7. Патент US 005291921 A, США, МПК: B 65 B 1/04. Дренажна платформа для зливу залишкового масла, збору та подальшої утилізації / Роберт Дивайн. 1994.
8. Патент US 005611377 A, США, МПК: B 65 B 1/04. Коробка – осушувач масляних фільтрів / Джон Р. Макгваєр. 1997.
9. Патент US 005884676 A, США, МПК: B 65 B 3/04. Коробка для зливу відпрацьованого масла / Гарі Сейдж. 1999.
10. Патент WO 2016068735 A1, Польща, МПК: B 03 C 1/14. Спосіб утилізації масляних і паливних фільтрів після вікористання / Ізабелла Богацька, Станіслав Левандовськи, Бартош Щитовський. 2016.
11. Патент на полезную модель 2163847 РФ, МПК7: B09B3. Способ переработки масляных фильтров и устройство для его реализации [Текст] / Бабенко Ю.И., Власов В.Н.; патентообладатель Бабенко Юрий Иванович, Власов Владимир Николаевич; № 2000103399/06; заявл.14.02.2000; опубл.10.03.2001.

12. Патент на корисну модель 20424 Украина, МПК: B09B 3/00 Процес рециклінгу масляних фільтрів, виконаний закаткою корпусу на кришку/ Колтунов Г.А.; заявник і власник патенту Київ. український інст. пром. власності. – № 200608913; заявл. 10.08.206; опубл. 15.01.2007, Бюл. № 1, 2007.
13. Мельникова Д. А. Об опыте решения проблемы твёрдых бытовых отходов - интернетжурнал "Технологии техносферной безопасности" (http://ipb.mos.ru/ttb). – Выпуск № 2 (43) –2012.

УДК 504.54

Шаміль Каміл Огли Гюльвердієв, студ. гр. ЕО-21М
Центральноукраїнський національний технічний університет

ВИЗНАЧЕННЯ ПЛИВУ БОБОВИХ ТРАВ НА ПОКРАЩЕННЯ АГРОЕКОЛОГІЧНИХ ПОКАЗНИКІВ ГРУНТУ

У статті висвітлено необхідність вирощування у сівозмінні бобових багаторічних трав задля покращення агроекологічного стану грунтів, а саме підвищення гумусу вмісту поживних речовин, симбіотичного азоту та зниження антропогенного тиску на грунтове середовище.
люцерна посівна, галега східна, гумус, азот, фосфор, калій

Актуальність теми. На сьогоднішній день, переважна більшість сільгоспвиробників орієнтована на вирошування економічно вигідних сільськогосподарських культур, які вирощують переважно в монокультурі, нехтуючи сівозміною.

Аналізуючи процеси, які проходять, в наслідок вирощування монокультур, необхідно відзначити, що відбувається зниження родючості грунтів, а саме зниження гумусу, поживних елементів таких, як азот, фосфор, калій, погіршення мікробіологічного стану грунту, зниження продуктивної вологи, створюються умови, що уможливлють розвиток різноманітних ерозійних процесів, погіршення структури грунту, зниження кількості агрономічноцінних агрегатів, накопичення токсинів та полютантів, зниження якості та урожайності сільськогосодарської продукції.

Тому, необхідно здійснювати відновлення аргоекологічних показників грунту, на основі біологічних процесів, а саме вирощування в сівозміні багаторічних бобових трав, що дозволить природнім шляхом накопичити органічну речовину у грунті, у вигляді післяжнивних та кореневих решток, накопичить симбіотичний азот та сприятиве відновленю грунтової мікрофлори, а також структуризації грунту.

Перспективною рослиною, в даному напрямку, є багаторічна бобова культура галега східна, яка має високі едифікаторні властивості, має високу врожайність, сімбіотизує біологічний азот в межах 108 кг/га за один укіс, атакож сприяє створенню накопиченню вологи в грунті, органічних решток та підвищення агроекологічних показників і поживних речовин [1].

Аналіз останніх досліджень і публікацій. В альтернативних системах землеробства, які виникли внаслідок екологізація галузі землеробства, пріоритетними напрямками розвитку є внесення органічних добрив з використанням максимально можливих ресурсів, а саме гною, компостів, післяжнивних решткив, зелена маса сидератів, а також ґрунтозахисні системи обробітку ґрунтів та екологічно обґрунтовані системи захисту рослин від шкодочинних організмів. В разі застосування екологічних систем землеробства виникає симбіоз природнього та антропогеного фактору, що робить галузь сільськогосподарського виробництва прийнятною і для людини, і для природи. Головними завданями екологічного землеробства можна класифікувати, як виробництво екологічно безпечної, економічно обґрунтованої сільськогосподарської продукції, а також збереження та підвищення

родючості грунту [2].

За рахунок інтенсифікації землеробства останні десятиліття спостерігається обтяження галузь землеробства екологічними проблемами, тому що основний засіб виробництва - грунт - має природне походження, та зарахунок антропогенного тиску, деградує, що проявляється у втраті ґрунтом структури, переущільненні, погіршенні водопроникності, накопиченні токсинів [3].

До основних законів землеробства, які сприятимуть покращенню та відновленню грунтового середовища, млжна віднести наступні: незамінності й рівнозначності факторів життя; мінімуму, оптимуму й максимуму; сукупної дії та взаємодії факторів життя; повернення поживних речовин у ґрунт; плодозміни; критичних періодів.

Про ведення біологічного або альтернативного землеробства, відзначав у своїх роботах Г. Кант, який і вважається його основоположником [4].

1) Переведення азоту повітря в рослинний білок здійснюється за участю бобових культур, специфічних бактерій ґрунту або ціанофітів, а не шляхом хемосинтезу азотних добрив;

2) Розпушення й оструктурення ґрунту здійснюється коренями рослин, дрібними ґрунтовими тваринами і мікроорганізмами, а не за допомогою знарядь та механізмів за великих втрат енергії;

3) Боротьба з бур'янами, хворобами та шкідниками ведеться в основному біологічним шляхом – правильним чергуванням культур в сівозміні, вибором видів і сортів відповідно до конкретних умов, активування природних ворогів шкідників, а не за рахунок застосування хімічних засобів захисту рослин (біоцидів).

Реалізація першого та другого принципу можлива за рахунок використання у сівозміні бобових культур, які завдяки своїм біологічним та кологічним особливостям здатні накопичувати біологічний азот, завдяки симбіозу з бульбочковими бактеріями, мають велику кількість післяжнивних решток і кореневих решток, чим створють сприятливі умови для накопичення гумусу та розвитку позитивної грунтової мікрофлори, структуризація та розпушення грунту здійснюється мичкуватою кореневою системою культур [5].

До бобових культур, можна віднести наступні культури: соя, горох, квасоля, боби, нут, арахіс, сочевиця, люцерна, еспарцет, буркун, галега східна, конюшина, люпин, вика тощо [6]

Постановка завдання. В наслідок, ведення сучасного сільського господарсва, відбувається порушення грунтового середовища: висока розораність, дисбаланс біохімічних речовин і енергії в агроекосистемах, неефективні протиерозійні системи, які зумовлюють зниження родючості гунтів та й порушення екологічної стійкості природнього навколишнього середовища, втрата продуктивних та якісних показників сільськогосподарських угідь.

Тому, головним завданням досліджень, є відновлення родючості грунтів із застосуванням максимально природніх методів та заходів, а саме вирощування багаторічних трав, що дозволять підвищити родючість грунту поліпшити агрономічноцінні показники, знизити ерозію та дозволить вирощувати екологічно безпечну сільськогосподарську продукцію.

Виклад основного матеріалу. Грунт – це система, яка задовольняє потреби рослиних організмів в поживних елементах і воді, є середовищем існування, та забезпечує кореневу систему необхідною кількістю повітря й тепла для повноцінного росту та розвитку. Також, від якісних показників грунту залежить урожайність сільськогосподарських культур.

В результаті наших досліджень, нами було встановлено, як люцерна посівна та галега східна вплинули на агроекологічні показники грунту.

Одним із вагомих показників, що відповідає за родючість грунту є гумус.

Як показали наші дослідженя, вміст гумусу залежав від культури, що вирощувалася на користь галеги східної (рис. 1).

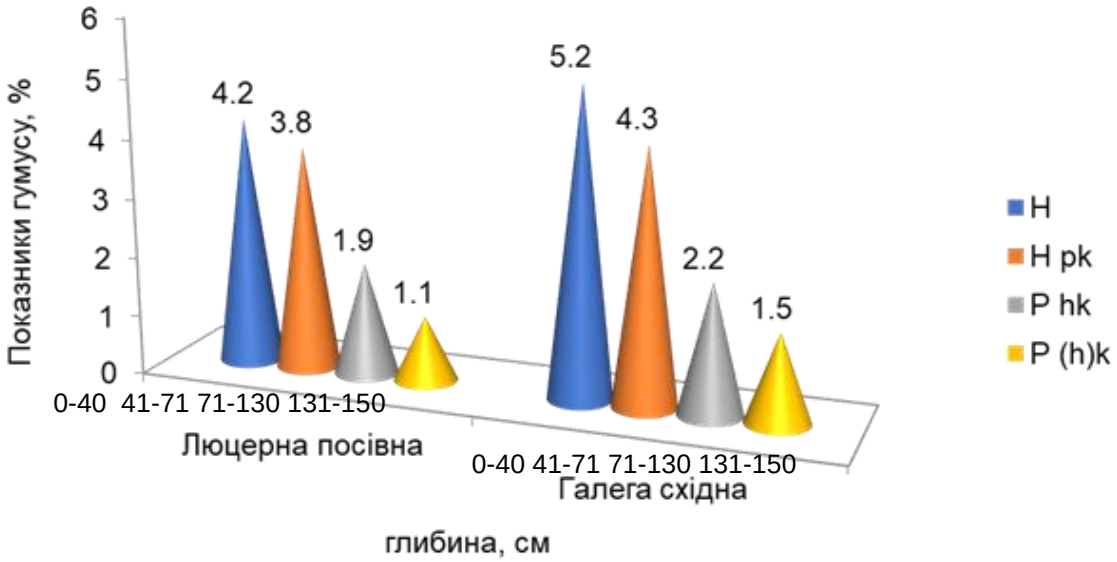


Рис.1-Показники гумусу залежно від люцерни посівної та галеги східної (в середньому за 2019-2021 р.), %

Встановлено, що за вирощування люцерни посівної показники гумусу були нижчими у порівнянні до варіантів, де вирощували галегу східну по горизонтам в залежності від глибини. Так, на глибині 0-40 см вміст гумусу за вирощування люцерни склало в межах 4,2%, тоді як за вирощування галеги східної досліджуваний показник склав 5,2%.

Зі збільшенням глибини горизонту спостерігається зниження досліджуваного параметру за обох культур, але перевага була за галегою.

В результаті аналізу данних було встановлено, що на глибині 41-70 см вміст гумусу склав 3,8 % (люцерна посівна) та 4,3% (галега східна).

Що стосується нижніх горизинтів показники були майже однаковими. Встановлено, що на глибині 71-130 см досліджуваний показник склав відповідно 1,9 %(люцерна посівна) та 2,2% (галега східна). Аналогічна тенденція була зафіксована і на глибині 131-150 см, що відповідно відповідно 1,1 % (люцерна посівна) та 1,5% (галега східна).

Аналіз одержаних данних, виявив, що за вирощування галеги східної відбувається відновлення гумусуваного шару у порівнянні до ділянок за вирощування люцерни посівної, особливо помітний ефект у верхньому шарі грунту.

В наших дослідженнях ми встановили, як впливали культури, що вирощували, а саме люцерна посівна та галега східна, на показник pH грунтового середовища (рис. 2)

Для більшості сільськогосподарських культур, оптимальним є pH близьке до нейтрального. Тоді відбувається максимально позитивний ріст та розвиток кореневої системи, а також повноцінне забезпечення процесу живлення.

Обидві досліджувані культури відносяться до бобових культур, які як відомо за рахунок своєї стержневої кореневої системи піднімають в верхні шари грунту мікро- та макроелементи, що вподальшому сприятиво відображається на стані самого грунту та наступних культур у сівозміні.

Наші дослідження показали, що за рахунок вирощування, що люцерни посівної, що галеги східної вглиб по горизонтах зміновалися.

Так, за вирощування люцерни посівної було зафіксовано слідуючі показники: у вехньому горизонті на глибині від 0-40 см показник pH склав 6,5, а на глибині 41-70 см pH дорівнював 6,9, тоді як на нижньому горизонті P_{hk} склав pH 7,0 і $P_{(h)k}$ – pH 7,1. Встановлено, що відбувалося збільшення у бік лужності, що буде негативно впливати на засвоєння рослинами наступних елементів таких як фосфор, залізо, марганець, мідь, цинк, бор за

рахунок утворення нерозчинних гідроксидів.

За вирощування галеги східної, pH грунтового середовища був на оптимальному рівні по горизонтах, а саме: у вехньому горизонті на глибині від 0-40 см показник pH був в межах 5,6, на глибині 41-70 см pH склав 6,1, тоді як у нижніх горизонтах Phk склав pH 6,5 і P(h)к – pH 6,8.

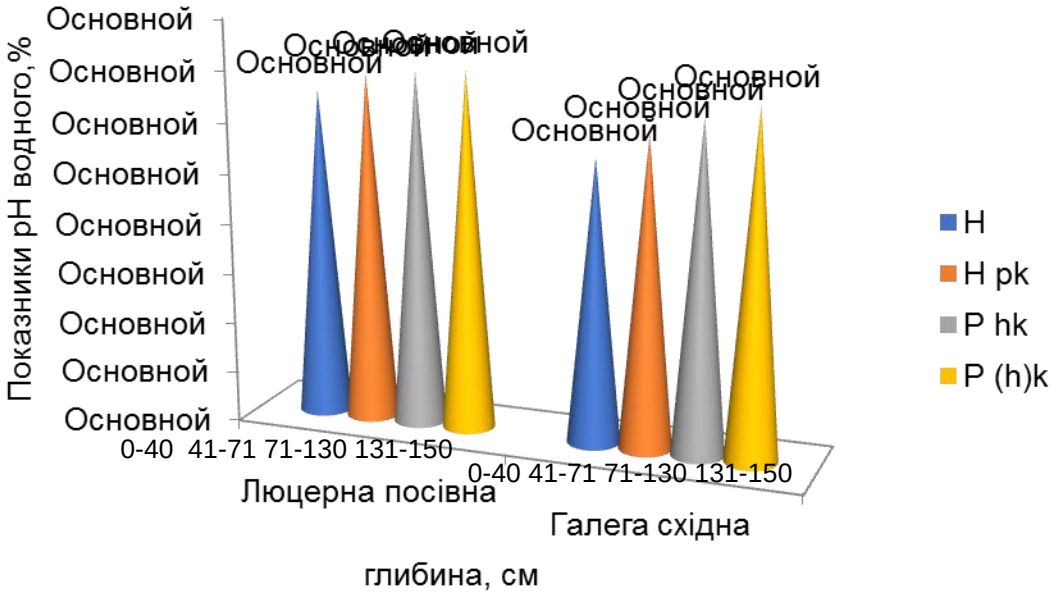
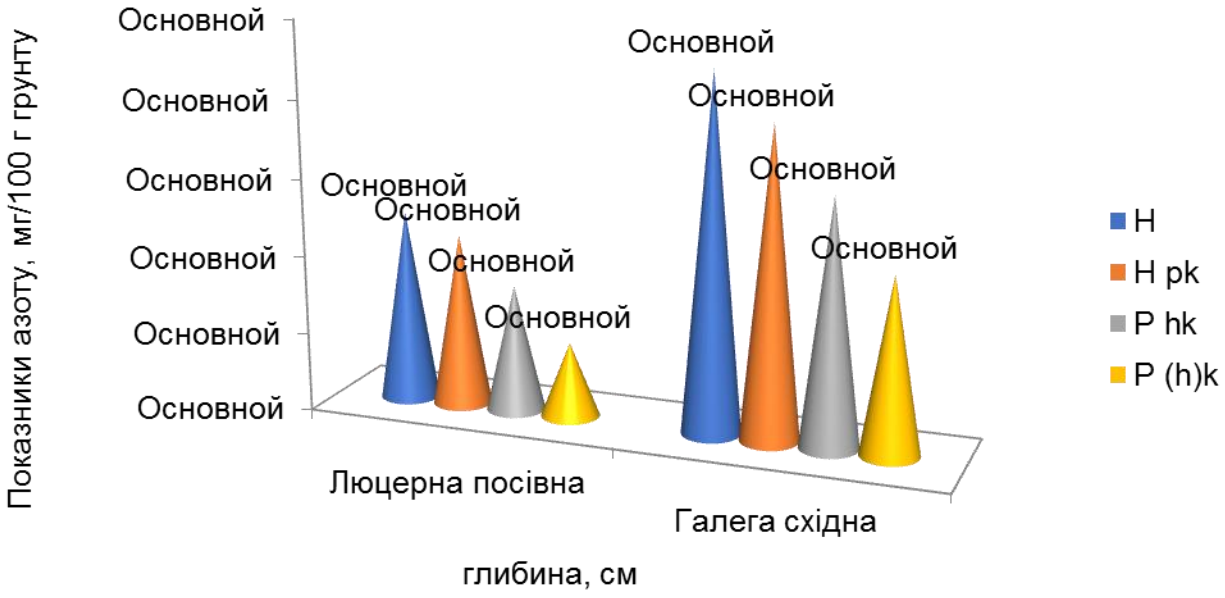


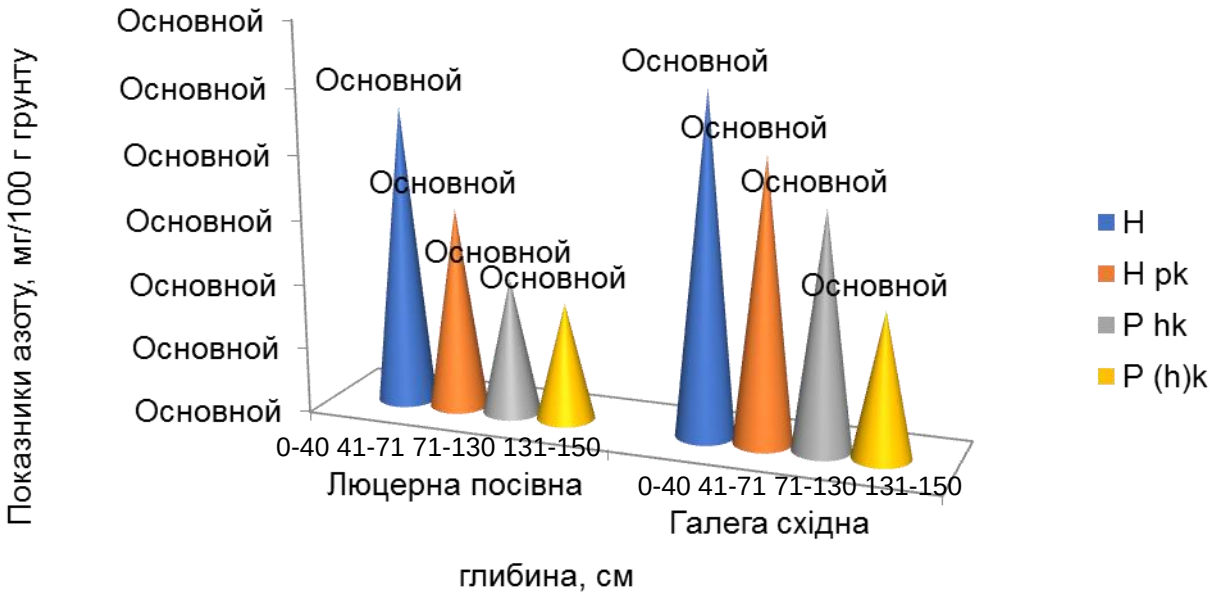
Рис 2 - Показники pH водного залежно від люцерни посівної та галеги східної (в середньому за 2019-2021 р.)

Отже, вирощування галеги східної сприяє знаходитися pH грунтового середовища на оптимальному рівні.

Також, в наших дослідженнях, ми звернули увагу, як впливало вирощування люцерни посівної та галеги східної на накопичення азоту (рис. 3).



а - лужногідролізований



б - загальний азот

Рис.3 - Показники азоту залежно від люцерни посівної та галеги східної (в середньому за 2019-2021 р.), мг/100г ґрунту

Як показали результати досліджень, кількісний показник лужногідролізованого та загального азоту зменшувався по досліджуваним горизонтам ґрунту, але на ділянках, де вирощувалася галега східна у порівнянні до ділянок , де вирощувалася люцерна посівна показники були вищими.

На глибині 0-40 см на варіанах, де вирощували люцерну посівну лужногідролізований азот склав 12,3 мг/100г ґрунту, а на варіантах за галеги східної було вищим і склало 22,7 мг/100г ґрунту, тоді як показники загального азоту були нижчим і склали відповідно 0,45 мг/100г ґрунту та 0,52 мг/100г ґрунту.

Зі збільшенням глибини, накопичення азоту зменшувалося. Так, на глибині 41-70 см досліджуваний показник склав 11,1 мг/100г ґрунту (люцерна посівна) і 19,8 мг/100г ґрунту (галега східна) лужногідролізованого азоту. Тоді як, показники загального азоту на цій же глибині склали 0,31 мг/100г ґрунту (люцерна посівна) і 0,43 мг/100г ґрунту (галега східна).

Встановлено, що показники лужногідролізованого азоту на ділянках за вирощування люцерни посівної на глибині 71-130 і 131-150 см склали відповідно 8,1 та 4,8 мг/100г ґрунту, а за вирощування галеги східної показники були вищими відповідно 7,6 та 6,5 мг/100г ґрунту.

Показники загального азоту при вирощуванні люцерни посівної на горизонтах P_{hk} та $P_{(h)к}$ складало 0,21 мг/100г ґрунту і 0,18 мг/100г ґрунту, а на варіантах де вирощували галегу східну показники були наступними на горизонті P_{hk}=0,36 мг/100г ґрунту та $P_{(h)к}$=0,22 мг/100г ґрунту, що було вищим від перших ділянок в середньому на 0,15 та 0,04 мг/100г ґрунту.

Отже, встановлено, що показники азоту, за вирощування галеги східної були вищими у порівнянні до варіантів, де вирощували люцерну посівну. Така тенденція прослідковувалася за рахунок накопичення більшої кількості органіки та більшої кількості симбіотичного азоту, що накопичувалися в посівах галеги східної.

Також, в наших дослідженнях ми звернули увагу, як накопичуався фосфор у посівах бобових трав.

Фосфор засвоюється рослинними організмами в менших кількостях ніж азот, хоча його роль теж надзвичайно велика у життєвому циклі рослин.

Нами було встановлено, що кількість фосфору змінювалася по горизонтам, а також залежно від досліджуваної культури (рис. 4).

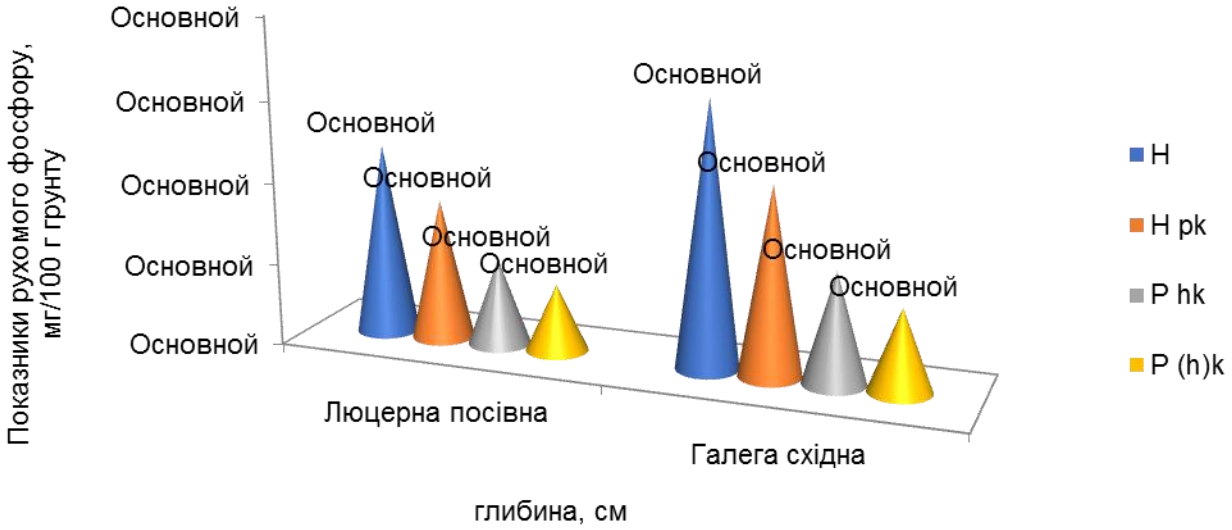


Рис.4. - Показники рухомий фосфор (за Мачигіним) залежно від люцерни посівної та галеги східної (в середньому за 2019-2021 р.), мг/100г ґрунту

Зі збільшеням глибини по горизонтам показники рухомого фосфору знижувалися за обох досліджуваних культур.

При вирощувані люцерни посівної, на горизонті Н-0-40 см, рухомий фосфор склав 11,7 мг/100г ґрунту, тоді як за вирощування галеги східної - був вищим на 4,4 мг/100г ґрунту.

На горизонті Нрк фосфор був у межах 8,6 мг/100г ґрунту (люцерна посівна) та 11,4 мг/100г ґрунту (галега східна), тоді як на Phk та P(h)к досліджуваний показник відповідно склав 5,3 та 4,1 мг/100г ґрунту (люцерна посівна) та 6,7 і 5,0 мг/100г ґрунту (галега східна).

Встановлено, що за вирощування галеги східної відбувається підвищення рухомого фосфору у грунті, що перевищувало аналогічні показники в посівах люцерни посівної в межах 18,0-27,3%.

Ми встановили, як змінювалася кількість калію погоризонтах грунту.

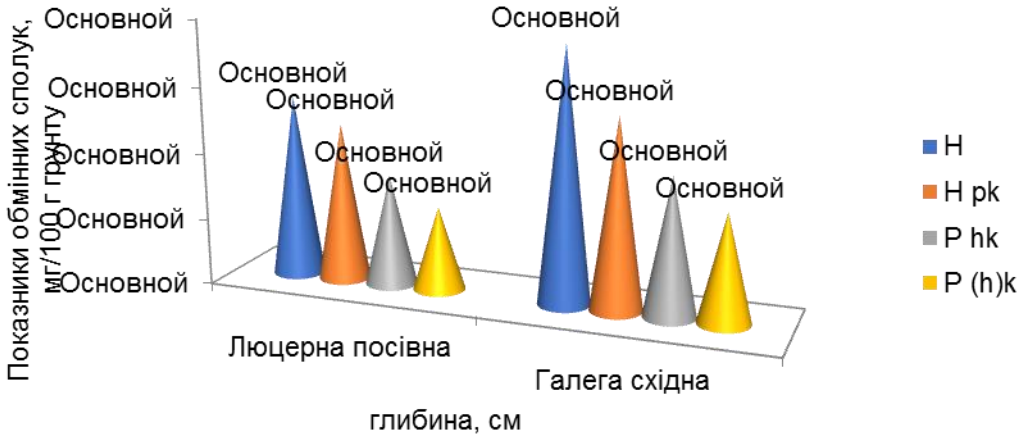


Рис. 5-Показники обмінних сполук калію (за Мачигіним) залежно від люцерни посівної та галеги східної (в середньому за 2019-2021 р.), мг/100г ґрунту

В результаті наших досліджень було встановлено, що показники обмінного калію за вирощування люцерни посівної на горизонті Н складало 13,7 мг/100г ґрунту, тоді як за вирощування галеги східної досліджуваний показник склав 19,0 мг/100г ґрунту і був вищим від попередніх ділянок в межах 5,3 мг/100г ґрунту.

Встановлено, що як і за дослідження інших показників, в глиб по горизонтах грунту показники зменшувалися.

Так, на горизонті Нрк вміст калію склав 11,9 мг/100г ґрунту (люцерна посівна) і 14,2 мг/100г ґрунту (галега східна), тоді як на горизонті Phk та P(h)к відповідно склали 8,3 та 6,3 мг/100г ґрунту (люцерна посівна) та 10,3 і 8,1 мг/100г ґрунту (галега східна).

Встановлено, що на варіантах де вирощували галегу східну, кількість обмінного калію була вищою, у порівнянні до ділянок, де вирощували люцерну посівну в межах 22,2-27,8 %.

В результаті наших досліджень, було встановлено, що вирощування галеги східної, сприяє відтворенню родючості ґрунту, за рахунок збільшення гумусу, азоту, азоту фосфору та калію, що сприятливо відзначиться на врожайності наступних культур у сівозміні, та поліпшить агроекологічні параметри ґрунтового середовища.

Висновки. Таким чином, підвищення агроекологічних параметрів ґрунту, за рахунок вирощування бобової багаторічної трави галеги східної, а саме відбувається накопичення гумусу та поживних речовин на горизонті Н: гумус - 5,2 %; pH – 6,7; азот лужногідролізований – 22,7 мг/100г ґрунту, азот загальний – 0,52 мг/100г ґрунту; рухомий фосфор – 16,1 мг/100г ґрунту; обмінний калій – 19,0 мг/100г ґрунту та зниження антропогенного тиску на середовище.

Список літератури

1. Кисіль В.І. Біологічне землеробство в Україні: проблеми і перспективи. Харків: «Штрих», 2000. 16с.
2. Шикула М. К., Балаєв А. Д. Родючість ґрунту та її відтворення в ґрунтозахисному землеробстві. Відтворення родючості ґрунтів у ґрунтозахисному землеробстві: монографія. Київ: Оранта, 1998. С. 208–219.
3. Гнатенко О.Ф., Капштик М.В., Петренко Л.Р., Вітвицький С.В. Ґрунтознавство з основами геології: навч. Посібник. К.: Оранта. 2005. 648 с.
4. Кант Г. Біологічне рослинництво: можливості біологічних агроекосистем. Г. Кант. М.: Агропромиздат, 1988. 207 с.
5. Резніченко В.П., Ковальов М.М. Забезпеченість азотом гумусного горизонту чорноземів типового та звичайного в умовах північного Степу України. Таврійський науковий вісник: Сільськогосподарські науки. Вип. 107. Видавничий дім «Гельветика», 2019. С. 303-311.
6. Бахмат О. М. Соя – культура майбутнього, особливості формування високого врожаю: [монографія]. Кам’янець-Подільський: Мошак М. І., 2009. 208 с.

УДК 330.16

О. Смоляник, магістр гр. ЕО-20М
Л. Коломієць, доцент кафедри ЕОНС та ЗСЖ
Центральноукраїнський національний технічний університет

ПЕРСПЕКТИВИ ТА РОЗВИТОК ВИРОБНИЦТВА ОРГАНІЧНОЇ ПРОДУКЦІЇ В УКРАЇНІ

Проаналізовано розподіл органічних операторів за областями країни. Окреслено перспективи розвитку органічного виробництва, доцільність збільшення його масштабів за зростаючого попиту.
сільське господарство, органічне виробництво, екологічні чинники, органічні оператори, розвиток органічного ринку

Актуальність. Останнім часом нераціональне землекористування і ведення сільського господарства без урахування необхідності відновлення грунтового покриву спричинило деградацію та зниження родючості грунтів, зміну їх водно-фізичних, агрохімічних властивостей, біологічної активності грунту. Одним зі способів вирішення екологічних проблем є запровадження органічного землеробства. Концепція сталого розвитку аграрного виробництва передбачає поєднання захисту довкілля, економічного

зростання й соціального розвитку, саме виробництво органічної продукції є практичною реалізацією, що дасть змогу отримати високу якість продовольства як важливої складової продовольчої безпеки.

Постановка проблеми. Ринок органічних продуктів є перспективним сегментом агропромислового ринку розвинених країн світу. За даними експертів Міжнародної федерації органічного сільського господарства IFOAM і науководослідного Інституту біоземлеробства FiBL нині виробництво екологічно чистої сільськогосподарської продукції розвивається у 153 країнах світу, а обсяг ринку досягає 50–60 млрд доларів США. Це є поштовхом до розвитку органічного землеробства в Україні, стимулом до вирощування екологічно чистої продукції українськими аграріями. Останніми роками в країні спостерігається збільшення обсягу внутрішнього ринку споживання органічної продукції, зростає попит на органічні продукти харчування. Виробництво органічної продукції є практичною реалізацією концепції сталого розвитку аграрного виробництва, що передбачає поєднання захисту довкілля, економічного зростання й соціального розвитку як взаємозалежних і взаємодоповнювальних елементів стратегічного розвитку держави, що гарантуватиме населенню високу якість продовольства як важливої складової продовольчої безпеки.

Мета дослідження. Вивчення стану органічного виробництва у світі, його розвиток в Україні, виявлення стримуючих чинників розвитку виробництва органічних продуктів, виділення проблемних аспектів.

Завдання: проаналізувати сучасний стан виробництва органічної продукції; окреслити перспективи розвитку органічного виробництва; виявити стримуючі фактори розвитку виробництва органічних продуктів; запропонувати заходи державного стимулювання виробництва органічної продукції

Результати дослідження. Проблеми екологічно чистого довкілля та здорового способу життя є надзвичайно важливими для суспільства. Сільськогосподарські екосистеми щорічно руйнуються внаслідок технологічних заходів – хімічних, механічних, біологічних. Зовнішні екологічні чинники, включаючи структуру агроландшафту, зокрема польові захисні пояси лісосмуг та екотони між ними, також мають певний вплив. Зважаючи на погіршення екологічних показників, деградація грунтових покривів зумовлює потребу застосування біологізації виробництва, і, зокрема, впровадження системи органічного землеробства, яка більше відповідає інтересам суспільства та не порушує біологічної рівноваги в землеробстві.

Переваги органічного сільського господарства полягають в економічному зростанні, захисті НПС, якості та безпеці продуктів харчування. Органічне землеробство зменшує використання агрохімічних засобів захисту завдяки поєднанню традиційних і сучасних технологій для боротьби зі шкідниками та хворобами, покращує властивості ґрунту, захищає водні ресурси від забруднення, мінімізує чинники, які безпосередньо впливають на зміну клімату, підтримує різноманіття мікрофлори ґрунту та підвищує врожайність. Запровадження сівозмін, використання посадкового матеріалу і порід, що адаптовані до місцевих умов, відновлення та розширення функціонального біорізноманіття сприяють подальшому зміцненню екологічної рівноваги.

У результаті наукових досліджень відмічено, що запровадження органічного виробництва має ряд переваг проти традиційного, зокрема екологічні, економічні і соціальні аспекти. Економічні обумовлюють зростання прибутку та підвищення конкурентоздатності. Екологічні переваги сприяють збереженню навколишнього середовища. Соціальні переваги базуються на забезпеченні ринку якісною та корисною продукцією, що є безпечною.

Виникнення першого органічного руху припадає на 40-і роки минулого сторіччя у Великобританії, тоді вперше було застосовано визначення «органічний» в науковій праці Єви Бальфур «Жива земля», у якій автор порівнює органічні і традиційні методи ведення сільського господарства. Середина XIX сторіччя характеризується стрімким розвитком «органічного» руху, було засновано велику кількість громадських організацій, які контролювали виробників та процеси виробництва продуктів харчування. Найбільшою та

найвагомішою організацією у світі є Міжнародна Федерація Органічного Руху (IFOAM), яка заснована в 1972 році [1-3].

У світі органічним виробництвом займаються в 172 країнах світу, із яких в Азії – 40%, Африці – 26, Латинській Америці – 17, Європі – 15, Північній Америці – 1, Океанії – 1 %. За ведення органічного виробництва перебуває приблизно 1 % світової площі сільськогосподарських угідь, а в країнах Євросоюзу – приблизно 3 % сільськогосподарських угідь. Лідером з виробництва органічної продукції є Австралія – понад 900 млн га сільськогосподарських земель, що використовують для вирощування за технологіями, адаптованими до органічного землеробства, також попереду Аргентина і Китай.

Відповідно до сприятливих природно-кліматичних умов, українські сільгоспвиробники мають змогу відмовитися від індустріалізованих методів виробництва та запровадити органічні підходи до ведення землеробства, без застосування пестицидів і генно-модифікованих організмів. Слід розуміти, що ГМО можуть бути не лише у рослинах та продукції, а й у допоміжних продуктах. Територія та земельний ресурс України є найбільшими у Європі – загалом більше 41 млн га угідь сільськогосподарського призначення.

Органічний рух в Україні розпочався наприкінці XX сторіччя (перше органічне підприємство з'явилося вже в 1970 році в Полтавській області), однак відчутним став у 2005 році. Нині Україна активно займає свою позицію на міжнародному ринку органічної продукції, розширюючи площі під виробництвом сільськогосподарської продукції, переробкою продукції, трейдерською діяльністю та експортом [2-4].

Відповідно до інформації Офісу підтримки реформ при Мінагрополітики у 2019 році в нашій державі сертифікованих операторів, що займаються виробництвом органічнох продукції, налічували 617, з них 470 – сільськогосподарські виробники органічної продукції, які пройшли перехідний період(рис.2). За останні п'ять років загальна площа під органікою в Україні зросла у 1,5 раза. Станом на 2019 рік загальна площа сертифікованих органічних сільськогосподарських земель становила 467 тис. га. (рис.1)

Рисунок 1. Органічна карта України

Джерело: https://kurkul.com/news/11018-v-ukrayini-prezentuvali-kartu-organichnoyi-produktsiyi

За загальною площею сільськогосподарських земель, що внаслідок сертифікаційного процесу набули статусу «органічні», Україна займає 11 місце серед країн Європи та 20 у світі. У східноєвропейському регіоні щодо обсягів сертифікованої площі органічної ріллі Україна посідає одне з провідних місць. Дикороси також є одним з перспективних напрямів ведення органічного виробництва, в Україні сертифіковано 570 тис. га дикоросів станом на 2017 рік. Майже половина сільськогосподарських угідь України, сертифікованих відповідно

до вимог органічного виробництва, зайняті під вирощуванням таких культур: зернові – 45,4 %, олійні – 18 % і бобові – 5,3 %. Слід відмітити що овочеві культури займають 1,6 % (10 місце) та фрукти – 0,7 %. Доволі стрімко почав розвиватися напрям виробництва органічних ягід в Україні, а саме вирощуваня малини, лохини, ожини.

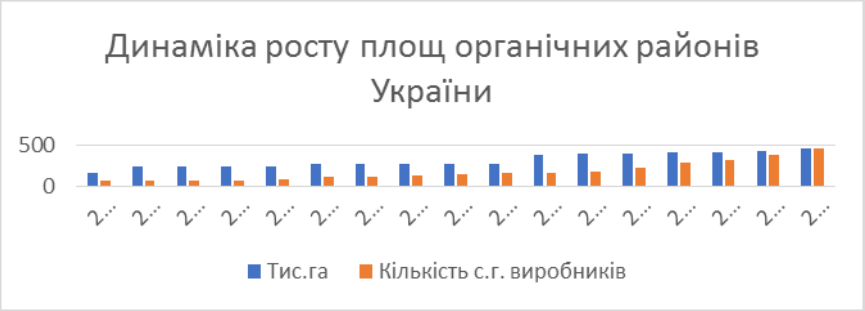


Рисунок 2. Динаміка росту площ органічних районів України

Територіально за виробництвом органічної продукції лідирують Київська, Одеська, Херсонська, Житомирська, Львівська, Хмельницька, Вінницька та Полтавська області. Оператори, що сертифікують свою діяльність, запроваджують її на площах до 10 га, як і в більшості країн Європи, до кількох тисяч гектарів орних земель. Продовжується зростання числа дрібних органічних господарств, значна частина яких спеціалізується на вирощуванні плодоовочевої та ягідної продукції.

Україна є потужним постачальником органічної продукції. Споживчий попит на органічну продукцію переважно зосереджений в економічно розвинутих країнах, оскільки така продукція є дорожчою, ніж звичайна, враховуючи вищу собівартість її виробництва та переробки [2,5-6].

З усього обсягу продажів органіки 90 % припадають на експорт і лише 10 % – на внутрішній ринок. Обсяги експортованої продукції постійно збільшуються, і станом на 2017 рік сягнули 99 млн євро. Найбільше споживають органічну продукцію вирощену в Україні у таких країнах: Швейцарія, Австралія, Італія, Болгарія, Нідерланди, Німеччина, Чехія, Велика Британія, Австрія, Польща, США, Бельгія, Угорщина, Канада, деякі країни Азії. Високим попитом українські органічні продукти користуються в Америці та країнах Азії, що займають приблизно 4 млн євро із загального експорту. Продуктами для експорту, що мають високий попит, є: дикороси (ягоди, гриби, горіхи), зернові, бобові, олійні, ягоди, фрукти, експортують також макуху соняшнику. Загалом українську органіку купують більше 40 країн світу.

Надалі попит на органічну продукцію у світі зростатиме з розвитком економіки країн, підвищенням рівня освіти да доходів населення, орієнтуючись на оцінку Продовольчої та сільськогосподарської організації ООН. Однак серед населення України попит на органічні продукти значно нижчий проти країн ЄС.

Обсяг внутрішнього ринку органічних продуктів у 2017 році в Україні становив 29,4 млн євро, а споживання на душу населення – лише 0,68 євро. Для порівняння, житель Європи споживає таких продуктів на 40,8 євро в рік, водночас у країнах ЄС на душу населення припадає 60,5 євро. Україна займає 25 місце в Європі відповідно до обсягу внутрішнього ринку органічних продуктів. Основними категоріями сертифікації органічної продукції, яку виробляють та споживають в Україні, є рослинництво: фрукти, овочі, зернові культури, тваринництво: м'ясо та молочні продукти, переробка: крупи та хлібобулочні вироби.

У розвитку органічного виробництва в Україні існує ряд стримуючих чинників, які гальмують подальший розвиток цього сегмента аграрного сектора: недосконалий науковий супровід щодо органічного сектору, переважання експорту органічної сировини, низький рівень обізнаності населення і виробників щодо специфіки органічної продукції, відсутність державної фінансової підтримки, відсутність ефективної системи державного контролю з

боку держави за виробництвом та якістю продукції, що спричиняє недобросовісну конкуренцію серед виробників і продавців, відсутність системи ефективного захисту прав споживачів та працюючої системи санкцій щодо недобросовісного вирощування сільськогосподарської продукції та підробок [6-8].

Висновок. Проведене дослідження дає змогу зробити висновки, що в Україні простежуються загальносвітові тенденції до популяризації виробництва органічної продукції. Органічне виробництво повільно, але розвивається: за період з 2003 до 2019 рр. кількість органічних господарств зросла майже у 7 разів і наразі становить 470 одиниць; площа, зайнята органічним виробництвом збільшилася майже в 3 рази й становить 469 тис. га (1,5 % від загальної площі сільськогосподарських угідь). Лідерами за кількістю органічних господарств є Київська, Херсонська області. Слід активізувати розвиток виробництва органічних продуктів у тих областях країни, де їхня кількість незначна – в Луганській, Сумській, ІваноФранківській, Чернівецькій областях.

На сьогоднішній день Україна має значний потенціал розвитку виробництва органічної продукції. Продукція органічного походження стає все більш привабливою як для європейського, так і для національного споживача. Враховуючи те, що Україна має потужний потенціал агропромислового комплексу, країна може стати одним із головних експортерів цієї продукції на ринку ЄС.

Для удосконалення й росту виробництва органічної продукції уже сертифікованими операторами органічного виробництва та заохочення й підтримки, створення нових підприємництв органічного ринку необхідне удосконалення нормативно-правової бази щодо виробництва органічної продукції та її гармонізація відповідно до вимог європейських Постанов, Регламентів, Директив. Розвиток органічного виробництва в Україні можливий лише за умови державної підтримки. Державне стимулювання може бути реалізоване через фінансову підтримку, пільгове оподаткування, підвищення розміру доплат до закупівельної ціни, пільгові ціни на послуги і засоби виробництва, державне страхування, популяризацію органічної продукції серед виробників і споживачів, створення розгалуженої інфраструктури ринку органічних продуктів. У такому випадку вітчизняний агровиробник здатний забезпечити виробництво достатньої кількості органічної продукції, що буде сприяти, з одного боку, покращанню стану навколишнього середовища, з іншого, – зростанню вітчизняного сільського господарства та стане досить значущою складовою підвищення рівня здоров’я нації.

Список літератури

1. https://ses-help.org.ua/dstu/%D0%94% 1.pdf
2. Офіційний сайт Міністерства аграрної політики та продовольства України. URL: http//minagro.gov.ua/.
3. https://dnaop.com/html/33894_3.html
4. https://gost-snip.su/download/dstu_4492_2005_oliya_sonyashnikova_tehnichni_umovi
5. Grabovska T. Eff ect of organic farming on insect diversity. Ukrainian Journal of Ecology. 2020. 10(3). P. 96–101.
6. https://kolosok.info/%D08B-4492:2005-_g2
7. https://lektsii.org/5-32824.html
8. Світ органічного сільського господарства. Статистика та тренди 2018: книга FiBL-IFOAM. Офіційний сайт International Federation of Organic Agriculture Muvements. URL: http://organicukraine.org.ua/congress/wp-content/uploads/prokopchyk-organicukraine-conggress2018-ua.pdf.

УДК 004

К. Авраменко, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ МІКРОКЛІМАТУ СКЛАДСЬКОГО КОМПЛЕКСУ

У статті розроблено програмне забезпечення, яке призначено для системи мікроклімату складського комплексу. Метою розробки є дослідження та програмна реалізація системи мікроклімату складського комплексу. Об'єктом дослідження є процес мікроклімату складського комплексу. Предметом дослідження є методи мікроклімату складського комплексу. Методи дослідження базуються на методах інтернету речей, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи мікроклімату складського комплексу. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення

комп'ютерна інженерія, мікроклімат, складський комплекс

Постановка проблеми. В Україні, зерно традиційно є стратегічною продукцією. За даними Міністерства агропромислового комплексу, у виробників зерна залишається більше 50% отриманого врожаю. Аналогічні тенденції прогнозуються й у майбутньому. Світова практика так само показує, що в аграрно розвинених державах до 80% урожаю зернових культур зберігається безпосередньо в його виробника. Однак, оснащення наших господарств, незалежно від форм їхньої власності, технологічним устаткуванням для забезпечення якісного післязбирального доведення до кондиції зерна, є вкрай незадовільним. До 70% господарств не мають зерноочисного й сушильного встаткування, що приводить до значних втрат урожаю. Тому в основу створення техніки для післязбиральної обробки зерна необхідно покласти такі напрямки [1]:

– обробка всього зібраного врожаю продовольчого й фуражного зерна на місці виробництва;

– будівництво складських приміщень для зберігання зерна й насіння із повністю укомплектованими засобами механізації вантажно-розвантажувальних операцій.

Таким чином, одним зі шляхів зменшення втрат і підвищення якості зерна, є безумовне забезпечення кожного господарства власним сучасним складським приміщенням для зберігання зерна, у якому існує автоматизована система управління вологістю та температурою. Підсистемою такої системи є мережа датчиків для визначення вологості, з можливістю обробки отриманих значень вологості та реагуванням на її зміну [1-5].

Це дасть можливість зерновиробникам:

– створити власну інфраструктуру якісного збереження врожаю й заощаджувати при цьому до 40-50% енергії на досушуванні й охолодженні зерна;

– зберігання свого зерна у своєму сховищі дозволить зерновласникам незалежно й упевнено розпоряджатися вирощеною продукцією;

– зберігати врожай на власній фермі й не витрачати час і засоби на транспортування й простій у чергах на зернозбиральних пунктах;

– не продавати зерно в період збирання, а притримати до часу, коли ціни на нього будуть найбільш сприятливими.

Слід зазначити, що більше 80% енергоносіїв Україна імпортує [6]. Тому усе більше актуальною проблемою є ефективне використання в технологічних процесах

альтернативних джерел енергії. Так як одним з найбільш енергоємних процесів у зерновиробництві є досушування зерна, то розробка й впровадження енергозберігаючих технологій доведення його до кондиції, є найважливішим і сучасним завданням.

В зв'язку з тим, що для досушки зерна необхідно періодично визначати значення вологості то впровадження системи виміру вологості є актуальною та вкрай необхідною мірою підвищення строків зберігання та якості зерна. Але для того, що ця система працювала необхідно розробити відповідне програмне забезпечення.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи мікроклімату складського комплексу.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи мікроклімату складського комплексу.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

Огляд існуючих систем мікроклімату складського комплексу.

Дослідження системи мікроклімату складського комплексу.

Програмна реалізація системи мікроклімату складського комплексу.

Об'єктом дослідження є процес мікроклімату складського комплексу.

Предметом дослідження є методи мікроклімату складського комплексу.

Методи дослідження базуються на методах інтернету речей, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Розглянемо більш детально один із затребуваних і актуальних на сьогоднішній день напрямків підрозділу сенсорного контролю – датчики вологості. Основні області застосування: клімат-контроль у промислових, складських і адміністративних приміщеннях, контроль технологічних процесів, екологічних параметрів, метеорологічний контроль і т.д.

Загальний опис

В основі побудови датчиків вологості лежить тришарова конденсаторна структура, що складається із платинових електродів і спеціального термореактивного полімерного ізолятора між ними (рисунок 1). Вся ця структура розміщена на підложці із кремнію, на якій також виконана інтегральна схема нормалізації й посилення сигналу. Через пори у верхньому електроді й завдяки конструктивній негерметичності корпуса датчика досягається рівноважний вміст води в навколишньому повітрі й міжелектродному просторі. Шар термореактивного полімеру, що покриває пористий платиновий електрод зверху, служить гарним захистом чутливого елемента від забруднення пилом, маслами. У той же час такий захист сприяє збільшенню часу відгуку датчика при зміні вологості.

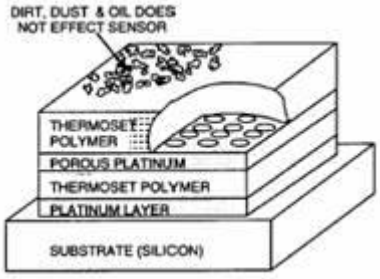


Рисунок 1 – Структура чутливого елемента датчика вологості

Для всіх типів датчиків інтегральна схема формує сигнал, що лінійно змінюється по напрузі, прямо пропорційний напрузі живлення й відносній вологості повітря:

$$Vout=Vs\ (0{,}0062(\%RH)+0{,}16) \quad (1)$$

де Vs – напруга живлення, %RH – відносна вологість при 25 0С.

Справжнє значення вологості при температурі, що змінюється, можна визначити по формулі:

$$RH=(\%RH)/(1,\ 0546-0,00216T) \quad (2)$$

де Т – температура в 0 С

Для найбільш точного виміру вологості в датчиках серії HIH-3602 у корпус вбудовані датчики температури (рисунок 2).

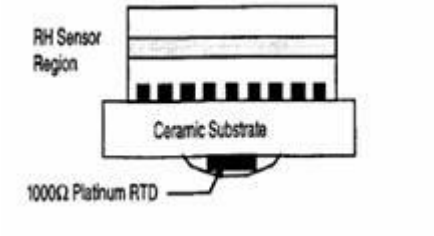


Рисунок 2 – Вбудований датчик температури

Особливості застосування датчиків серії HIH-3602

Для роботи в суворих кліматичних умовах і найбільш точного визначення значення вологості поблизу точки роси фірма Honeywell випускає серію HIH-3602. Датчики випускаються в корпусі TO-5, у який вбудований датчик температури, для моделі HIH-3602-A це NTC термістор номіналом 100 ком, для HIH-3602-C платиновий датчик температури номіналом 1 ком. Можливі ситуації, коли при високому значенні відносної вологості (більше 95%) при зниженні температури можлива конденсація пар води, що спричиняє залипання показань на рівні 100% відносної вологості. Для запобігання цього випадку в датчиках серії HIH-3602 передбачений гідрофобний фільтр, виконаний зі спеціально обробленої нержавіючої сталі. Вивести зі сплячого стану також допомагає подача на датчик трохи підвищеної напруги живлення, однак при цьому можливо деякий зсув калібрування на 2:3 %. Сама по собі конденсація й наступний випар вологи не робить впливу на калібрування датчиків.

Основні переваги серії HIH-3602

– наявність вбудованого датчика температури

– корпус із гідрофобним фільтром

– кожний датчик постачений паспортом з індивідуальними даними

– лазерне припасування елементів

– висока стійкість до впливу хімічних речовин (крім пар ацетону або етанола).

Особливості застосування датчиків серії HIH-3610

Датчики серії HIH-3610 є самими масовими по виробництву й застосуванню в схемах до яких не пред'являються особливі вимоги по точності виміру на граничній вологості (більше 95%). Датчики виготовлені в SIP корпусі, причому моделі HIH-3610-001(3) мають формовані виводи із кроком 2,54 мм, у моделей HIH-3610-002(4) крок між виводами 1,27 мм. Також датчики серій HIH-3610-003(4) поставляються з індивідуальними паспортними даними. Вихідний сигнал (напруга) з датчика прямо пропорційний напрузі живлення й відносній вологості повітря.

До переваг даної серії можна віднести:

– високу точність, компактність габаритів;

– простоту у використанні;

– мале енергоспоживання;

– малий час відгуку;

– можливість проведення вимірів у широкому діапазоні тисків.

Недоліки:

– світлочутливі, вимагають екранування від яскравого світла.

Вимірювач вологості та температури ІВТМ

Прилад ІВТМ складається із блоку виміру, керування й індикації (БРІУ) і первинних перетворювачів. До приладу може підключатися ІВМ РС-сумісний комп'ютер із принтером. Конструктивно блок керування виконується в пластмасовому або металевому корпусах. Первинний перетворювач також виконується в пластмасовому або металевому корпусі залежно від обраної моделі й складається з вимірювальної камери, у якій розташовуються сенсори й корпуси, у якому розташовується схема попередньої обробки сигналів.

Для виміру вологості використовуються сорбційно-ємнісні мікроелектронні сенсори. Для виміру температури в приладі застосовані сенсори резистивного типу. Сигнал від сенсорів обох типів перетвориться в частотний сигнал за допомогою первинних перетворювачів.

Опис схеми перетворювача ІПВТ-03

Перетворювач виконаний за схемою RC-генератора на таймері типу 555. У якості R елемента в каналі температури використовується терморезистор, а в якості С-елемента каналу вологості використовується ємнісної сенсор вологості.

Підключення датчиків до таймера виробляється за допомогою електронного комутатора. Крім вимірювальних елементів комутатор робить підключення до таймера зразкових RC-елементів (як зразкові елементи застосовуються термостабільні резистори й конденсатори). Застосування подібної вимірювальної схеми дозволяє робити автокомпенсацію перетворювача при зміні температури навколишнього середовища. Керування комутатором, підрахунок частоти з таймера, обчислення температури й вологості здійснюється логічним блоком перетворювача, виконаним на базі PIC-контролера. По програмі, закладеної в мікроконтролері, здійснюється вимір частоти від сенсорів, зразкових елементів і обчислення значень температури й вологості по індивідуальних калібруваннях, що перебуває в пам'яті обчислювального пристрою перетворювача. Обчислені значення параметрів вологості й температури в послідовному цифровому коді надходять на вихідний пристрій.

Живлення перетворювача здійснюється постійним струмом з напругою від 6 до 9В. Живлення складових частин перетворювача здійснюється за допомогою внутрішнього стабілізатора 5В. Вихідний пристрій необхідно для передачі даних про обмірювані значення в приладі. Передача даних здійснюється по напівдуплексному каналі диференціальним методом. Застосування даного способу передачі дозволяє перетворювачу працювати на довгих лініях при великому рівні електромагнітних перешкод. Відстань, на якій стійко працює перетворювач, становить не менш 300 метрів.

Опис БРІУ

БРІУ виконаний у вигляді мікропроцесорної системи. Він призначений для роботи з перетворювачем вологості й температури типу ІПВТ-03 і його модифікаціями. До приладу може бути підключене до 8 перетворювачів ІПВТ-03. Перетворювачі підключаються до приладу в будь-якому порядку і є взаємозамінними з перетворювачами свого типу. Прилад може бути пов'язаний з комп'ютером по послідовному каналу зв'язку з інтерфейсом RS-232 або RS-485. До приладу може бути підключений блок реле. У приладі передбачений аналоговий вихід.

Робота БРІУ визначається програмою, записаною в постійний запам'ятовувальний пристрій. Внутрішні змінні, а також константи калібрування й інші оперативні параметри (при необхідності) зберігаються в FLASH-пам'яті, що є енергонезалежною й зберігає інформацію при відключеному живленні протягом усього терміну служби приладу. Одним з режимів приладу є режим нагромадження даних. У даному режимі прилад із заданою періодичністю записує дані про обмірювані значення вологості й температури із прив'язкою до реального часу. Інформація зберігається в спеціальній енергонезалежній пам'яті реєстрації. Мінімальний обсяг пам'яті 32 Кб (максимальний 192 Кб), що дозволяє восьмиканальному приладу запам'ятати до 455 (2730) відрахунків інформації. Установка внутрішніх годиників приладу, періодичності запам'ятовування даних і їхній перегляд здійснюються за допомогою комп'ютера.

БРІУ робить послідовне опитування перетворювачів, і дані про температуру й вологість, обмірювані й розраховані в кожному перетворювачі, відображаються на індикаторі приладу. У тому випадку якщо один або обидва параметри в каналі замасковані, замість значень температури й вологості виводяться символи ---і-. При перевищенні встановленого порога для даного каналу БРІУ видає звуковий сигнал і виводить номер каналу, у якому виявлений вихід за поріг, на індикатор.

Роз'єм RS232/RS485 призначений для підключення до комп'ютера по інтерфейсу RS232 і об'єднанню приладів у мережу по інтерфейсу RS485.

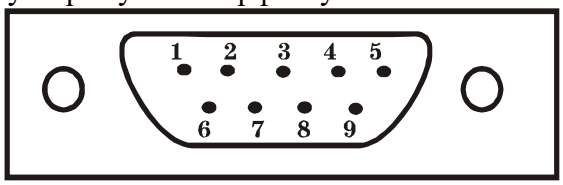


Рисунок 3 – Вид вилки з боку монтажу.

1 – сигнал А лінії RS485 4, 6, 7, 8 – не використовуються
2 – сигнал Rx лінії RS232 5 – загальний (земля) RS232, RS485
3 – сигнал Tx лінії RS232 9 – сигнал У лінії RS485

Роз'єм "ЛІНІЇ КЕРУВАННЯ" для підключення блоку реле призначені для підключення 16-ти канального блоку реле. У розніманні з маркуванням "Лінії керування 0-F" роз'єм № 1 відповідає лінії керування 0, роз'єм № 2 відповідає лінії керування 1 і так далі до роз'єм № 16, що відповідає лінії F. Роз'єм з № 17 по 25 загальний (-).

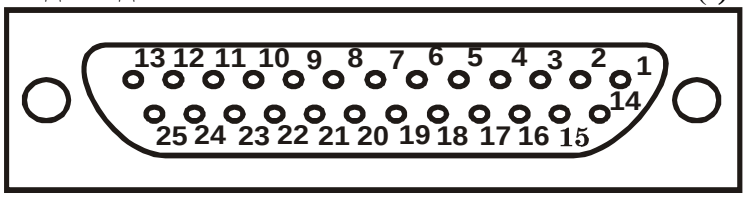


Рисунок 4 – Вид розетки для підключення блоку реле з боку монтажу

Аналоговий вихід. При перевищенні максимальних значень вологості й температури або відсутності імпульсів у відповідних каналах вихідний струм залишається рівним максимальному.

При зниженні вологості або температури нижче мінімального значення вихідний струм дорівнює нулю. При маскуванни каналів вологості й температури вихідний струм може приймати довільне значення в межах діапазону його зміни. У режимах настроювання приладу вихідний струм залишається рівним останньому обмірюваному значенню. Вихідний сигнал – струм прямо пропорційний вимірюваній відносній вологості й температурі й може змінюватися залежно від замовлення межах від 0 до 20, від 4 до 20 і від 0 до 5 мА. Значення вологості й температури розраховуються по формулах:

$$H = (Ih - \mathrm{Im}in)\frac{(H\max - H\min)}{(\mathrm{Im}ax - \mathrm{Im}in)} + H\min \text{, \%} \quad (3)$$

$$T = (It - \mathrm{Im}in)\frac{(T\max - T\min)}{(\mathrm{Im}ax - \mathrm{Im}in)} + T\min \text{, oc} \quad (4)$$

де Ih – значення струму, що відповідає вимірюваної вологості, Imin – мінімальне значення вихідного струму, Imax – максимальне значення вихідного струму, Hmin – мінімальне значення вологості, Hmax – максимальне значення вологості, It – значення струму, що відповідає вимірюваній температурі, Tmin – мінімальне значення температури, Tmax – максимальне значення температури. Imin, Imax, Hmin, Hmax, Tmin, Tmax – параметри аналогового виходу, що задаються при замовленні.

Опис багатоканальної системи визначення вологості

Запропонована в проекті система, можлива для реалізації в трьох різних варіантах:

– Багатоканальна система контролю й керування вологістю й температурою.
– Вимірювальна мережа зі збором інформації на FLASH-card.
– Вимірювальна мережа з передачею інформації через GPRS.

Розглянемо їх докладніше.

Багатоканальна система контролю й керування вологістю й температурою

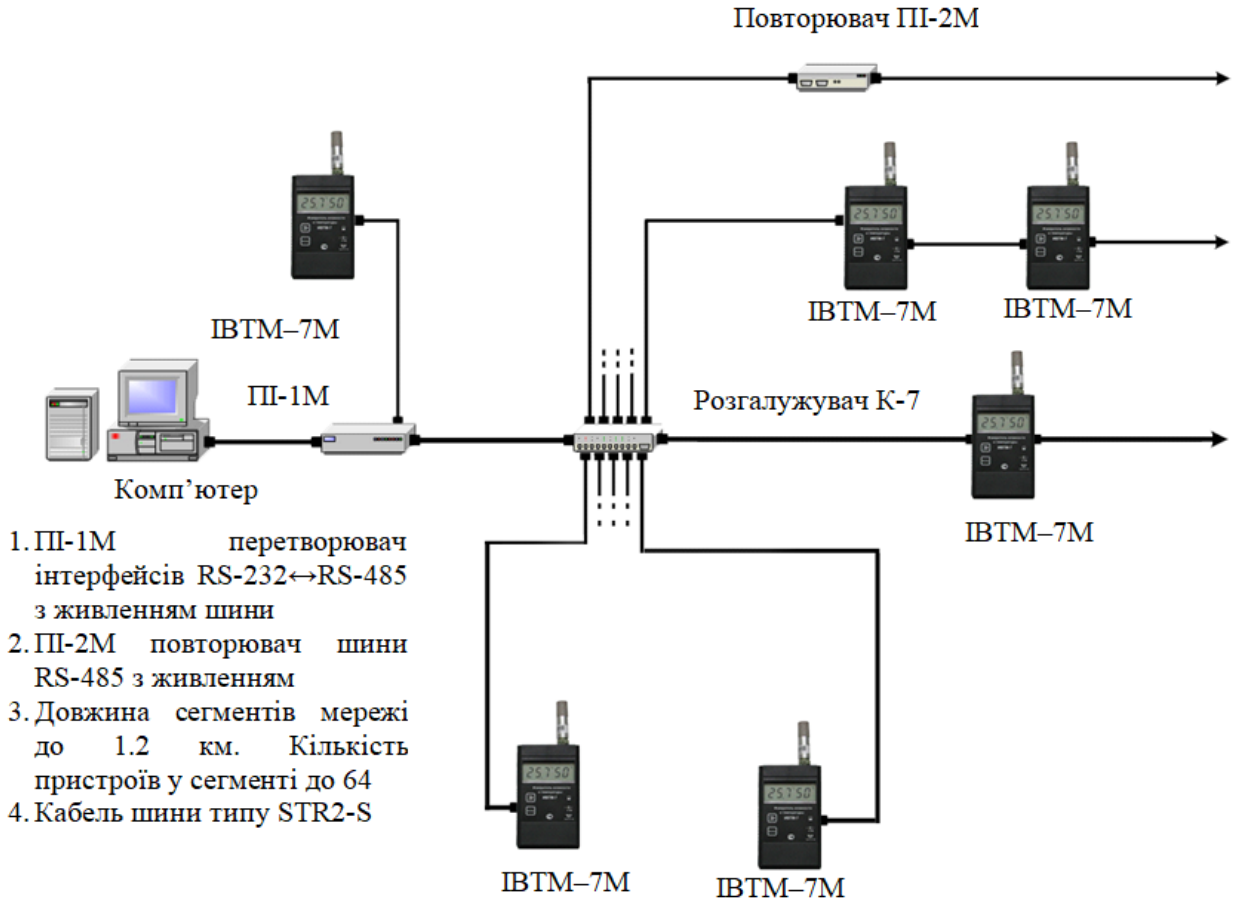


Рисунок 5 – Багатоканальна система контролю й управління мікрокліматом

Багатоканальна система контролю й управління мікроклімату. Призначена для контролю (і керування) температурою й відносною вологістю на різних, вилучених друг від друга об'єктах.

У систему можуть поєднуватися прилади ІВТМ-7 МК-3, ІВТМ-7 МК-3-М, ІВТМ-7 Р-МК, ІВТМ-7 Р-МК-М, ІВТМ-7/8 Р-МК, які призначені для виміру вологості й температури в складському приміщенні. Як допоміжні пристрої в системі, при необхідності, використовуються перетворювачі інтерфейсу ПІ-1, ПІ-2, комутатор RS-232, і адаптери. Інформація від всіх приладів виводиться на комп'ютер. З нього ж може здійснюватися керування приладами, що мають функцію регулювання. Для роботи із системою використовується комп'ютерна програма. Діапазон виміру відносної вологості 0...99%, температури -20...+60 ОС (-50...+150 ОС по спеціальному замовленню). Прилад має можливість підключення до комп'ютера по інтерфейсу RS232 і RS485.

Вимірювальна мережа зі збором інформації на FLASH-card

Можливості мережі

- До складу мережі можуть входити будь-які портативні й мережні одно- і багатоканальні прилади, що підтримують інтерфейс RS-485 (або інтерфейс RS-232 с зовнішнім перетворювачем інтерфейсу ПІ-7). Кількість приладів у мережі обмежується параметрами інтерфейсу RS-485 (64 приладів в мережі без додаткових підсилювачів).

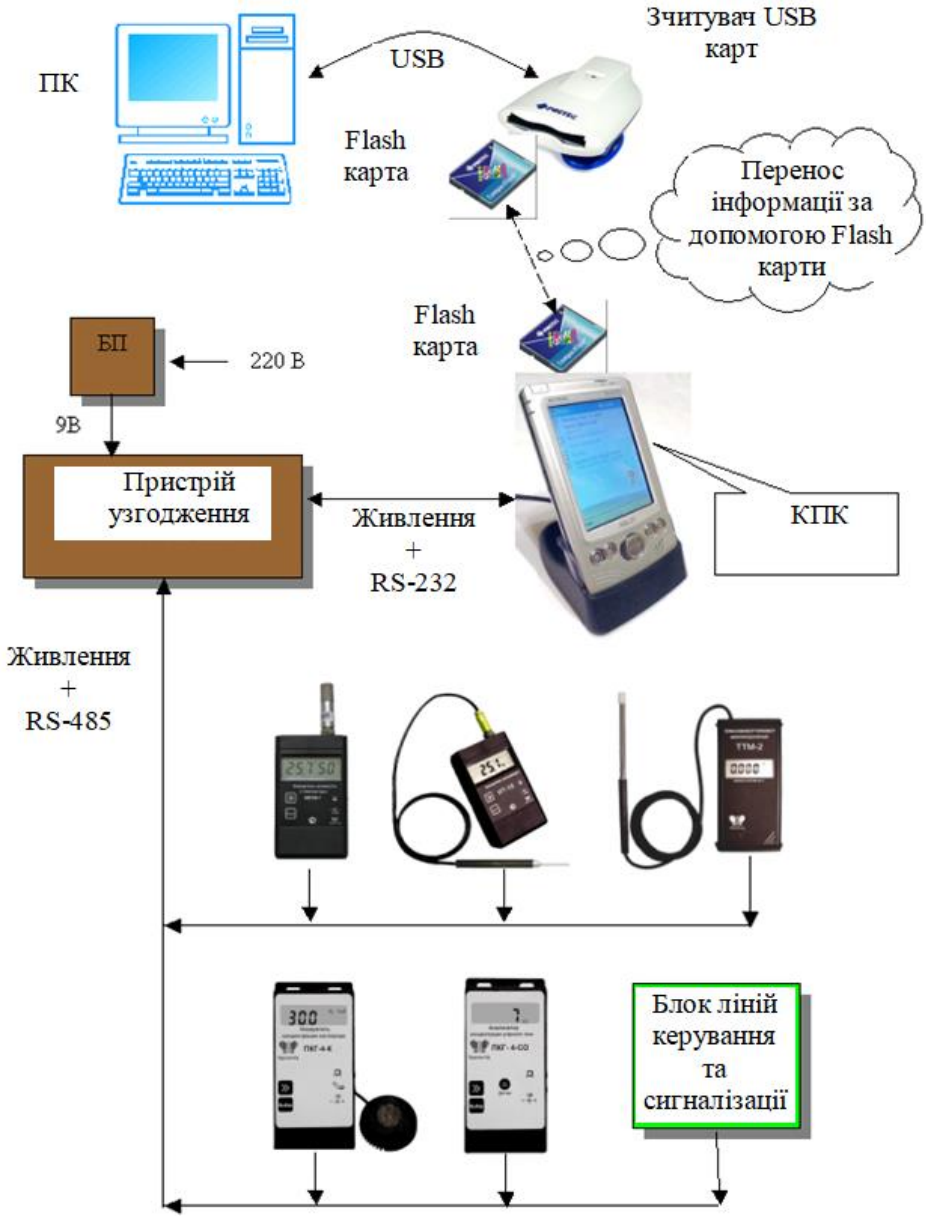


Рисунок 6 – Вимірювальна мережа зі збором інформації на FLASH-карту

- У вимірювальну мережу можуть входити також виконавчі пристрої (блоки реле й т.д)
- Збір інформації здійснюється КПК, що підключається до вимірювальної мережі через спеціальний пристрій узгодження. Для КПК написане спеціальне програмне забезпечення.
- Програма на КПК здійснює функції сервера, надаючи клієнтам інформацію, зчитувальну із приладів, по запиту із глобальної мережі Інтернет. Доступ серверної частини в Інтернет організується за допомогою встановлюваного в КПК модуля GPRS через операторів мобільного зв'язку GSM, що підтримують цей сервіс. ПЗ дозволяє відслідковувати стан зв'язку й при її обриві автоматично відновлювати з'єднання. Реалізовано функцію реєстрації на сервері динамічної IP адреси, виділюваного КПК у момент з'єднання, що дозволяє підтримувати постійний зв'язок між клієнтами й сервером.
- Для ПК написана спеціальна програма-клієнт, що дозволяє шляхом запиту через мережу Інтернет одержати доступ з будь-якої точки миру до інформації, зчитувальної КПК. Клієнтське ПЗ також підтримує технологію реєстрації на сервері динамічного IP адреси, надаваного КПК.

Вимірювальна мережа з передачею інформації через GPRS

Можливості мережі

- До складу мережі можуть входити будь-які портативні й мережні одно- і багатоканальні прилади, що підтримують інтерфейс RS-485 (або інтерфейс RS-232 с зовнішнім перетворювачем інтерфейсу ПІ-7). Кількість приладів у мережі обмежується параметрами інтерфейсу RS-485 (64 приладів в мережі без додаткових підсилювачів).
- У вимірювальну мережу можуть входити також виконавчі пристрої (блоки реле й т.д)

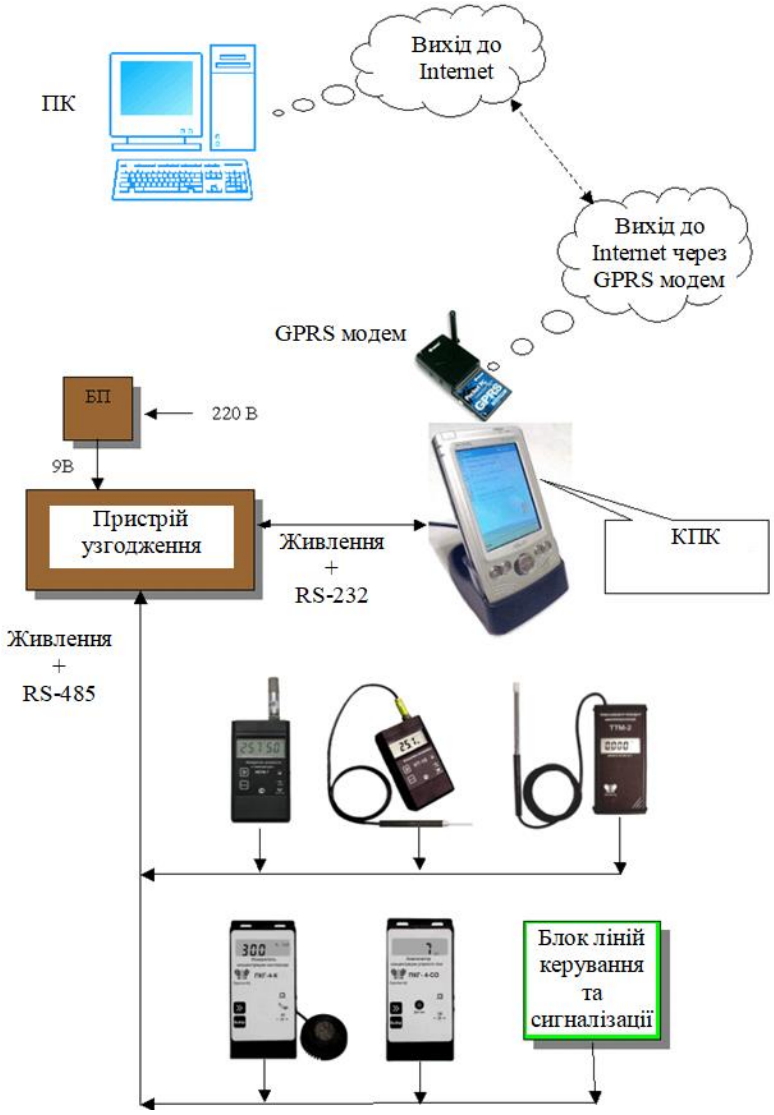


Рисунок 7 – Вимірювальна мережа з передачею інформації через GPRS

- Збір інформації здійснюється КПК, що підключається до вимірювальної мережі через спеціальний пристрій узгодження. Для КПК написане спеціальне програмне забезпечення.
- Програма на КПК здійснює функції сервера, надаючи клієнтам інформацію, зчитувальну із приладів, по запиту із глобальної мережі Інтернет. Доступ серверної частини в Інтернет організується за допомогою встановлюваного в КПК модуля GPRS через операторів мобільного зв'язку GSM, що підтримують цей сервіс. ПЗ дозволяє відслідковувати стан зв'язку й при її обриві автоматично відновлювати з'єднання. Реалізовано функцію реєстрації на сервері динамічної IP адреси, виділюваного КПК у момент з'єднання, що дозволяє підтримувати постійний зв'язок між клієнтами й сервером.
- Для ПК написана спеціальна програма-клієнт, що дозволяє шляхом запиту через мережу Інтернет одержати доступ з будь-якої точки миру до інформації, зчитувальної КПК.

Клієнтська ПЗ також підтримує технологію реєстрації на сервері динамічного IP адреси, надаваного КПК.

Розробка структурної схеми

Розглянемо розроблену структурну схему (рисунок 8.) з об`єкту спостереження до кінцевої точки оператора.

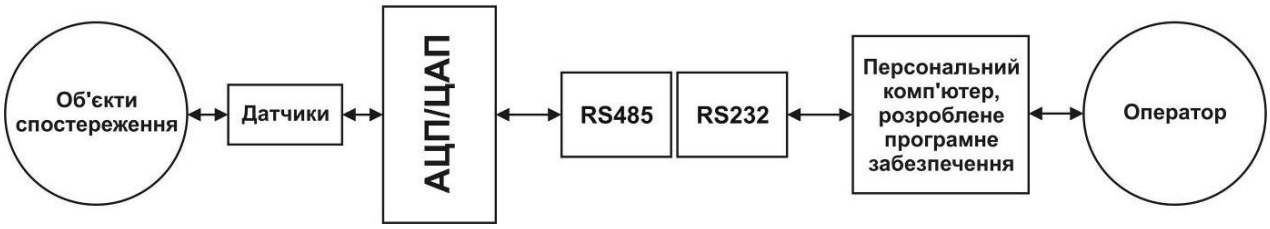


Рисунок 8 – Структурна схема системи

Взаємодія оператора з об'єктами спостереження в системі мікроклімату складського комплексу відбувається за наступною схемою Спочатку у складському приміщенні у відповідних бункерах встановлюються датчики вимірювання вологості в залежності від моделі бункера зберігання використовуються різні датчики в нашому випадку це RKF/A далі за допомогою 32 канального АЦП/ЦАП дані обробляються та за допомогою перетворювача інтерфейсів з RS485 – RS232 відбувається взаємодія з персональним комп'ютером та розробленим програмни забезпеченням яким керує оператор.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів мікроклімату складського комплексу. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем мікроклімату складського комплексу; Досліджена система мікроклімату складського комплексу; На основі отриманих результатів досліджень створена програмна реалізація системи мікроклімату складського комплексу; Розроблені під час виконання роботи алгоритми дозволяють успішно вирішувати завдання мікроклімату складського комплексу; Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.
2. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.
3. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.
4. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.
5. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – С. 85-88.
6. Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 1(126). – С. 75-79.
7. Коваленко А.С. Обґрунтування набору даних для оцінки технічного стану інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2015. – Вип. 1(42). – С.39-41.

8. Коваленко А.С. Експертна система технічного діагностування інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2015. – № 1(41). – С. 106-111.
9. Коваленко А.С. Удосконалення методу технічного обслуговування об'єктів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко, О.П. Доренський // Системи озброєння і військова техніка. – Х.: ХУПС, 2016. – № 2(46). – С. 109-114.
10. Коваленко А.С. Метод визначення оптимального комплексу робіт з відновлення працездатності інтегрованої системи технічної діагностики в умовах ресурсних обмежень / А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2016. – Вип. 3(140). – С. 69-72.
11. Kovalenko A.S. Information model and its element for displaying information on technical condition of objects of integrated information system / A.S. Kovalenko, A.A. Smirnov, A.V. Kovalenko, A.P. Dorensky // International Journal of Computational Engineering Research (IJCER). – India: Delhi, 2016. – Volume 6, Issue 1. – P. 21-27.
12. Кожанова А.С. Система технічної діагностики інтегрованих інформаційних систем – обґрунтування необхідності створення, визначення понятійного апарату та напрямів досліджень / А.С. Кожанова, О.А. Смірнов, М.П. Савченко, Д.М. Ізосімов, В.В. Мороз // Створення та модернізація озброєння і військової техніки в сучасних умовах: Тринадцята наук.-техн. конф., 5-6 вер. 2013 р., м. Феодосія: тези доп. – Феодосія: ДНВЦ, 2013. – С. 187-188.
13. Кожанова А.С. Визначення основних напрямків досліджень щодо створення системи технічної діагностики інтегрованих інформаційних систем / А.С. Кожанова, О.А. Смірнов, А.В. Челпанов // Проблемні питання розвитку озброєння та військової техніки Збройних Сил України: IV наук.-техн. конф., 16-20 груд. 2013 р., м. Київ: зб. тез. – Київ: ЦНДІ ОВТ ЗСУ, 2013. – С. 293.
14. Коваленко А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформатика та системні науки : V Всеукр. наук.-практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 2014. – С. 292-294.
15. Коваленко А.С. Задачи распознавания ситуаций в системах организационной стратегии интеграции производства и операций / А.С. Коваленко, А.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVI міжнар. наук.-практ. сем., 11-12 квіт. 2014 р., м. Кіровоград: зб. тез. – Кіровоград: КНТУ, 2014. – С. 53-55.
16. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2014. – С. 241.
17. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп'ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 28-31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 2014. – С. 190-193.
18. Коваленко А.С. Основні складові та функції системи технічної діагностики інтегрованих інформаційних систем / Коваленко А.С. // Інформаційні технології та комп'ютерна інженерія: наук.-практ. конф., 4 груд. 2014 р., м. Кіровоград: зб. тез доп. – Кіровоград: КНТУ, 2014. – С. 236.
19. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2015. – С. 15.
20. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

УДК 004

В. Большов, магістр гр. КІ-21М-1,4
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ТЕРМІНАЛІВ МЕРЕЖІ ПЛАТІЖНИХ АВТОМАТІВ

У статті розроблено програмне забезпечення, яке призначено для системи терміналів мережі платіжних автоматів. Метою розробки є дослідження та програмна реалізація системи терміналів мережі платіжних автоматів. Об'єктом дослідження є процес терміналів мережі платіжних автоматів. Предметом дослідження є методи терміналів мережі платіжних автоматів. Методи дослідження базуються на методах теорії комп'ютерних систем та мереж, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи терміналів мережі платіжних автоматів. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерна інженерія, платіжні автомати

Постановка проблеми. Сучасний світ неможливо уявити без мобільного зв'язку. Він проник в усі ніши нашого життя. Причому сучасний мобільний зв'язок виконує не тільки роль телефонного зв'язку, але є й могутнім центром телекомунікаційних засобів передачі інформації. GSM, CDMA, GPRS, EDGE, 3G, 4G, 5G – це далеко не повний перелік стандартів стільникового зв'язку та технологій передачі різного виду даних, починаючи від суто телефонної розмови і відеозв'язку й закінчуючи мобільним з'єднанням з мережею INTERNET [1-3]. Але як відомо за усе в житті приходиться платити. І тут виникає наступна задача, що потребує вирішення: як можна спростити процедуру оплати мобільного зв'язку. Звісно можна зайти до магазину й поповнити рахунок у продавця. Та на даному етапі розвитку техніки, існує й альтернативний варіант оплати витрат з мобільного зв'язку – це застосування автоматів самообслуговування. Термінал мережі платіжних автоматів (ТПМА) дозволяє повністю автоматизувати різні елементи процесів торгівлі й обслуговування залежно від розглянутої галузі, представляючи надійне й функціональне рішення, що дозволяє приймати наявні платежі [1-5]. Термінал мережі платіжних автоматів поставляється у вандалостійкому виконанні, що припускає здатність витримувати агресивні впливи з боку зовнішнього середовища зі збереженням повної працездатності [6]. Термінал мережі платіжних автоматів ідеально підходить для установки на частково охоронюваних територіях адміністративних будинків, торгових центрів і т.д.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи терміналів мережі платіжних автоматів.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи терміналів мережі платіжних автоматів.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем терміналів мережі платіжних автоматів.
- Дослідження системи терміналів мережі платіжних автоматів.
- Програмна реалізація системи терміналів мережі платіжних автоматів.

Об'єктом дослідження є процес терміналів мережі платіжних автоматів.

Предметом дослідження є методи терміналів мережі платіжних автоматів.

Методи дослідження базуються на методах теорії комп'ютерних систем та мереж, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Опис GSM/GPRS

Як було відмічено вище, для організації обміну інформацією між автоматом і віддаленим комп'ютером (сервером), використовується технологія бездротового зв'язку GPRS або GSM. Розглянемо ці технології.

Опис стандарту GSM

Стандарт GSM відноситься до другого покоління стандартів для стільникового зв'язку, заснованому на цифрових технологіях. Реалізоване в системах GSM повношвидкісне кодування мови дозволяє зробити її якість порівнянною з якістю стаціонарних телефонних мереж. Радіотелефон стандарту GSM можна умовно розділити на дві частини: абонентський модуль SIM (SIM-карта) і безпосередньо сам телефон, що містить апаратне й програмне забезпечення. SIM-карта служить підтвердженням дійсності абонента й містить у своїй пам'яті всі необхідні дані, пов'язані з повноваженнями конкретного абонента. Щоб викрадач не зміг нею скористатися, у неї вводять спеціальний ідентифікаційний номер (PIN-код). Використання SIM-карти також зручно тим, що при зміні апарата абонентові не потрібно міняти свій мобільний номер, він просто переставляє карту, і всі збережені на ній дані, включаючи записну книжку, стають доступними в новому апарату. Коли SIM-карти немає в апарату, доступ до абсолютної більшості послуг закритий, за винятком екстрених викликів (якщо дозволяє мережа). Виготовити дублікат SIM-карти дуже складно й у сукупності з функціями захисту, вона дає високий рівень захисту користувачів і мереж від несанкціонованого доступу.

Можливості GSM

У стандарті GSM уведено кілька функцій захисту. У першу чергу це шифрація радіоканалу, що виключає прослуховування третьою стороною, а також захист номера абонента (для запобігання розкриття його місцезнаходження). Крім стандартних можливостей, надаваних операторами стільникового зв'язка – місцевий, міжміська й міжнародний зв'язок, переадресація виклику й інших, телефони стандарту GSM дають своїм власникам ряд додаткових функцій: збереження мовних повідомлень, що надійшли в період, коли абонент був недоступний (голосова пошта), прийом повідомлення про факс, що прийшов (факс-пошта), визначення номера що дзвонить. Передбачено можливість передачі коротких повідомлень "із точки в точку" (пейджингу), тобто абоненти при бажанні можуть обмінюватися простими короткими (кілька десятків символів) повідомленнями (тарифи на цю послугу нижче, ніж на звичайні переговори). Функція мобільного модему/факсу поряд з повсюдним поширенням портативних комп'ютерів дає можливість доступу до Інтернету й електронної пошти через мережу GSM. Ці послуги значно збільшують привабливість використання телефонів GSM для користувачів. Так, приміром, факс-пошта може бути дуже корисна діловій людині, тому що дозволяє не пропустити інформацію про факс у будь-який час, незалежно від місцезнаходження абонента. Мобільний телефон сповістить свого власника, а той може одержати факс коли завгодно й де завгодно, тому що факс автоматично надходить у його електронну поштову скриньку.

Технологія GPRS

Нові можливості GPRS дозволили ввести принципово нові послуги, які раніше не були доступні. Насамперед це мобільний доступ до ресурсів Internet із задовольняючого споживача швидкістю й з дуже вигідною системою тарифікації. Приміром, за допомогою системи GPRS абонент може переглядати WEB-сторінки в Internet стільки, скільки йому необхідно, оскільки плата стягується тільки за обсяг прийнятої інформації, а не за час знаходження в мережі Internet. При введенні погодинної оплати на фіксованих телефонних лініях, тарифи на доступ в Internet з мобільного GPRS-телефону будуть ще більш конкурентоспроможні. Технологія GPRS дозволить швидко передавати й одержувати

більші обсяги даних, відеозображення, музичні файли стандарту MP3 і іншу мультимедійну інформацію. Для корпоративних користувачів система GPRS може послужити відмінним інструментом для забезпечення безпечного й швидкого доступу співробітників до корпоративних мереж підприємств, до поштових, інформаційних серверів, віддаленим базам даних. Технології GPRS може застосовуватися в системах телеметрії: пристрій може бути увесь час підключений , не займаючи при цьому окремий канал. Така послуга може бути затребувана службами охорони, банками для підключення банкоматів і в інших областях, у тому числі й промислових.

Мережі GPRS-GSM

Щоб підтримувати GPRS, мережі GSM повинні бути доповнені двома основними функціональними елементами: Serving GPRS Support Node (SGSN) – вузол, яким грає в мережі GPRS ту ж роль, що й комутатор MSC і гостьовий регістр VLR у мережі GSM: і Gateway GPRS Support Node (GGSN) – еквівалент блоку IWF (interworking unit) у мережі GSM:

- SGSN – відповідає за доступ абонентів у мережу GPRS (автентифікацію), а також моніторинг абонентів і трафика. Він служить для автентифікації встаткування й фіксує всі «дії» мобільного абонента в мережі.
- GGSN служить гейтом, тобто, з'єднує мережу GPRS і зовнішні мережі загальнодоступні або частні.

Втім, говорити про два основні елементи можна лише з деякою натяжкою, оскільки кожен контролер у мережі GSM повинен бути дообладнаний пристроєм PCU (блок керування пакетним зв'язком).

Додамо, що в мережі GSM-GPRS з'являється кілька нових інтерфейсів, кожний із префіксом «G».

- Gp – це протокол Frame Relay, однак, перехід до технології UMTC дозволить впровадити замість Frame Relay протокол ATM (Asynchronous Tr.mfer Mode) – це, до речі, викличе перехід до комутаторів ATM від сьогоднішніх комутаторів MSC.
- Gr – протокол з комутацією з'єднань типу C7.
- Gn – протокол пакетної передачі типу TCP/IP (Transmisson Control Protocol і Internet Protocol).

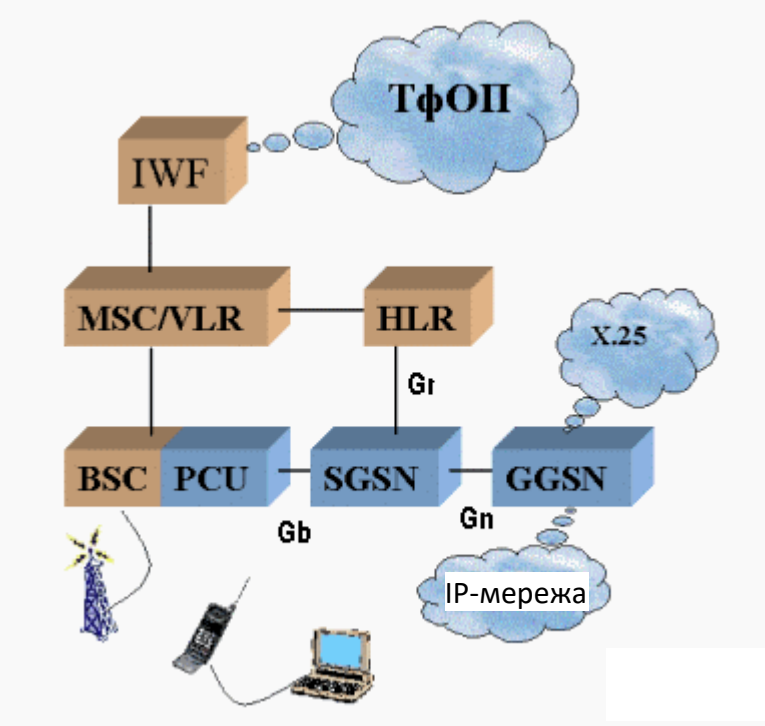


Рисунок 1 – Структура мережі GSM-GPRS

Підключення по GPRS

Набір додатків по оцінках фахівців може включати:

- e-mail;
- доступ до Internet;
- телеметрію;
- підтримка протоколу бездротових додатків (WAP);
- підтримка протоколу передачі файлів (FTP).

Електронна пошта – це послуга, що забезпечує абонентів стільникової мережі (клієнтів) можливістю одержувати й відсилати повідомлення по мережі Internet без необхідності доступу до комп'ютера. Кожний, хто має адресу в Internet, зможе відіслати повідомлення будь-кому, чия електронна адреса вам відомийа. Джерелом адреси для вас може бути ISP (Internet Service Provider – провайдер) або оператор стільникової мережі, якщо він візьме на себе ці функції. У технічному плані – це комп'ютер, підключений до Internet, що буде зберігати отримані на ваше ім'я повідомлення поки ви їх не заберете (наприклад, до пам’яті телефону або вашого домашнього комп'ютера).

Доступ до Internet. Web – це розподілена інформаційна система, заснована на технології сервер-клієнт. Клієнтська програма, це ваш браузер (Internet Explorer, Netscape, Linux або який-небудь інший). Ви можете підключитися браузером до сайту, де зберігаються сторінки, що цікавлять вас, включаючи текст, графіку, а також мультимедіа (відеофрагменти, музика).

Телеметрія. Віддалений моніторинг сайтів в Internet або якого-небудь устаткування – створені нібито спеціально для GPRS, оскільки «підривний» характер передачі даних надзвичайно вдало вписується в технологію пакетного зв'язку GPRS.

WAP. Протокол Бездротових Додатків – WAP зараз рекламується як якийсь прорив, що дозволить забезпечити тотальний доступ до інтернет-ресурсам з мобільних пристроїв, насамперед телефонів. Число WAP-сумісних ресурсів в Internet зараз росте швидкими темпами, з'явилися й перші українські ресурси.

FTP. Це сервіс, призначений для копіювання файлів з одного комп'ютера на інший. Може використовуватися в торговельних й бізнес-додатках.

Стандарти GPRS-терміналів

Клас А. Термінал дозволяє одночасно здійснювати передачу мови й даних у режимі GPRS.

Клас В. Термінал підтримує й голосове з'єднання, і передачу даних у пакетному режимі (GPRS), але або режими використовуються не одночасно (під час передачі даних через GPRS абонент не може робити й приймати голосові дзвінки й навпаки).

Клас С. Термінал забезпечує тільки передачу даних у пакетному режимі. Найбільш імовірне виконання – у вигляді PCMCIA-карти, установлюваної в портативний комп'ютер.

Швидкість прийому й передачі інформації залежить від можливостей конкретної моделі мобільного термінала, а саме від кількості каналів, що підтримують прийом і передачу даних. При цьому один канал підтримує передачу інформації з максимальною швидкістю 13.4 кб/с. Повна мобільність має на увазі, що людині повсюдно стають доступні всі можливості, які він має на своєму робочому місці, такі як швидкісний доступ в Internet. GPRS (General Packet Radio Service) – технологія, що робить це реальним уже сьогодні. Суть послуги полягає в організації постійного підключення через GPRS-телефон до мережі Internet. Для роботи в мережі можливо використовувати комп'ютер (наприклад, ноутбук) або електронний органайзер (Palm Pilot, Psion, Cassiopea). При цьому абонент має можливість переглядати HTML-сторінки, перекачувати файли, працювати з електронною поштою й будь-якими іншими ресурсами Internet. Чим приваблива ця технологія:

- GPRS надає негайний доступ до послуг, без необхідності додзвонюватися до інтернет-провайдеру.
- Користувачі GPRS одержують доступ до Internet у повному обсязі, як при провідному з'єднанні.

- Можна працювати з WAP-сайтами безпосередньо з телефонного апарата GPRS.
- Оплачується тільки обсяг посланої/отриманої інформації, а не ефірний час. Дотепер у стільникових мережах для передачі або прийому даних абонентом займався цілий канал на час від установлення з'єднання до його розриву, що оплачувалося поза залежністю від його завантаження.
- В GPRS максимально можлива швидкість передачі даних становить 171,2 Кбіт/с – це більш ніж в 3 рази швидше, ніж режим роботи провідних ліній, і майже в 12 разів швидше роботи передачі даних у звичайних мережах GSM (9,6 Кбіт/с). Уже сьогодні доступна швидкість до 33 Кбіт/с.

EDGE

EDGE (Enhanced Data rates for Global Evolution) – у перекладі з англійського – це вдосконалена технологія передачі даних для глобального розвитку, що забезпечує високошвидкісну передачу великих обсягів інформації, характерну для мереж третього покоління мобільного зв'язку.

Технологія EDGE надається абонентам на базі послуги GPRS . Технологія EDGE забезпечує швидкість передачі даних, у середньому в три рази перевищуючу можливості GPRS, а також – більш ефективне використання частотних ресурсів і поліпшення покриття мережі в порівнянні зі звичайною мережею стандарту GSM. Технічні особливості EDGE дозволять абонентам більш оперативно працювати з Інтернет-ресурсами (web, wap, e-mail, ICQ) і користуватися найширшим спектром неголосових послуг, що вимагають високих швидкостей передачі даних. Виявившись у зоні дії базової станції з підтримкою EDGE, телефон абонента буде автоматично використовувати EDGE замість традиційного GPRS. Швидкість з'єднання залежить від кількості одночасних користувачів, класу телефону з підтримкою EDGE, умов радіоприйома й співвідношення сигнал/шум. Середня швидкість у мережі 80-120 Кбіт/с.

Опис програмного забезпечення

Програмне забезпечення повинне забезпечувати наступні функції:

1. Первісне налаштування.
2. Сервісне обслуговування автоматів.
3. Установка й налаштування таймера.
4. Налаштування додаткових параметрів через командний рядок.
5. Моніторинг функціонування терміналів.
6. Робочий режим.

Для коректної роботи ПЗ необхідно, щоб на автоматі самообслуговування було попередньо встановлене наступне програмне забезпечення:

- Операційна система Windows 10/11.
- MS Internet Explorer версії 5 або вище.

Крім того, перед початком роботи треба переконатися, що в операційній системі автомата настроєне з'єднання з Інтернет. У якості Інтернет-з'єднання може використовуватися як канал GPRS, так і будь-який інший доступний спосіб з'єднання. Для доступу в Інтернет ПЗ використовує налаштування Internet Explorer, тому у випадку використання proxy-сервера, при підключенні автомата в локальну мережу, необхідно додати proxy у налаштування «Internet Explorer».

Розглянемо перераховані вище функції більш докладно.

1. Первісне налаштування.

При першому запуску системи вам необхідно настроїти дані для авторизації автомата в системі оплати послуг мобільних операторів. Спершу треба перейти в параметри авторизації, тобто ввести значення параметрів:

– Номер термінала – ідентифікатор термінала в системі оплати послуг оператора мобільного зв'язку (ТПМА). У модулі адміністратора системи ТПМА повинен бути зареєстрований термінал, під яким будуть проводитися платежі. Терміналу повинен бути

заданий тип Автомат ТПМА. Неприпустиме використання того самого номера термінала декількома пристроями. Для кожного автомата, реєструйте в системі новий термінал:

– Ім'я користувача – логін користувача, від імені якого будуть відбуватися платежі.

– Пароль – пароль користувача, від імені якого будуть відбуватися платежі.

– Підтвердження пароля – у дане поле необхідно повторно ввести пароль користувача для підтвердження його коректності.

У якості логіна й пароля повинні використовуватися дані персони, зареєстрованої в системі ТПМА із правами «Продавець». Припустимо використання однакових даних користувача в різних автоматах.

Після цього необхідно настроїти параметри сервісного доступу автомата, тобто ввести налаштування для входу в режим обслуговування й інкасації автомата:

- Секретний номер телефону – задається номер телефону, що дозволяє перевести автомат у сервісний режим і потрапити в панель налаштування.

- Ім'я користувача – логін користувача, що буде здійснювати сервісне обслуговування автомата.

- Пароль – пароль користувача, що буде здійснювати сервісне обслуговування автомата.

- Підтвердження пароля – у дане поле необхідно повторно ввести пароль користувача для підтвердження його коректності.

Користувач, під ім'ям якого буде здійснюватися сервісне обслуговування автомата не повинен бути зареєстрований у системі ТПМА. Тому, для логіна й пароля ви можете використовувати будь-яку комбінацію символів. Реєструвати в ТПМА даного користувача не потрібно.

Після цього етапу відбувається налаштування системи Параметри з'єднання з Інтернет і опції автомата:

- З'єднання з Інтернет – у даному полі прапором відзначте те з'єднання, через яке автомат буде здійснювати комунікацію із системою ТПМА.

- Задати рядок ініціалізації модему – дозволяє вказати нестандартний рядок ініціалізації модему для поточного обраного з'єднання.

- Включити стиск даних, переданих через Інтернет – дозволяє включити додатковий стиск даних, переданих через Інтернет з метою економії трафика.

- Включити мультиканальний режим – включення мультиканального режиму.

- Опції автомата – панель додаткових опцій автомата:

- Зупиняти автомат при помилках купюроприймача – даний прапор дозволяє припинити прийом платежів при виявленні яких-небудь проблем у роботі купюроприймача.

- Зупиняти автомат при помилках принтера – даний прапор дозволяє припинити прийом платежів при виявленні яких-небудь проблем у роботі принтера чеків. При знятому прапорі автомат буде продовжувати приймати платежі, навіть якщо в принтері закінчився папір для печаті чеків.

- Зупиняти автомат при відсутності засобів на рахунку агента – даний прапор дозволяє припинити прийом платежів на автоматі при відсутності засобів в агента.

- Зупиняти автомат при відсутності зв'язку – даний прапор дозволяє припинити прийом платежів при відсутності зв'язку з автоматом протягом зазначеного періоду часу (за замовчуванням, 15 хвилин).

- Не перезапускати програмно модем у випадку відсутності сторожового таймера – відзначте прапор, щоб відключити опцію програмного перезавантаження модему у випадку відсутності сторожового таймера.

Після цього автомат переходить у робочий режим. Якщо в автомат не завантажені файли інтерфейсу, автомат скачує їх автоматично із сервера ТПМА. Даний процес може зайняти досить тривалий час (мінут 30-40 при використанні GPRS з'єднання), при цьому на екрані буде показане повідомлення «Вибачте, автомат тимчасово не працює». У робочому

режимі в деяких випадках можлива поява наступного напису: «Вибачте, автомат тимчасово не працює» Даний напис означає, що були виявлені проблеми з купюроприймачом або принтером чеків.

2. Сервісне обслуговування автоматів.

Воно складається з наступних підрозділів:

- Перехід у режим обслуговування й інкасації.
- Сервісне меню.
- Зміна провайдеру GPRS зв'язку.
- Мультиканальний режим роботи модему.

Розглянемо їх більш докладно.

Перехід у режим обслуговування й інкасації. Для цього необхідно вибрати опцію Оплата послуг, ввести номер телефону, що задали при налаштуванні автомата й розблокувати сервісний режим за допомогою імені користувача й пароля.

Сервісне меню. Містить такі дані як:

- Версія інтерфейсу.
- Кількість купюр у купюроприймачі й суму.
- Статус купюроприймача.
- Статус принтера.
- Статус з'єднання.
- Статус сторожового таймера.
- Статус платежів.

За допомогою кнопок у вікні сервісного меню можна виконувати також наступні дії:

- Змінити номер термінала, логін і пароль ТПМА.
- Змінити параметри входу в сервісний режим.
- Змінити параметри Інтернет і опції автомата.
- Налаштування інтерфейсу.
- Налаштування e-mail оповіщень.
- Налаштування Multi SIM підключення.
- Видалити файл конфігурації й запустити знову програму.
- Переглянути лог.
- Запустити відновлення.
- Видалити файл конфігурації й запустити знову програму.
- Вийти й запустити знову програму.
- Настроїти параметри безпеки при роботі з додатком.

Крім цього, дозволяє виконувати й контролювати наступні дії:

- Проведення інкасації.
- Налаштування параметрів принтера.
- Рівень сигналу GSM мережі.
- Налаштування одержання балансу.
- Налаштування оповіщень.
- Налаштування параметрів сторожового таймера.
- Налаштування інтерфейсу.
- Режим підтримки двох SIM-карт.
- Безпека.
- Режим блокування системних комбінацій клавіатури.

Зміна провайдеру GPRS зв'язку. Призначено для введення параметрів передачі даних по мережі GSM, операторів, які не прописані в автоматі ТПМА.

Мультиканальний режим роботи модему. Даний режим дозволяє здійснювати одночасно наступні дії:

- передача даних по Інтернет;
- відправлення SMS;
- одержання рівня сигналу й інших даних про стан GSM мережі;

- відправлення USSD запитів балансу SIM карти.

Таким чином, використання даного режиму дозволяє, не розриваючи модемне з'єднання з Інтернет, одержувати безупинно дані про баланс SIM карти, коливання рівня сигналу й дані про стан GSM мережі. Підключення мультиканального режиму відбувається в кілька етапів:

- Установка драйвера для мультиканального режиму.
- Установка стандартного драйвера модему.
- Створення модемного з'єднання.
- Налаштування ПЗ автомата.

3. Установка й налаштування таймера.

Таймер дозволяє здійснювати дві основні функції:

- Перезавантаження модему, при відсутності відгуку від модему протягом деякого часу. Даний режим роботи включений за замовчуванням.
- Перезавантаження комп'ютера при відсутності сигналу від комп'ютера протягом 30 мінут.

Таким чином, за допомогою плати відслідковується «зависання» комп'ютера або модему й коректуються дані проблеми шляхом перезавантаження модему або комп'ютера. Період в 30 мінут для перезавантаження комп'ютера обраний з метою виключення ситуацій помилкового спрацьовування: при відновленні даних із сервера й т.п.

4. Налаштування додаткових параметрів через командний рядок.

За допомогою командного рядка ви можете вказати параметри, відмінні від параметрів за замовчуванням для принтера й купюроприймача. Вони підрозділяються на:

- Параметри принтера й купюроприймача.
- Додаткові параметри.

5. Моніторинг функціонування терміналів.

Для зручності відстеження роботи ваших автоматів самообслуговування, у системі ТПМА передбачена можливість віддаленого моніторингу в реальному часі. Дана функціональність доступна користувачам із правами головного менеджера, і містить у собі:

- Моніторинг через Інтернет.
- Моніторинг за допомогою мобільного телефону.

6. Робочий режим.

Його робота полягає у виконанні трьох основних функцій:

- Прийом платежів. Інтерфейс для прийому платежів розроблений таким чином, щоб максимально спростити процес звернення клієнта до автомату і зробити прийом платежів за допомогою автомата інтуїтивно зрозумілим без додаткових інструкцій.
- Відновлення ПЗ. Після виходу нової версії ПЗ, автомат автоматично завантажує відновлення, не припиняючи роботу. Також автоматично відбувається відновлення зовнішнього вигляду інтерфейсу автомата: додавання можливості оплати послуг нових провайдерів і т.п.
- Режим автоматичного включення автомата. Призначений для того, щоб настроїти автоматичне включення автомата після збоїв електрики в мережі.

Розробка структурної схеми

Створене програмне забезпечення призначене для забезпечення користувачів послугою поповнення рахунку мобільного телефону. Воно універсальне, може бути використана терміналах фірми «Об'єднана система швидких платежів» серії: ОСМП-2; ОСМП-МІНІ; ОСМП-ВУЛИЦЯ; ОСМП 2 з лайт-боксом; ОСМП 2 з двома моніторами. На рисунку 2 показана структурна схема програмного забезпечення, розглянемо її з верху долілиць.

Рисунок 2 – Структурна схема програмної частини

Коли на термінал надходить запит обслуговування, невідомо користувач є адміністратором чи ні. Для проходження автентифікації крім магнітного ключа адміністраторові необхідно знати системний 9 значний номер телефону перекладу термінала в інтерфейс адміністратора.

Коли магнітний ключ приєднаний до термінала, й користувач ввів системний номер телефону, на екрані з'являється форма автентифікації, що запитує ім'я користувача й пароль, при правильному введенні даних користувачеві привласнюється статус адміністратора й автомат переходить у режим налаштування, у противному випадку користувачеві дається ще дві спроби після відбування, яких відбувається включення тривоги, передача сигналу на базовий ПК і повне блокування термінала.

Адміністратор повністю управляє комп'ютером, він може управляти всіма налаштуваннями й опціями термінала: статус з'єднання; встановлення рядка ініціалізації модему; встановлення архівації та кодування даних; кількість купюр у купюроприймачі й суму; статус купюроприймача; статус принтера; статус сторожового таймера; статус платежів; налаштування інтерфейсу; налаштування e-mail оповіщень; налаштування Multi SIM підключення; видалення файлу конфігурації й перезапуск терміналу.

Також за допомогою кнопок у вікні сервісного меню можна управляти наступними опціями: зупиняти авт. при помилках купюроприймача; зупиняти автомат при помилках принтера; зупиняти автомат при відсутності засобів на рахунку користувача; зупиняти автомат при відсутності зв'язку; змінювати номер термінала, логін і пароль; змінити параметри входу в сервісний режим; переглядати лог. файл.

Також адміністратор може зробити інкасацію коштів. Виконання даної операції здійснюється через комп'ютерний відсік шляхом зняття грошової касети із кріплення купюроприймача, попередньо запустивши процес інкасації в програмному забезпеченні термінала.

Якщо системний номер адміністратора не був введений, термінал переходить в інтерфейс користувача, перемикається на режим обслуговування клієнтів, і користувач,

натискаючи на сенсорній панелі кнопки, вибирає необхідні розділи, після чого відбувається видача чека з поповненням рахунку мобільного телефону.

На рисунку 3 можна побачити складові частини термінала, а саме: TFT 17" вандалостійкий сенсорний монітор; Вандалостійкий корпус; IBM PC сумісний комп'ютер; Пристрій для печатки бланків; Пристрій для прийому грошей.

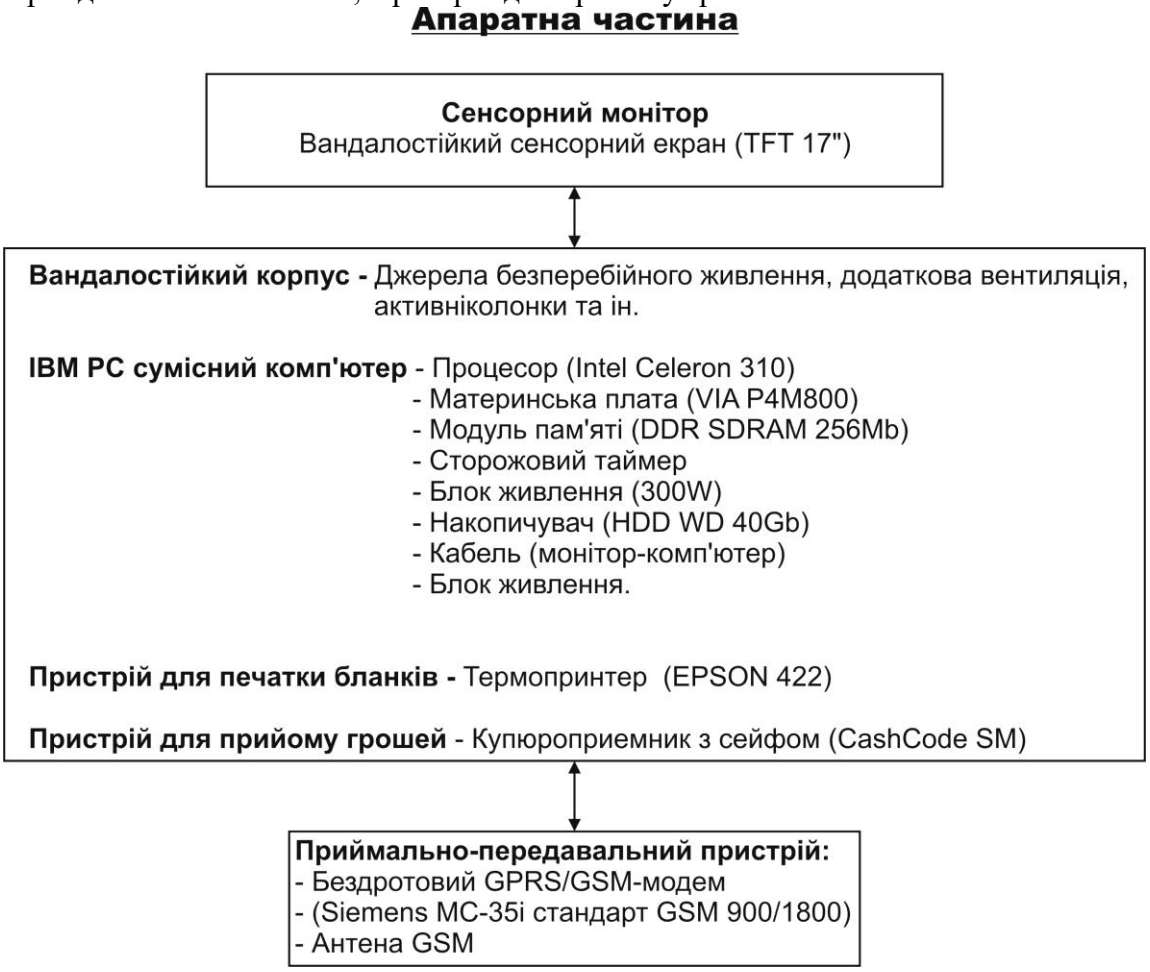


Рисунок 3 – Структурна схема апаратної частини

Розглянемо докладніше основні елементи терміналу. Серце термінала комп'ютерний відсік, що знаходиться у вандалостійкому корпусі. Являє собою комп'ютерну частину термінала й вузол об'єднання всіх пристроїв у єдину систему. Включає у свій состав IBM PC сумісний комп'ютер, розташований у верхній частині термінала (рисунок 4), що складається з наступних складових елементів**:**

– Процесор (Intel Celeron 310);
– Материнська плата (VIA P4M800);
– Модуль пам'яті (DDR SDRAM 256Mb);
– Сторожовий таймер;
– Блок живлення (300W);
– Накопичувач (HDD WD 40Gb);
– Кабель (монітор-комп'ютер)
– Блок живлення.

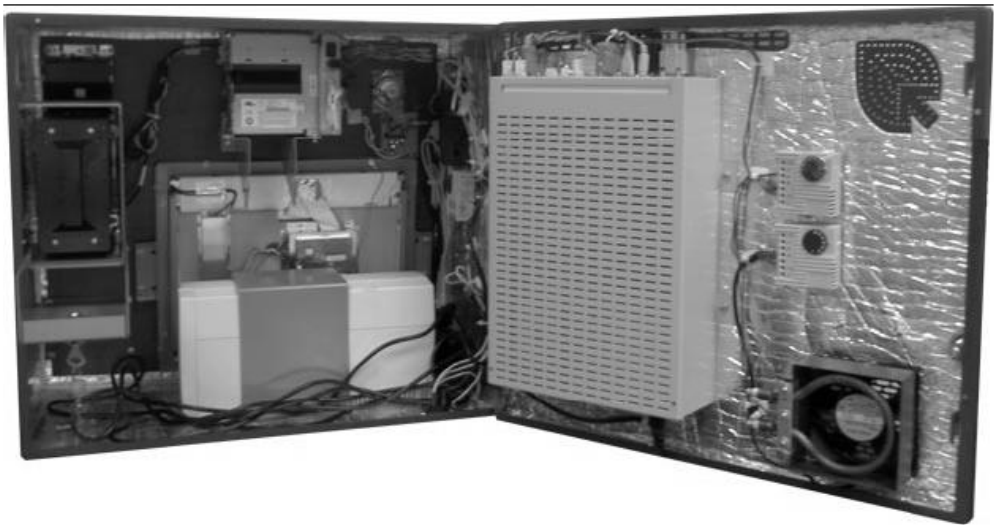


Рисунок 4 – Комп'ютерний відсік

Як видно з основних характеристик комп'ютерного відсіку комп'ютер повністю сполучимий з настільним ПК, у ньому встановлена операційна система Windows 10/11 (залежно від конфігурації складових елементів) з відповідними перевагами й недоліками. Всі вузли термінала (сенсорний монітор, принтер, GPRS/GSM модем, і т.д.) підключаються через стандартні роз'ми ПК. Основна відмінність від настільного ПК є форма корпуса, який поміщено у вандалостійкий корпус (рисунок 4).

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів терміналів мережі платіжних автоматів. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем терміналів мережі платіжних автоматів; Досліджена система терміналів мережі платіжних автоматів; На основі отриманих результатів досліджень створена програмна реалізація системи терміналів мережі платіжних автоматів; Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Kovalenko Oleksandr Qualitative risk analysis of software development / Oleksandr Kovalenko, Jamil Al-Azzeh, Oleksii Smirnov, Anna Kovalenko, Serhii Smirnov // Asian Journal of Information Technology. – Volume 17 Issue 3. – Medwell Journals. – 2018. – P. 218-230. ISSN: 1682-3915. URL: http://medwelljournals.com/abstract/?doi=ajit.2018.218.230 Doi: ajit.2018.218.230
2. Kovalenko Oleksandr, The mathematical model of the testing technology for DOM XSS vulnerabilities / O. Kovalenko, O. Smirnov, A.Kovalenko, S. Smirnov, V. Vialkova // Scientific & practical cyber security journal (SPCSJ) Volume 2 Issue 1, P. 22-28. Georgia. Tbilisi. Scientific Cyber Security Association (SCSA), 2018 ISSN: 2587-4667. URL: https://journal.scsa.ge/wp-content/uploads/2018/12/04-3-o.kovalenko-a.kovalenko-o.smirnov-s.smirnov-v.vialkova.pdf
3. Коваленко А.В. Технология тестирования DOM XSS уязвимости / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // Scientific & practical cyber security journal (SPCSJ) Volume 1. Issue 1. P. 38-45 Georgia. Tbilisi. Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667. URL: https://journal.scsa.ge/wp-content/uploads/2018/12/8-dom-xss-testing-technology-vulnerabilities.pdf
4. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.
5. Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.
6. Коваленко А.В. Комплекс математических моделей технологии тестирования web-

приложений / А.В. Коваленко, А.А. Смирнов // Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.
7. Коваленко А.В. Задачи распознавания ситуаций в erp системах / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник наукових праць "Системи обробки інформації". – Випуск 4(120). – Х.: ХУПС – 2014. – С. 161-164.
8. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник наукових праць "Системи обробки інформації". – Випуск 5(142). – Х.: ХУПС – 2016. – С. 153-157.
9. Коваленко А.В. Проблемы анализа и оценки рисков информационной деятельности / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць "Системи обробки інформації". – Випуск 3(140). – Х.: ХУПС – 2016. – С. 40-42.
10. Коваленко А.В. Метод качественного анализа рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 2(23). – Харків: ХУПС. – 2016. – С. 150-158.
11. Коваленко А.В. Метод количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 128-133.
12. Коваленко А.В. Использование псевдобулевых методов бивалентного программирования для управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Системи управління, навігації та зв'язку. – Випуск 1 (37). – Полтава: ПолтНТУ. – 2016. – С. 98-103.
13. Коваленко А.В. Проблемы анализа и оценки рисков информационной деятельности / А.В. Коваленко, А.А. Смирнов // Збірник наукових праць II міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 24-27 лютого 2016 р. – Київ: Європейський університет. – 2016. – С. 138-139.
14. Коваленко А.В. Анализ и оценка рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез «Securitea informationala 2015-2016». Conferenta internationala (editia a XII-a). Chisinau. Moldova. 3 martie 2016. – Chisinau: ADSEM. – 2016. – P. 96-102.
15. Коваленко А.В. Исследование источников и причин риска разработки программного обеспечения, этапов и работ, при выполнении которых возникает риск / А.В. Коваленко, А.А. Смирнов // Збірник тез VII всеукраїнської науково-практичної конференції "Інформатика та системні науки (ІСН-2016)". м. Полтава. 10-12 березня 2016 р. – Полтава.: ПУЕТ – 2016. – С. 264-266.
16. Коваленко А.В. Оценка показателя чистой приведенной стоимости для количественной оценки рисков проекта разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез науково-практичної конференції "Проблеми кібербезпеки інформаційно-телекомунікаційних систем". м. Київ. 10-11 березня 2016 р. – Київ: КНУ ім. Тараса Шевченко – 2016. – С. 81-82.
17. Коваленко А.В. Методика структурной идентификации рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології» (IS&CT). м. Кіровоград. 24-25 березня 2016 р. – Кіровоград: КНТУ. – 2016. – С. 71-72.
18. Коваленко А.В. Методы качественного анализа рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез першої міжнародної науково-практичної конференції «Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2016). м. Харків. 30 березня – 1 квітня 2016 р. – Харків: НТУ «ХПІ». – 2016. – С. 6-7.
19. Коваленко А.В. Структурная идентификация рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез XVIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 15-16 квітня 2016 р. – Кіровоград: КНТУ. – 2016. – С. 175-182.
20. Коваленко А.В. Исследование разработанной методики структурной идентификации рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез VIII міжнародної науково-практичної конференції "Проблеми і перспективи розвитку IT-індустрії". м. Харків. 28-29 квітня 2016 р. – Харків: ХНЕУ. – 2016. – С. 49.

УДК 004

В. Борзенко, магістр гр. КІ-21М-1,4
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДОВІДКОВО-ІНФОРМАЦІЙНОГО СЕРВІСУ КОМП'ЮТЕРНОЇ МЕРЕЖІ ВІДДІЛУ ПІДПРИЄМСТВА

У статті розроблено програмне забезпечення, яке призначено для системи довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства. Метою розробки є дослідження та програмна реалізація системи довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства. Об'єктом дослідження є процес довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства. Предметом дослідження є методи довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства. Методи дослідження базуються на методах теорії телекомунікацій, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, довідково-інформаційний сервіс, комп'ютерна мережа

Постановка проблеми. Сучасний розвиток IT технологій вимагає розробки системи довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства, яка призначення для обліку товарів і послуг. Облік визначатиме товари та послуги, показуючи їх особливу економічну роль, як результати виробничого процесу. Слід зазначити, що увага до товарів і послуг у платіжному балансі приділяється не точці виробництва, а точці, коли вони обмінюються між резидентом і нерезидентом, тобто експортуються чи імпортуються.

У роботі показано взаємозв'язок рахунку товарів і послуг і його балансової статті з іншими міжнародними рахунками. Товари та послуги будуть визначені як результати виробничого процесу, на відміну від доходу та трансфертів.

Обсяг товарів і послуг відповідатиме визначенню виробництва як процесу об'єднання ресурсів для виробництва продукції або надання вироблених основних засобів у розпорядження іншого суб'єкта господарювання.

Виробництво протиставляється доходу від власності, який передбачає передачу невиробленого активу в розпорядження іншого суб'єкта.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завданнь:

– Огляд існуючих систем довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства.

– Дослідження системи довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства.

– Програмна реалізація системи довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства.

Об'єктом дослідження є процес довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства.

Предметом дослідження є методи довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства.

Методи дослідження базуються на методах теорії телекомунікацій, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу

Проектування алгоритмів програмного забезпечення системи довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства.

Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства, який розробляється у даній роботі, дозволяє всьому вашому підприємству мати доступ до тих самих даних у реальному часі на одній платформі. Наприклад, системи продажів, людського капіталу, фінансів, бюджетування, прогнозування та ланцюгів поставок, якими користуються окремо, можна оптимізувати на одній хмарній платформі.

Це не тільки зменшує потребу в IT-підтримці на різних платформах, але й дозволяє вашій організації бути більш стратегічними, оскільки менше часу витрачається на збір даних. Цей спрощений підхід зосереджується на всій організації підприємства, інтегруючи дані, щоб дати організаціям можливість належним чином планувати, продумано реагувати та змінювати те, як вони реагують на зміни в організації.

Окрім підвищення ефективності роботи, впровадження платформи Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства, який розробляється у даній роботі, скеровує організацію до підвищення ефективності загальних бізнес-процесів, інформування про найкращі практики та може призвести до зниження витрат на інтеграцію IT.

Висвітлимо деякі особливості довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства, який розробляється у даній роботі:

Інформаційна панель фінансового директора: традиційно керівництво бореться з керуванням даними. Або необхідні дані недоступні, або повільно надходять до керівника через велику кількість різних відділів, через які вони повинні пройти. Інформаційна панель фінансового директора забезпечує консолідований доступ до даних у реальному часі. Це «єдине вікно» забезпечує численні переваги, зокрема:

– Дані можна аналізувати в режимі реального часу без попереднього агрегування вручну. Консолідовані дані можна використовувати, щоб зосередитися на тому, «що ми пропустили, чому ми це пропустили та як це виправити?» а не маніпулювання даними між платформами Ці зміни мають ще один позитивний результат: щасливіші працівники. Фінансові аналітики, які раніше були просто «жокеями з електронними таблицями», тепер мають чудові сучасні інструменти, що дозволяє їхнім ролям стати більш стратегічними та особистими.

– Керівництво має доступ у режимі реального часу до фінансових даних, у тому числі до грошових потоків. Це особливо цінно в періоди невизначеності, коли компаніям може знадобитися швидко реагувати на збої, щоб забезпечити їхню роботу.

– Завдяки безперервному моніторингу даних у режимі реального часу компанії можуть визначити, чи щось у їхніх даних не відображається. Помилки можна швидко помітити, а проблеми можна пом'якшити раніше, ніж у більш «традиційних» базах даних.

Можливість деталізації: довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства дозволяє користувачам деталізувати дані в реальному часі, щоб визначити, що працює, а що ні. Ця здатність допомагає керівництву відповідати на важливі запитання, зокрема:

– Чи варто нам використовувати наш баланс для злиття та поглинання?.

– Чи регулярно ми оцінюємо дані, щоб виявити шаблони інформації, які можуть вплинути на майбутнє?.

– Чи маємо ми уявлення про галузеві тенденції, які можуть допомогти нам консолідувати наших постачальників?.

Можливості моделювання та прогнозування: довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства Analytics Cloud (OAC) дозволяє користувачам залучати кілька зовнішніх наборів даних, щоб допомогти компаніям прогнозувати майбутні проблеми та тенденції. Наприклад, заклад вищої освіти може підключити базу даних, що містить історичну кількість заявок, щоб оцінити майбутні тенденції заявок на основі поєднання різних програм, які оцінює установа. Здатність моделювати та прогнозувати є величезною цінністю для керівництва, оскільки використання історичних даних спрощено за допомогою OAC, а система є гнучкою, що дозволяє користувачам використовувати дані, які найбільше відповідають їхнім потребам. OAC також дозволяє користувачам запускати сценарій і аналіз чутливості, дозволяючи їм легко визначити, «якщо ми зробимо X, що станеться з Y?» Функція бухгалтерського обліку організації може створювати додаткову цінність за допомогою OAC, оскільки замість інтерпретації даних у кількох електронних таблицях для визначення бюджетів вони можуть аналізувати всю необхідну фінансову інформацію в одному місці.

Завдяки сучасним хмарним інструментам організації не лише виживають у невизначеності, але й реагують на збої за допомогою інновацій та стратегічного мислення, які можуть досягти результатів змін на довгі роки.

Система, яка розробляється, допомагаємо компаніям успішно орієнтуватися у все більш складному бізнес-ландшафті, який продовжує змінюватися та розвиватися. Вона веде клієнтів через виклики сучасного бізнесу, орієнтуючись на переосмислення того, як працює організація, і готуємо їх до використання можливостей завтрашнього зростання за допомогою визначення наступних понять:

– Залучення та ефективність робочої сили.

– Використання фінансових даних і даних про ефективність підприємства для прийняття рішень на основі великих даних.

– Покращення взаємодії з клієнтами на всіх етапах клієнтського шляху.

– Хмарна технологія та ландшафт даних, що розвиваються.

Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства Analytics Cloud

Змістовне використання як структурованих, так і неструктурованих даних може надати цінну інформацію про конкурентоспроможність, стимулювати конкурентну перевагу та сприяти зростанню. Але багато компаній борються з тим, з чого почати.

Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства Analytics допомагає керівникам IT-підрозділів і бізнесів конкурувати, використовуючи довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства Analytics Cloud, бізнес-аналітику та сучасні аналітичні можливості. Система, яка розроблена у даній роботі, допомагає організаціям досягти успіху, дозволяючи їм використовувати та використовувати інформацію у внутрішніх, зовнішніх, структурованих і неструктурованих даних для прийняття рішень, покращення процесів і організації, зниження ризиків і задоволення своїх клієнтів.

Рішення, які пропонуються:

– Дорожні карти аналітики.
– Дизайн аналітики.
– Можливості реалізації OAC.
– Аналітичні можливості Essbase.
– Відкриття великої батареї та прогнозна аналітика.
– Розширена аналітика.
– Звітність та аналітика.

Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства, який розробляється у даній роботі, блок EPM

Фінансові керівники стикаються з безпрецедентними змінами та тиском, щоб зберегти та покращити свою конкурентну перевагу, вимагаючи нового погляду на свої дані.

Вони повинні мати можливість трансформувати свої можливості та перейти на ефективну та гнучку платформу, щоб отримати 360-градусний огляд своїх даних, що дозволить їм приймати обґрунтовані та своєчасні бізнес-рішення, які допоможуть їм збільшити прибутковість і випередити конкурентів.

Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства EPM Cloud допомагає компаніям успішно орієнтуватися у складнощах широкомасштабних ініціатив трансформації EPM і розробляти процеси та процедури потрібного розміру для виконання складних ініціатив трансформації підприємства.

Рішення, які пропонуються:

– Хмарні можливості звітування про продуктивність підприємства.
– Хмарні можливості планування та бюджетування підприємства.
– Фінансова консолідація та близькі можливості.
– Хмарні можливості управління прибутковістю та витратами.
– Можливості стратегічного планування робочої сили.
– EPM Cloud керування та інфраструктура/інструменти управління.
– Оцінки та оптимізація EPM Cloud.
– Рішення для фінансового планування Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства.
– Звірка рахунків.

Компанії постійно стикаються з наслідками швидко мінливих ділових обставин і невизначеності, і їм потрібні моделі прогнозів і аналіз впливу, щоб допомогти передбачити ці зміни та орієнтуватися в них. Рішення для фінансового планування та моделювання сценаріїв вирішують проблему нескінченної зміни клімату та допомагають розвивати динаміку бізнесу. Хоча зміни впливають на кожен бізнес по-різному, кожна організація може використовувати дані для моделювання конкретних сценаріїв для прийняття обґрунтованих бізнес-рішень.

Грошовий потік – це сфера, якою кожна організація хоче керувати як ніколи в цей безпрецедентний час. Деякі зі змінних, які можуть вплинути на ваші прогнози грошових потоків, а також запитання, на які модуль може допомогти відповісти для вашого бізнесу, включають:

– Як відбуватиметься нарощування грошових коштів від операційної діяльності?
– Який грошовий потік необхідний протягом наступних трьох місяців для підтримки наших запланованих бізнес-операцій?
– Що станеться, якщо ми змінимо нашу структуру капіталу?
– Який відсоток нашої дебіторської заборгованості ми стягнемо?
– Як зміниться наш COGS?
– Як ми ставимося до нашої зарплати та змін, які ми внесли або можемо внести до неї?

Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства, який розробляється у даній роботі, блок ERP

Інструменти та структури, які добре служили фінансовим керівникам у минулому, представляють безліч проблем і пасток, оскільки вони намагаються досягти оптимальних позитивних фінансових результатів у передбачуваний спосіб із низьким ризиком.

Пропонуються перевірені методи, які поєднують випробувані та справжні принципи з гнучкими методами для ефективного впровадження. Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства, глибокий досвід вирішення проблем і широка проникливість процесів вносять безпрецедентний досвід у вашу трансформацію.

Рішення, які пропонуються:

- Можливості фінансової хмари.
- Хмарні можливості управління портфелем проектів.
- Хмарні можливості управління ризиками.
- Хмарні можливості закупівель.
- Інфраструктура/інструменти управління хмарою ERP.
- Оцінки та оптимізація ERP Cloud.
- Рішення бухгалтерського центру.

Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства, який розробляється у даній роботі, блок HCM

Сучасні HR системи, процеси та технології створюють новий спосіб ведення бізнесу. Здатність швидко зібрати талановитий потенціал вашої організації, а потім мобілізувати таланти, щоб опинитися на потрібному місці в потрібний час, має вирішальне значення для успіху.

Система, яка розроблена у даній роботі, допомагає клієнтам збільшувати свої інвестиції в програмне забезпечення та вдосконалювати роботу своїх талантів, модернізуючи та оптимізуючи їхні HR-стратегії, процеси та технології. Використовуємо свої знання у сфері кадрових систем, процесів і технологій, щоб надати клієнтам інформацію, необхідну для управління їхнім наскрізним життєвим циклом талантів, розкриття потенціалу своїх співробітників і початку трансформації кадрової сфери за допомогою хмарної технології HCM.

Рішення, які пропонуються:

– Наскрізні можливості довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства HCM Cloud.

- Дорожня карта трансформації HR та розробка бізнес-кейсів.
- Інфраструктура/інструменти управління хмарою HCM.
- Оцінки та оптимізація HCM Cloud.

– Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства Strategic Workforce Planning Cloud.

– Рішення довідково-інформаційного сервісу комп'ютерної мережі відділу підприємства Workforce Health & Safety Solution.

Оптимізація вашої робочої сили відбувається завдяки узгодженню вашої стратегії роботи з людьми та вашої бізнес-стратегії. Надзвичайний тиск на організації через політичні, глобальні чи економічні проблеми вимагає швидких змін і реагування, щоб повернути бізнес на правильний шлях за допомогою плану відновлення. Цей план повинен не тільки відповідати поточним вимогам, але й водночас позиціонувати їх на майбутнє.

Хмарне рішення Strategic Workforce Planning Cloud Solution (SWPCS) є потужним інструментом, який може допомогти організаціям стати спритними та динамічними в ці часи, коли компаніям необхідно заздалегідь передбачати як поточні, так і майбутні потреби в робочій силі. Інструмент має можливості переглядати коротко-, середньо- та довгострокові плани, визначати прогалини в навичках попиту та пропозиції. Це дозволяє компаніям приймати більш обґрунтовані рішення, наприклад, щодо планування спадкоємності, активів або розподілу, варіантів злиття та поглинання та визначення пріоритетів проектів. Нижче наведено кілька сценаріїв, за якими SWPCS можна використовувати для допомоги організаціям:

– Як нам змінити наші поточні демографічні показники на «xyz» у наступні три роки?

– Де нам слід розширити наш географічний слід?

– Як людський капітал із об'єкта придбання змінює рентабельність інвестицій?

– Які нові складські одиниці (SKU) або послуги ми повинні виробляти або продавати?

– Як цифрова трансформація змінює наш розподіл ресурсів?

– Де наш бізнес може стати більш динамічним, враховуючи руйнування, спричинені автоматизацією?

– Як ми ефективно використовуємо додаткові ресурси?

Довідково-інформаційний сервіс комп'ютерної мережі відділу підприємства, який розробляється у даній роботі, блок SCM

Система може допомогти створити та реалізувати план трансформацій, унікальний для вашої галузі та відповідний пріоритетам і культурі вашої організації.

Пропонуються комплексні послуги та рішення з трансформації SCM Cloud та бізнес-інтеграції:

– Хмарні можливості ланцюга поставок.
– Хмарні оцінки та оптимізація.
– Визначення дорожньої карти та плану трансформації.
– Удосконалення процесів, глобальне розгортання, мережа постачання та дизайн.
– Сталі, інноваційні та безпечні бізнес-операції.
– Цілісний підхід до ланцюга постачання та підприємства.
– Бізнес-процес для досягнення операційної досконалості.
– Інфраструктура/інструменти хмарного управління та управління SCM.

Розробка структурної схеми

Автоматизований довідково-інформаційний сервіс комп'ютерної мережі відділу логістики (АДІС) "Облік поставок" призначена для автоматизації процесів керування матеріальними, інформаційними й фінансовими потоками на підприємстві. До задач підсистеми ставляться організація й контроль діяльності підрозділів підприємства за висновком договорів, організації надходження, переміщення й складування матеріальних цінностей, а також збуту готової продукції.

Структурна схема автоматизованої довідково-інформаційної системи комп'ютерної мережі відділу логістики (АДІС) "Облік поставок" зображена на рисунку 1.

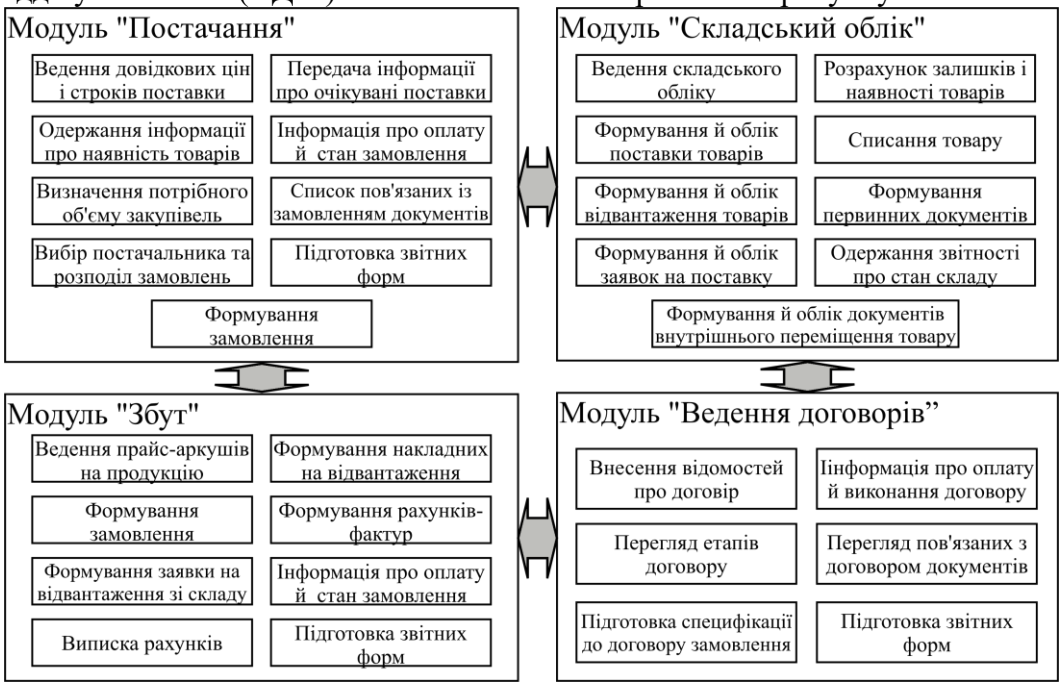


Рисунок 1 – Структурна схема системи

Підсистема "Логістика" включає у свій склад наступні модулі:

–Модуль "Ведення договорів".
–Модуль "Постачання".
–Модуль "Збут".
–Модуль "Складський облік".

Модуль "Постачання" призначений для автоматизації діяльності підрозділів, що займаються забезпеченням підприємства товарами, матеріалами й комплектуючими.

Модуль виконує наступні функціональні задачі:

–Ведення довідкових цін і строків поставки матеріалів і комплектуючих по постачальниках.

–Одержання інформації про наявність товарів на складах.

–Визначення об'єму закупівель на основі планів діяльності, стану складу й запасів у цехах.

–Вибір постачальника, розподіл замовлень на закупівлю між постачальниками.

–Формування замовлення, прив'язка замовлення до договору.

–Передача інформації про очікувані поставки на склад.

–Одержання інформації про оплату й поточний стан замовлення на закупівлю.

–Перегляд списку пов'язаних із замовленням документів.

–Підготовка звітних форм.

Модуль "Збут" призначений для автоматизації діяльності підрозділів, що займаються збутому продукції підприємства.

Модуль виконує наступні функціональні задачі:

–Ведення прайс-аркушів на продукцію.

–Формування замовлення, прив'язка замовлення до договору.

–Формування заявки на відвантаження зі складу.

–Виписка рахунків (формування документів-підстав).

–Формування накладних на відвантаження продукції.

–Формування рахунків-фактур.

–Одержання інформації про оплату й поточний стан замовлення.

–Підготовка звітних форм.

Модуль "Складський облік" призначений для проведення складських операцій (приходу, витрати й внутрішнього переміщення товару), а також для обліку наявності товарів на логічних, фізичних складах і всього підприємства в цілому.

Модуль виконує наступні функціональні задачі:

–Ведення складського обліку в розрізі підприємства, фізичних і/або логічних складів.

–Формування й облік поставки товарів.

–Формування й облік відвантаження товарів.

–Формування й облік заявок на поставку товарів.

–Формування й облік документів внутрішнього переміщення товару.

–Розрахунок залишків і наявності товарів на складах.

–Списання товару при проведенні складських операцій приходу, витрати й внутрішнього переміщення товару.

–Можливість формування первинних документів (прибуткова накладна, видаткова накладна, накладна внутрішнього переміщення).

–Одержання звітності про стан складу.

Модуль "Ведення договорів" призначений для автоматизації діяльності підрозділів підприємства, що займаються роботою з договорами. Даний модуль дозволяє проводити весь цикл робіт за договором, починаючи від етапу розгляду комерційної пропозиції й підготовки договору, до моменту його закриття. Крім цього контролюються етапи проходження договору, його оплата, існує можливість знаходження всіх пов'язаних з договором документів.

Модуль "Договору" виконує наступні функціональні задачі:

–Внесення відомостей про договір.

–Виділень етапів договору.

–Підготовка специфікації до договору у формі замовлення.

–Перегляд інформації про оплати й виконання договору.

–Перегляд пов'язаних з договором документів.

–Підготовка звітних форм.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів довідково-інформаційного сервісу комп’ютерної мережі відділу підприємства. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем довідково-інформаційного сервісу комп’ютерної мережі відділу підприємства; Досліджена система довідково-інформаційного сервісу комп’ютерної мережі відділу підприємства; На основі отриманих результатів досліджень створена програмна реалізація системи довідково-інформаційного сервісу комп’ютерної мережі відділу підприємства. Розроблені під час виконання роботи алгоритми дозволяють успішно вирішувати завдання довідково-інформаційного сервісу комп’ютерної мережі відділу підприємства. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Коваленко О.В. Метод тестування DOM XSS уразливості / О.В. Коваленко, О.А. Смірнов, А.С. Коваленко, С.А. Смірнов // Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп’ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті». м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.
2. Коваленко О.В. GERT-модель технології тестування DOM XSS уразливості / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов, С.А. Смірнов // Збірник наукових праць IV міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 21-24 лютого 2018 р. – Київ: Європейський університет. – 2018. – С. 65-70.
3. Коваленко О.В. Технології тестування уразливостей Web-застосунків з використанням GERT-моделі / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов, С.А. Смірнов // Збірник тез всеукраїнської науково-практичної конференції "Комп'ютерні інтелектуальні системи та мережі (КІСМ-2018)". м. Кривий Ріг. 21-23 березня 2018 р. – Кривий Ріг.: ДВНЗ КНУ – 2018. – С. 227-230.
4. Коваленко А.В. Тестирование уязвимости Web-приложений к атаке вида межсайтовый скриптинг / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // Збірник тез «Securitea informationala 2018». Conferenta internationala (editia a XIV-a). Chisinau. Moldova. 20-21 martie 2018. – Chisinau: ADSEM. – 2018. – P. 54-56.
5. Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // Збірник тез X міжнародної науково-практичної конференції “Проблеми і перспективи розвитку IT-індустрії”. м. Харків. 19-20 квітня 2018 р. – Харків: ХНЕУ. – 2018. – С. 38.
6. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Сборник тезисов XII международной конференции "Стратегия качества в промышленности и образовании". г. Варна. Болгария. 30 мая – 02 июня 2016 г – Варна. ТУВ. – 2016. – С. 585-589.
7. Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Матеріали Всеукраїнської науково-практичної конференції «Кібербезпека в Україні: правові та організаційні питання». м. Одеса, 21 жовтня 2016 р. – Одеса: ОДУВС, 2016. – С.146-148.
8. Коваленко А.В. Метод управления рисками разработки программного обеспечения с использованием псевдобулевых методов бивалентного программирования / А.В. Коваленко, А.А. Смирнов // Матеріали Всеукраїнської науково-практичної конференції «Актуальні задачі та досягнення у галузі кібербезпеки». м. Кропивницький, 23-25 листопада 2016 року – Кропивницький: ЦНТУ, 2016. – С. 162.
9. Коваленко А.В. Псевдобулевые методы бивалентного программирования для управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко, С.А. Смирнов // Збірник наукових праць III міжнародної науково-практичної конференції

«Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 22-25 лютого 2017 р. – Київ: Європейський університет. – 2017. – С. 158-162.

10. Коваленко А.В. Метод управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез II науково-практичної конференції "Проблеми кібербезпеки інформаційно-телекомунікаційних систем". м. Київ. 23-24 березня 2017 р. – Київ: КНУ ім. Тараса Шевченко – 2017. – С. 203-205.
11. Коваленко А.В. Алгоритмы анализа уязвимостей при управлении рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Conferenta internationala (editia a XIII-a). «Securitea informationala 2017». Chisinau. Republic of Moldova. 4-5 aprilie 2017. – Chisinau: ADSEM. – 2017. – P. 19-22.
12. Коваленко А.В. Алгоритм анализа DOM XSS уязвимости при управлении рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник тез дев'ятнадцятого міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кропивницький 7-8 квітня 2017 р. – Кропивницький: ГЛА НАУ. – 2017. – С. 125-127.
13. Коваленко А.В. Алгоритм анализа уязвимости SQL Injection для управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник тез другої міжнародної науково-технічної конференції «Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2017). м. Харків.10-12 квітня 2017 р. – Харків: НТУ «ХПІ». – 2017. – С. 27.
14. Коваленко А.В. Метод управления рисками разработки программного обеспечения на основе алгоритмов анализа уязвимостей / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник тез Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології» (IS&CT). м. Кіровоград. 20-22 квітня 2017 р. Кіровоград: КНТУ. – 2017. – С. 92.
15. Коваленко А.В. Алгоритмы анализа DOM XSS уязвимости и уязвимости SQL Injection при управлении рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник тез IX міжнародної науково-практичної конференції "Проблеми і перспективи розвитку IT-індустрії". м. Харків. 20-21 квітня 2017 р. – Харків: ХНЕУ. – 2017. – С. 61.
16. Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.
17. Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений / А.В. Коваленко, А.А. Смирнов // Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.
18. Коваленко А.В. Задачи распознавания ситуаций в erp системах / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник наукових праць "Системи обробки інформації". – Випуск 4(120). – Х.: ХУПС – 2014. – С. 161-164.
19. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник наукових праць "Системи обробки інформації". – Випуск 5(142). – Х.: ХУПС – 2016. – С. 153-157.
20. Коваленко А.В. Проблемы анализа и оценки рисков информационной деятельности / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць "Системи обробки інформації". – Випуск 3(140). – Х.: ХУПС – 2016. – С. 40-42.

УДК 004

В. Бурлаченко, магістр гр. КІ-21М-1,4
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ СЕРВІСУ ДІАГНОСТУВАННЯ ТА ТЕСТУВАННЯ СКЛАДОВИХ ПК

У статті розроблено програмне забезпечення, яке призначено для системи сервісу діагностування та тестування складових ПК. Метою розробки є дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК. Об’єктом дослідження є процес сервісу діагностування та тестування складових ПК. Предметом дослідження є методи сервісу діагностування та тестування складових ПК. Методи дослідження базуються на методах схемотехніки, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи сервісу діагностування та тестування складових ПК. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп’ютерна інженерія, діагностування, тестування

Постановка проблеми. На перший погляд більшості домашніх користувачів зовсім не потрібно розбиратися в залізній начинці свого комп'ютера – але лише за умови, якщо хто-небудь інший візьме на себе рішення питань, пов'язаних з «залізом». А якщо таких не найдеться, то ознайомитися з деякою початковою інформацією із цієї області зовсім не зашкодить. Справа в тому, що рано або пізно вам доведеться проводити апгрейд вашого ПК, що далеко не завжди припускає банальну зміну одного системного блоку з начинкою на іншій, більш сучасний. Можливо, у багатьох випадках виявиться досить замінити (або доповнити) його окремі компоненти, наприклад просто докупити пам'ять. Але, щоб вибрати вірну стратегію апгрейду, варто чітко знати, яке апаратне забезпечення у вас установлено, і зрозуміти, чого конкретно не вистачає вашому комп'ютеру для більш швидкої роботи: потужності процесора, пам'яті, швидкості вінчестера й т.п. Але це лише одна із проблем. Інша, не менш важлива пов'язана з тим, що любе встаткування в якийсь момент починає працювати зі збоями або взагалі виходить із ладу. Сумні наслідки подібних ситуацій очевидні, тому набагато надійніше контролювати стан важливих апаратних систем, що дозволить із певною ймовірністю прогнозувати можливе поводження апаратури в найближчому майбутньому, вчасно виявити ті або інші неполадки з комп'ютерними компонентами й вчасно подбати про запобіжні заходи. І нарешті, припустимо, що ви придбали новий комп'ютер або провели апгрейд старого – очевидно, що вам необхідно швидко зорієнтуватися й зрозуміти, чи дійсно комп'ютер стабільно працює й чи відповідає начинка його системного блоку заявленій при покупці, причому не розбираючи сам блок, на якому може стояти пломба.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи сервісу діагностування та тестування складових ПК.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи сервісу діагностування та тестування складових ПК.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем сервісу діагностування та тестування складових ПК.
– Дослідження системи сервісу діагностування та тестування складових ПК.

– Програмна реалізація системи сервісу діагностування та тестування складових ПК.

Об'єктом дослідження є процес сервісу діагностування та тестування складових ПК.

Предметом дослідження є методи сервісу діагностування та тестування складових ПК.

Методи дослідження базуються на методах схемотехніки, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Windows API – найбільш важлива й потужна додаткова бібліотека функцій, доступна кожному програмістові. Багато хто з них, у тому числі й досвідчених розроблювачах працюють із ними, використовуючи прості готові рішення, почерпнуті в різних книгах і журналах, не дуже замислюючись про суть цієї технології. Такий підхід є достатнім при рішенні простих задач, але для серйозної роботи переважніше більш детально розібратися з основними принципами використання функцій Windows API. Чим ми зараз і займемося.

Windows API – набір функцій операційної системи

Абревіатура API для багатьох починаючих програмістів виглядає досить таємниче. Насправді ж Application Programming Interface (API) – це просто деякий готовий набір функцій, що можуть використовувати розроблювачі додатків. У загальному випадку дане поняття еквівалентно тому, що раніше частіше називали бібліотекою підпрограм. Однак найчастіше під API мається на увазі деяка особлива категорія таких бібліотек.

У ході розробки практично будь-якого досить складного додатка (MyAppication) для кінцевого користувача формується набір специфічних внутрішніх функцій, використовуваних для реалізації даної конкретної програми, що називається MyApplication API. Часто виявляється, що ці функції можуть ефективно використовуватися також для створення інших додатків, у тому числі іншими програмістами. У цьому випадку автори виходячи зі стратегії просування свого продукту повинні вирішити питання – чи відкривають вони доступ до цього набору для зовнішніх чи користувачів ні? При позитивній відповіді на нього в описі програмного пакета, як його переваг, з'являється фраза про те, що "комплект включає відкритий набір API-функцій" (але іноді за додаткові гроші).

Таким чином, найчастіше під API мається на увазі набір функцій, що є частиною одного додатка, але при цьому доступних для використання в інших програмах. Наприклад, Excel крім інтерфейсу для кінцевого користувача має набір функцій Excel API, що може використовуватися, зокрема, при створенні додатків за допомогою Delphi.

Відповідно, Windows API – це набір функцій, що є частиною самої операційної системи й у той же час – доступної для будь-якого іншого додатка, у тому числі написаного за допомогою Delphi. І в цьому плані цілком виправдана аналогія з набором системних переривань BIOS/DOS, що фактично являє собою DOS API

Відмінність полягає в тім, що склад функцій Windows API, з однієї сторони значно ширше, у порівнянні з DOS, з іншого боку – не включає багато засобів прямого керування ресурсами комп'ютера, які були доступні програмістам у попередньої ОС. Крім того, звертання до Windows API виконується за допомогою звичайних процедурних обігів, а виклик функцій DOS – через спеціальну машинну команду процесора, що називається Interrupt ("переривання").

Далі під терміном API буде матися на увазі Win API і більше того, за замовчуванням – Win64 API.

Навіщо потрібний Win API

Незважаючи на те, що Delphi має величезну множину різноманітних функцій, при більш-менш серйозній розробці виявляється, що їхніх можливостей часто не вистачає для рішення необхідних задач. При цьому програмісти-новачки часто починають скаржитися на недоліки Delphi і подумувати про зміну інструмента, не підозрюючи, що на їхньому комп'ютері є величезний набір засобів і потрібно тільки вміти ним скористатися.

При знайомстві з Win API виявляється, що багато вбудованих Delphi-функцій – не що інше, як звертання до відповідних системних процедур, але тільки реалізовані у вигляді синтаксису даної мови. З огляду на це, необхідність використання API визначається наступними варіантами:

1. API-функції, які повністю реалізовані у вигляді убудованих Delphi-функцій. Проте, іноді й у цьому випадку буває корисним перейти до застосування API, тому що це дозволяє часом істотно підвищити продуктивність (зокрема, за рахунок відсутності непотрібних перетворень переданих параметрів).

2. Вбудовані Delphi-функції реалізують лише окремий випадок відповідної API-функції. Це досить звичайний варіант. Наприклад, API-функція CreateDirectory має більше широкі можливості в порівнянні з убудованим Delphi-оператором MkDir.

3. Величезне число API-функцій взагалі не мають аналогів в існуючому сьогодні варіанті мови Delphi. Наприклад, видалити каталог не можна засобами Delphi – для цього потрібно використовувати функцію DeleteDirectory.

Варто також підкреслити, що деякі API-функції (їхня частка в Win API досить незначна) не можуть викликатися з Delphi-програм через ряд обмежень мови, наприклад відсутність можливості роботи з адресами пам'яті. Але в ряді випадків можуть допомогти нетривіальні прийоми програмування (зокрема, у випадку з тими ж адресами).

Як вивчати Win API

Це не таке просте питання, якщо врахувати що число функцій Win64 API оцінюється величиною порядку 10 тисяч (точної величини ніхто не знає, навіть Microsoft).

До складу Delphi (версій 4-6) входить файл із описом оголошень Win API – WIN64API.TXT (про його застосування ми розповімо докладніше пізніше). Але, по-перше, з його допомогою можна одержати відомості про призначення тої або іншої функції і її параметрів тільки по використовуваних мнемонічних іменах, а по-друге – перелік функцій у цьому файлі далеко не повний. У свій час в Delphi були спеціальні довідкові файли, з описом функцій Win16 API. Вичерпна інформація про Win64 API перебуває в довідковій системі Platform Software Development Kit, що зокрема перебуває на компакт-дисках MSDN Library, включених до складу Delphi 5.0 і 6.0 Enterprise Edition і Office 2000 Developer Edition. Однак потрібно зізнатися, що знайти там потрібну інформацію й розібратися в ній зовсім не просто. Не говорячи вуж про те, що всі описи там приводяться стосовно до мови C.

Win API і Dynamic Link Library (DLL)

Набір Win API реалізований у вигляді динамічних DLL-бібліотек. У такий спосіб далі зараз ми будемо фактично говорити про технологію використання DLL у середовищі Delphi на прикладі бібліотек, що входять до складу Win API. Однак говорячи про DLL необхідно зробити кілька важливих зауважень.

У цьому випадку під DLL ми маємо на увазі традиційний варіант двійкових динамічних бібліотек, які забезпечують прямий обіг додатків до потрібних процедур – підпрограмам або функціям (приблизно також як це відбувається при виклику процедур усередині Delphi-проекту). Такі бібліотеки можуть створюватися за допомогою різних інструментів – VC++, Delphi, Fortran, Assembler. Delphi може робити Active DLL, доступ до яких виконується через інтерфейс OLE Automation.

Звичайно файли динамічних бібліотек мають розширення .DLL, але це зовсім не обов'язково. Для Win16 часто застосовувалося розширення .EXE, драйвери зовнішніх пристроїв позначаються за допомогою .DRV.

Як ми вже відзначали, визначити точне число API-функцій Windows і файлів їх утримуючих – досить складно (але всі вони перебувають у системному каталозі). У цьому плані краще виділити склад бібліотек, що становлять ядро операційної системи, і основних бібліотек із ключовими додатковими функціями.

Бібліотеки Win64 API ядра операційної системи Windows 95/98:

–KERNEL64.DLL: низькорівневі функції керування пам'яттю, задачами й іншими ресурсами системи;

–USER64.DLL: тут в основному перебувають функції керування користувальницьким інтерфейсом;

–GDI64.DLL: бібліотека Graphics Device Interface – різноманітні функції висновку на зовнішні пристрої;

–COMDLG64.DLL: функції, пов'язані з використанням діалогових вікон загального призначення.

Основні бібліотеки з функціями розширення:

–COMCTL64.DLL: набір додаткових елементів керування Windows, у тому числі Tree List і Rich Text;

–MAPI64.DLL: функції роботи з електронною поштою;

–NETAPI64.DLL: елементи керування й функції роботи з мережею;

–ODBC64.DLL: функції цієї бібліотеки потрібні для роботи з різними базами даних через протокол ODBC;

–WINMM.DLL: операції доступу до системних засобів мультимедіа.

Програми написані на Win API мають більшу продуктивність і невеликий розмір. За допомогою функцій Win API можна одержати доступ до різних об'єктів Windows.

Функція CreateWindowEx

Створює вікно із заданими властивостями.

Функція CreateWindowEx виглядає так:

```
function CreateWindowEx(
dwExStyle: DWORD;
lpClassName: PChar;
lpWindowName: PChar;
dwStyle: DWORD;
X, Y, nWidth, nHeight: Integer;
hWndParent: HWND;
hMenu: HMENU;
hInstance: HINST;
lpParam: Pointer
): HWND;
```

Параметри CreateWindowEx:

dwExStyle– визначає розширений стиль вікна, може бути комбінацією (за допомогою оператора or):

– WS_EX_ACCEPTFILES– на вікно можна перетаскувати файли із Провідника, коли користувач опускає на вікно з таким стилем файли, посилає повідомлення WM_DROPFILES;

– WS_EX_APPWINDOW– на панелі задач для вікна з таким стилем з'являється кнопка;

– WS_EX_CLIENTEDGE– навколо клієнтської частини буде рамка, причому клієнтська частина буде втиснена усередину;

– WS_EX_CONTEXTHELP– у заголовку вікна з'явиться кнопка контекстної допомоги, це прапор не може використовуватися одночасно з WS_MAXIMIZEBOX і WS_MINIMIZEBOX;

– WS_EX_CONTROLPARENT;

– WS_EX_DLGMODALFRAME;

– WS_EX_LEFT;

– WS_EX_LEFTSCROLLBAR;

– WS_EX_LTRREADING;

– WS_EX_MDICHILD;

– WS_EX_NOINHERITLAYOUT;

– WS_EX_NOPARENTNOTIFY;

– WS_EX_OVERLAPPEDWINDOW;

– WS_EX_PALETTEWINDOW;

– WS_EX_RIGHT;

– WS_EX_RIGHTSCROLLBAR;

– WS_EX_RTLREADING;

– WS_EX_STATICEDGE;

– WS_EX_TOOLWINDOW– створюється вікно з маленьким заголовком, як у панелі інструментів;

– WS_EX_TOPMOST– створюване вікно буде перебувати поверх інших;

– WS_EX_TRANSPARENT;

– WS_EX_WINDOWEDGE.

Визначення інших констант ви зможете знайти в довідці по Win64api.

lpClassName– Ім'я класу вікна. Ви можете створювати свої класи за допомогою функції RegisterClassEx або використовувати визначені: edit, button, static, scrollbar, combobox і інші;

lpWindowName– текст, що з'явиться в заголовку вікна (якщо вікно із заголовком), на кнопці (якщо клас вікна button), у поле уведення тексту (якщо клас вікна edit);

dwStyle– список основних стилів вікна. Містить кілька наступних констант, з'єднаних оператором or:

– WS_BORDER– вікно буде мати тонку рамку;

– WS_CAPTION– вікно буде мати заголовок;

– WS_CHILD або WS_CHILDWINDOW– вікно буде дочірнім, тобто цілком розташовуватися усередині деякого іншого вікна;

– WS_CLIPCHILDREN– площа займана дочірніми вікнами не буде перемальовуватися;

– WS_CLIPSIBLINGS– перемальовування одного дочірнього вікна не впливає на інші;

– WS_DISABLED– вікно створюється недоступним, його можна розблокувати за допомогою функції EnableWindow;

– WS_DLGFRAME– створюється вікно з рамкою як у діалогових вікон;

– WS_GROUP– для дочірнього вікна (зі стилем WS_CHILD) визначає перший елемент у групі, при натисканні на Tab саме він одержить фокус, група простирається до наступного дочірнього вікна з тим же стилем, усередині групи можна переміщатися за допомогою клавіш керування курсором;

– WS_HSCROLL– створюється вікно з горизонтальною смугою прокручування;

– WS_ICONIC або WS_MINIMIZE– створюване вікно споконвічно мінімізоване;

– WS_MAXIMIZE– створюване вікно споконвічно максимізоване;

– WS_MAXIMIZEBOX– створюване вікно має кнопку максимізації;

– WS_MINIMIZEBOX– створюване вікно має кнопку мінімізації;

– WS_OVERLAPPED– створюється вікно, що перекривається, має заголовок і рамку;

– WS_OVERLAPPEDWINDOW– комбінація прапорів WS_OVERLAPPED, WS_CAPTION, WS_SYSMENU, WS_THICKFRAME, WS_MINIMIZEBOX і WS_MAXIMIZEBOX;

– WS_POPUP– створюється вікно не має споконвічно рамки й заголовка, не може використовуватися зі стилем WS_CHILD;

– WS_SIZEBOX або WS_THICKFRAME– створюється вікно, розмір якого можна змінювати;

– WS_SYSMENU– створюється вікно зі значком системного меню, повинен уживатися із прапором WS_CAPTION;

– WS_TABSTOP– створюється дочірнє вікно, що може одержувати фокус уведення при натисканні на Tab;

– WS_TILEDWINDOW– комбінація прапорів WS_OVERLAPPED, WS_CAPTION, WS_SYSMENU, WS_THICKFRAME, WS_MINIMIZEBOX і WS_MAXIMIZEBOX;

– WS_VISIBLE– створюється вікно, що споконвічно видиме. Якщо ви не вкажете це прапор для вікна, то ви його ніколи не побачите (якщо тільки не скористаєтеся функцією ShowWindow);

– WS_VSCROLL– створюване вікно буде мати вертикальну смугу прокручування;

– X-X– горизонтальна координата верхнього лівого кута вікна; якщо ви хочете надати windows можливість розташувати вікно за замовчуванням, укажіть тут CW_USEDEFAULT, у цьому випадку наступний параметр ігнорується;

– Y-Y– вертикальна координата верхнього лівого кута вікна.

nWidth– ширина вікна (в одиницях пристрою, для монітора– у пікселях), якщо ви хочете надати Windows вибрати положення вікна, то виставите тут CW_USEDEFAULT, у цьому випадку наступний параметр ігнорується;

nHeight– висота вікна (в одиницях пристрою, для монітора– у пикселах);

hWndParent– описувач батьківського вікна, якщо вікно створюється зі стилем WS_CHILD, то тут обов'язково повинен стояти коректний описувач;

для дочірнього вікна (із прапором стилю WS_CHILD) визначає ідентифікатор цього дочірнього вікна, для звичайного вікна визначає описувач головного меню вікна (якщо дорівнює нулю, то використовується меню з визначення класу вікна);

hInstance– описувач додатка, з яким вікно зв'язується (для Windows NT/2000/XP ігнорується);

lpParam– покажчик на що-небудь, він передається в повідомленні WM_CREATE при створенні вікна, і в ньому можуть бути переправлені які-небудь дані для копії вікна.

Функція ShowWindow

Ця функція показує або ховає вікно.

```
function ShowWindow(
      hWnd: HWND;
      nCmdShow: Integer
): BOOL;
```

Параметри функції:

hWnd– Описувач потрібного вікна;

nCmdShow– Константа, що визначає, що буде зроблено з вікном:

– SW_HIDE– вікно буде приховано;

– SW_SHOWNORMAL– вікно буде показане й активоване, якщо вікно було мінімізовано або максимізоване, то воно буде відновлено у вихідну позицію й розмір;

– SW_SHOWMINIMIZED– активізує й звертає (мінімізує) вікно;

– SW_SHOWMAXIMIZED– активізує й максимізує вікно;

– SW_MAXIMIZE– максимізує вікно;

– SW_SHOWNOACTIVATE– те ж саме, що SW_SHOWNORMAL, тільки вікно не активізується;

– SW_SHOW– відображає вікно в його поточній позиції;

– SW_MINIMIZE– мінімізує вікно й активізує наступне по Z-Списку;

– SW_SHOWMINNOACTIVE– те ж саме, що й SW_SHOWMINIMIZED, тільки вікно не активізується;

– SW_SHOWNA– те ж саме, що SW_SHOW, тільки вікно не активізується;

– SW_RESTORE– відновлює вікно з максимізованого або мінімізованого стану;

– SW_SHOWDEFAULT– відображає вікно так, як воно було відображено при старті відповідного додатка;

– SW_MAXIMIZE– максимізує вікно.

Структура типу TWndClassEx

Структура типу TWndClassEx має такий вигляд:

```
tagWNDCLASSEXA = packed record
cbSize: UINT;
style: UINT;
lpfnWndProc: TFNWndProc;
cbClsExtra: Integer;
cbWndExtra: Integer;
hInstance: HINST;
hIcon: HICON;
```

```
hCursor: HCURSOR;
hbrBackground: HBRUSH;
lpszMenuName: PAnsiChar;
lpszClassName: PAnsiChar;
hIconSm: HICON;
end;
```

Обробка повідомлень в Win API

Повідомлення – інформація про деяку зміну в користувальницькому інтерфейсі, наприклад переміщення вікна або натискання клавіші на клавіатурі. Повідомлення також можуть розсилатися іншими додатками.

Цикл обробки повідомлень виглядати так:

```
while GetMessage(Mmsg,0,0,0) do
begin
  TranslateMessage(Mmsg);
  DispatchMessage(Mmsg);
end;
```

TranslateMessage– ця функція переводить повідомлення віртуальних клавіш у символьні повідомлення.

DispatchMessage– ця функція повідомлення віконному оброблювачеві подій. Як віконний оброблювач подій служить функція WindowProc.

Бібліотека візуальних компонентів (VCL)

Бібліотека візуальних компонентів (VCL) – об’єктно-орієнтована бібліотека для розробки програмного забезпечення, розроблена компанією «Borland» для підтримки принципів візуального програмування. VCL входить у комплект поставки «Delphi», «C++ Builder» і «Borland Developer Studio» і є, по суті, частиною середовища розробки, хоча розробка додатків цих середовищах можлива й без використання VCL.

VCL представляє величезну кількість готових до використання компонентів для роботи в самих різних областях програмування, таких, наприклад, як інтерфейс користувача (екранні форми й т.зв. «контроли»), робота з базами даних, взаємодія з операційною системою, програмування мережевих додатків та інше.

Розробка структурної схеми

На рисунку 1 зображена структурна схема системи тестування та діагностики вузлів ЕОМ, яка була розроблена у результаті виконання проекту. З цієї схеми ми бачимо, щ о система складається з наступних структурних модулів:

– Панель «Головне вікно програми», з яким зв’язані усі інші панелі виконані у вигляді закладок;
– Панель «Загальне»;
– Панель «Процесор»;
– Панель «Диски»;
– Панель «Принтери»;
– Панель «Пам'ять»;
– Панель «Клавіатура»;
– Панель «Шрифти»;
– Панель «Відеокарта»;
– Панель «Діагностика».

Усі панелі, крім останньої відображають інформацію, яка відноситься до параметрів відповідного структурного блоку.

Остання панель відображає інформацію діагностування, й включає в себе наступні підпрограми тестування:

– Перевірка CD/DVD-ROM;
– Тестування пам’яті;
– Тестування процесора.

Крім того програма надає інформацію про автора програми та місце, де вона створена.

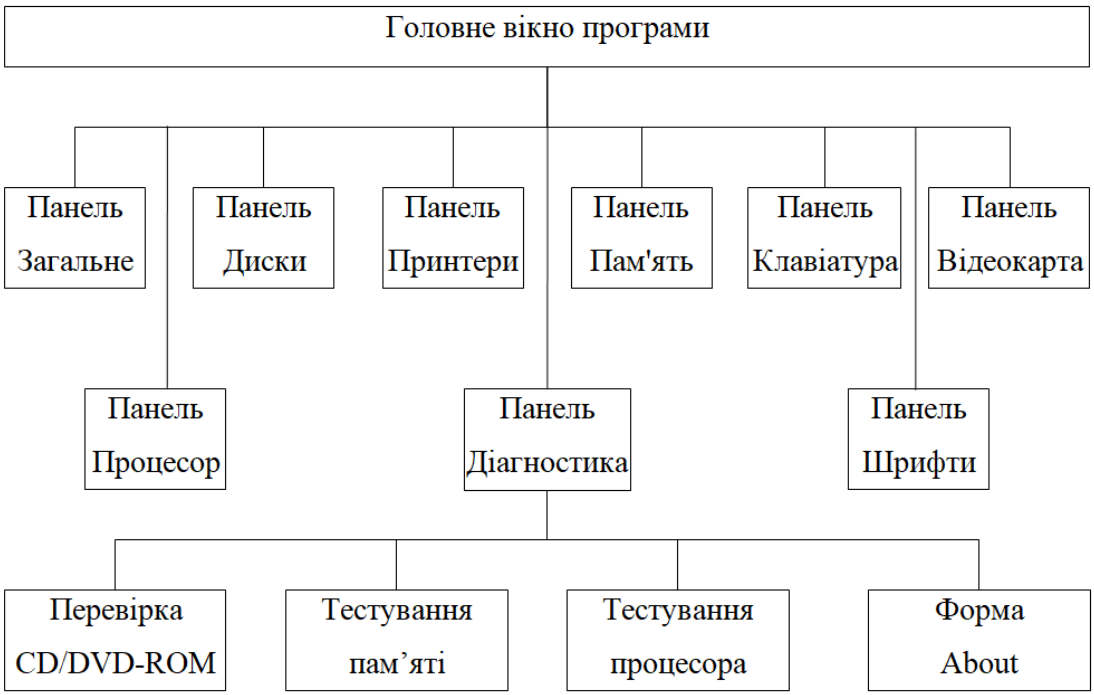


Рисунок 1 – Структурна схема системи

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів сервісу діагностування та тестування складових ПК. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем сервісу діагностування та тестування складових ПК. Досліджена система сервісу діагностування та тестування складових ПК. На основі отриманих результатів досліджень створена програмна реалізація системи сервісу діагностування та тестування складових ПК. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання сервісу діагностування та тестування складових ПК. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Кожанова А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. Кожанова, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2013. – Вип. 6(113). – С. 255-257.
2. Коваленко А.С. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко ., А.А. Смірнов, А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2014. – Вип. 4(120). – С. 161-164.
3. Коваленко А.С. Підсистема технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко , О.А.Смірнов, О.В. Коваленко // Системи озброєння і військова техніка.– Х.: ХУПС, 2014. – № 1(37). – С. 126-129.
4. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.
5. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.
6. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.
7. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного

технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.
8. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – С. 85-88.
9. Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 1(126). – С. 75-79.
10. Коваленко А.С. Обґрунтування набору даних для оцінки технічного стану інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2015. – Вип. 1(42). – С.39-41.
11. Коваленко А.С. Експертна система технічного діагностування інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2015. – № 1(41). – С. 106-111.
12. Коваленко А.С. Удосконалення методу технічного обслуговування об’єктів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко, О.П. Доренський // Системи озброєння і військова техніка. – Х.: ХУПС, 2016. – № 2(46). – С. 109-114.
13. Kovalenko A.S. Information model and its element for displaying information on technical condition of objects of integrated information system / A.S. Kovalenko, A.A. Smirnov, A.V. Kovalenko, A.P. Dorensky // International Journal of Computational Engineering Research (IJCER). – India: Delhi, 2016. – Volume 6, Issue 1. – P. 21-27.
14. Кожанова А.С. Система технічної діагностики інтегрованих інформаційних систем – обґрунтування необхідності створення, визначення понятійного апарату та напрямів досліджень / А.С. Кожанова, О.А. Смірнов, М.П. Савченко, Д.М. Ізосімов, В.В. Мороз // Створення та модернізація озброєння і військової техніки в сучасних умовах: Тринадцята наук.-техн. конф., 5-6 вер. 2013 р., м. Феодосія: тези доп. – Феодосія: ДНВЦ, 2013. – С. 187-188.
15. Кожанова А.С. Визначення основних напрямків досліджень щодо створення системи технічної діагностики інтегрованих інформаційних систем / А.С. Кожанова, О.А. Смірнов, А.В. Челпанов // Проблемні питання розвитку озброєння та військової техніки Збройних Сил України: IV наук.-техн. конф., 16-20 груд. 2013 р., м. Київ: зб. тез. – Київ: ЦНДІ ОВТ ЗСУ, 2013. – С. 293.
16. Коваленко А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформатика та системні науки : V Всеукр. наук.-практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 2014. – С. 292-294.
17. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2014. – С. 241.
18. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп’ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 28-31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 2014. – С. 190-193.
19. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2015. – С. 15.
20. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

УДК 004

Є.Водзинський, магістр гр. КН-21М-1,4
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АДАПТИВНОЇ ОПТИМІЗАЦІЇ МАРШРУТИЗАЦІЇ МЕРЕЖІ ІНФОРМАЦІЙНИХ ТА КОМП'ЮТЕРНИХ СИСТЕМ

У статті розроблено програмне забезпечення, яке призначено для системи адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем. Метою розробки є дослідження та програмна реалізація системи адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем. Об'єктом дослідження є процес адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем. Предметом дослідження є методи адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем. Методи дослідження базуються на методах побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, маршрутизація, мережі інформаційних та комп'ютерних систем

Постановка проблеми. Мережеву маршрутизацію, процес вибору шляхів маршрутизації в мережі зв'язку, можна оптимізувати для зниження витрат, покращення використання мережі та максимізації продуктивності.

У мережах з комутацією пакетів, таких як Інтернет, постачальники засобів зв'язку (включно з операторами зв'язку) мають багато варіантів – часто з кількома переходами та варіантами передачі – щодо того, як пакет переміщається до пункту призначення. У міру надходження пакетів програмне забезпечення мережевої маршрутизації оцінює ці варіанти та приймає рішення в режимі реального часу щодо того, як пакет повинен дістатися до місця призначення.

Мережі динамічні, умови змінюються щосекунди, що вводить новий рівень складності. Система мережевої маршрутизації має оптимізуватися для кожного пакета. Оптимізація може включати вартість, пропускну здатність, угоди про рівень обслуговування, кількість переходів, доступність тощо.

Для вирішення цих складнощів постачальникам послуг потрібен механізм прийняття рішень на основі правил, який базується на високопродуктивній базі даних із низькою затримкою та постійно оновлюється реальними вхідними даними, включаючи доступність мережі, пропускну здатність і вартість. Механізм прийняття рішень розглядає кожен пакет і його призначення, а потім запитує базу даних, щоб застосувати свої правила прийняття рішень для прийняття рішення про маршрутизацію.

Бізнес-вимоги бази даних для мережевої маршрутизації:

– Великі набори даних постійно оновлюються з мережевими вхідними даними, параметрами топології та вартістю.

– Рішення щодо маршрутизації в реальному часі на основі оптимізації механізму правил.

– Постійна доступність, завжди 24x7x365.

– Надзвичайно низька затримка, щоб не перешкоджати роботі мережі.

Система, яка розроблена у даній роботі, є гарним вибором для мережевої маршрутизації, оскільки він забезпечує:

– Передбачувану, високу продуктивність проти надзвичайно великих обсягів транзакцій.

– Провідну в галузі доступність і час безвідмовної роботи (п'ять 9с).

– Масштабування для обробки величезних обсягів даних.

– Значно нижчу ТСО порівняно з іншими технологіями.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем.

– Дослідження системи адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем.

– Програмна реалізація системи адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем.

Об'єктом дослідження є процес адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем.

Предметом дослідження є методи адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем.

Методи дослідження базуються на методах побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу

За останні п'ятнадцять років Інтернет став важливою частиною світової комунікаційної інфраструктури. Інтернет складається з десятків тисяч доменів або автономних систем (AS) – частин інфраструктури, кожною з яких керує окрема установа, наприклад університет, корпорація або постачальник послуг Інтернету (ISP). Повідомлення, надіслане одним комп'ютером, зазвичай проходить через кілька AS, перш ніж досягти пункту призначення, завдяки чому продуктивність зв'язку залежить від потоку трафіку всередині та між AS на шляху. У цій роботі розглянемо роль оптимізації в управлінні маршрутизацією трафіку через Інтернет. Пояснемо, як складність показників продуктивності та протоколів мережевої маршрутизації призводить до проблем оптимізації, які не піддаються аналітичному вирішенню, що змушує використовувати ефективні методи пошуку для дослідження великого простору налаштовуємих параметрів.

Повідомлення, надіслані одним комп'ютером (скажімо, веб-клієнтом) на інший комп'ютер (скажімо, веб-сервер), поділяються на окремі дейтаграми або пакети, які проходять через мережу незалежно. Рішення базувати Інтернет-протокол (IP) на комутації пакетів сягає корінням у перші дні ARPAnet наприкінці 1960-х років [1,2]. На відміну від традиційного підходу комутації каналів телефонної мережі, комутація пакетів визнає, що передача даних часто відбувається з перебоями, коли користувачі та програми чергують періоди високої активності мережі та відносної тиші. Комутація пакетів має привабливість, що дозволяє зв'язкам у мережі мультиплексувати трафік між декількома парами відправників і отримувачів. Поки одна пара відправник-одержувач неактивна, інша може використовувати незатребувану пропускну здатність шляхом обміну трафіком на вищій швидкості.

Однак ця гнучкість має свою ціну. Якщо занадто багато хостів обмінюються пакетами одночасно, сукупний трафік може перевантажити пропускну здатність каналу, що призведе до затримок у передачі даних і, зрештою, до втрати пакетів у разі переповнення буфера, що

живить канал. Таким чином, Інтернет має відносно просту модель обслуговування найкращих зусиль без гарантії, що пакет даних досягне свого кінцевого пункту призначення вчасно.

Хоча багато програм (наприклад, електронна пошта) можуть допускати затримку в отриманні даних, відсутність інформації часто є неприпустимим. Замість створення надійної впорядкованої доставки даних в IP, кінцеві хости несуть відповідальність за повторну передачу втрачених пакетів і реконструкцію впорядкованого потоку даних на одержувачі. Ці функції виконує протокол керування передачею (TCP), реалізований в операційній системі на кінцевих комп'ютерах [1]. Крім того, відправник TCP визначає швидкість передачі даних, відстежуючи успішність (або невдачу) надсилання пакетів одержувачу. Коли пакети втрачаються, хост-відправник зменшує швидкість передачі, щоб допомогти зменшити явне перевантаження; коли пакети успішно доставлені, хост-відправник оптимістично збільшує швидкість надсилання, щоб отримати вигоду від доступної смуги пропускання на шляху до одержувача. Ця децентралізована схема контролю перевантаження забезпечує форму справедливого розподілу пропускної здатності каналів між конкуруючими парами відправників і отримувачів.

Однак транспортні протоколи, такі як TCP, не гарантують ефективну роботу мережі. Наприклад, одне посилання може бути сильно перевантаженим, тоді як інші посилання в мережі залишаються незначно завантаженими. Або виклик VoIP (VoIP) може проходити довгий шлях із високою затримкою розповсюдження, якщо доступний шлях із низькою затримкою. Відповідальність за вибір шляху, яким пакет слідує через мережу, лежить на протоколах маршрутизації, реалізованих окремими маршрутизаторами в мережі. Замість того, щоб використовувати жорстко з'єднані таблиці для пересилання пакетів, маршрутизатори обмінюються один з одним керуючими повідомленнями для розподіленого обчислення шляхів через мережу.

Розподілений підхід дозволяє набору маршрутизаторів автоматично адаптуватися до змін у топології мережі. Це робить IP-мережі надійними за наявності збоїв зв'язку та маршрутизатора, а також легко дозволяє розгортати нове обладнання в міру зростання мережі.

Однак протоколи маршрутизації, які використовуються в більшості IP-мереж, не включають інформацію про мережеве навантаження та продуктивність у вибір шляхів. Надані власним пристроям, маршрутизатори продовжують пересилати пакети через сильно завантажені канали зв'язку, що призводить до втрати пакетів, що змушує кінцеві хости зменшувати швидкість надсилання у відповідь.

Перші спроби розробки протоколів маршрутизації на основі затримки викликали занепокоєння щодо стабільності адаптивної маршрутизації в мережах з комутацією пакетів. При маршрутизації, чутливій до навантаження, пересилання пакетів даних відбуватиметься за такий же малий часовий проміжок, що й доставка інформації про переважне навантаження на окремі канали. Початковий досвід роботи в ARPAnet показав, що дія на основі застарілої інформації про завантаження каналів спонукає маршрутизатори спрямовувати трафік на, здавалося б, недостатньо використовувані канали, що призводить до перевантажень, які зрештою змушують маршрутизатори перемикатися на інші шляхи.

Незважаючи на деякі пропозиції, які намагаються запобігти подібним коливанням [1], звичайні протоколи IP-маршрутизації адаптуються лише до змін у топології мережі та конфігурації маршрутизатора. У той час як дослідження та стандартизація тривали (і все ще тривають) над більш гнучкими протоколами маршрутизації [1,6], практикам потрібен був ефективний спосіб впливати на потік трафіку через їхні мережі. На щастя, протоколи IP-маршрутизації мають різні налаштовуємі параметри, які мережеві оператори можуть налаштувати, щоб змінити шляхи, які маршрутизатори використовують для пересилання пакетів даних. Наприклад, оператор може налаштувати цілочисельні ваги зв'язку, які маршрутизатори використовують для обчислення найкоротших шляхів для передачі трафіку даних.

Однак визначення правильного налаштування цих параметрів у великій мережі є складним для людини-оператора, створення основи для застосування методів оптимізації.

Як альтернатива адаптації протоколів розподіленої маршрутизації до переважного трафіку, мережеві оператори або автоматизовані системи керування можуть змінювати конфігурацію «статичних» параметрів, які керують роботою протоколів маршрутизації [7]. Це призводить до «контура керування», де вимірювання робочої мережі служать вхідними даними для моделі «що, якщо» фіксує наслідки змін.

Вимірювання фіксують запропоноване навантаження на мережу, а також поточну топологію, тоді як модель передбачає результат розподіленого процесу вибору шляху, що виконується на маршрутизаторах для заданого налаштування налаштовуємих параметрів. Це дозволяє використовувати методи оптимізації для визначення налаштувань параметрів, які відповідають цілям продуктивності мережі. Після визначення хороших значень оператор-людина або автоматизована система можуть зв'язатися з маршрутизаторами, щоб змінити конфігурацію налаштовуємих параметрів. Потім маршрутизатори спільно діють на основі нових значень параметрів, щоб обчислити нові шляхи для пересилання пакетів даних через мережу, як передбачено моделлю.

Підхід має кілька переваг перед розширенням протоколів маршрутизації для автоматичної адаптації до трафіку.

По-перше, мережа може продовжувати використовувати звичайні протоколи IP-маршрутизації, а не розгортати або вмикати нові (потенційно складні) протоколи, які автоматично адаптуються до переважного трафіку.

По-друге, мережа уникає коливань протоколу, які можуть виникати, коли маршрутизатори реагують на локально створені (потенційно застарілі) уявлення про переважний трафік. Натомість зміни маршрутизації ретельно плануються на основі точного уявлення про трафік і топологію всієї мережі.

По-третє, маршрутизаторам не потрібно підтримувати статистику щодо навантаження на окремі канали, поширювати цю інформацію по всій мережі або перераховувати шляхи, коли ця інформація змінюється; уникнення цих операцій зменшує пропускну здатність і обчислювальні вимоги до маршрутизаторів.

По-четверте, і, мабуть, найважливіше, вибір параметрів маршрутизації може залежати від широкого спектру обмежень продуктивності та надійності, які може бути важко включити в розподілений протокол маршрутизації.

Ці обмеження та методи оптимізації для їх задоволення можуть продовжувати розвиватися з часом. Незважаючи на ці переконливі переваги, виниклу проблему оптимізації важко вирішити через складність базових протоколів маршрутизації та різноманітність цілей мережі.

Звичайні протоколи IP-маршрутизації не були розроблені з урахуванням оптимізації, а оптимізація налаштовуємих параметрів є складною в обчислювальному плані навіть для найпростіших показників (наприклад, таких як мінімізація максимального використання каналу) [1]. Простір параметрів досить великий, і невеликі зміни значення параметра можуть призвести до значних відмінностей у потоці трафіку через мережу.

У ядрі Інтернету зміни маршрутизації в одній AS можуть вплинути на те, як трафік залишає цю мережу та потрапляє в наступну AS на шляху до місця призначення [9]. Це вносить додаткову складність у моделювання наслідків змін налаштовуємих параметрів. Нарешті, проблема оптимізації повинна враховувати численні цілі, такі як навантаження на канал, затримку розповсюдження та кількість переходів. Більш складні показники також беруть участь [10], такі як бажання обмежити загальний трафік, спрямований на кожну сусідню AS, або частоту змін конфігурації, оскільки трафік коливається з часом. У процесі оптимізації може також знадобитися врахувати вплив несправностей обладнання та планового технічного обслуговування на придатність налаштувань параметрів; крім того, кілька маршрутизаторів або каналів можуть вийти з ладу або відновитися разом, якщо вони мають спільні ризики, наприклад загальне джерело живлення або оптичний підсилювач.

Ці практичні проблеми ускладнюють (а часто й унеможливлюють) отримання аналітичних рішень проблеми оптимізації. Натомість вирішення проблеми оптимізації залежить від наявності ефективних моделей протоколів маршрутизації та ефективних способів дослідження великих частин простору параметрів. Щоб зменшити накладні витрати

на обчислення, моделі «що, якщо» не повинні імітувати детальну роботу протоколів маршрутизації з часом, коли маршрутизатори обмінюються гіпотетичними повідомленнями керування. Замість цього моделі повинні визначати результат протоколів маршрутизації – шляхи, які маршрутизатори в кінцевому підсумку виберуть для передачі трафіку.

Крім того, процес оптимізації не повинен вимагати повторного обчислення цих результатів з нуля для кожного можливого налаштування налаштовуємих параметрів; натомість слід використовувати інкрементні алгоритми, щоб мінімізувати витрати на обчислення для дослідження простору пошуку. Нарешті, експериментальні результати на реальних вхідних даних слід використовувати для керування евристичними методами, які обмежують простір пошуку з точки зору кількості параметрів та діапазону їхніх значень. У наступному розділі ми досліджуємо ці принципи в контексті оптимізації маршрутизації всередині однієї AS на основі звичайних протоколів маршрутизації за найкоротшим шляхом із регульованими вагами країв. Ця проблема широко вивчалася протягом останніх кількох років, і рішення були включені в інструменти управління, які використовуються в багатьох оперативних мережах IP [9,11,12].

Далі ми опишемо, як розширити модель маршрутизації, щоб точно охопити роботу великих магістральних мереж, які мають кілька точок виходу для спрямування трафіку до кожного пункту призначення. Потім ми досліджуємо роль оптимізації в налаштуванні політик міждоменного маршрутизації, які контролюють те, як трафік проходить між AS. У висновку ми коротко обговорюємо останні дослідницькі роботи з розробки протоколів розподіленої маршрутизації, які легше налаштовувати, а також централізованих підходів для прямого обчислення маршрутів від імені оперативних маршрутизаторів.

Багато автономних систем (AS) запускають протоколи маршрутизації, такі як Open Shortest Path First (OSPF) [13] або Intermediate System-Intermediate System (IS-IS) [14], які обчислюють найкоротші шляхи на основі налаштовуємих ваг зв’язку. У цьому розділі ми формулюємо задачу оптимізації для вибору ваг зв’язку на основі топології мережі, матриці трафіку та цільової функції. На перший погляд, налаштування єдиного цілого значення ваги для кожного посилання може здатися недостатньо гнучким, щоб задовольнити різноманітні цілі продуктивності. Тим не менш, експериментальні результати показують, що можливо досягти майже оптимальної маршрутизації для реальних топологій і матриць трафіку. Незважаючи на те, що проблема оптимізації є NP-складною, методи локального пошуку є напрочуд ефективними у пошуку хороших рішень і легко підтримують різноманітні цілі продуктивності та операційні обмеження.

Дослідження різних методів пошуку показали, що можна знайти майже оптимальні налаштування ваг зв’язку на графах із сотнями вузлів за розумний проміжок часу [1,13,14]. Крім того, правильне налаштування вагових коефіцієнтів зв’язку працює майже так само добре, як і теоретична верхня межа розв’язання проблеми потоку кількох товарів, яка передбачає більшу гнучкість у маршрутизації трафіку, ніж дозволяють алгоритми маршрутизації за найкоротшим шляхом. На практиці різниця в продуктивності між хорошим налаштуванням ваги та рішенням для багатокомпонентного потоку особливо мала для реальних мережевих топологій, які зазвичай мають відносно малий діаметр і вузький діапазон пропускної здатності зв’язку. Хоча виконання локального пошуку обчислювально недороге для невеликих графіків, визначення хороших налаштувань ваги у великих графіках може свідчити про альтернативний підхід. Багатообіцяючою альтернативою є двофазний алгоритм, де на першому етапі кожен елемент матриці трафіку розподіляється на один шлях, а на другому етапі намагається знайти налаштування вагових коефіцієнтів зв’язку, які можуть досягти цих шляхів [15].

Використання локального пошуку для оптимізації ваг зв’язку забезпечує величезну гнучкість у вирішенні багатьох практичних обмежень щодо встановлення ваг зв’язку в робочих мережах. Зокрема, мінімізація кількості та частоти змін ваги є дуже важливою, щоб уникнути перебоїв у роботі мережі. Коли вага зв’язку змінюється, нова інформація розповсюджується по всій мережі, і маршрутизатори перераховують свої найкоротші шляхи. Протягом цього періоду конвергенції маршрутизатори в мережі не мають узгодженого представлення найкоротших маршрутів для деяких пунктів призначення; тим часом пакети

можуть бути втрачені або мати тривалі затримки. Хоча нещодавня робота значно скоротила час конвергенції [16], затримки в кілька секунд не є рідкістю. Таким чином, оператори не змінюють ваги каналів, якщо поточна конфігурація маршрутизації не спричиняє проблеми з продуктивністю. Стійкість до варіацій у матриці трафіку:

Налаштування ваг зв'язку не повинно бути чутливим до невеликих варіацій у матриці трафіку M. Існуючі методи вимірювання або висновку матриці трафіку [15, 17] мають обмеження щодо їх точності, а сама матриця трафіку коливається через деякий час. Найпростішим способом врахування невизначеності було б збільшити елементи Mi,j матриці трафіку на деяку цільову величину.

Загалом, алгоритм локального пошуку міг би оцінити параметр-кандидат для вагових коефіцієнтів посилання на набір матриць трафіку, віддаючи перевагу рішенню, яке добре працює для кожної матриці трафіку, а не тому, яке найкраще працює для однієї за рахунок інших. Цей підхід надзвичайно ефективний для вибору вагових коефіцієнтів зв'язку, які враховують добові цикли трафіку, дозволяючи мережевим операторам мати єдине призначення вагових коефіцієнтів каналів як для денного, так і для нічного трафіку [10]. Фактично, нещодавня робота [17] показує, що можна знайти рішення маршрутизації, які добре працюють незалежно від матриці трафіку або в надзвичайно широкому діапазоні вимог трафіку. Пережиті збої обладнання та планове технічне обслуговування: в ідеалі призначення вагових зв'язків було б стійким до звичайних збоїв у мережі, таких як збої одного з'єднання. У разі збою зв'язку інформація розповсюджується по всій мережі, і маршрутизатори обчислюють нові найкоротші шляхи по краях, що залишилися. Запобігання перевантаженню після збою може вимагати від алгоритму локального пошуку оцінки параметрів кандидата вагових коефіцієнтів зв'язку для кожного сценарію збою [18]. Однак оцінка вагових коефіцієнтів за всіма помилками є непомірною для обчислень у великих мережевих налаштуваннях.

На щастя, можна ідентифікувати та оцінити набагато менший набір критичних сценаріїв [19]. На практиці ваги каналів, які добре працюють у вихідній топології мережі, продовжують працювати добре після більшості збоїв одного каналу. Однак для деяких сценаріїв збою може знадобитися змінити ваги посилань, щоб уникнути перевантаження. На щастя, зміни однієї або двох ваг зв'язку зазвичай достатньо, щоб зменшити перевантаження. Як профілактичний захід, необхідні зміни ваги можуть бути обчислені заздалегідь до будь-якого збою з'єднання та збережені системою керування мережею. Ці самі попередньо обчислені зміни ваги надзвичайно корисні, коли мережеві оператори повинні навмисно «вивести з ладу» канал, щоб виправити або оновити обладнання; у цьому випадку зміни ваги можна внести заздалегідь, до відключення обладнання для обслуговування.

Підтримка різноманітних обмежень щодо налаштувань шляхів: коли між двома маршрутизаторами існує кілька найкоротших шляхів, маршрутизатори вздовж шляхів розподіляють трафік між кількома вихідними посиланнями – практика, відома як рівноцінна багатошляхова маршрутизація. Замість того, щоб чергувати ці посилання на рівні пакетів, маршрутизатори зазвичай намагаються пересилати пакети для тієї самої пари джерело-одержувач по одному шляху; це зменшує ймовірність того, що пакети з того самого TCP-з'єднання надходять до одержувача не в порядку. Балансування навантаження зазвичай досягається шляхом виконання геш-функції для IP-адрес джерела та призначення кожного пакета; значення геш-функції визначає, яке вихідне посилання має нести пакет. Як наслідок, зв'язки вносять невизначеність у точний розподіл трафіку за посиланнями; з двома вихідними посиланнями трафік не обов'язково ділиться точно навпіл. Локальний пошук може легко пояснити вплив невизначеності 10, штрафуючи рішення, які мають зв'язки (наприклад, встановивши Pi,j на 0,6, а не на 0, 1, неявно припускаючи, що обидва канали повинні нести 60% вихідного трафіку).

З іншого боку, наявність кількох найкоротших шляхів є перевагою в деяких налаштуваннях. Зокрема, якщо одне з двох з'єднань вийде з ладу, інший найкоротший шлях усе ще залишається, дозволяючи мережі об'єднуватися швидше, спричиняючи менше втрат пакетів і затримок [10]. Алгоритм локального пошуку може легко схилитися до рішень із більшою кількістю зв'язків, включивши кількість зв'язків у цільову функцію. Загалом,

використання локального пошуку дозволяє самій цільовій функції змінюватися з часом, оскільки важливі різні показники. Наприклад, для найкращого Інтернет-трафіку мінімізація максимального використання каналу (або суми f() для всіх каналів) є дуже ефективним способом максимізації пропускної здатності TCP. Однак ситуація зовсім інша, коли на сцену з'являються такі інтерактивні програми, як Voice-over-IP (VoIP) і відеоігри. Для цих програм важливо підтримувати низьку затримку розповсюдження (наприклад, нижче 100 мс), а збоїв під час конвергенції протоколу маршрутизації слід уникати шляхом зменшення частоти змін маршрутизації та вибору рішень, які мають кілька найкоротших шляхів. Базовий механізм локального пошуку є надзвичайно гнучким для включення складних показників і зміни їх важливості в оцінці налаштувань кандидатів на ваги посилань.

Переважна більшість Інтернет-трафіку має пройти кілька автономних систем (AS) на шляху до місця призначення. У той час як попередній розділ зосереджувався на одній мережі окремо, у цьому розділі розглядаються додаткові проблеми, з якими стикаються транзитні AS (наприклад, великі Інтернет-провайдери), які підключаються до інших мереж Інтернету. Оптимізація вагових коефіцієнтів зв'язку в цих мережах потребує складнішої моделі, яка фіксує вплив маршрутизації з раннім виходом, а також показників оптимізації, які віддають перевагу рішенням, які уникають переміщення трафіку з однієї точки виходу в іншу.

Пункти призначення з кількома точками виходу

Хоча багато мереж здебільшого автономні, постачальники послуг транзиту підключаються до багатьох інших мереж у різних географічних точках. Наприклад, AS A дозволяє клієнтській мережі (наприклад, компанії або університетському містечку), підключеній до маршрутизатора i, досягати пунктів призначення (наприклад, веб-серверів) в AS B і C.

Великі транзитні AS часто взаємодіють один з одним у декілька місць для підвищення надійності та продуктивності AS. Дедалі частіше клієнти підключаються до своїх постачальників у кількох місцях для підвищення надійності. Кожне з'єднання складається з двох маршрутизаторів – по одному в кожному домені – які обмінюються інформацією про доступність за допомогою протоколу Border Gateway Protocol (BGP). По суті, BGP – це клей, який утримує разом різні частини Інтернету. Наприклад, маршрутизатор в AS B оголошуватиме шлях призначення dB до сусіднього маршрутизатора в AS A. Оскільки дві мережі є одноранговими у двох місцях, маршрутизатори в AS A мають дві можливі точки виходу (посилання на AS B у маршрутизаторах j і k) для спрямування трафіку до місця призначення дБ. На високому рівні два міждоменні шляхи здебільшого еквівалентні, наприклад, обидва шляхи проходять однакову кількість AS (і, фактично, ту саму AS) на шляху до місця призначення. Коли маршрутизатор має два «однаково хороші» міждоменні маршрути до пункту призначення, рішення щодо маршрутизації BGP залежить від вартості внутрішньодоменного шляху до кожного вихідного маршрутизатора. Наприклад, маршрутизатор і направляв би трафік, призначений до дБ, через маршрутизатор j, оскільки внутрішньодоменний шлях вартості 10 до маршрутизатора j коротший, ніж шлях вартості 20 до маршрутизатора k. Загальна практика раннього виходу або гарячої маршрутизації намагається звести до мінімуму використання внутрішніх мережевих ресурсів, перекидаючи трафік до сусідньої AS якомога раніше. Насправді AS B буде виконувати маршрутизацію раннього виходу у зворотному напрямку для трафіку від dB до клієнта в мережі ASA, що призведе до асиметричної маршрутизації, коли трафік від клієнта залишає AS A через маршрутизатор j, а трафік до клієнта входить через маршрутизатор k. Враховуючи велику кількість пунктів призначення, розповсюджених по всьому Інтернету, транзитній AS доведеться вибирати між кількома точками виходу для значної частини пунктів призначення, що робить вкрай важливим фіксувати вплив маршрутизації раннього виходу на потік трафіку.

Потік трафіку через транзитну мережу залежить від маршрутів BGP, рекламованих сусідніми AS, а також від локальних політик, налаштованих на окремих маршрутизаторах. У цьому розділі ми розглянемо, як налаштування політик BGP впливає на пересилання трафіку даних. Далі ми опишемо, як розширити модель маршрутизації з попереднього розділу, щоб відобразити роль політик маршрутизації. Потім ми обговорюємо підходи до дослідження

дуже великого простору пошуку конфігурацій політики BGP, а також фундаментальні обмеження на здатність моделей передбачати навантаження на посилання в AS.

Політики BGP, що впливають на процес вибору шляху

У найпростішому випадку маршрутизатори в AS вибирають маршрути BGP із найкоротшим шляхом AS, розриваючи зв'язки на основі близькості точок виходу. Загалом, маршрутизатор можна налаштувати для застосування політики, яка призначає локальні переваги маршруту, щоб вибрати один маршрут замість іншого з коротшим шляхом AS або ближчою точкою виходу. Сучасні маршрутизатори забезпечують надзвичайно гнучку «мову програмування» для вказівки правил для призначення атрибута локальних переваг. Наприклад, політики можуть розрізняти маршрути, отримані від різних сусідніх AS, на основі комерційних відносин. У транзитних мережах загальноприйнятою практикою є надання більшої переваги маршрутам BGP, отриманим від клієнтів, ніж маршрутам, отриманим від постачальників вищестоящого зв'язку, щоб гарантувати, що трафік даних проходить через сусідів, які платять клієнтам, навіть якщо шлях через постачальника коротший [14, 15]. Подібним чином мережа може призначити нижчу локальну перевагу маршрутам BGP, отриманим через канали з низькою пропускною здатністю, які існують лише для надання резервної служби [16], щоб гарантувати, що трафік даних перетинає високу пропускну здатність. основні ланки, за винятком випадків, коли вони вийшли з ладу.

На додаток до призначення переваг на основі відносин із сусідньою AS, оператори конфігурують політики, щоб впливати на навантаження на мережеві з'єднання [12, 17, 18]. Наприклад, припустімо, що AS вивчає маршрути BGP до пункту призначення від двох постачальників вищестоящого зв'язку. Призначаючи нижчі локальні переваги для одного маршруту, AS вирішує спрямувати трафік через маршрут, отриманий від іншого постачальника. Ретельне призначення локальних переваг над діапазоном пунктів призначення допомагає збалансувати навантаження на два посилання. У деяких випадках один провайдер може стягувати вищу ціну за пересилання трафіку або загалом мати нижчу продуктивність. Щоб зменшити фінансові витрати або підвищити продуктивність, AS може віддавати перевагу маршрутам через іншого провайдера, аж до моменту, коли канал стає занадто сильно завантаженим. Загалом, AS може підключатися до однієї AS у кількох географічних місцях. Якщо одне з посилань до сусідньої AS стає перевантаженим, оператори Provider local-pref 90 первинний шлях Transit AS local-pref 100 Customer вторинний шлях d можуть змінити конфігурацію сусіднього маршрутизатора, щоб призначити нижче локальне значення переваги деяким із отриманих маршрутів BGP. на цьому місці. Це гарантує, що маршрутизатори в AS спрямовують трафік для цих пунктів призначення через інші, малозавантажені посилання до тієї ж AS.

Моделювання впливу на потік трафіку

Щоб відобразити вплив політики маршрутизації, потрібні розширення моделі, представленої раніше [19]. Змінюючи налаштування атрибута local-preference, налаштування політик BGP впливає на те, як маршрутизатор на периферії мережі вибирає найкращий маршрут. Це, у свою чергу, впливає на вихідний набір для кожного блоку адреси призначення (або префікса). Моделювання впливу політик BGP на потік трафіку вимагає:

– Маршрутів, отриманих BGP від сусідніх AS: вхідними даними для обчислення є набір маршрутів, отриманих BGP від сусідніх AS. На практиці маршрути, отримані за допомогою BGP, можна зафіксувати шляхом «скидання» таблиці маршрутизації BGP на кожному маршрутизаторі або моніторингу маршрутів BGP, коли вони оголошуються сусідами.

– Специфікація політик маршрутизації: політика маршрутизації BGP – це послідовність пунктів, де кожен пункт визначає набір маршрутів (наприклад, на основі префікса призначення або елементів у шляху AS) і призначає значення локального параметра для відповідності маршрути. Наприклад, політика може призначити локальне налаштування 100 для всіх маршрутів з AS «1234» як другий стрибок і 90 для всіх інших.

– Модель вибору шляху BGP: обчислення вихідного набору вимагає застосування політик до маршрутів BGP, отриманих із сусідніх запитів, а потім вибору «найкращого» зі змінених шляхів. Зокрема, модель повинна вибирати маршрути з найвищими локальними

перевагами та, серед них, маршрути з найкоротшою довжиною шляху AS. Зрештою, кожен маршрутизатор вибере маршрут із «найближчою» точкою виходу.

– Пропонований трафік від точок входу до пункту призначення: оцінка потоку трафіку вимагає накладення запитів трафіку vi,d поверх шляхів через AS для досягнення вибраної точки виходу для кожної точки входу та префікса призначення. Маючи точне уявлення про визначені BGP маршрути та вимоги до трафіку, інструмент оптимізації може проводити локальний пошук можливих змін у політиках маршрутизації та оцінювати вплив на потік трафіку через мережу.

Обмеження простору пошуку Значна гнучкість у визначенні політики маршрутизації BGP у поєднанні з великою кількістю префіксів призначення робить простір пошуку надзвичайно великим – надто великим для повного вивчення. На щастя, евристика корисна для обмеження накладних витрат процесу оптимізації [12]:

– Дослідження поступових змін політики: локальний пошук може досліджувати поступові зміни конфігурації маршрутизатора, а не розглядати всі можливі політики з нуля. Наприклад, оптимізація може, помітивши, що одне межове з'єднання перевантажене, зосередитися на зміні політики в цьому одному місці, щоб призначити менші локальні переваги маршрутам BGP для деяких пунктів призначення. Цей підхід не тільки обмежує простір пошуку, але й зменшує накладні витрати на оцінку наслідків зміни конфігурації, оскільки інструменту оптимізації потрібно враховувати лише вимоги трафіку, які переміщуються до нових точок виходу.

– Зосередження лише на популярних префіксах призначення: типова магістральна мережа IP має маршрути BGP для більш ніж 150 000 блоків адрес призначення. На практиці переважна більшість цих напрямків сприяють отриманню лише невеликої частини трафіку [10]. Зміна значень локальних параметрів для невеликої кількості популярних префіксів місць призначення перемістить значну кількість трафіку з однієї точки виходу в іншу, мінімізуючи при цьому кількість змін маршруту та зменшуючи накладні витрати на локальний пошук. Крім того, ці популярні напрямки, як правило, мають більш стабільні кандидати BGP 17AS C AS A інший провайдер AS B після після нижнього сусіда до попереднього маршрути [11], що робить імовірним те, що нова конфігурація політики BGP буде ефективною в майбутньому.

– Зосередження на групах блоків адрес призначення: на практиці мережа має однакові (або схожі) параметри маршрутизації BGP для кількох префіксів призначення. Наприклад, великий клієнт може мати дюжину блоків адрес, які рекламуються точно так само, в тих самих місцях мережі. Визначення політик маршрутизації, які збігаються в групі пунктів призначення (скажімо, шляхом зіставлення загальних аспектів їхніх ASpaths), зменшує простір пошуку, а також робить маршрутизацію в мережі менш чутливою до невеликих змін в інших атрибутах маршруту. Крім того, групи пунктів призначення, як правило, мають більш стабільний сукупний обсяг трафіку, що робить прогнози навантаження трафіку в мережі більш точними.

Поєднання цих трьох методів суттєво зменшує накладні витрати на дослідження простору пошуку та, як правило, надає перевагу надійним рішенням. Однак у деяких випадках вплив зміни політики BGP може бути важко передбачити через побічні ефекти на сусідніх доменах [12].

У прикладі AS A і B рекламують шляхи до пунктів призначення в AS C. Спочатку існує п'ять «найкращих» маршрутів – два через AS A і три через AS B. Маршрутизатори на маршруті західного узбережжя через AS A і маршрутизатори на маршруті східного узбережжя через AS B. Припустімо, що крайнє ліве з'єднання до AS Bis перевантажене (як показано пунктирною лінією), і політику BGP для цієї точки виходу змінено 18, щоб призначити нижчу локальну перевагу маршрутам, що походять від AS C

Після цієї зміни деякі маршрутизатори можуть перемикатися з маршруту через крайнє ліве посилання до AS B на маршрут через крайнє праве посилання до AS A.

Ці маршрутизатори будуть рекламувати новий найкращий шлях до сусідніх мереж. Залежно від політики маршрутизації сусіда, нове оголошення може змусити сусіда вибрати іншу наступну AS (тобто іншого провайдера) для досягнення цього префікса. Це може

призвести до непередбачуваного зменшення обсягу трафіку, що надходить у домен через цей маршрутизатор. Подібним чином зміна маршруту може спровокувати збільшення трафіку, якщо інші сусіди віддадуть перевагу маршруту (A,C) над маршрутом (B,C).

Такі побічні ефекти ускладнюють прогнозування того, як зміна політики маршрутизації вплине на обсяг трафіку в мережевих посиланнях. Щоб запобігти поширенню змін маршрутизації на сусідні домени, інструмент оптимізації повинен, коли це можливо, досягати цілей балансування навантаження шляхом коригування політики маршрутизації для пунктів призначення, для яких кожен потенційно найкращий маршрут має однаковий шлях AS. На практиці AS матиме багато таких адресатів. Наприклад, пункт призначення, який знаходиться в AS B, буде доступним через три точки виходу до ASB, які пропонують той самий шлях ASpath з одним стрибком.

Зміна політики маршрутизації для одного (або кількох) із цих префіксів призначення перемістить трафік від пунктирного посилання до одного з інших посилань до AS B, не змінюючи шляху AS, який бачить сусідній вузол. Залежно від реалізації BGP подальша AS може навіть не отримати нове оголошення BGP, оскільки шлях AS не змінився. Це дозволяє мережі зменшувати перевантаження на каналі, переміщуючи трафік на інші, малозавантажені канали, не змінюючи інформацію про маршрутизацію BGP, що надсилається до сусідніх доменів.

Алгоритм ARTCP

Запропонований у даній роботі, удосконалений протокол Adaptive Rate Transmission Control Protocol (ARTCP) системи адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем, позичає деякі механізми від протоколу TCP. В ARTCP повністю переглянутий алгоритм керування потоком, що і відрізняє його від TCP. Разом з тим, запропонований протокол може забезпечувати сумісність із TCP.

Протокол ARTCP, запропонований у цій роботі, здатний працювати більш ефективно і якісно, ніж TCP, однак можна виділити кілька напрямків подальших досліджень нового протоколу, які можуть, по-перше, дати можливість ефективно використовувати його в асиметричних системах, а по-друге, досягти рівноправності між потоками з різною довжиною маршруту.

Асиметричні системи

Оскільки в протоколі ARTCP усунута ACK-синхронізація, властива TCP, то відправлення сегментів відбувається незалежно від прибуття підтверджень аж до вичерпання максимального вікна, то на відміну від TCP, ARTCP може бути вдосконалений так, щоб ефективно працювати в системах з асиметричними каналами.

Для використання ARTCP у таких системах необхідно зменшити частоту підтверджень. Оскільки штучна затримка підтверджень викличе збільшення затримки в петлі зворотного зв'язку, то, вимір затримки передачі сегментів потрібно також зв'язати з одержувачем. Оскільки важко домогтися гарної синхронізації системних годин одержувача й відправника, то одержувач може лише помічати зміну часу передачі сегментів, якщо відправник використовує стандартне поле часової мітки [6]. Якщо різниця значень мітки в потоці й системних годинах одержувача змінюється, значить змінюється й абсолютне значення затримки. У цьому випадку одержувач повинен збільшити частоту підтверджень, щоб відправник міг зреагувати на зміну навантаження в мережі. Коли значення швидкості прибуття потоку й затримки передачі не міняються, частота підтверджень може бути знову зменшена.

Алгоритм ARTCP не містить перешкод для цього вдосконалення, крім того, модифікований таким способом ARTCP для асиметричних каналів збереже сумісність із представленим тут протоколом.

З'єднання з різними RTT

Для з'єднань, що володіють різним часом RTT, через розходження довжин їхніх маршрутів, середнє значення коефіцієнта F обмежується деяким числом, меншим 1, як показано в роботах [10, 11, 12, 13]. Це вірно як для ARTCP, так і для TCP. Щоб досягти рівноправності поділу ПрЗд між ARTCP потоками з різною довжиною маршруту, необхідно

усунути залежність коефіцієнта *speedup* від мінімального часу RTT. Цього результату можна досягти шляхом модифікації алгоритму адаптації таким чином, щоб у режимі **FT** повністю відмовитися від використання обмірюваного RTT як індикатора перевантаження, використовуючи лише значення міжсегментних інтервалів потоку.

Імітаційна модель

Для дослідження можливостей протоколу й відпрацьовування його механізмів була розроблена програмна імітаційна модель самого протоколу й мережних компонентів, у середовищі яких повинен функціонувати протокол ARTCP. Модель являє собою набір компонентів реальні об'єкти, що імітують, у складі мережі й об'єкти протоколів ARTCP і CBR (Constant Bit Rate). Досліджувана мережа може мати топологічну схему досить великої складності, що будується з необхідного числа екземплярів наявних класів.

Основними способами що дозволяють досліджувати й верифікувати складні мережні протоколи є стенди для тестування (експериментальні мережі), програмні емулятори й системи автоматизованої верифікації. Верифікацію набору процедурних правил протоколу можна здійснювати за допомогою програмного перебору станів кінцевого автомата представляючого протокол [16, 17, 18], наприклад, в інтерпретаторі мови PROMELA, такому як SPIN [1]. Однак верифікація набору процедурних правил, і вивчення ефективності протоколу переслідують різні цілі й, відповідно, для них застосовуються різні інструменти. Верифікація протоколу означає застосування до його набору процедурних правил формального методу, що дозволяє довести, що цей набір або моделююча його система кінцевих автоматів (з обміном даними) повна, не містить недосяжних станів, вільна від статичних і динамічних блокувань. Верифікація протоколу не ставить задачею навіть визначення кількісних характеристик його ефективності, з погляду верифікації важливо лише те, що прогресивний обмін даними взагалі відбувається. Внаслідок цього методи верифікації побудовані на повному або частковому аналізі доступних станів КА, що представляє протокол. Для систем з великим числом станів ($>10^5$) верифікація заснована на повному аналізі утруднена на практиці.

Моделювання протоколу не дає гарантії повного аналізу досяжних станів, але зате дозволяє досліджувати кількісні характеристики системи. Разом з тим моделювання дозволяє зробити статистично обґрунтований висновок про надійність протоколу, принаймні, щодо відсутності в ньому блокувань. Складність імітаційного моделювання в тому, що крім реалізації процедурних правил самого досліджуваного протоколу до складу моделі повинні входити всі його компоненти: середовище функціонування, словник і способи кодування повідомлень, модель сервісу протоколу. Однак при цьому моделювання вимагає все-таки менших витрат, ніж розгортання експериментальної мережі для дослідження властивостей протоколу. У цьому випадку наше завдання полягає не у верифікації протоколу ARTCP, а у визначенні чисельних значень його характеристик у різних умовах.

Розробка структурної схеми

У протоколі ARTCP повністю перероблені всі механізми керування потоком.

Механізм корекції помилок передачі в ARTCP не впливає на швидкість передачі. Від TCP збережені віконний механізм для керування завантаженням одержувача, алгоритми визначення RTT і установки таймера ТПП. Ознакою втрати сегмента служить спрацьовування ТПП або прихід двох послідовних підтверджень одного сегмента. Алгоритм керування швидкістю містить у собі: функції диспетчеризації сегментів, виміру швидкості й адаптації швидкості (рисунок 1).

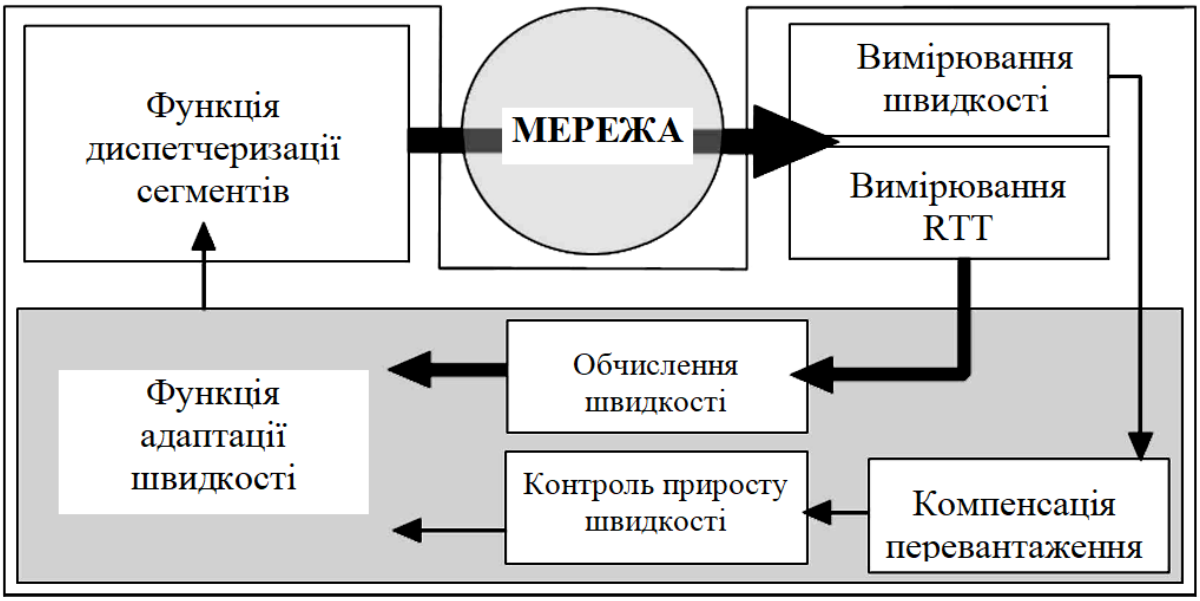


Рисунок 1 – Структурна схема системи

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем; Досліджена система адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем; На основі отриманих результатів досліджень створена програмна реалізація системи адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем; Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання адаптивної оптимізації маршрутизації мережі інформаційних та комп'ютерних систем. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12. (Scopus).
2. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022. (Scopus).
3. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34. (Scopus).
4. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477. (Scopus).
5. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 137, 2021. https://doi.org/10.1007/s42979-021-00739-w (Scopus).
6. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
7. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data

of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).
8. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
9. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
10. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
11. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
12. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
13. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
14. Смірнов О.А., Дрєєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.
15. Смирнов А.А., Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений. Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.
16. Смирнов А.А., Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения. Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.
17. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95
18. Смірнов, О.А., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю.Усік П.С., «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.
19. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.
20. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

УДК 004

В.Гут, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНОЇ ПЛАТФОРМИ НА БАЗІ РІШЕНЬ CISCO

У статті розроблено програмне забезпечення, яке призначено для системи інформаційно-комунікаційної платформи на базі рішень Cisco. Метою розробки є дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco. Об’єктом дослідження є процес інформаційно-комунікаційної платформи на базі рішень Cisco. Предметом дослідження є методи інформаційно-комунікаційної платформи на базі рішень Cisco. Методи дослідження базуються на методах теорії обробки телекомунікаційного трафіку, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп’ютерні науки, інформаційно-комунікаційна платформа, Cisco

Постановка проблеми. В наш час існує множина різних програм, що дозволяють вести телефонні переговори через Інтернет або локальну мережу. Така можливість уже нікого не дивує, для цього потрібні лише комп'ютер, підключений до мережі, відповідна програма й мікрофон з навушниками. Звичайно, таке рішення явно не підходить для організації телефонії в серйозній фірмі (все-таки подібні засоби носять скоріше розважальний характер), однак ідея передачі голосу через мережу передачі даних дуже приваблива, особливо якщо фірма має множину офісів у різних містах. І в цьому випадку рано або пізно виникає питання про впровадження IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco.

IP-телефонія інформаційно-комунікаційної платформи на базі рішень Cisco, по суті, є способом організації телефонного зв'язку з використанням мережі передачі даних для передачі голосу. Переваги такої організації телефонного зв'язку очевидні, і головне з них – істотне зниження витрат на дзвінки між офісами, розташованими в різних містах. Крім цього, даний підхід дозволяє ввести єдиний номерний план для всієї організації, коли не потрібно пам'ятати телефонні коди міст, у яких перебувають філії компанії. Ну й звичайно, не варто забувати про впровадження додаткових сервісів.

Корпоративна IP-телефонія інформаційно-комунікаційної платформи на базі рішень Cisco дозволяє об'єднати вже існуюче в організації телефонне встаткування (звичайні телефони, підключені до АТМУ) і спеціалізовані IP-телефони в одну систему, що використовує для передачі голосового трафіку мережі передачі даних.

В зв’язку з тим, що багато фірм мають корпоративну мережу передачі даних, побудовану з використанням активного мережного устаткування фірми Cisco Systems, у цьому магістерському проекті, особлива увага приділена рішенням, які пропонує саме ця компанія.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи інформаційно-комунікаційної платформи на базі рішень Cisco.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем інформаційно-комунікаційної платформи на базі рішень Cisco.

– Дослідження системи інформаційно-комунікаційної платформи на базі рішень Cisco.

– Програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco.

Об'єктом дослідження є процес інформаційно-комунікаційної платформи на базі рішень Cisco.

Предметом дослідження є методи інформаційно-комунікаційної платформи на базі рішень Cisco.

Методи дослідження базуються на методах теорії обробки телекомунікаційного трафіку, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Cisco AVVID

Рішення Cisco для побудови мереж IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco засновано на використанні архітектурної моделі Cisco AVVID (Architecture for Voice, Video and Integrated Data) і призначено для рішення наступних основних задач:

побудова сучасної багатофункціональної системи цифрової телефонії на базі корпоративної IP-мережі;

підключення системи корпоративної IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco до телефонної мережі загального користування й стикування з існуючими ділянками традиційної телефонної мережі компанії;

забезпечення широкого кола сучасних сервісів для абонентів корпоративної мережі IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco.

Крім того, дане рішення дозволяє створити мережу відеотелефонії, що може бути частиною корпоративної IP телефонної системи.

За допомогою засобів IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco можна побудувати й невелику мережу в кілька десятків користувачів малого підприємства або віддаленого офісу компанії, і мережа великої корпорації в кілька сотень тисяч абонентів.

Архітектура пропонованого рішення дозволяє технологічно й економічно ефективно створити географічно розподілену мережу корпоративної телефонії.

Рішення IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco складається з наступних основних компонентів:

Інтелектуальна мережна інфраструктура на базі протоколу IP, що включає маршрутизатори, комутатори, шлюзи й інше мережне встаткування. IP-інфраструктура є основою для подальшого впровадження користувальницьких додатків і повинна забезпечувати підтримку таких життєво важливих для мережі сервісів, як безпека, мережне керування й механізми якості обслуговування (QoS). У рамках архітектури Cisco AVVID інтелектуальна мережна інфраструктура використовується поряд з передачею даних для функціонування корпоративної телефонної й відеотелефонної системи.

Інтелектуальні клієнтські пристрої з підтримкою протоколу IP, у тому числі цифрові IP-телефони Cisco, відеопристрої, персональні комп'ютери зі спеціалізованим програмним забезпеченням для рішення різних бізнес-задач, програмні емулятори телефонів (наприклад, IP Communicator) і так далі.

Керування корпоративною системою IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco, а також відеотелефонії Cisco здійснюється спеціалізованим додатком Cisco Call Manager або кластером Cisco Call Manager. Крім того, у системі можуть використовуватися додаткові службові пристрої й додатки, такі як корпоративна служба

каталогів, що служить централізованим сховищем інформації про абонентів у телефонній і відеосистемі, а також службові пристрою для забезпечення аудіо– і відеоконференцій, H.323-гейткіпери й т.д.

Сучасні телефонні додатки, що виникли завдяки розвитку інтегрованих систем з підтримкою голосу, відео– і даних, наприклад, система уніфікованої обробки повідомлень (Unified Messaging), інтелектуальні центри обробки викликів (Contact Center), мультимедійні системи організації конференцій. Впровадження подібних додатків створює додаткові можливості для користувачів/абонентів корпоративної телекомунікаційної мережі, підвищує зручність і ефективність використання системи.

Керуючий сервер Cisco Call Manager забезпечує керування встановленням телефонних з'єднань і відеоз'єднань у системі. Call Manager також управляє наданням додаткових функцій абонентам, що використовують як IP-телефони, так і відеопристрої. Він також забезпечує адміністратора мережі засобами для налаштування й керування взаємодією різних компонентів системи IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco.

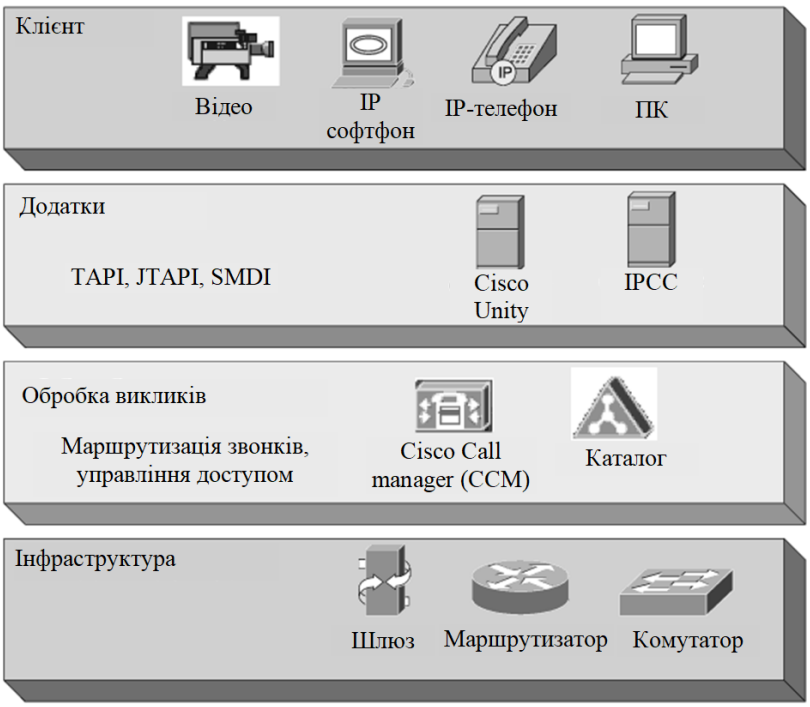


Рисунок 1 – Рівні архітектури AVVID

Спеціалізовані цифрові IP-телефони Cisco підключаються в локальну мережу, що комутирується, Ethernet 10/100 і забезпечують як традиційну функціональність цифрових телефонів, так і ряд нових можливостей.

Для стикування із системами традиційної телефонії, у тому числі із установленими раніше АТМУ, і підключення до телефонної мережі загального користування застосовуються голосові шлюзи. Дана можливість реалізована на базі цілого ряду мультисервісних маршрутизаторів Cisco. Існують також голосові модулі для деяких моделей комутаторів Cisco Catalyst і самостійні пристрої, що забезпечують функціональність голосових шлюзів.

Переваги застосування Cisco AVVID:

–швидкість впровадження нових сервісів;

–надійність;

–можливість взаємодії різних мереж;

–зниження матеріальних витрат.

Архітектура AVVID складається із чотирьох рівнів:

1. Інфраструктурний рівень – це фундамент мережі.

2. Рівень обробки викликів, що виконує функції комутації викликів. Його функції схожі з функціями АТМУ при використанні традиційних технологій телефонії.

3. Рівень додатків, що забезпечують додаткову функціональність.

4. Клієнтський рівень, на якому розташовуються пристрої й додатки, з якими користувач безпосередньо взаємодіє.

Моделі розгортання

Cisco підтримує наступні моделі розгортання IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco:

– однооб'єктна;

– с централізованою обробкою викликів;

– с розподіленоюоднокластерною обробкою викликів;

– с розподіленою багатокластерною обробкою викликів.

Однооб'єктна модель

В однооб'єктній моделі розгортання (рисунок 2) всі додатки CCM і DSP-ресурси фізично розташовані в одному місці.

Відмітні риси моделі:

– CCM, DSP і додатки перебувають в одному місці;

– підтримка приблизно 36 тис. IP-телефонів на кластер;

– кілька кластерів можуть бути з'єднані за допомогою транків;

– для зовнішніх дзвінків використовується телефонна мережа загального призначення (PSTN).

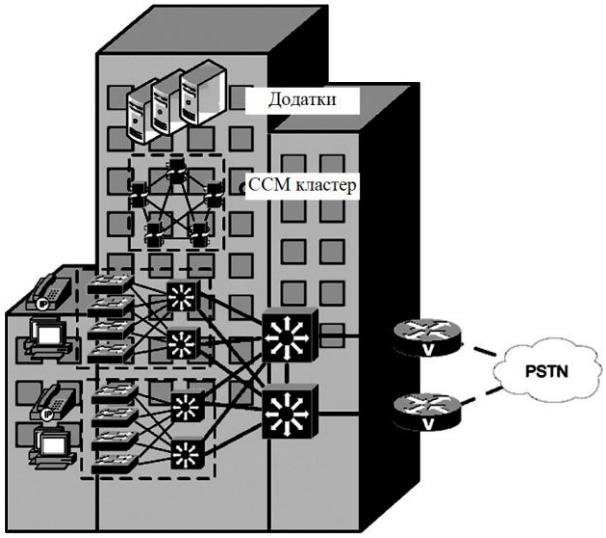


Рисунок 2 – Однооб'єктна модель AVVID

Модель із централізованою обробкою викликів

Дана модель (рисунок 3) має на увазі CCM-кластер на центральному вузлі й з'єднання до віддалених вузлів через мережу передачі даних (IP-мережа з дотриманням QoS). Віддалені вузли звертаються до центрального CCM-кластеру для обробки викликів. Додатка, такі як голосова пошта й автовідповідач, звичайно розташовуються на центральному вузлі. Така організація зменшує витрати на зміст устаткування й забезпечує централізоване адміністрування й обслуговування.

Мережа передачі даних може бути побудована на технологіях виділених ліній, Frame Relay, ATM. На маршрутизаторах, що стикуються з мережею передачі даних, повинні бути реалізовані механізми QoS, такі як установка пріоритетів на чергах і контроль трафіку, для "захисту" голосового трафіку від інших типів трафіку в мережі.

У даній моделі для захисту мережі передачі даних від перевантаження може знадобитися контроль доступу (CAC). У версії CCM Release 3.3 використовується можливість автоматичного вибору маршруту (AAR – Automated Alternate Routing). AAR

дозволяє CCM динамічно перенаправляти виклики через телефонну мережу загального користування, коли мережа передачі даних перевантажена, для запобігання погіршення якості встановлених викликів.

Телефонна мережа загального призначення використовується як резервна. Можна також застосовувати технологію ISDN як резервний канал передачі даних, але ISDN не годиться для передачі голосу, тому що не підтримує вимоги QoS. Навіть якщо віддалений офіс втратить зв'язок із кластером CCM, обробка викликів може бути здійснена за допомогою технології безвідмовної телефонії віддаленого вузла (SRST – Survivable Remote Site Telephony), доступної при використанні маршрутизаторів Cisco IOS. На час втрати зв'язку з CCM ця технологія буде забезпечувати внутрішню комутацію викликів у віддалених точках.

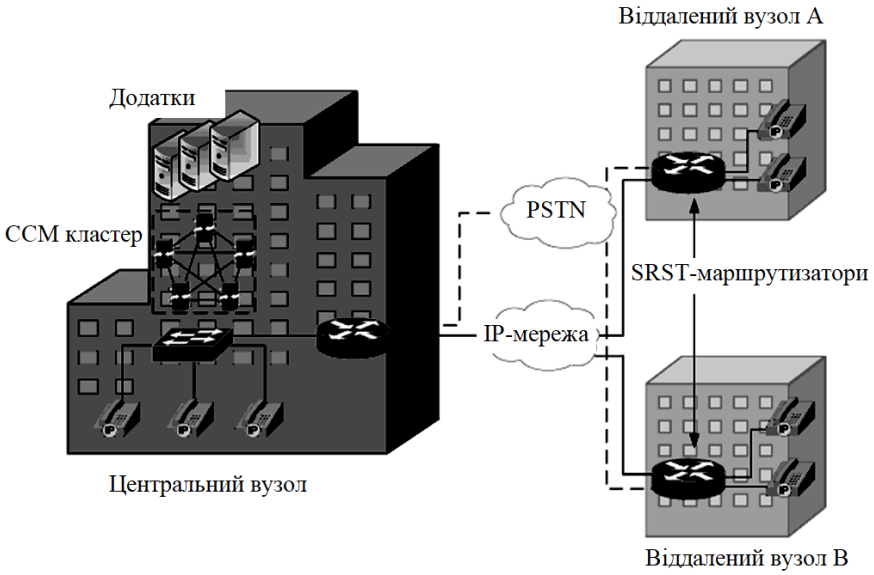


Рисунок 3 – Структура моделі із централізованою обробкою викликів

Моделі з розподіленою обробкою викликів

CCM-кластери можуть бути присутнім у всіх вузлах, у цьому випадку локальна комутація викликів буде вироблятися без участі центрального вузла. Також є можливість рознести компоненти одного CCM-кластера по різних точках (рисунок 4 і рисунок 5).

Процес реєстрації IP-телефонів

Щораз, коли IP-телефон завантажується, відбувається наступний процес:

1. Якщо використовується комутатор Cisco з підтримкою живлення, комутатор посилає спеціальний FLP (Fast Link Pulse) сигнал. Комутатор використовує цей сигнал, щоб визначити, чи підключений до нього IP-телефон, що вимагає живлення. У знеструмленому стані IP-телефон Cisco повертає сигнал, запитуючи тим самим комутатор подати напругу 48 вольтів постійного струму.

2. Після того як IP-телефон одержав живлення й завантажився, комутатор посилає йому CDP (Cisco Discovery Protocol) пакет з інформацією віртуальної голосової локальної мережі (якщо зконфігуровано).

3. IP-телефон надсилає широкомовний запит DHCP-серверу. DHCP-сервер повертає телефону, як мінімум, IP-адресу, маску підмережі й IP-адреса Cisco TFTP-сервера.

4. Телефон з'єднується з TFTP-сервером і одержує з його свою конфігураційну інформацію, що містить перелік CCM (до трьох CCM).

5. Телефон намагається зареєструватися з першим CCM з переліку, отриманого з TFTP-сервера.

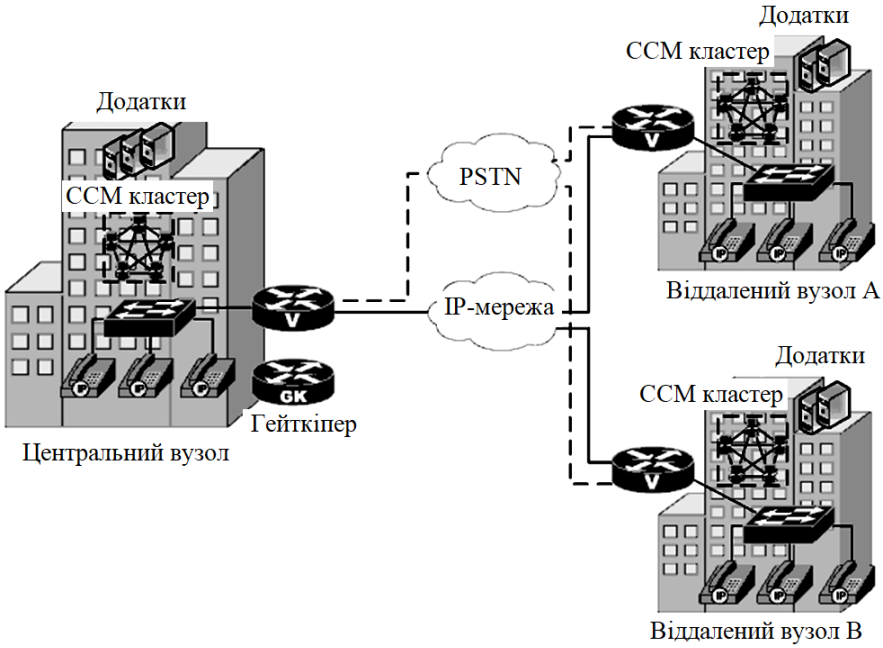


Рисунок 4 – Багатокластерна модель із розподіленою обробкою викликів

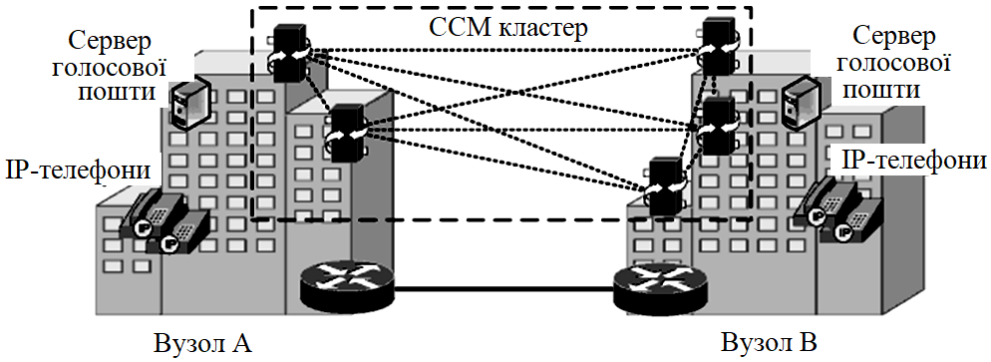


Рисунок 5 – Однокластерна модель із розподіленою обробкою викликів

Комутація на Cisco Call Manager

CCM маршрутизує два типи викликів:

1. Внутрішні (on-cluster).
2. Зовнішні (off-cluster).

Комутація внутрішніх викликів

Коли надходить виклик з IP-телефону, CCM аналізує набраний номер. Якщо він відповідає DN (Directory Number), зареєстрованому на тому же CCM-кластері, CCM направляє виклик на IP-телефон призначення, асоційований з відповідної DN. Це внутрішній (on-cluster) виклик. CCM дозволяє обробляти такі виклики без напрямку його на зовнішній шлюз. Не тільки IP-телефони можуть виступати в ролі пристроїв, здатних ініціювати й приймати внутрішні виклики, це може бути будь-який пристрій із зареєстрованим на CCM DN. Наприклад, такими пристроями можуть бути Cisco софтфони й аналогові телефони, підключені до MGCP-шлюзів або шлюзів, що працюють по протоколі Skinny.

Комутація зовнішніх викликів

Якщо на IP-телефоні набирається номер, для якого не найшлося відповідного DN, виходить, має місце зовнішній (off-cluster) виклик. CCM у цьому випадку переглядає свою таблицю зовнішніх маршрутів, щоб визначити, куди направити виклик. CCM використовує концепцію таблиць маршрутів і шаблонів трансляцій для визначення, куди і як направляти зовнішній виклик.

Можна створювати плани маршрутизації для зовнішніх викликів, використовуючи

триярусну архітектуру, що надає кілька рівнів маршрутизації й маніпуляцій із цифрами. Шаблон маршруту (Route Pattern) визначає по номеру дозвону список маршрутів (Route List), що вибере доступний шлях для вихідного дзвінка на основі пріоритетів. Ці шляхи Cisco визначає як "групи маршрутів" (Route Group). Рівні вибору маршруту показані на рисунку 6.

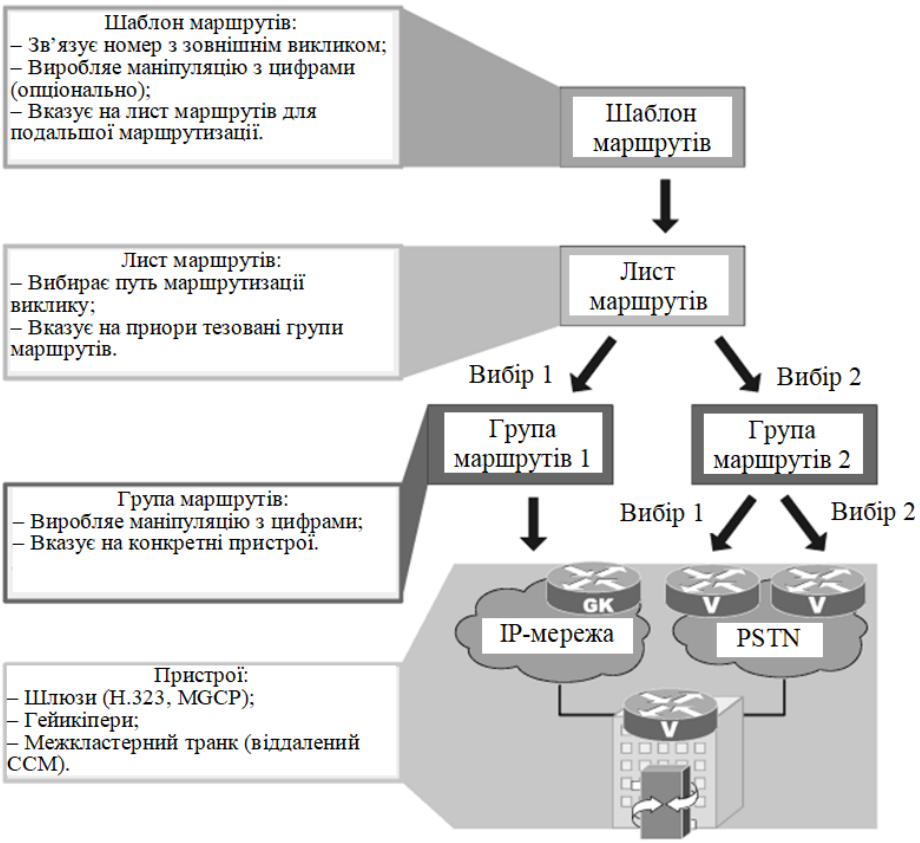


Рисунок 6 – Елементи маршрутизації зовнішніх викликів в CCM

Процес конфігурування маршрутів для зовнішніх викликів містить наступні етапи:

– додавання шлюзів;

– створення груп маршрутів з доступних пристроїв;

– створення списків маршрутів з доступних груп маршрутів;

– створення шаблона маршруту й асоціювання його з доступним списком маршрутів або шлюзом.

Шаблон маршруту є ключовим елементом у плані маршрутизації. Він визначає набраний номер і направляє виклик на підходящий шлюз. Коли набраний номер відповідає шаблону маршруту, CCM направляє виклик на відповідний список маршрутів або шлюз.

Шлюзи

Шлюзи – це пристрої, що дозволяють CCM взаємодіяти з не-IP-мережами, такими як телефонна мережа загального користування (PSTN). Cisco розділяє свої шлюзи на дві головні категорії – аналогові й цифрові. Аналогові шлюзи можуть бути шлюзами станцій або транковими шлюзами.

Шлюзи станцій використовують порти FXS для підключення кінцевих пристроїв, таких як аналогові телефони й факси. Транкові шлюзи використовують порти FXO і підключаються до телефонної мережі загального користування або АТМУ для забезпечення зв'язку з іншими телефонними системами по аналогових лініях.

Цифрові шлюзи забезпечують те ж підключення до телефонної мережі загального користування або АТМУ, однак вони використовують цифрові технології підключення, такі як PRI CCS і транки T1 CAS.

CCM підтримує три типи шлюзів:

–MGCP-шлюзи. Використовує модель клієнт-сервер, у якій CCM управляє шлюзом. MGCP-шлюзи підтримують всі додаткові сервіси CCM, надмірність CCM і безперебійність викликів. Додатковою перевагою таких шлюзів є їхнє нескладне конфігурування.

–Non-IOS MGCP-шлюзи. Аналогічні MGCP-шлюзам, але не підтримують безперебійність викликів.

–H.323-шлюзи. Використовують однорангову модель. Більша частина конфігурації виробляється безпосередньо на шлюзі. При одноранговій моделі CCM не має контролю над шлюзом, що приводить до зменшення кількості доступних сервісів CCM при використанні таких шлюзів. Зате H.323-шлюзи підтримують додаткову функціональність Cisco IOS – CAC і SRST.

Крім перерахованих тут, існує ще один тип шлюзів – міжкластерний транк. Це логічний шлюз, що використовується для комунікації між кластерами CCM.

Конфігурація абонентських пристроїв

Для того щоб зконфігурувати POTS Dial Peer (традиційний телефонний пристрій), потрібно:

1. Зконфігурувати абонентський пристрій типу POTS.
2. Зконфігурувати телефонний номер.
3. Указати, до якого порту пристрій підключений.

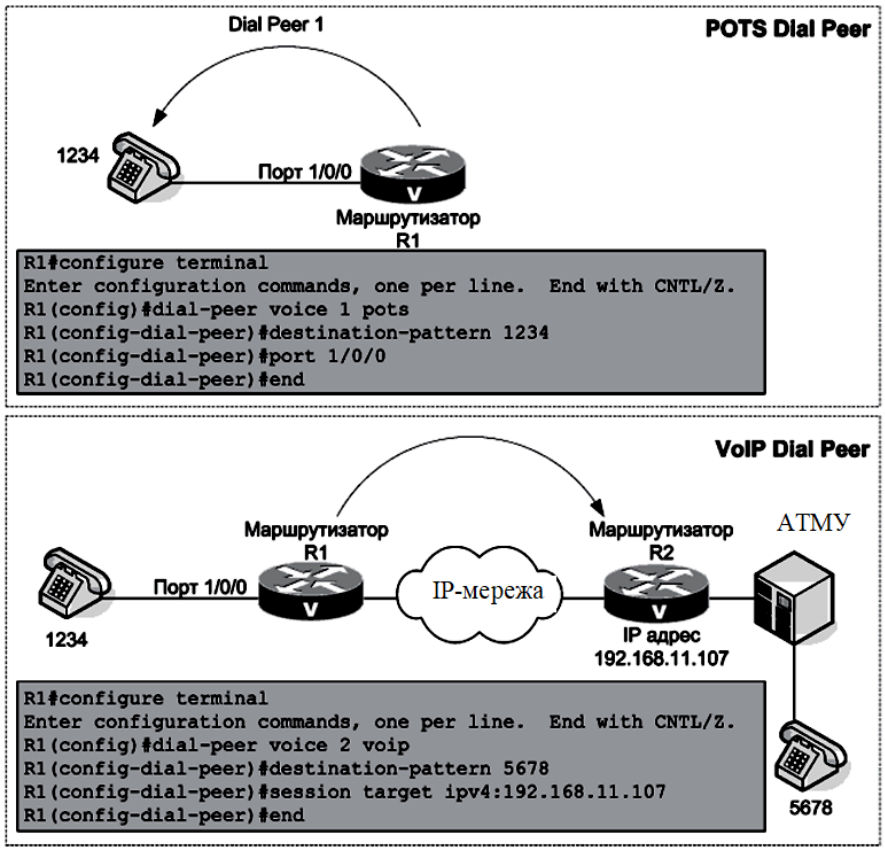


Рисунок 7 – Приклади конфігурації

Якщо до голосового порту підключена АТМУ, то замість телефонного номера може бути зконфігурований діапазон номерів.

Для конфігурування VoIP Dial Peer адміністратор повинен знати, як ідентифікувати пристрій на іншому кінці лінії – це може бути IP-адреса пристрою, а може бути Cisco Call Manager або гейткіпер (контролер зони H.323) (gatekeeper), як використовують для дозволу адрес і управління доступу викликів (CAC – call admission control) для завершення встановлення виклику.

Для конфігурування VoIP Dial Peer необхідно виконати наступні кроки:

1. Зконфігурувати шлях через мережу для голосових даних.
2. Визначити абонентський пристрій як VoIP Dial Peer.
3. Зконфігурувати телефонні номери, доступні через віддалений маршрутизатор (шлюз).
4. Зконфігурувати IP-адресу маршрутизатора (шлюзу), на якому встановлення телефонного з'єднання закінчується.
5. У якості IP-адреси використовувати адресу самого пристрою, а не адресу його порту.

Методи конфігурування діапазонів телефонних номерів

Для конфігурування телефонних номерів використовуються так звані шаблони призначення (Destination Pattern).

Шаблон призначення асоціює телефонний номер з абонентським пристроєм. Він також визначає набираються цифри, що, які маршрутизатор накопичує й перенаправляє на віддалений телефонний інтерфейс (АТМУ, Cisco Call Manager або телефонну мережу загального призначення). Шаблон призначення може вказувати на цілий телефонний номер або на частину номера з підстановочними символами; він може вказувати як на один конкретний номер, так і на діапазон номерів.

Як було видно із прикладів, шаблон призначення задається командою destination-pattern. Параметр команди може складатися з наступних елементів:

–Плюс (+): опціонний символ, що вказує на номер стандарту E.164. E.164 – це рекомендації ITU-T (International Telecommunication Union Telecommunication Standardization Sector) для інтернаціонального плану нумерації. Знак плюс відноситься поперед рядка, що визначає шаблон призначення, і означає, що рядок повинен відповідати рекомендаціям E.164.

–Рядок: набір цифр, що визначають телефонний номер. У рядку крім цифр можуть використовуватися спеціальні символи:

–Зірочка (*) і ґрати (#). Ці символи завжди є на стандартних клавіатурах кнопкових телефонів. Вони можуть використовуватися в автоматичних телефонних системах, таких як автовідповідач.

–Кома (,) вставляє односекундну паузу між цифрами, що набираються. Кома може бути використана, наприклад, коли набирається 9 для виходу в телефонну мережу загального призначення через АТМУ – пауза дає АТМУ час для комутації з телефонною мережею загального призначення.

–Точка (.) – відповідає будь-якій цифрі. Використовується для завдання діапазону телефонних номерів.

–Квадратні дужки ([]) – позначають діапазон. Наприклад "20[0-4]." відповідає діапазону номерів від 2000 до 2004.

–T. Опціонний контрольний символ, що позначає, що значення шаблона є рядком змінної довжини. Маршрутизатор накопичує набираються цифри, що, доти, поки інтервал між ними не перевищить зконфігурованого значення (яке за замовчуванням становить 10 секунд). Після закінчення набору, щоб не чекати, поки мине таймаут, можна набрати ґрати й тоді маршрутизатор почнуть обробляти запит негайно.

Розробка структурної схеми

Систему IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco розглянемо на прикладі бездротової мережі передачі даних і голосу, побудованої з використанням мостів радіо-Ethernet Cisco AIR-BRI342 і системи IP Telephony AVVID.

На рисунку 8 зображена структурна схема розроблювальної системи. Дана радіомережа поєднує центральний офіс компанії з її філіями. Для виключення конфліктів голосового трафіку з локальним трафіком даних у ЛОМ необхідно розділити пристрою IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco й інші мережні пристрої по різним коллізійним сегментах шляхом організації VLAN за допомогою Ethernet-

Комутаторів. Ця міра особливо актуальна, якщо у ЛОМ працюють напівдуплексні пристрої Ethernet. Що стосується властиво радіомережі, те всі радіомосты по визначенню перебувають в одному коллізійном сегменті.

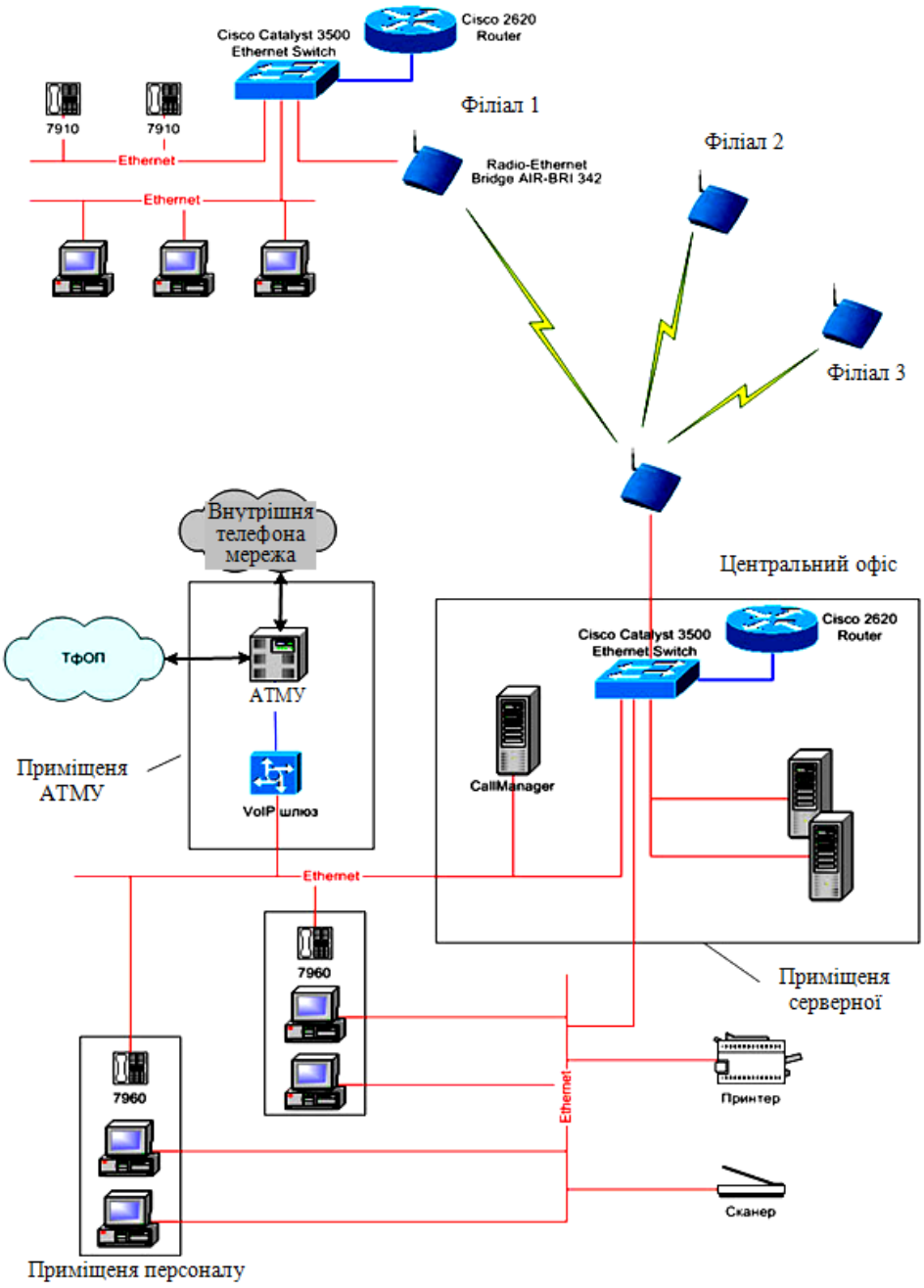


Рисунок 8 – Структурна схема системи

Відомо що, при передачі голосу по VoIP у радіомережах множинного доступу з використанням коллізійних протоколів MAC-рівня типу CSMA/CA, основна проблема, що приводить до тимчасових затримок, ця відсутність гарантованої пропускної здатності каналу для даного з'єднання. Протокол CSMA/CA припускає випадкову установку з'єднання й, як наслідок, асинхронний механізм передачі радіокадрів, що зовсім неприйнятно для передачі голосового трафіку, критичного до тимчасових затримок. Випадковий характер установки з'єднань приводить до також випадкового значення пропускної здатності каналу для даного з'єднання. З ростом числа абонентів мережі ймовірність появи колізій зростає й, відповідно, зростають коливання швидкості передачі даних щодо середньостатистичного значення. Ці коливання можуть становити десятки відсотків, у результаті чого виникають непередбачені затримки трафіку. Як показали

численні експерименти по передачі VoIP у середовищі радіо-Ethernet, затримка трафіку тут значно перевищує суму інших складових загальної затримки, характерних для технології VoIP: затримки кодека, затримки в чергах, затримки в буфері, що згладжує. Коли загальна затримка в каналі, по якому встановлений сеанс голосового зв'язку, перевищить 150 мс – суб'єктивну межу слухового сприйняття затримки, відбувається катастрофічна втрата якості мови.

Типовий метод рішення проблеми, наприклад, засобами математичного забезпечення Cisco Systems містить у собі:

–усунення перевантажень у мережі шляхом профілювання трафіку за допомогою GTS (Generic Traffic Shaping);

–пріоритезація трафіку на основі інструментів класифікації PBR, BGP, CAR;

–забезпечення необхідного рівня сервісу в мережі за допомогою сигналізації QoS, що використовує протокол маркування кадрів Ethernet відповідно до рекомендації 802.1p/Q або протокол RSVP.

Для реалізації перерахованих функцій у кожній точці мережі необхідно встановити прикордонний маршрутизатор, як мінімум, серії 2600. Однак, слід зазначити принаймні два недоліки наведеного методу. Перший полягає в тому, що механізм GTS обмежує швидкість вихідного потоку даних, тобто, виконує свою функцію тільки на ділянці маршрутизатор – радіомост у даній точці мережі. Але, оскільки в радіосегмент дані однаково надходять несинхронно, імовірність нерівномірного завантаження мережі не вдається звести до нуля.

Другий недолік полягає в тім, що математичне забезпечення, що дозволяє реалізувати маркування кадрів Ethernet, не входить у базову комплектацію маршрутизатора. Для цього необхідно доповнити його розширеною версією програмного забезпечення IP Plus, а також збільшити об'єм flash-пам'яті до 16 Мбайт і RAM – до 48 Мбайт на загальну суму 2400 USD. Якщо взяти до уваги, що вартість маршрутизатора Cisco 2610 у базовій комплектації становить 1995 USD, а радіомоста Cisco AIR-BRI342 – 1949 USD, те одне тільки забезпечення QoS у мережі радіо-Ethernet підвищує витрати на 60%.

Таким чином, у бездротовій мережі передачі даних і голосу на базі радіо-Ethernet основна робота із забезпечення якості мови виконується маршрутизатором. Нагадаємо, що все вищесказане відноситься до переважної більшості мостів радіо-Ethernet, що використовує протоколи множинного доступу типу CSMA/CA. Однак, у цей час з'являється встаткування цього типу, що використовують неколлізійні канальні протоколи.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів інформаційно-комунікаційної платформи на базі рішень Cisco. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем інформаційно-комунікаційної платформи на базі рішень Cisco; Досліджена система інформаційно-комунікаційної платформи на базі рішень Cisco; На основі отриманих результатів досліджень створена програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco; Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інформаційно-комунікаційної платформи на базі рішень Cisco. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022, pp. 1-12. **(Scopus).**
2. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland)* Volume 22, Issue 16, 6223, 2022. (**Scopus**).
3. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». *In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. **Springer**, Singapore. pp. 21-34. **(Scopus).**
4. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. **Springer**, Cham. 2022, pp. 2463-2477. **(Scopus).**
5. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». *SN Computer Science*, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w **(Scopus).**
6. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». *2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*, 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 **(Scopus).**
7. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». *CEUR Workshop Proceedings* Volume 3101, 2021, Pages 192-207. **(Scopus).**
8. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings* Volume 2805, 2020, Pages *44-58.* **(Scopus)**.
9. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». *International Journal of Computing*; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. **(Scopus)**.
10. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings.* Volume 2740, 2020, Pages 102-114. **(Scopus)**.
11. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346. **(Scopus)**.
12. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131. **(Scopus)**.
13. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у *Кібербезпека та інформаційні технології: монографія.* – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
14. Смірнов О.А., Дрєєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.
15. Смирнов А.А., Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений. Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.
16. Смирнов А.А., Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения. Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.
17. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95
18. Смірнов, О.А., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю.Усік П.С., «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». *Проблеми телекомунікацій*. № 1(26). С. 83-96. 2020.
19. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки.* №4. С. 103-110. 2020.
20. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.

УДК 004

В.Глобенко, магістр гр КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ LAN МЕРЕЖ ІНФОРМАЦІЙНИХ ТА КОМП'ЮТЕРНИХ СИСТЕМ

У статті розроблено програмне забезпечення, яке призначено для системи моніторингу LAN мереж інформаційних та комп'ютерних систем. Метою розробки є дослідження та програмна реалізація системи моніторингу LAN мереж інформаційних та комп'ютерних систем. Об'єктом дослідження є процес моніторингу LAN мереж інформаційних та комп'ютерних систем. Предметом дослідження є методи моніторингу LAN мереж інформаційних та комп'ютерних систем. Методи дослідження базуються на методах побудови мереж інформаційних та комп'ютерних систем, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи моніторингу LAN мереж інформаційних та комп'ютерних систем. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, моніторинг, LAN, мережі інформаційних та комп'ютерних систем

Постановка проблеми. Розвиток засобів обчислювальної техніки відбувається в багатьох напрямках, що розширюють сферу застосування ЕОМ і підвищують ефективність їхнього використання. Найбільше застосування ЕОМ відбувається при побудові різного виду комп'ютерних мереж, як локальних або корпоративних, так і глобальних. Сучасний світ неможливо представити без використання комп'ютерних мереж, які у тому або іншому вигляді застосовуються усюди, починаючи від елементарних хатніх комп'ютерних мереж і закінчуючи промисловими мережами, глобальними мережами, банківськими мережами, мережами для проведення наукових досліджень.

Актуальність теми магістерського дослідження обумовлена необхідністю підвищення ймовірності достовірного надання даних, які передаються по комп'ютерним мережам. Ці задачі є одними із ключових при розробці сучасних систем моніторингу локальних мереж.

Терміном **моніторинг мережі** називають роботу системи, що виконує постійне спостереження за комп'ютерною мережею в пошуках повільних або несправних систем і яка при виявленні збоїв повідомляє про їх мережному адміністраторові за допомогою пошти, телефону або інших засобів оповіщення. Ці задачі є підмножиною задач керування мережею.

У той час, як система виявлення вторгнень стежить за появою погроз ззовні, система моніторингу мережі виконує спостереження за мережею в пошуках проблем, викликаних перевантаженими й/або серверами, що відмовили, іншими пристроями або мережними з'єднаннями.

Наприклад, для того, щоб визначити стан веб-серверу, програма, що виконує моніторинг, може періодично відправляти запит HTTP на одержання сторінки; для поштових серверів можна відправити тестове повідомлення по SMTP і одержати по IMAP або POP3.

Невдалі запити (наприклад, у тому випадку, коли з'єднання не може бути встановлено, воно завершується по таймауту, або коли повідомлення не було доставлено) звичайно викликають реакцію з боку системи моніторингу.

Як реакція може бути:

–відправлено сигнал тривоги системному адміністраторові;

–автоматично активована система захисту від збоїв, що тимчасово виведе проблемний сервер з експлуатації, доти, поки проблема не буде вирішена, і так далі.

Проведений критичний аналіз існуючих методів обробки даних, дозволив структурувати область застосування таких технологій і виявити ряд обмежень. У зв'язку із цим виникла необхідність у розробці перспективного, однак ще недостатньо дослідженого теоретично й апробованого на практиці, динамічного багатопоточного методу обробки даних і програмного забезпечення для реалізації цього підходу.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи моніторингу LAN мереж інформаційних та комп'ютерних систем.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи моніторингу LAN мереж інформаційних та комп'ютерних систем.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем моніторингу LAN мереж інформаційних та комп'ютерних систем.

– Дослідження системи моніторингу LAN мереж інформаційних та комп'ютерних систем.

– Програмна реалізація системи моніторингу LAN мереж інформаційних та комп'ютерних систем.

Об'єктом дослідження є процес моніторингу LAN мереж інформаційних та комп'ютерних систем.

Предметом дослідження є методи моніторингу LAN мереж інформаційних та комп'ютерних систем.

Методи дослідження базуються на методах побудови мереж інформаційних та комп'ютерних систем, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Розробка системи моніторингу мережі з використанням стандартних інструментів

Джерела телеметричних даних

Головне джерело телеметричної інформації про сервери Windows – журнал подій Security, а найважливіше джерело виробничої телеметрії – журнали подій System і Application. Досвідчені користувачі оснащення Event Viewer консолі Microsoft Management Console (MMC) знають, що у всіх журналах подій Windows застосовується один формат файлів (.evt). Запис про кожну подію містить стандартні поля (наприклад, дата, час, джерело події, категорія, ID події), за якими знаходиться поле опису з даними у вільній формі, унікальними для конкретної події. Будь-який додаток моніторингу, сумісний з журналами подій Windows, дозволяє генерувати попередження й звіти на основі джерела, категорії або ID події, але в ідеальному випадку потрібно мати можливість відфільтровувати записи за даними в описі події.

Мережні пристрої, такі як маршрутизатори, комутатори, бездротові AP і брандмауери, незмінно передають телеметричні дані через протокол SNMP або Syslog. SNMP був спроектований наприкінці 1980-х для керування множиною пристроїв у мережі Internet, що бурхливо розвивається,. Диспетчери SNMP збирають телеметричну інформацію від агентів через UDP-порт 162. Диспетчери можуть використовувати SNMP-команди Get для запиту конкретних телеметричних даних, називаних змінними (variable), або пасивно чекати звіту про важливі події від агентів через повідомлення Trap. Для моніторингу в сферах виробництва й безпеки досить збирати повідомлення Trap. Для поглибленого аналізу тенденцій і планування ресурсів варто опитувати агентів за допомогою команд Get.

Syslog

Syslog – стандарт протоколювання подій для UNIX. Перевага Syslog перед механізмом протоколювання подій Windows полягає в тому, що весь процес консолідації

потоків подій від численних систем – невід'ємна частина Syslog. У дійсності Syslog одночасно мережний протокол і формат журналу, і за замовчуванням він використовує UDP-порт 514. Кожне повідомлення Syslog містить поля дати, часу, пріоритету, ім'я хосту й тексту повідомлення. З технічної точки зору пріоритет – число від 0 до 191. Однак більшість додатків Syslog відображають пріоритет у вигляді двох складових: Facility і Level.

Facility. Спочатку Syslog проектувався для моніторингу BSD Unix, і величина Facility використовувалася для ідентифікації процесу Unix про який свідчить подія. Значення від 0 до 15 відповідають найважливішим процесам Unix, а значення від 16 до 23 (з іменами від Local0 до Local7) призначені для додатків і пристроїв. Більшість мережних пристроїв використовують значення від Local0 до Local7 (наприклад, пристрої Cisco задіють Local6 і Local7), але не всі. Маршрутизатор Xincom Twin Wan використовує майже всі низькі значення Facility.

Level. Інший елемент пріоритету повідомлень Syslog – Level, значення якого перебувають у межах від 0 до 7. Level характеризує ступінь важливості повідомлення.

Продуктивність і стан

Для повного функціонального моніторингу корисно контролювати об'єкти продуктивності (performance-object) і стан серверів з окремого комп'ютера або від провайдеру послуг. Адміністратори, не знайомі з об'єктами продуктивності, можуть досліджувати їх за допомогою оснащення Performance консолі MMC. Різниця між моніторингом журналу подій і об'єкта продуктивності наступна: з журналів подій можна витягти інформацію про будь-яку частину системи, у якій відбулися неполадки, а об'єкт продуктивності дозволяє переконатися, що конкретні параметри перебувають у припустимих межах. Наприклад, за допомогою об'єктів продуктивності можна стежити за простором жорсткого диска, так як системний журнал видає попередження, тільки коли тім заповнюється настільки, що користувач уже починає випробовувати незручності.

Ще одна типова перевірка із застосуванням об'єкта продуктивності – моніторинг коефіцієнта використання центрального процесора з відстеженням певних рівнів протягом тривалих періодів часу (наприклад, понад 90% протягом 10 хвилин). Однак при перевірках коефіцієнта використання центрального процесора варто проявляти обережність; неважко переплутати корисне навантаження з некерованим процесом і згенерувати помилкове повідомлення про проблему. Чудова властивість об'єктів продуктивності полягає в тому, що інші додатки можуть створювати власні об'єкти продуктивності й публікувати телеметричні дані, специфічні для даного додатка. Наприклад, Active Directory (AD), SQL Server і Exchange Server мають у своєму розпорядженні власні об'єкти продуктивності.

Відсутність повідомлень про помилки в журналі й показники продуктивності в межах припустимих порогів – гарні індикатори коректного функціонування. Однак деякі проблеми не знаходять відбиттів в індикаторах. Перевірки стану серверів – найефективніший спосіб переконатися в тому, що сервери й додатки працюють у мережі й успішно обробляють запити. Перевірки стану серверів надійні, так як вони виконують тестову транзакцію. Багато провайдерів додатків і служб в Internet дозволяють регулярно проводити тестові транзакції із сервером через обрані споживачем інтервали часу. Для Web-серверу можна періодично запитувати дану Web-сторінку й перевіряти, чи успішно вона передана. Для системи SQL Server можна періодично виконувати запит і перевіряти результати.

Однак навіть перевірки стану не розкривають всіх проблем. Наприклад, простий сигнал ping, переданий через кожні 5 мінут, дозволяє переконатися, що операційна система й набір протоколів активні, але не містять ніякої інформації про стан самого додатка. Мені доводилося зустрічати завислі сервери, які відповідали на сигнали ping. Аналогічно простий запит HTML-сторінки із сервера не доводить, що відповідний додаток електронної комерції на базі Active Server Pages (ASP) працює коректно.

Тому перевірки стану повинні бути як можна більш функціональними. Ще одне застереження: програму перевірки стану варто розміщати поза контрольованим виробничим середовищем. Якщо помилково розмістити програму перевірки стану на контрольованому

сервері, то, наприклад, не вдасться визначити відмову сервера або серверного з'єднання, так як додаток не зможе передати адміністраторові відповідне повідомлення. Але якщо додаток перевірки працює на окремому сервері (і якщо цей сервер доступний з Internet), то єдиний випадок, коли важливий додаток буде не готовий до роботи без відома адміністратора, – одночасна відмова виробничого й контролюючого середовища.

Необхідний інструментарій

Отже, що потрібно для моніторингу всіх пристроїв, серверів, журналів, пасток SNMP і подій Syslog? Очевидно, необхідні один-два інструмента за доступною ціною, що охоплюють всі елементи, які потрібно відслідковувати. Продукти моніторингу високого рівня, такі як Argent Guardian і Microsoft Operations Manager (MOM), дозволяють контролювати всі об'єкти продуктивності, журнали подій Windows, пастки SNMP, потоки подій Syslog і навіть виконувати різні перевірки стану. Деякі не настільки великі, менш дорогі пакети, такі як Sentry II компанії Engagent, EventTracker компанії Prism Microsystems і комплекс Event Log Management компанії Dorian, охоплюють підмножину телеметричних джерел і обмежений набір об'єктів продуктивності.

Збираючись придбати інструмент, варто скласти список всіх характеристик, які необхідно знати, і підібрати інструмент, що контролює їх всі. Якщо інструмент не забезпечує моніторинг важливого параметра, наприклад SNMP, то заповнити пробіл можна за допомогою безкоштовної або недорогої умовно безкоштовної утиліти. Далі буде розглянутий ряд таких інструментів, з яких можна скласти ефективний комплекс моніторингу.

Розглянемо три корисних інструменти, які можна додати до арсеналу мережного моніторингу: Log Parser, безкоштовний інструмент Microsoft; tail, відмінну утиліту з миру UNIX; утиліту Kiwi Syslog Daemon, що представлена безкоштовною й могутнішою, але проте недорогою версією. Інформація буде корисна навіть тим адміністраторам, які вже мають або мають намір придбати інструменти: більшість інструментів на ринку розташовують лише функціями попереджень і звітності, доповненими шаблоновими звітами й зразковими правилами розсилання попереджень. Методи проектування й аналізу, описані в розділі, будуть надзвичайно корисні навіть для власників розгорнутої на підприємстві програми моніторингу.

Моніторинг текстових журналів

Більшість серверних продуктів Microsoft і компонентів Windows протоколюють будь-які важливі події в журналах System або Application, але кожна служба або основний компонент операційної системи (наприклад, IIS, DHCP, SMTP, Internet Authentication Service, IAS) записують більш докладну інформацію у власний текстовий журнал у спеціальному форматі. Для моніторингу інформації, наявної тільки в цих текстових файлах, зручно використовувати Log Parser. Log Parser розпізнає будь-який формат текстового файлу з розмежувачами, наприклад із символами табуляції або комами (CSV), і дозволяє задіяти ту ж команду SQL Select для опитування текстових файлів журналів.

Таким чином, за допомогою Log Parser можна вирішити задачу підготовки звітів на основі текстових журналів. Але що робити із попередженнями про критичні події, що поступають у реальному часі, інформація про які зберігається в текстових файлах? Я рекомендую скористатися інструментом tail, запозиченим з UNIX. Tail відслідковує додавання нових рядків у зазначені користувачем текстові файли. Як тільки виявляються нові дані, tail посилає їх у стандартний вихідний потік (stdout). Вихідні дані tail можна направити в сценарій, що аналізує нові записи в міру їхнього протоколювання й при необхідності генерує попередження. Наприклад:

```
tail /f logfile.txt |LoopOnNewMessages.cmd
```

виявляє нові повідомлення, додані у файл logfile.txt, і направляє їх в LoopOnNewMessages.cmd. Loop OnNewMessages.cmd передає кожне повідомлення за адресою rsmith@ultimatewindowssecurity.com, але замість нього можна вказати будь-яку іншу поштову адресу. Щоб не пересилати не занадто істотні повідомлення, можна доповнити сценарій фільтруючою логікою.

Tail

Витягти інформацію з журнальних файлів або інших текстових файлів можна за допомогою широко розповсюдженої програми grep. Але є й інший корисний інструмент із миру UNIX, гідний зайняти місце в інструментальному наборі адміністратора Windows, – програма tail. По суті, tail показує останні кілька рядків текстового файлу – це особливо корисно при аналізі файлів журналів. Наприклад, якщо адміністратор становить нові правила для брандмауера, tail покаже, як правила відбиваються на файлі журналу.

Tail є в більшості систем UNIX, а версію Win32 можна завантажити в рамках прийняття умов ліцензії GNU з Web-вузла Sourceforge (*http://unxutils.sourceforge.net*). Спочатку потрібно завантажити файл UnxUpdates.zip, а потім витягти tail.exe у своєму комп'ютері.

Автономне використання Tail

При використанні поза комбінацією з іншими програмами, tail показує кілька останніх рядків текстового файлу. За допомогою декількох параметрів можна змінити подання інформації на екрані. Особливо корисний параметр follow (-f), що дозволяє безупинно відслідковувати й виводити на екран зміни в текстовому файлі. Наприклад, команда

tail -f ex050410.log

показує останні 10 рядків журнального файлу з ім'ям ex050410.log і буде відслідковувати й відображати нові записи в міру їхньої появи. Якщо файл являє собою журнал Web-служби Microsoft IIS і хто-небудь звертається до Web-вузла, IIS зробить у журналі новий запис. Нові додавання негайно відображаються на консолі, у якій працює tail. Цей параметр спрощує діагностику, дозволяючи негайно побачити нові записи.

Спільне застосування Tail і Grep

Як відомо, grep – програма, що веде пошук зазначених послідовностей символів у цільовому текстовому файлі. Наприклад, при діагностиці комп'ютера, що працює з Windows Firewall, потрібно відшукати в журналі брандмауера дії, зроблені в певний день. Журнал не розділений по датах і досить великий.

За допомогою grep можна витягти рядки даних від 7 березня 2009 року й записати їх у новий текстовий файл:

```
grep « 2009-03-07» p-firewall.log
> 030705 p-firewall.log
```

Як щодо tail? Команду можна використовувати для обробки журналів брандмауера в процесі діагностики або відстеження атак у реальному часі. Але можна застосувати tail разом з grep, щоб виводити на екран тільки певні дані.

Для початку варто настроїти брандмауер на запис журналів у текстовий файл. Всі системи UNIX використовують syslog для протоколювання подій; більшість комерційних брандмауерів також підтримують syslog. Якщо на системі UNIX використовуються grep і tail, то варто настроїти брандмауер на пересилання даних syslog у хост-машину syslog. Користувачі Windows можуть установити й працювати із сервером syslog на базі Windows. Я рекомендую Kiwi Syslog Daemon фірми Kiwi Enterprises, відмінний інструмент для збереження даних syslog у текстовому файлі.

Потім потрібно побудувати шаблон на основі синтаксису постачальника брандмауера. Наприклад, адміністратор використовує брандмауера Cisco PIX і хоче одержувати оповіщення щораз, коли хтось звертається до Web-служб через брандмауера. За допомогою tail і grep можна в реальному часі виявляти в журналах символи «/80» (представляють Web-трафік у журналі PIX), наприклад:

tail -f pix.log | grep «/80»

Більш вдалий підхід – використовувати метасимволи регулярних виражень, які забезпечують більше складну фільтрацію, чим звичайні текстові рядки:

tail -f pix.log | grep /80[[:space:]]

Освоєння регулярних виражень вимагає часу, але в нагороду ви одержуєте бібліотеку корисних і ефективних шаблонів, які можна використовувати для пошуку майже будь-яких даних, – безсумнівно, це виправдує витрачені зусилля.

Ускладнений Tail

Grep і tail – прості у використанні й дуже гнучкі програми. При роботі з консольними додатками обидва інструменти значно спрощують аналіз журнальних файлів і повсякденне адміністрування. Версія командного рядка tail – швидка й проста в експлуатації, і, імовірно, прихильники строгих правил нададуть їй перевагу завдяки простоті й можливості пересилати вихідні дані в інші програми, такі як grep. Але існують версії tail із графічним інтерфейсом Windows, причому деякі з них наділені більш складними функціями, наприклад кольоровим виділенням співпадаючих послідовностей. Таке форматування допомагає відзначати важливі файли.

Зразок безкоштовної графічної програми tail для Windows – BareTail компанії Bare Metal Software (її можна завантажити за адресою *http://www.baremetalsoft.com/baretail*). Як і tail, BareTail відображає текстовий файл і відслідковує доповнення до файлу, але оскільки BareTail працює із графічним інтерфейсом, вона має у своєму розпорядженні функції виділення.

Завдяки таким функціям простіше виявити певний текст (наприклад, конкретну IP-адресу або порт) «на ходу», спостерігаючи за журналом брандмауера. Можна також змінити шрифт, без праці скопіювати рядок тексту й відкрити недавно переглянуті файли журналів за допомогою списку недавно використаних файлів Windows.

Моніторинг SNMP і журналів Syslog

Контролювати телеметричні джерела SNMP і Syslog легко завдяки безкоштовній версії програми Kiwi Syslog Daemon компанії Kiwi Enterprises. Цей диспетчер серверів Windows Syslog і SNMP дозволяє зібрати всі телеметричні дані про мережні пристрої в одній програмі. Із графічного інтерфейсу програми можна настроїти фільтри для збору повідомлень, що відповідають певним критеріям, а потім указати одне або кілька дій, що вживаються у відповідь на повідомлення. Можна побудувати фільтри для видалення непотрібних повідомлень і вказати, що повідомлення, що залишилися, повинні генерувати попередження або зберігатися в базі даних для наступних звітів. За допомогою Kiwi Syslog Daemon можна фільтрувати повідомлення за часом дня, днем тижня, пристроям, рівню, IP-адресі звітного агента або рядкам у повідомленні. Крім того, інструмент може виконувати різноманітні дії – оповіщення по електронній пошті, збереження в базі даних по ODBC, запуск програми й інші – у відповідь на зазначені події.

Безкоштовна версія Kiwi Syslog Daemon інтерактивно працює в настільному комп'ютері, тому адміністратор повинен зареєструватися, щоб контролювати пристрої. Але розширена версія продукту функціонує як служба, а її вартість – усього 100 дол. для одного сервера. Якщо активізувати моніторинг пасток SNMP, необхідно також указати поля Facility і Level, які використовуються інструментом при перетворенні пастки в повідомлення Syslog. Наприклад, можна вказати пастки SNMP як Facility Local4 і Level 3 Error. Потім можна скласти правила розсилання попереджень спеціально для пасток SNMP шляхом фільтрації повідомлень Local4.

Отже, існують ресурси для моніторингу різноманітних джерел телеметричних даних. Перш ніж почати проектувати власне рішення для моніторингу, корисно познайомитися з інструментами, які є на ринку. Вони доступні й повнофункціональні. Однак не можна одержати повне рішення, просто здобуваючи інструмент. Потрібно визначити критерії для звітів і попереджень, щоб не одержувати занадто багато повідомлень про незначні події, але не слід упадати в іншу крайність і задавати настільки строгі критерії, що рішення моніторингу може перешкодити виконанню тої самої задачі, для якої воно призначалося. Для досягнення балансу варто становити критерії, відтинаючи незначні, а не вибираючи важливі події. Єдине виключення із цього правила – журнал Security, що набагато коротше, а крім того, краще документовано. Повна база даних подій журналу Security і їхніх значень

опублікована в Security Log Encyclopedia на сайті Ultimate Windows Security (*www.ultimatewindowssecurity.com*).

Для підготовки ефективної й вичерпної процедури моніторингу потрібно прикласти певні зусилля, але вони не пропадуть впусту. Немає нічого гірше, ніж довідатися про проблему від користувачів і після перегляду журналів виявити, що попередження надходили трьома днями раніше.

Вимоги бізнесу й законодавства щодо інформаційної безпеки й звітності дуже високі. Порушення безпеки можливі, але адміністратор і його компанія набагато успішніше переборють труднощі, якщо добре підготуються до критичної ситуації. Ефективному моніторингу немає повноцінної заміни.

Розробка удосконаленого методу моніторингу мережі

Як було відмічено вище, однією з основних проблем, що стоять зараз перед розроблювачами систем керування комп'ютерними мережами є проблема достовірного надання даних про їхній стан. Потреба в надійно працюючих великих комп'ютерних мережах усе вища з кожним днем. Тому при проектуванні сучасних систем керування дуже важливу роль відводять розробкам оптимізованих за часом алгоритмам збору й обробки даних.

Імовірність збереження актуальності інформації на момент її використання чисельно дорівнює:

$$P = \frac{c^2}{(c+b)(c+q)}, \quad (1)$$

де c – середній час значимої зміни реальної інформації щодо інформації, збереженої в БД;

b – середній час підготовки, передачі й уведення інформації для відновлення БД;

q – середній час між двома послідовними опитуваннями того самого пристрою.

Ідеальним випадком є ситуація, коли інформація про будь-яку зміну стану досягає БД у момент зміни. З використанням механізму багатопоточності можна «скоротити» час очікування інформації від джерела, використавши його для обробки інформації від іншого джерела.

На представленому нижче графіку (рисунок 1) видна залежність імовірності збереження актуальності даних від відношення часу очікування відповіді до часу обробки даних.

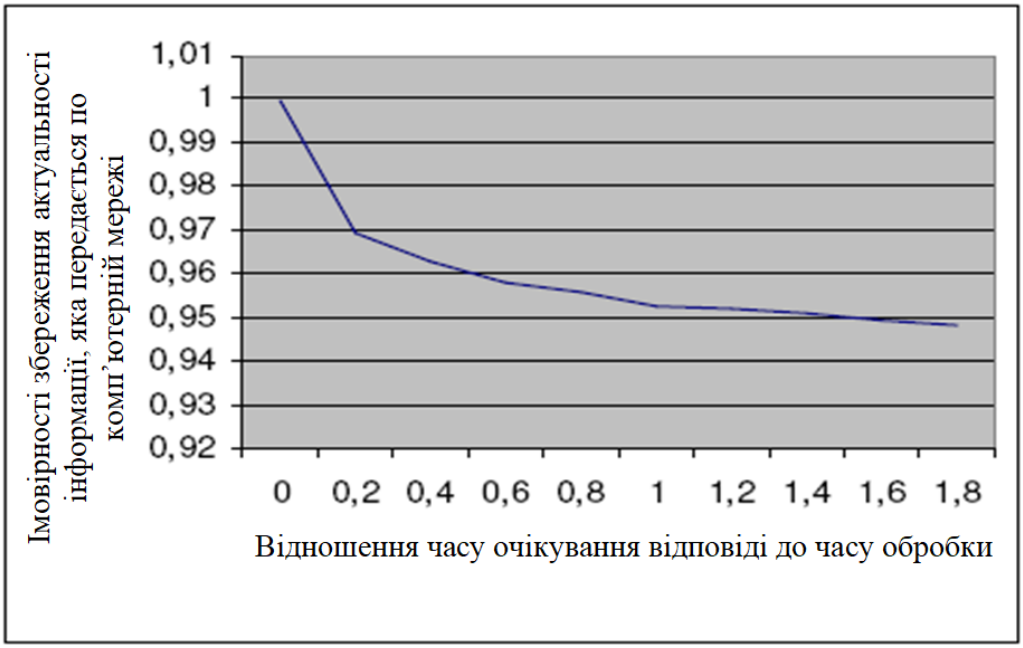


Рисунок 1 – Залежність імовірності збереження актуальності від відношення часу очікування відповіді до часу обробки

Із графіка видно, що навіть незначне скорочення часу очікування даних може привести до істотного збільшення ймовірності збереження актуальності даних, якщо час очікування приблизно дорівнює часу обробки. Можна зробити висновок, що для мереж, у яких при роботі системи моніторингу час обробки інформації приблизно дорівнює часу очікування після запиту, їсти можливість істотно підвищити актуальність інформації при використанні алгоритмів, що дозволяють використовувати час простою для обробки інформації.

Сучасні операційні системи дозволяють використовувати багатопоточну схему роботи додатків. Це досягається за рахунок розподілу робочого часу процесора між різними додатками, що дозволяє декільком програмам працювати «одночасно».

Весь процесорний час персонального комп'ютера (сервера) можна розділити на періоди. В один такий період всі додатки одержують по «шматочку» процесорного часу для своїх потреб. Їхні розміри залежать від пріоритетів додатків в операційній системі. Чим вище пріоритет – тим довший часовий інтервал, у якому додаток використовує процесор. У рамках даного інтервалу додаток також може й не використовувати процесор (перебуває в стані очікування). Чим більше потоків «всередині» у додатка, тим більше додаток в цілому одержує цього процесорного часу.

Для складних математичних задач введення механізму паралельного розрахунку даремно, тому що час роботи додатка приблизно дорівнює часу процесора для розрахунку задачі. Для задач моніторингу дана схема навпроти є досить вигідною. Вона дозволяє одержати виграш за часом за рахунок того, що в процесі опитування робочої станції є часові інтервали, у яких не використовуються ресурси процесора, пам'яті, мережі. Це інтервали очікування відповіді від віддаленого пристрою. Тим самим, якщо використовувати ці інтервали для роботи інших потоків, то можна буде зменшити час «простою» процесора. А за рахунок цього відбувається оптимізація за часом. Очевидно, що якщо змусити процесор працювати без «простою» у рамках відведеного для додатка процесорного часу й при цьому не допускати переповнення пам'яті або надлишкового мережного трафіку, така система буде максимально ефективна.

Як уже було визначено раніше, збільшення кількості паралельно опитуваних робочих станцій спочатку збільшує продуктивність системи, а після створення n+1 потоку, навпаки, сповільнюють її. Це пов'язане з закінченям обчислювальних ресурсів.

На продуктивність системи впливають три основних фактори: завантаженість центрального мікропроцесора, оперативний пам'яті, а також мережний трафік.

Очевидно, що поки всі три ресурси будуть використані не повністю, додавання нового потоку в систему буде збільшувати її продуктивність. Але як тільки один з ресурсів буде повністю вичерпаний, продуктивність системи або впаде, або втратить динаміку росту. Приміром, якщо при неповному завантаженні центрального процесора буде повністю зайнята доступна оперативна пам'ять, при додаванні нового потоку в систему, частину процесорного часу буде витрачатися на керування задачами по перевантаженню даних з оперативної пам'яті на жорсткий диск. Таким чином, одна з основних задач, яку необхідно вирішити при створенні систем моніторингу формулюється в такий спосіб: необхідно визначити максимальне число потоків, при якому система опитування робочих станцій буде працювати з максимальною ефективністю.

Щоб формально описати дану задачу необхідно визначити, як залежить продуктивність системи від значення вищеописаних факторів.

При запуску опитування мережі відбувається формування запиту і його відправлення віддаленому пристрою. Після цього потік переходить у стан очікування й до відповіді робітник станції майже не займає ресурсів процесора, пам'яті й не створює мережного трафіку. Природно, що в момент очікування відповіді одним потоком, використовувати ресурси процесора може інший потік.

Виходячи із усього вищесказаного й за умови необмежених ресурсів оперативної пам'яті й мережного трафіку, були визначені наступне співвідношення для визначення оптимальної кількості потоків, що додаються в систему моніторингу:

$$N = \frac{t_t}{t_p}, \tag{2}$$

де tt – тривалість потоку (від запиту даних до закінчення їхньої обробки);

tp – тривалість використання потоком ресурсів центрального процесора.

Дане співвідношення справедливо для ідеального випадку. У реальній ситуації процесор не може надати всі свої ресурси для системи моніторингу. Частина його ресурсів іде на керування операційною системою й іншими, а також використовується іншими додатками.

У зв'язку із цим протягом часу, за яке працює потік, до ресурсів процесора звертаються не тільки потоки системи моніторингу, але й інші додатки.

У такий спосіб вищенаведене співвідношення можна обмежити цією умовою:

$$N = \frac{t_t(1-P)}{t_p}, \tag{3}$$

де tt – тривалість потоку (від запиту даних до закінчення їхньої обробки);

tp – тривалість використання потоком ресурсів центрального процесора;

P – коефіцієнт завантаженості процесора іншими додатками (0 < P < 1).

Таке уточнення співвідношення дозволяє в будь-який момент часу визначити має сенс чи ні в цей момент часу додати додатковий потік у систему.

Але існують і інші фактори, що впливають на продуктивність системи моніторингу. Другим по значимості є завантаженість оперативної пам'яті. Усе раніше наведені міркування дійсні за умови, що оперативна пам'ять не повна, тобто система не використовує файл підкачування. У випадку ж його використання існують додаткові часові витрати за часом на перевантаження даних з файлу до пам’яті й обернено.

При цьому варто пам'ятати, що перевантаження даних відбувається тільки в тому випадку, коли потік готовий до виконання. А це відбувається не на кожному циклі ітерації в керуючому потоці.

Тому для випадку, коли всі потоки не будуть міститися в оперативній пам'яті, то попереднє співвідношення не може бути використана. У випадку недостачі оперативної пам'яті необхідно використовувати наступну формулу:

$$N = \frac{t_t(1-P)}{t_p + M \cdot t_s}, \tag{4}$$

де tt – тривалість потоку (від запиту даних до закінчення їхньої обробки);

tp – тривалість використання потоком ресурсів центрального процесора;

P – поточна завантаженість процесора (0 < P < 1);

ts – час перезавантаження даних з оперативної пам'яті у файл;

M – число таких перезавантажень.

Визначення значення коефіцієнта M не представляє особливої праці. Перевантаження відбудеться тільки тоді, коли потік перебуває в стані роботи, а не очікування. У зв'язку із цим значення M можна визначити за формулою:

$$M = \frac{t_p}{t_i}, \tag{5}$$

де tp – тривалість використання потоком ресурсів центрального процесора;

ti – час, на який надається доступ до ресурсу процесора потоку, при передачі йому керування в одній ітерації

Немаловажним фактором є завантаженість мережі. Очевидно, що при повному завантаженні мережі про паралельність також безглуздо говорити. Потоки будуть формуватися в послідовні черги, і при цьому мережа буде практично непрацездатна для інших додатків і користувачів. У двох попередніх випадках, перевантаження параметра вело лише до істотного вповільнення роботи одного комп'ютера, а для мережного трафіку може паралізувати роботу всієї мережі. Через це для кожної конкретної мережі встановлюється гранично припустимий мережний трафик, що може створювати система моніторингу. Його розрахунок ведеться з розрахунку розмірів мережі, її швидкості, кількості мережних додатків, часу доби й т.д.

Поєднуючи все вищесказане в єдину задачу одержуємо, що:

якщо оперативна пам'ять не переповнена:

$$N = \frac{t_t(1-P)}{t_p}, \tag{6}$$

−якщо оперативна пам'ять переповнена:

$$N = \frac{t_t(1-P)}{t_p + \frac{t_p}{t_i} \cdot t_s}, \tag{7}$$

−Обидва співвідношення обмежені умовою, що не перевищена установлена межа мережного трафіку, створюваного системою моніторингу.

−Виконання цієї умови, а також визначення переповнення оперативної пам'яті відбувається відповідно до раніше певних співвідношень.

−Всі параметри, необхідні для розрахунку оптимальної кількості потоків надаються операційною системою.

−Визначивши оптимальну кількість потоків у будь-який момент часу, була вирішена тільки половина поставленої задачі. У класичній багатопоточній схемі опитування мережі, нам потрібно розподілити весь діапазон IP-адрес на N груп, і провести опитування. Але класична схема не враховує того, що в сучасному житті під один додаток у мережі не виділяється сервер. Завантаженість сервера, на якому встановлена система моніторингу, постійно міняється через використання інших додатків, розташованих на ньому. А зі зміною завантаженості сервера міняється оптимальна кількість потоків, у рамках яких проходить опитування кінцевого або мережного устаткування. Друга проблема класичної схеми полягає в тому, що час опитування одного пристрою залежить від його типу, об'єму збирається информации, що, часу, необхідного на її обробку, а також швидкості каналу зв'язку. При «класичному» розподілі IP-адрес на групи всі ці моменти не враховуються, а це приводить до того, що в деякий момент часу одна група повністю оброблена, а інша ні.

−З обліком цих двох причин для рішення поставленої задачі дана модель була дороблена в такий спосіб:

−Визначаємо оптимальне число потоків у цей момент

−Запускаємо п потоків у яких відбувається опитування перших п адрес діапазону.

−Як тільки в якому-небудь потоці опитування закінчиться (або буде встановлена його неможливість) відбувається визначення оптимального числа потоків у цей момент.

−Якщо оптимальне число потоків менше поточного, то потік (у якому закінчилася робота) знищується. Якщо більше, то створюється ще к потоків (к = оптимальне число потоків – існуюче число потоків).

−Така схема дозволяє відслідковувати зміну стану системи в часі й дозволяє рівномірно розподіляти пристрою між потоками. На рисунку 2 представлений описаний вище алгоритм у графічній формі. Для того щоб оцінити запропонований вище алгоритм системи моніторингу, було проведено його моделювання, а також моделювання класичних

алгоритмів, у системі GPSS WORLD. Всі необхідні для такого моделювання параметри були визначені на реальній мережі. На їхній підставі були виведені наступні базові значення:

–Середнє число потоків для опитування мережі дорівнює 4.

–Середній час опитування 1 пристрою становить 80 секунд.

–Кількість перевантажень процесора або пам'яті серверів, на яких установлені засоби керування мережею рівнялося в середньому 6 разів за годину.

–На підставі цих даних було зроблене моделювання опитування великої локальної мережі (більше 250 комп'ютерів) для трьох реалізацій моніторингу (однопотоковий, багатопоточний і динамічний багатопоточний). За отриманим даними були побудовані графіки продуктивності різних реалізацій моніторингу (рисунок 3).

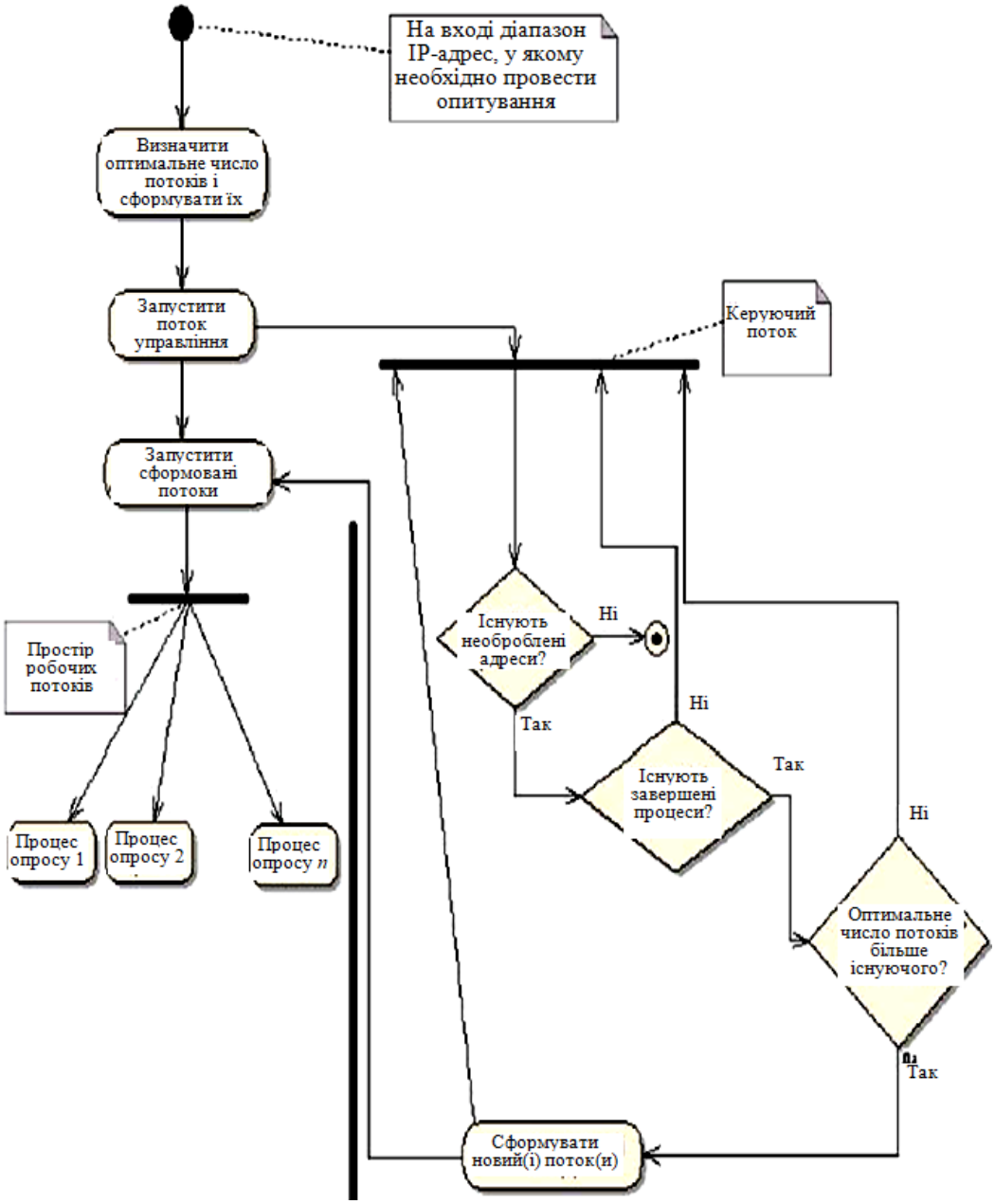


Рисунок 2 – Схематичне представлення удосконаленого алгоритму моніторингу мережі

Результати моделювання підтвердили раніше висунуті припущення. Модернізований багатопоточний алгоритм був реалізований у системі моніторингу. Після проведення тестових випробувань системи на мережі отримані дані підтвердили результати моделювання. Порівняння результатів отриманих при моделюванні й випробуванні системи

моніторингу, побудованої на модернізованому багатопоточному алгоритмі, представлені на рисунку 4.

На підставі аналізу наведених вище даних установлено, що час збору й обробки інформації для мережних або кінцевих пристроїв було скорочено більш ніж на 20%. Це означає, що час між повторними опитуваннями однієї й тієї ж робочої станції при круговому безперервному опитуванні змінилося, і стало становити 80% від того, котре забезпечує система моніторингу, побудована по класичній моделі. Таким чином, імовірність збереження актуальності даних для системи моніторингу, побудованої на алгоритмах розроблених у даній роботі для мереж збільшився з 92% до 95,5%.

Даний результат перевершує 95%, що у цей момент вважається мінімальним рівнем для вимог по актуальності даних.

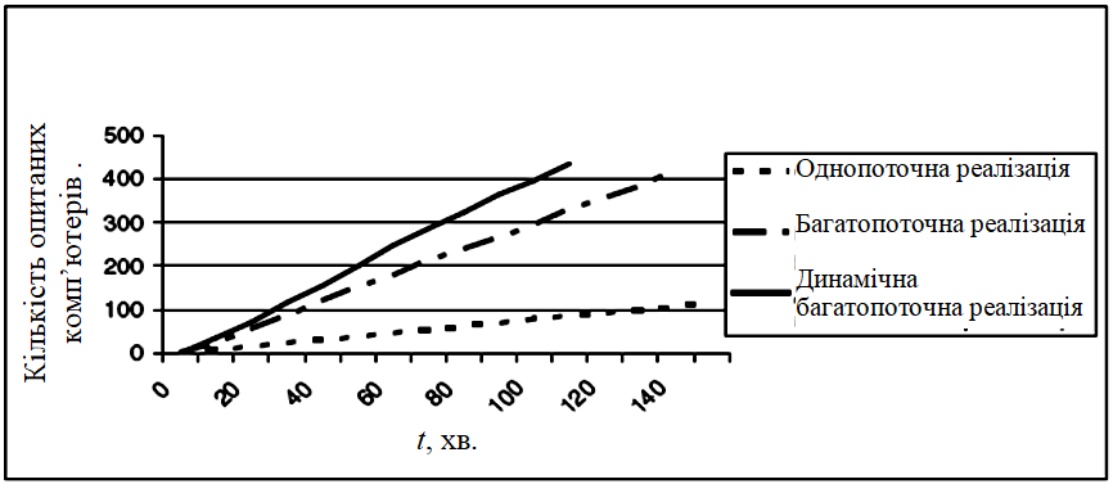


Рисунок 3 – Продуктивність різних реалізацій моніторингу при моделюванні

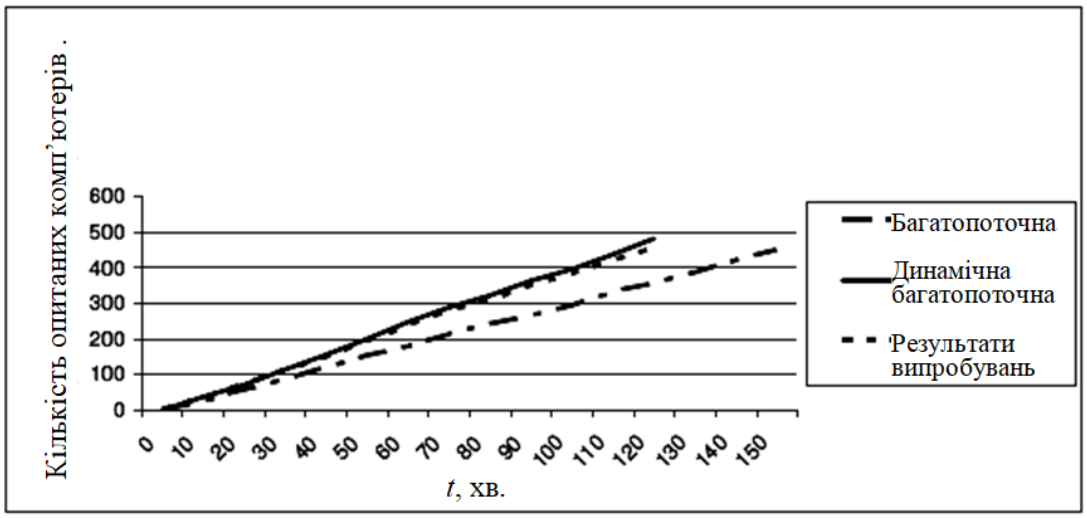


Рисунок 4 – Продуктивність різних реалізацій моніторингу при реальному випробуванні

Розробка структурної схеми

Структурна схема системи зображена на рисунку 5. З рисунку видно, що моніторинг локальної мережі здійснюється по трьох напрямках: Моніторинг обладнання; Моніторинг ресурсів; Моніторинг трафіку.

Моніторинг обладнання включає в себе побудову списку наявного обладнання та здійснення його контролю. До мережного обладнання, що підлягає моніторингу, відносяться: персональні комп'ютери, ноутбуки, сервери, принтери, ip-телефони.

Моніторинг ресурсів дозволяє переглядати та завантажувати наявні в мережі ресурси, а також розміщювати чи приховувати для загального доступу свої ресурси. До ресурсів

локальної мережі відносяться: файли, мультимедіа, бази даних, сервіси інформаційної безпеки, список користувачів.

Моніторинг трафіку використовується для контролю вхідного та вихідноготрафіку. Він включає у себе контроль підключених інтерфейсів, статистику подій по основним мережним протоколам: TCP, UDP, IP та ICMP.

TCP – один з основних мережних протоколів Інтернету, призначений для управління передачею даних в мережах і підмережах TCP/IP.

UDP – один із протоколів в стеку TCP/IP. Від протоколу TCP він відрізняється тим, що працює без встановлення з'єднання. UDP – це один з найпростіших протоколів транспортного рівня моделі OSI, котрий виконує обмін дейтаграмами без підтвердження та гарантії доставки.

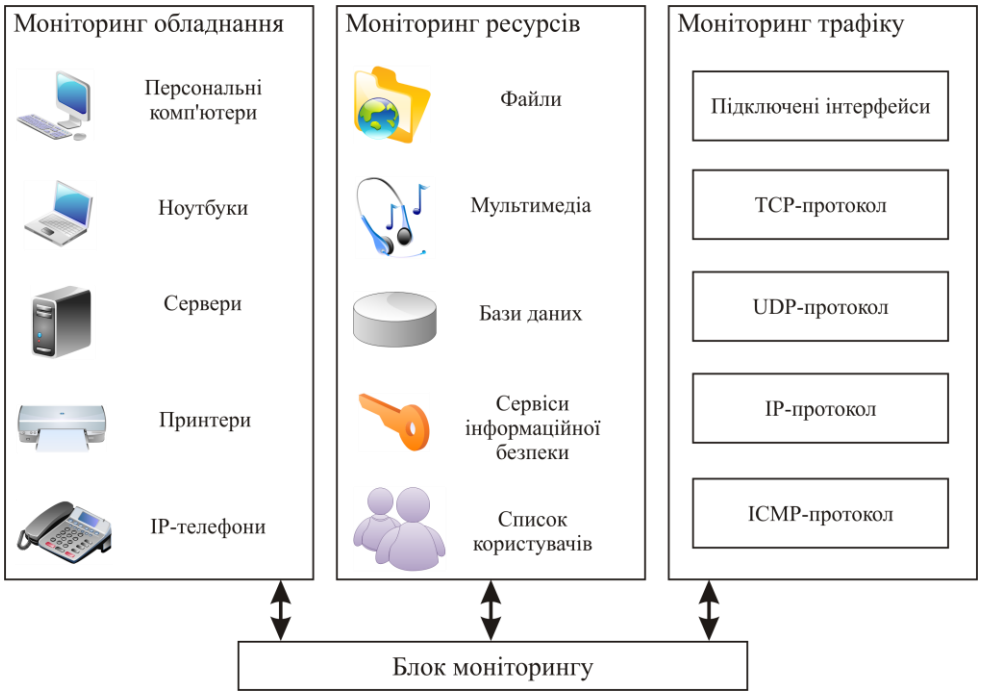


Рисунок 5 – Структурна схема системи

IP – найбільш широко розповсюджена реалізація ієрархічної схеми мережної адресації. Використовуваний в мережі Інтернет, протокол відповідає за адресацію пакетів, але не відповідає за встановлення з'єднань, не є надійним і дозволяє реалізувати тільки негарантовану доставку даних.

ICMP – мережний протокол, що входить в стек протоколів TCP/IP. В основному ICMP використовується для передачі повідомлень про помилки й інші виняткові ситуації, що виникли при передачі даних. Також на ICMP покладають деякі сервісні функції, зокрема на основі цього протоколу заснована дія таких загальновідомих утиліт як ping та traceroute.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів моніторингу LAN мереж інформаційних та комп’ютерних систем. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем моніторингу LAN мереж інформаційних та комп’ютерних систем; Досліджена система моніторингу LAN мереж інформаційних та комп’ютерних систем; На основі отриманих результатів досліджень створена програмна реалізація системи моніторингу LAN мереж інформаційних та комп’ютерних систем; Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання моніторингу LAN мереж інформаційних та комп’ютерних систем. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної

діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
2. Смірнов О.А., Дрєєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.
3. Смирнов А.А., Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений. Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.
4. Смирнов А.А., Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения. Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.
5. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95
6. Смірнов, О.А., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю.Усік П.С., «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.
7. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.
8. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.
9. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12. (Scopus).
10. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022. (Scopus).
11. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34. (Scopus).
12. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477. (Scopus).
13. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w (Scopus).
14. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
15. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).
16. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
17. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
18. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
19. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
20. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).

УДК 004

Я.Іщак, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ СКЛАДСЬКОГО АУДИТУ АВТОТРАНСПОРТНОГО ПІДПРИЄМСТВА

У статті розроблено програмне забезпечення, яке призначено для системи складського аудиту автотранспортного підприємства. Метою розробки є дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства. Об'єктом дослідження є процес складського аудиту автотранспортного підприємства. Предметом дослідження є методи складського аудиту автотранспортного підприємства. Методи дослідження базуються на методах теорії аудиту, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи складського аудиту автотранспортного підприємства. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, складський аудит

Постановка проблеми. Дана робота присвячена реалізації системи, яка складається з послідовності кроків, які використовуються для аудиту запасів запасних частин, включаючи перевірку закупівлі, зберігання та розподілу запасних частин.

Етапи роботи над проектом включають:

– проведення планування проекту;

– визначення обсягу та запити/координацію аудиторів;

– розробляти робочу програму та інші шаблони звітності;

– координувати зустрічі з ключовим персоналом процесу;

– перегляд відомих найкращих практик для початкового порівняльного аналізу у фокусній сфері;

– розуміти та документувати процес;

– управління запасами;

– отримати та переглянути письмові та «настільні» політики та процедури та організаційні схеми для кожної сфери;

– тощо…

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи складського аудиту автотранспортного підприємства.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи складського аудиту автотранспортного підприємства.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем складського аудиту автотранспортного підприємства.

– Дослідження системи складського аудиту автотранспортного підприємства.

– Програмна реалізація системи складського аудиту автотранспортного підприємства.

Об'єктом дослідження є процес складського аудиту автотранспортного підприємства.

Предметом дослідження є методи складського аудиту автотранспортного підприємства.

Методи дослідження базуються на методах теорії аудиту, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу

Керувати складом – завдання не з легких. З огляду на численні рухомі частини, від управління запасами до безпеки персоналу, для керівників складів надзвичайно важливо проводити регулярні аудити. Аудит складу дозволяє визначити сфери, які потрібно вдосконалити, оцінити, що працює добре, і вирішити будь-які проблеми з запасами чи безпекою.

У цьому вичерпному посібнику ми дослідимо важливість аудиту складу та надамо вам покроковий підхід до проведення ефективного аудиту. Незалежно від того, чи є ви досвідченим менеджером складу чи новачком, цей посібник озброїть вас знаннями та інструментами, необхідними для оптимізації ваших складських операцій.

Розуміння складських аудитів

Аудити складів передбачають систематичні перевірки операцій і об’єктів для підвищення ефективності, відповідності та безпеки, що призводить до оптимізації процесів, зниження витрат і безпечнішого робочого середовища.

Що таке аудит складу?

Аудит складу – це систематична перевірка роботи складу та обладнання для оцінки продуктивності, ефективності та безпеки. Він передбачає оцінку різних аспектів складу, включаючи управління запасами, протоколи безпеки, процеси отримання та відвантаження, зберігання тощо. Проводячи регулярні аудити, керівники складів можуть визначати сфери, які потребують покращення, впроваджувати коригувальні дії та підвищувати загальну ефективність роботи.

Переваги складських аудитів

Аудити складів пропонують підприємствам кілька переваг. По-перше, вони допомагають виявити неефективність і вузькі місця в складських операціях, дозволяючи впорядкувати та оптимізувати. По-друге, перевірки забезпечують дотримання правил безпеки, знижуючи ризик нещасних випадків і травм. Крім того, перевірки допомагають підтримувати точність інвентаризації, мінімізувати скорочення та підвищити задоволеність клієнтів, забезпечуючи своєчасне та точне виконання замовлень. Зрештою, проведення регулярних аудитів складу призводить до підвищення ефективності, зниження витрат і безпечнішого робочого середовища.

Види складських аудитів

Аудити складів можуть охоплювати різні сфери уваги. Деякі поширені типи аудитів включають:

1. **Аудити запасів:** ці аудити спрямовані на забезпечення точного підрахунку запасів, виявлення розбіжностей і вирішення проблем, пов’язаних з управлінням запасами.
2. **Аудити безпеки:** аудити безпеки оцінюють дотримання протоколів безпеки, визначають потенційні небезпеки та пропонують заходи для підвищення безпеки на робочому місці.
3. **Операційні аудити:** Операційні аудити оцінюють ефективність складських процесів, таких як отримання, комплектування, пакування та відвантаження, щоб визначити сфери, які потрібно вдосконалити.
4. **Аудит об’єкта:** Аудит об’єкта зосереджується на фізичних аспектах складу, включаючи планування, системи зберігання, стан обладнання та загальну організацію. Важливо пристосувати ваші аудити до конкретних потреб і завдань вашого складу.

Проведення аудиту складу

Проведення складського аудиту передбачає визначення цілей, планування аудиту, виконання комплексної оцінки безпеки, запасів, операцій, об’єктів і документації, а також виконання рекомендацій щодо постійного вдосконалення.

Крок 1: Визначте цілі аудиту

Перш ніж приступати до аудиту складу, важливо визначити чіткі цілі. Визначте, чого ви сподіваєтеся досягти за допомогою аудиту. Наприклад, ваші цілі можуть включати виявлення розбіжностей в інвентаризації, покращення протоколів безпеки, оптимізацію операційних процесів або покращення планування та організації об'єкта. Визначення ваших цілей надає дорожню карту для аудиту та забезпечує цілеспрямований підхід.

Крок 2: Плануйте аудит

Правильне планування має вирішальне значення для успішного аудиту складу. Зверніть увагу на такі фактори:

1. **Частота аудиту:** визначте, як часто ви будете проводити аудити. Частота може змінюватися залежно від розміру вашого складу та складності операцій. Щомісячні або щоквартальні аудити є звичайними, але ви також можете вибрати проведення аудитів на щорічній основі.

2. **Аудиторська група:** Зберіть групу осіб, які відповідатимуть за проведення аудиту. Це може включати керівників складів, наглядачів, офіцерів безпеки та інший відповідний персонал.

3. **Контрольний список аудиту:** розробіть повний контрольний список, який охоплює всі сфери, які ви збираєтеся оцінити під час аудиту. Контрольний список повинен узгоджуватися з вашими визначеними цілями та служити керівництвом для аудиторської групи.

4. **Розподіл ресурсів:** виділіть достатньо часу, персоналу та ресурсів для забезпечення ретельного та ефективного аудиту. Розгляньте будь-які додаткові інструменти чи обладнання, які можуть знадобитися для збору чи аналізу даних.

Крок 3: Виконайте аудит

Після того, як ви завершили етап планування, настав час виконати аудит. Виконайте наведені нижче дії, щоб забезпечити комплексну оцінку:

1. **Оцінка безпеки:** почніть з оцінки протоколів і процедур безпеки. Перевірте відповідність нормам, перевірте запасні виходи та вогнегасники, оцініть стан обладнання безпеки та визначте будь-які потенційні небезпеки чи ризики.

2. **Управління запасами:** перевірте процеси контролю запасів, включаючи методи підрахунку запасів, точність записаних даних і дотримання найкращих практик управління запасами. Проведіть фізичні підрахунки та порівняйте їх із зареєстрованою інвентаризацією, щоб виявити будь-які розбіжності.

3. **Операційні процеси:** оцініть ефективність ключових операційних процесів, таких як отримання, комплектування, пакування та відправлення. Шукайте можливості оптимізувати робочі процеси та усунути вузькі місця. Оцінити ефективність технологій і систем автоматизації на місці.

4. **Приміщення та обладнання:** Оцініть фізичне розташування складу, системи зберігання та стан обладнання. Переконайтеся, що складські приміщення організовані належним чином, оптимізуйте використання простору та визначте будь-які потреби в технічному обслуговуванні або ремонті обладнання.

5. **Документація та ведення записів:** переглядайте складську документацію, включаючи стандартні операційні процедури (SOP), посібники з техніки безпеки та записи про навчання. Переконайтеся, що документація є актуальною та легкодоступною для працівників.

6. **Аналіз даних:** аналізуйте дані, зібрані під час аудиту, щоб визначити тенденції, закономірності та області, які потребують покращення. Використовуйте ключові показники ефективності (KPI), щоб оцінити продуктивність, точність та інші відповідні показники.

Крок 4: Реалізація рекомендацій

Після завершення аудиту ви отримаєте список висновків і рекомендацій. Визначте пріоритетність визначених проблем на основі їх впливу та можливостей реалізації. Розробіть план дій для виконання кожної рекомендації, розподіливши обов'язки та встановивши

кінцеві терміни. Регулярно перевіряйте прогрес і контролюйте ефективність впроваджених рішень.

Підтримання відповідності складського аудиту

Підтримання відповідності складського аудиту передбачає проведення регулярних аудитів, впровадження запланованих аудитів, забезпечення навчання співробітників, моніторинг ефективності та сприяння відкритому спілкуванню для постійного вдосконалення та забезпечення оптимізованого та безпечного складського середовища.

Регулярні перевірки та постійне вдосконалення

Аудити складів не слід розглядати як одноразові заходи. Щоб підтримувати оптимальну операційну ефективність і безпеку, важливо проводити регулярні аудити та постійно прагнути до вдосконалення. Розгляньте застосування таких практик:

1. **Заплановані аудити:** встановіть графік регулярних аудитів, щоб забезпечити постійну оцінку та вдосконалення роботи складу.
2. **Навчання співробітників:** Проводьте регулярні тренінги для співробітників щодо протоколів безпеки, управління запасами та операційних процесів. Це допоможе підтримувати відповідність і покращити загальну продуктивність.
3. **Моніторинг продуктивності:** запровадьте механізми відстеження продуктивності для моніторингу ключових показників і виявлення будь-яких відхилень або областей, які потребують уваги.
4. **Зворотній зв'язок і комунікація:** заохочуйте відкриті канали зв'язку з персоналом складу для збору відгуків, вирішення проблем і визначення потенційних областей для покращення.

Застосовуючи проактивний підхід до аудиту складу та впроваджуючи стратегії постійного вдосконалення, ви можете забезпечити добре оптимізоване та безпечне складське середовище.

Аудит складів необхідний для підтримки ефективної роботи та забезпечення безпеки працівників. Проводячи регулярні перевірки, визначаючи чіткі цілі та впроваджуючи необхідні вдосконалення, ви можете підвищити продуктивність, зменшити витрати та створити безпечніше робоче середовище. Не забувайте пристосовувати ваші аудити до конкретних потреб вашого складу та віддавати пріоритет постійному вдосконаленню.

Завдяки добре спланованому та виконаному процесу аудиту ви матимете все необхідне для оптимізації свого складу та досягнення довгострокового успіху.

Нижче наведемо перелік політик та процедур, які використовуються при управлінні запасами та матеріалами системи складського аудиту автотранспортного підприємства.

Робоча програма аудиту запасів

Зразки, включені в цей документ, допоможуть вам розробити план оцінки процесів, необхідних для закупівлі, зберігання та управління запасами, а також для оцінки ефективності та ефективності поточних процесів управління запасами.

Приклади кроків у цих робочих програмах включають перегляд останнього попереднього аудиторського звіту та відповідних прес-релізів, отримання заповненої анкети внутрішнього контролю інвентаризації від диспетчера заводу, документування будь-яких потенційних слабких місць контролю або незвичайної практики та розслідування після прибуття, отримання та документування пояснень від керівника підприємства. Контролер для всіх відкритих робочих нарядів старше двох місяців, а також забезпечення наявності програми перехресного навчання для всього складського персоналу.

Основні рекомендації та процедури інвентаризації

У цій Політиці щодо запасів викладаються керівні принципи та облікова політика, щоб забезпечити належний контроль і оцінку запасів, а також запобігання втратам або нестачам.

Згідно з цією політикою витрати на запаси визначаються за методом повного поглинання, який включає матеріальні, прямі витрати на оплату праці, змінні та постійні непрямі виробничі (накладні) витрати. Стандартні витрати оновлюються відповідно до вимог

ринку та щорічно переглядаються на обґрунтованість. Закупівельна ціна відрізняється від стандартної собівартості, а фактичні витрати реєструються на рахунку відхилень, який використовується для визначення відповідних стандартних витрат.

Найпопулярніші процедури автоматизованого керування ІТ

Визначте процеси тестування внутрішнього контролю вашої компанії та частоту тестування її автоматизованих засобів контролю.

Відповідно до цієї політики ІТ-відділ відповідає за підтримку цієї процедури, а відділ контролю документації – за підтримку її конфігурації; принаймні ця процедура переглядатиметься ІТ-відділом щорічно або за потреби для вдосконалення та змін процесу; ІТ-персонал і бухгалтерія відповідають за оновлення матриці автоматизованих елементів керування System X у міру виконання запитів на зміни; ІТ-спеціалісти та бухгалтерії відповідають за тестування та документування всіх автоматизованих засобів контролю раз на рік.

Найкращі методи обробки замовлень

Ця політика визначає процедури, яких має дотримуватися команда обслуговування клієнтів під час обробки замовлень. Він підсумовує підготовку документів, документообіг, а також обов'язки осіб і відділів під час обробки замовлень на продаж.

У цьому зразку всі замовлення клієнтів оброблятимуться ефективним і організованим способом, щоб забезпечити точні та оперативні відправлення. Після отримання замовлення клієнта (ЗП) команда обслуговування клієнтів несе відповідальність за те, щоб переконатися, що ЗП надійшло від дійсного клієнта. Якщо клієнта немає в основній формі клієнта, CS має зв'язатися з торговим представником, щоб отримати нову форму додавання клієнта.

Аудит від купівлі до платежу: покращення дотримання фінансових вимог

Оцініть середовище внутрішнього контролю вашої організації та визначте можливості внутрішнього контролю та вдосконалення процесів за допомогою цього зразка аудиторського звіту.

Процедури, виконані в цьому аудиторському звіті, включають: опитано персонал, відповідальний за сфери діяльності; аналіз даних було виконано для всіх замовлень на купівлю, пов'язаних з рахунками-фактурами від (Дата) до (Дата); аналіз даних було виконано для всіх рахунків-фактур, датованих між (Дата) і (Дата); проведено обмежене тестування транзакцій на точність, належну підтримку, обґрунтованість і належну авторизацію; і всі розглянуті області були розглянуті. До цього зразка включено зведення потенційних можливостей для вдосконалення, зазначених під час цього огляду.

Оптимізуйте процес керування своїм ІТ-постачальником за допомогою цієї СММ

Цю модель зрілості можливостей можна використовувати для вимірювання зрілості процесу управління постачальником ІТ організації та сприяння його просуванню від початкового/спеціального стану до оптимізованого стану.

Модель зрілості можливостей описує криву зрілості на таких рівнях можливостей: ПОЧАТКОВИЙ, який описує погано узгоджену функцію з незадокументованими стратегіями, ручними процесами управління, відсутністю інтегрованих систем і сильною залежністю від електронних таблиць/ручних документів; ПОВТОРЕНИЙ, який описує слабко узгоджену функцію, що підтримується неофіційними політиками, застосованими до процесів, що виконуються персоналом зі змішаним рівнем кваліфікації; ВИЗНАЧЕНИЙ, який описує наявну стратегічну структуру управління з чітко визначеними процесами, які підтримуються організованою та висококваліфікованою командою; КЕРОВАНИЙ, який описує функцію, узгоджену зі стратегічним планом організації та персоналом; і ОПТИМІЗОВАНИЙ, який описує процес управління, що виконується на оптимальному рівні з повним використанням найкращих практик.

У цьому прикладі стандарти та критерії управління постачальниками ОПТИМІЗОВАНОЇ організації загалом встановлено та повідомлено.

Модель зрілості можливостей – це структура, яка описує шлях вдосконалення від спеціального, незрілого процесу до зрілого, дисциплінованого процесу, зосередженого на

постійному вдосконаленні. CMM визначає стан процесу за допомогою спільної мови, яка базується на моделі зрілості можливостей Інституту розробки програмного забезпечення Карнегі-Меллона.

Застосовуйте найкращі методи роботи з нашою внутрішньою політикою використання запасів

Метою цієї політики є встановлення вказівок щодо придбання продукції компанії для внутрішнього використання.

Згідно з цією політикою, процес внутрішніх замовлень передбачає, що під час ініціювання замовлення для підрозділу відділу продажів, підрозділу внутрішнього використання, підрозділу роздачі або підрозділу, що оплачується, запитувач повинен заповнити внутрішнє замовлення на закупівлю. Форма внутрішнього замовлення на закупівлю має бути авторизована відповідно до політики авторизації зобов'язань і витрат на основі стандартних витрат.

Випереджайте нашу політику безпеки виробничого обладнання

Керуйте безпекою виробничого обладнання вашої організації, яке використовується в середовищі з підключенням до Інтернету, за допомогою найкращих процедур у цій політиці.

Ця політика розроблена для мінімізації потенційного ризику для компанії через втрату чутливих або конфіденційних даних компанії, інтелектуальної власності, шкоди суспільному іміджу тощо, які можуть виникнути внаслідок несанкціонованого використання ресурсів компанії. Ця політика охоплює всі пристрої, які є частиною виробничої інфраструктури з підключенням до Інтернету, якою володіє та керує компанія. Ці пристрої (мережа та хост) особливо вразливі до атак з Інтернету, оскільки вони добре помітні та є частиною рекламованої послуги.

Анкета для самооцінки попереднього контролю

Цей документ містить 11 зразків анкет, які внутрішній аудит може використовувати для оцінки погляду керівництва на середовище внутрішнього контролю організації.

Охоплено наступні сфери: виставлення рахунків, виплата готівки, основні засоби, загальний контроль інформаційних технологій, нарахування заробітної плати, закупівлі та управління запасами та казначейство. Приклади запитань включають: чи існують засоби керування системою, які сповіщають користувача про те, що певні рахунки-фактури наближаються до терміну їх виконання та їх потрібно оплатити? Які стандартні процедури перевірки для забезпечення своєчасної оплати рахунків? Чи існує стандартна процедура, згідно з якою хтось гарантує, що утримувані кошти виплачуються вчасно, коли вони належать?

Політика оцінки запасів

У цій Політиці оцінки запасів викладено набір процедур для забезпечення належного контролю та визначення вартості запасів, а також запобігання втратам або нестачам.

Відповідно до цього зразка всі запаси повинні бути належним чином проконтрольовані та оцінені, щоб забезпечити точність записів щодо матеріалів; робота в процесі; готові або частково готові нові або вживані товари, запчастини та лише ті запаси, які були придбані для продажу або які стануть частиною товарів, призначених для продажу. У цій політиці обговорюється метод нижчої вартості або ринкової оцінки та стверджується, що його слід застосовувати до фактичної кількості запасів під час визначення вартості кожного елемента запасів.

Політика авторизації затвердження витрат

Цей інструмент містить два зразки політик, які документують вимоги щодо затвердження та авторизації для використання коштів або активів компанії.

Згідно з цією політикою, усі співробітники повинні знати та дотримуватися своїх обмежень повноважень щодо затвердження; фінансова організація несе відповідальність за забезпечення дотримання цих інструкцій; президент або головний фінансовий директор може схвалити винятки з цієї політики, але не виходячи за межі своїх повноважень; ліміти авторизації повинні бути конвертовані у функціональну валюту операції; і місцевий

фінансовий менеджер несе відповідальність за оновлення місцевих рівнів, якщо це необхідно, через коливання курсу іноземної валюти.

Політика підрахунку циклу інвентаризації

Використовуйте цю політику підрахунку циклу інвентаризації, щоб підтримувати високий рівень точності записів інвентаризації у вашій організації.

Згідно з цією політикою корпоративний контролер відповідає за визначення тих операційних підрозділів у своїй групі, які виграють від програми підрахунку циклів, і співпрацює з операційним керівництвом і контролером підрозділу для своєчасного та ефективного впровадження такої програми. Операційне керівництво та контролер установки несуть відповідальність за впровадження та підтримання належної постійної системи інвентаризації та підрахунку циклів. Усі запити на використання постійної інвентаризації замість проведення фактичної інвентаризації слід направляти головному фінансовому директору компанії.

Анкета розподілу обов'язків

Фундаментальним елементом внутрішнього контролю є розподіл певних ключових обов'язків. Адекватний розподіл обов’язків зменшує ймовірність того, що помилки (навмисні чи ненавмисні) залишаться непоміченими, забезпечуючи окрему обробку різними особами на різних етапах транзакції та незалежні перевірки виконаної роботи. Цей інструмент містить 14 зразків анкет, які аудитори можуть використовувати для забезпечення належного розподілу обов’язків для різноманітних процесів, включаючи нарахування заробітної плати, основні засоби та інвентар.

Звіт аудиту процесу скидок

У цьому звіті містяться результати аналізу процесу скидок компанії. Переглянуті процеси включають процес знижок групи дистриб’юторів, процес знижок групи покупців, а також процес обліку знижок і головну книгу. Мета огляду полягала у виявленні потенційних сильних і слабких сторін контролю в рамках цього процесу, тестуванні засобів контролю, що стосуються процесу знижок групи дистриб’юторів, і надання рекомендацій, які могли б допомогти зменшити бізнес-ризики, пов’язані зі знижками. Загальна мета була зосереджена на покращенні середовища контролю. Цей звіт надає керівництву інформацію про стан ризиків і внутрішнього контролю в певний момент часу.

Політика претензій щодо вантажних перевезень

Ця політика визначає дії, які необхідно вживати під час обробки претензій щодо вантажу для відправлень, якщо перевізник втратив та/або пошкодив товари під час транспортування або якщо в доставлених товарах виявилася нестача.

Деякі з процедур, викладених у цій політиці, включають: перевірку коносамента (BOL) на назву перевізника та дату відправлення, щоб переконатися, що претензію буде оброблено для правильного перевізника, і кожна претензія на фрахт має бути окремим набором документів із BOL, підтвердження доставки (POD), рахунок-фактура тощо нижче та додається. Кредит-ноти слід друкувати поверх претензій (у тому ж порядку, що й вимоги).

Політика нестандартних операцій

Цей інструмент містить два зразки політики, які встановлюють уніфіковані процедури для виявлення та реєстрації нестандартних операцій компанії.

У цих зразках кожен контролер підприємства несе відповідальність за виявлення нестандартних операцій. Усі нестандартні транзакції мають бути офіційно задокументовані в письмовій формі та передані корпоративному контролеру або фінансовому директору після їх виникнення. Будь-які нестандартні транзакції будуть повністю розкриті в щоквартальному пакеті розкриття інформації для кожного сайту, а про будь-які нестандартні транзакції, ініційовані в США та впливаючи на закордонні місцезнаходження (наприклад, реструктуризація), буде офіційно повідомлено в письмовій формі іноземним контролерам до ініціювання угоди.

Робоча програма аудиту основних засобів

Шість зразків робочих програм, включених до цього інструменту, висвітлюють детальні процедури, які слід враховувати під час проведення аудиту основних засобів.

Зразкові процедури включають отримання та перегляд результатів фізичного останнього основного засобу; підготовка робочого процесу та/або опису для документування циклу основних засобів, включаючи процес затвердження, ідентифікацію перевиконання проекту, додавання активів і закриття проекту, вибуття активів, передачу активів і фізичну перевірку активів; отримання списку основних засобів та перевірка наявності значних активів; підтвердження підтверджуючих рахунків-фактур, відзначаючи належний запис сум; та визначення процесу фізичного обслуговування виробничого обладнання.

Політика закупівель

Цей документ містить чотири приклади політики, які встановлюють стандарти та вказівки, яких організація повинна дотримуватися під час свого циклу закупівель.

Політика закупівель зосереджена на забезпеченні закупівлі якісних матеріалів, витратних матеріалів і послуг уповноваженими особами від уповноважених постачальників з урахуванням відповідності кількості, конкурентоспроможних цін і своєчасної доставки. Ця комерційна діяльність завжди повинна здійснюватися в найкращих інтересах компанії. Керівні принципи також повинні враховувати та уникати перебоїв у бізнес-операціях через неналежні або несвоєчасні закупівлі та неефективне використання готівки, спричинене надмірними закупівлями.

Політика фізичної інвентаризації

У своїй діяльності компанії використовують різні види запасів. Підтримання належного контролю над запасами є важливим для забезпечення належної балансової оцінки та визнання витрат на продану продукцію. Крім того, фактична інвентаризація дає можливість підтримувати точність записів постійної інвентаризації.

Ціль цієї політики полягає в тому, щоб надати компанії та її дочірнім компаніям рекомендації щодо необхідності проводити регулярну фізичну інвентаризацію сировини, незавершеного виробництва, готової продукції та запасів. Згідно з цією політикою, процедури повинні забезпечувати узагальнення всіх результатів фізичної інвентаризації, розслідування розбіжностей і реєстрацію необхідних коригувань.

Політика закриття книг

Метою цієї політики є встановлення процедур своєчасного надання інформації щодо вимог компанії до звітності.

Згідно з цією політикою корпоративна бухгалтерія несе відповідальність за публікацію календаря закриття кожного місячного звітного періоду, який розповсюджується серед усіх бухгалтерів і кожного відділу, задіяного в щомісячному процесі закриття, публікацію списку необхідних звірок та аналізу рахунків, які необхідно підготувати та надіслані бухгалтерським персоналом для перевірки корпоративним контролером або менеджером з фінансової звітності, а також перегляд балансів фінансових звітів, щоб переконатися, що баланси записуються відповідно до загальноприйнятих принципів бухгалтерського обліку (GAAP).

Політика управління активами

Цей інструмент містить два приклади політик, які зосереджені на підтримці системи управління активами для моніторингу та управління інвестиціями, зробленими в основні засоби та інвентар.

У цих зразках загальна мета управління компанією полягає в тому, щоб отримати максимальний прибуток від активів, що використовуються в бізнесі. Для цього потрібні не тільки прибуткові операційні результати, але й рентабельність має відповідати розміру операційних інвестицій та використаних активів. Кожна операційна одиниця компанії повинна впроваджувати та підтримувати системи управління активами, спрямовані на збереження операційних інвестицій на найнижчому, економічно вигідному рівні, що відповідає довгостроковим цілям отримання прибутку.

Робоча програма аудиту знижок на придбання

Кроки найкращої практики, включені в цю програму аудиту, можуть бути використані організаціями для проведення аудиту знижок на купівлю.

Приклади робочих кроків включають спостереження та документування робочого процесу та/або опису для документування розуміння процесу купівельних знижок, документування методології, що лежить в основі коригування інвентаризації та припущень, що використовуються під час запису коригувань/записів інвентаризації, відстеження та узгодження даних про закупівлю, що використовуються в розрахунки знижок до вихідних даних закупівлі/постачальника та перевірка наявності підтверджувальної документації для кожної грошової та кредитної знижки, застосованої до кредиторської заборгованості.

Робоча програма аудиту закупівель

У цьому зразку робочої програми ми пропонуємо кроки найкращої практики для процесу аудиту закупівель. Детальні кроки в цьому документі дають змогу зрозуміти процес закупівлі, зокрема, як створити карти процесу, переглянути політику відділу та порівняти поточну практику з найкращою.

Етапи аудиту включають: створення підсумкової картки показників для кожного перевіреного процесу; використовуючи стандартний шаблон звіту, створити звіт внутрішнього аудиту (включно зі зведенням результатів виконаної роботи); провести заключну нараду з ключовим керівництвом компанії для розгляду проекту звіту внутрішнього аудиту та висновків; і зібрати тестову роботу та ключові дані підтримки.

Опис створення бази даних

Визначення моделі даних передбачає вказівку множини припустимих інформаційних конструкцій, множини припустимих операцій над даними й множини обмежень для збережених значень даних.

Модель даних, з одного боку, являє собою формальний апарат для опису інформаційних потреб користувачів, а з іншого боку – більшість СУБД орієнтуються на конкретну модель даних, і, таким чином, якщо інформаційні потреби вдається точно виразити засобами однієї з моделей даних, те відповідна СУБД дозволяє відносно швидко створити працездатний фрагмент ІС.

Інформаційні конструкції, операції й обмеження моделей даних вибираються з досить невеликої множини варіантів, що характеризує "великі" інформаційні об'єкти й операції. Зокрема, не допускається розгляд окремих символів даних, операцій додавання атрибутів, обмеження на відповідність типів даних і т.п., що характерно для мов програмування.

Інформаційні об'єкти послужили основою для об’єктно-орієнтованого проектування систем, коли фіксується множина інформаційних об'єктів і дій над об'єктами. Типовий список дій містить у собі створення/знищення об'єкта, редагування об'єкта, фіксацію одного об'єкта як частина іншого об'єкта, зв'язування об'єктів, синхронізацію дій над об'єктами.

Таки часто всі названі об'єкти вбудовуються в структуру відносин, які можна вважати найпростішими універсальними об'єктами.

Кількість істотно, що розрізняються моделей, даних визначається наявністю різних множин інформаційних конструкцій.

Логічна структура бази даних визначає:

– таблиці і їхні імена, також називані *сутностями* (entities);

– імена полів, також називані *атрибутами* (attributes) кожної таблиці;

– характеристики полів, наприклад унікальність їхнього значення й допустимість значень NULL, а також тип даних, збережених у полі;

– первинний ключ кожної таблиці – поле (кілька полів) зі значеннями, що унікально ідентифікують кожний запис у таблиці. У таблиці також можуть існувати інші унікальні поля, але тільки одне з них розглядається як унікальний ключ доступу для пошуку записів – первинний ключ. У таблиці не обов'язково повинен існувати первинний ключ, однак рекомендується визначати його для кожної таблиці.

Розробка структурної схеми

Структурна схема розробленої системи наведена на рисунку 1. З структурної схеми ми бачимо, що у рамках магістерського проекту реалізовано клієнт-серверну архітектуру доступу до бази даних, де зберігаються дані по паливо-мастильним матеріалам та по запасним частинам для техніки механізованої колони підприємства.

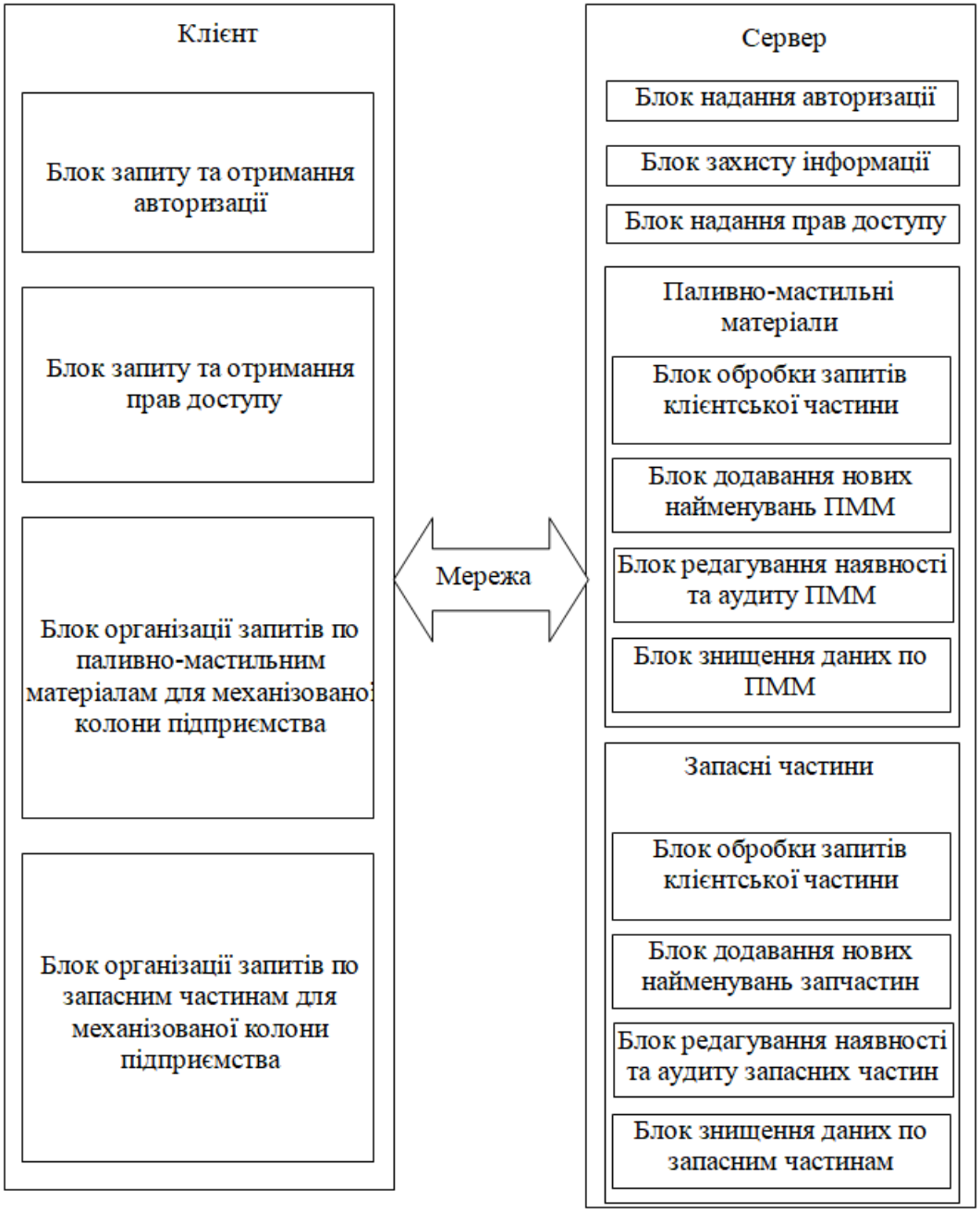


Рисунок 1 – Структурна схема розробленої системи

На клієнтській стороні реалізуються наступні функції:

– Блок запиту та отримання авторизації. У цьому блоці реалізується авторизація користувача. Також у цьому блоці реалізовано можливість встановлення та зміни логіну.

– Блок запиту та отримання прав доступу. У цьому блоці реалізується процедура запиту на права доступу, й відповідно отримання прав користувача або адміністратора, з наданими їм повноваженями.

– Блок організації запитів по паливно-мастильним матеріалам для механізованої колони підприємства. Запити можуть організовуватися або користувачем, у цьому випадку він має право доступу тільки до кількості матеріалів, або адміністратором, у цьому випадку

адміністратор має право редагування складу та кількості паливно-мастильних матеріалів.

– Блок організації запитів по запасним частинам для авто техніки механізованої колони підприємства. Запити можуть організовуватися або користувачем, у цьому випадку він має право доступу тільки до кількості запасних частин, або адміністратором, у цьому випадку адміністратор має право редагування складу та кількості запасних части для автотехніки.

Перейдемо до розгляду серверної частини. Доступ до серверної частини має тільки адміністратор та відповідний програміст, який веде цю розподілену базу даних.

На серверній частині реалізовані наступні функції:

– Блок надання авторизації. У цьому блоці на запит зі сторони сервера відбувається перевірка чи є авторизованим користувач. У випадку, якщо відбувається нова авторизація, то перевіряється те, хто проводить авторизацію. Якщо її проводить адміністратор, то нова авторизація дозволяється у іншому випадку, вона заборонена.

– Блок захисту інформації. Цей бок призначен для захисту інформації, яка зберігається у базі даних, та недопущенню відносно неї несанкціонованих дій. Для цього відбувається шифрування усіх даних які зберігаються у базі даних, та перевіряється цифровий підпис при кожному зверенені до бази даних, для встановлення авторства користувача при тій, або іншій дії.

– Блок надання прав доступу. Цей блок надає відповідні права доступу при запиті зі сторони адміністратора. Також у цьому блоці перевіряються права доступу користувача та адміністратора й у випадку порушення переривається робота з базою даних.

Також серверна частина містить два великі блоки. Один з яких обробляє усі запроси, які відносяться до паливо-мастильних матеріалів, а другий обробляє усі запроси, які відносяться до запасних частин автотехніки механізованої колони підприємства.

Розглянемо їх більш детальніше.

Блок паливно-мастильних матеріалів підрозділяється на наступні блоки:

– Блок обробки запитів клієнтської частини. У цьому блоці обробляються усі запити які мають відношення до дій з рухом паливо-мастильних матеріалів.

– Блок додавання нових найменувань паливно-мастильних матеріалів. У цьому блоці розширюється діапазон тих паливно-мастильних матеріалів, з якими мають справу у механізованій колонні підприємства.

– Блок редагування наявності та аудиту ПММ. У цьому блоці адміністратором редагується наявність кількості тих або інших видів паливно-мастильних матеріалів, та реалізована можливість проведення аудиту наявності паливно-мастильних матеріалів на складі.

– Блок знищення даних по паливно-мастильним матеріалам. У цьому блоці відбувається знищення тих, або інших видів паливно-мастильних матеріалів.

Блок запасних частин підрозділяється на такі блоки:

– Блок обробки запитів клієнтської частини. У цьому блоці обробляються усі запити які мають відношення до дій з рухом запасних частин.

– Блок додавання нових найменувань запасних частин. У цьому блоці розширюється діапазон тих запасних частин, з якими мають справу у механізованій колонні підприємства.

– Блок редагування наявності та аудиту запасних частин. У цьому блоці адміністратором редагується наявність кількості тих або інших видів запасних частин, та реалізована можливість проведення аудиту наявності запасних частин на складі.

– Блок знищення даних по запасним частинам. У цьому блоці відбувається знищення тих, або інших видів запасних частин.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів складського аудиту автотранспортного підприємства. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем складського аудиту автотранспортного підприємства. Досліджена система складського аудиту автотранспортного підприємства. На основі отриманих результатів

досліджень створена програмна реалізація системи складського аудиту автотранспортного підприємства. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання складського аудиту автотранспортного підприємства. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – С. 85-88.
2. Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 1(126). – С. 75-79.
3. Коваленко А.С. Обґрунтування набору даних для оцінки технічного стану інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2015. – Вип. 1(42). – С.39-41.
4. Коваленко А.С. Експертна система технічного діагностування інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2015. – № 1(41). – С. 106-111.
5. Коваленко А.С. Удосконалення методу технічного обслуговування об’єктів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко, О.П. Доренський // Системи озброєння і військова техніка. – Х.: ХУПС, 2016. – № 2(46). – С. 109-114.
6. Кожанова А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. Кожанова, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2013. – Вип. 6(113). – С. 255-257.
7. Коваленко А.С. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко., А.А. Смірнов, А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2014. – Вип. 4(120). – С. 161-164.
8. Коваленко А.С. Підсистема технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А.Смірнов, О.В. Коваленко // Системи озброєння і військова техніка.– Х.: ХУПС, 2014. – № 1(37). – С. 126-129.
9. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.
10. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.
11. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.
12. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.
13. Kovalenko A.S. Information model and its element for displaying information on technical condition of objects of integrated information system / A.S. Kovalenko, A.A. Smirnov, A.V. Kovalenko, A.P. Dorensky // International Journal of Computational Engineering Research (IJCER). – India: Delhi, 2016. – Volume 6, Issue 1. – P. 21-27.
14. Кожанова А.С. Система технічної діагностики інтегрованих інформаційних систем – обґрунтування необхідності створення, визначення понятійного апарату та напрямів досліджень / А.С. Кожанова, О.А. Смірнов, М.П. Савченко, Д.М. Ізосімов, В.В. Мороз // Створення та модернізація озброєння і військової техніки в сучасних умовах: Тринадцята наук.-техн. конф., 5-6 вер. 2013 р., м. Феодосія: тези доп. – Феодосія: ДНВЦ, 2013. – С. 187-188.
15. Кожанова А.С. Визначення основних напрямків досліджень щодо створення системи технічної діагностики інтегрованих інформаційних систем / А.С. Кожанова, О.А. Смірнов, А.В. Челпанов // Проблемні питання розвитку озброєння та військової техніки Збройних Сил України: IV наук.-техн. конф., 16-20 груд. 2013 р., м. Київ: зб. тез. – Київ: ЦНДІ ОВТ ЗСУ, 2013. – С. 293.
16. Коваленко А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформатика та системні науки : V Всеукр. наук.-практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 2014. – С. 292-294.

17. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2014. – С. 241.
18. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп'ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 28-31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 2014. – С. 190-193.
19. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2015. – С. 15.
20. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

УДК 004

В.Ковальчук, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ХМАРНИХ СЕРВІСІВ З ВИКОРИСТАННЯМ ЦСК

У статті розроблено програмне забезпечення, яке призначено для системи хмарних сервісів з використанням ЦСК. Метою розробки є дослідження та програмна реалізація системи хмарних сервісів з використанням ЦСК. Об'єктом дослідження є процес хмарних сервісів з використанням ЦСК. Предметом дослідження є методи хмарних сервісів з використанням ЦСК. Методи дослідження базуються на методах хмарних технологій та хмарних технологій та захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи хмарних сервісів з використанням ЦСК. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, хмарні сервіси, центр сертифікації ключів

Постановка проблеми. Робота присвячена питанням забезпечення безпеки інформації в сервіс-орієнтованих хмарних архітектурах, за рахунок розробки центру сертифікації та розподілу ключів (ЦСК). При згадуванні абревіатури SOA (Service-Oriented Architecture) більшість ІТ-фахівців першою справою згадують Web-сервіси й протокол HTTP, які є складовими хмарних сервісів, хоча цей термін позначає набагато більше широке поняття.

Хмарна архітектура SOA зовсім незалежить від мов програмування, платформ або протокольних специфікацій, за допомогою яких сервіси розробляються, а також від того, де й за допомогою чого вони розгорнуті.

Практично хмарна архітектура SOA вимагає наявності не тільки сервісів, але й засобів, за допомогою яких ці сервіси можуть бути виявлені й підключені незалежно від нижчележачої інфраструктури. SOA – це не продукт або специфікація. Хмарна архітектура ретельно вибудовується – складається з множини компонентів, таких, як сервери додатків, що зв'язують ПЗ, репозиторій і навіть спеціалізовані пакети централізованого управління SOA.

Строго говорячи, SOA не можна відносити ні до нової реалізації CORBA, ні до оновленої хмарної архітектури RMI (Remote Method Invocation). Ключовий компонент SOA – сервіс. Сервіси тут є бізнес-функціями, призначеними для забезпечення погодженої роботи великих, що складаються з множини частин додатків.

По суті, це будівельні блоки для відбиття бізнес-логіки в розроблювальних додатках. А кінцевим місцем, де сервіси “живуть”, є сервер додатків, будь то WebLogic від BEA Systems, WebSphere від IBM, Application Server від Oracle або Java AS від Sun Microsystems.

Функції, або операції, в SOAP (Simple Object Access Protocol) повинні бути інтуїтивно зрозумілими й відповідати своїм назвам – наприклад, submitPurchaseOrder (“підтвердити замовлення на покупку”) або validateCustomerAccounts (“перевірити особовий рахунок замовника”).

На відміну від звичайних додатків сервіс в хмарній архітектурі SOA призначається для використання всім реалізованим бізнес-функціям. У той час як звичайні корпоративні додатки містять у собі схожі фрагменти бізнес-логіки або навіть дублюють окремі об'єкти – наприклад, об'єкт клієнтського замовлення, – в хмарній архітектурі SOA вам потрібно запустити лише єдиний екземпляр такої бізнес-функції.

Таким чином, можливо повторно використовувати функціональність у середовищі із множинними додатками й швидко коректувати бізнес-логіку, для того щоб мати можливість пристосовуватися до мінливих умов ринку. У цьому й складається головна перевага SOA.

Зі зміною єдиного екземпляра бізнес-функції в SOA автоматично вносяться корективи й в усі додатки, що опираються на цю функцію. Так що, наприклад, будь-які зміни в правилах ціноутворення або політики знижок застосовуються у всіх додатках.

Аналогічно будь-які зміни в підтримуючій інфраструктурі залишаються прозорими для всіх додатків, що використовують сервіси. Наприклад, якщо ви переходите з однієї версії бази даних на іншу, то будуть модифіковані лише пов'язані з нею сервіси, оскільки додатки в хмарній архітектурі SOA працюють із усіма інфраструктурними додатками тільки за допомогою сервісів. У недавнім минулому зміни в сполучному ПЗ або в базі даних змушували переробляти всю систему, включаючи клієнтські настільні додатки.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи хмарних сервісів з використанням ЦСК.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи хмарних сервісів з використанням ЦСК.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем хмарних сервісів з використанням ЦСК.
– Дослідження системи хмарних сервісів з використанням ЦСК.
– Програмна реалізація системи хмарних сервісів з використанням ЦСК.

Об’єктом дослідження є процес хмарних сервісів з використанням ЦСК.

Предметом дослідження є методи хмарних сервісів з використанням ЦСК.

Методи дослідження базуються на методах хмарних технологій та захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Опис технології WS Security

Надаючи вільно зв'язані сервіси, сервіс-орієнтована хмарна архітектура дозволяє гнучко реагувати на постійно мінливі ділові процеси. При цьому необхідно приділити увагу не тільки функціональним аспектам, але й створенню гнучкої інфраструктури безпеки, оскільки зміни ділових процесів роблять на неї серйозний вплив. Приміром, залучення нових ділових партнерів або включення конфіденційних відомостей у важливі корпоративні процеси вимагає адекватного стандартизованого рішення для забезпечення безпеки.

Як основна технологія забезпечення безпеки повідомлень на базі SOAP (Simple Object Access Protocol) міцно закріпився стандарт безпеки служб Web (Web Services Security, WS Security), ратифікований OASIS, організацією по розвитку стандартів структурованої інформації. WS Security складається із цілого пакета специфікацій і множини механізмів, які комбінуються відповідно до необхідного сценарію застосування.

До честі творців стандартів у рамках SOA вони приділили підвищену увагу безпеки при розробці цих стандартів. Механізми безпеки органічно вбудовуються в концепцію Web-сервісів і дозволяють не тільки уникнути основних проблем, але й істотно підвищити ефективність як механізмів захисту, так і засобів керування політикою безпеки.

Стандарти

Основний пул стандартів безпеки Web-сервісів розробляється в рамках консорціуму OASIS. Структуру специфікацій безпеки SOA можна зобразити у вигляді наступної ієрархічної конструкції (рисунок 1).

Розглянемо ці стандарти:

– Базові стандарти (SOAP Foundation) містять у собі специфікації XML Signature і XML Encryption, які визначають відповідно формати ЕЦП і шифрування SOAP-транзакцій. Дані специфікації ніяк не обмежують список алгоритмів шифрування й ЕЦП, що робить вбудовування українського ДСТУ в SOA-архітектуру неважким завданням. Також до базових понять можна віднести інформацію в складі SOAP-заголовка (security-token, маркер безпеки), використовувану для автентифікації й авторизації запиту. Наприклад, security-token може містити в собі сертифікат X.509 і/або ім'я/пароль. Одним з видів security-token є SAML (Security Assertion Markup Language), що включає в себе інформацію про статус автентифікації, авторизації й атрибутах учасників транзакції. Це дозволяє забезпечити побудову відносин довіри (trust) в SOA-хмарній архітектурі й виключити необхідність автентифікації/авторизації для кожного запиту.

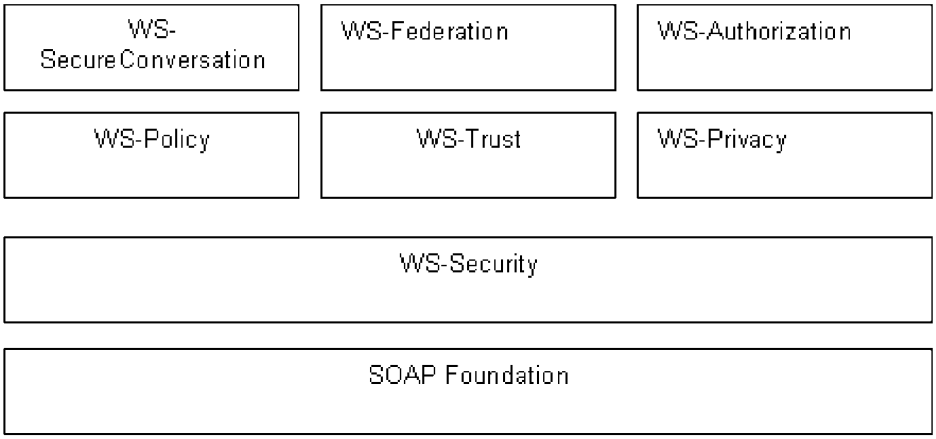


Рисунок 1 – Структура пула стандартів безпеки Web-сервісів

– WS-Security визначає базові механізми й формати використання security-token у складі SOAP-запитів. Основною метою WS-Security є абстрагування реалізації політик безпеки Web-сервісів від конкретних методів (наприклад, протоколів автентифікації й авторизації). За допомогою уточнюючих специфікацій, описаних нижче, WS-Security дозволяє досягти сумісності методів реалізації політик безпеки, описаних з використанням даних стандартів.

– WS-Policy визначає шаблони й правила опису політики безпеки для Web-сервісів.

– WS-Trust описує правила організації довірених відносин між учасниками Web-взаємодії.

– WS-Privacy визначає формати політики конфіденційності при обміні SOAP-Повідомленнями.

– WS-SecureConversation регламентує правила безпечного обміну повідомленнями в SOA-хмарній архітектурі.

– WS-Federation є специфікацією, що визначає встановлення довірених відносин між різними доменами безпеки.

– WS-Authorization описує формати опису правил розмежування доступу до Web-сервісів.

Отже, стає ясно, що безпека в SOA-хмарній архітектурі описується досить великим набором специфікацій. Втішно, однак, що даний набір є невід'ємною частиною пула стандартів SOA і розробляється одночасно з ним. Це дає підстави думати, що додатки в складі Web-сервісної хмарної архітектури можуть створюватися безпечними вже на стадії проектування.

Хмарна архітектура

Розглянемо тепер типову архітектуру безпеки Web-сервісів, що застосовується в більшості рішень корпоративного рівня.

Ключові завдання, покладені на таку архітектуру:

– Керування доступом до Web-сервісів і однократна автентифікація (Single Sign-on, SSO). Призначено для забезпечення однократної автентифікації, авторизації й аудита Web-сервісів.

– Централізоване керування політикою безпеки. Дозволяє мінімізувати необхідність дублювання зусиль для застосування політики безпеки для кожного Web-сервісу за допомогою використання централізованої інфраструктури безпеки, не вимагаючи при цьому переробки самих Web-сервісів.

– Уніфікація процесу моніторингу. Дозволяє проводити аудит роботи Web-сервісів, що показує, які користувачі (додатка) здійснювали доступ до Web-сервісів, які дії вони виконували і які дані при цьому передавали.

– Маршрутизація запитів до Web-служб. Дозволяє, аналізуючи вміст запиту, проводити його перетворення й перенапрямок до того або інший Web-сервісу.

Схема керування захистом в SOA-хмарній архітектурі

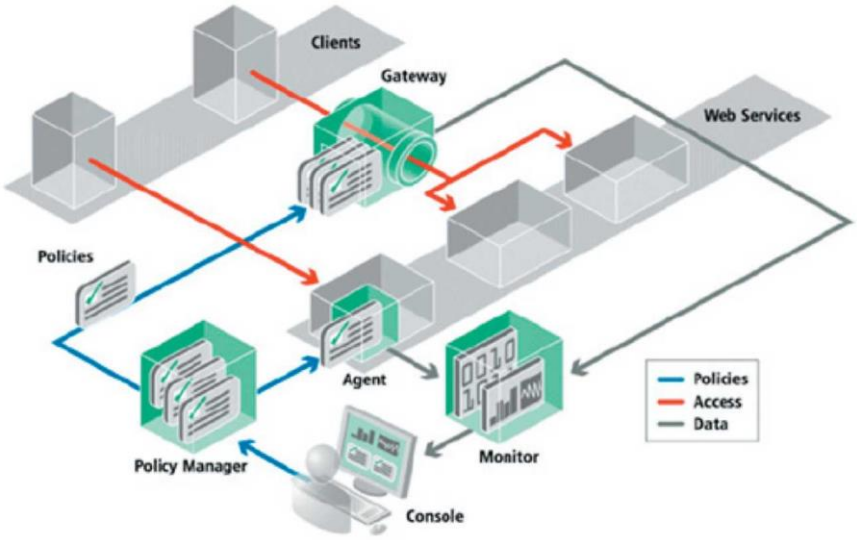


Рисунок 2 – Схема керування захистом в SOA-хмарній архітектурі

До складу такої схеми входять наступні компоненти:

– менеджер політик (Policy Manager);

– компоненти застосування політики: агенти (Agents) і шлюзи (Gateways);

– панель моніторингу (Monitor).

Менеджер політик – це графічний інструмент для визначення нових політик безпеки й експлуатації, зберігання політик, а також для керування поширенням і відновленням політик на агентах і шлюзах.

Компоненти застосування політик діляться на шлюзи (Policy Gateways) і агенти (Policy Agents). Шлюзи політик установлюються перед групою додатків або сервісів, перехоплюючи запити до цих додатків з метою застосування політик, підвищуючи безпеку вже встановлених додатків і додаючи в них нові правила. Агенти політик забезпечують

додатковий диференційований рівень безпеки й розміщаються на серверах додатків, що забезпечують виконання додатка або сервісу. Таким чином, забезпечується можливість автентифікації й авторизації запитів до Web-сервісів по наявним на підприємстві репозитаріям користувачів (наприклад, LDAP-каталог).

На панелі моніторингу адміністратор може задати рівні якості обслуговування для кожного додатка, визначити правила видачі попереджень і повідомлень, якщо додаток перевищить заданий рівень якості обслуговування.

Таким чином, архітектуру безпеки SOA можна побудувати без переробки безпосередньо Web-сервісів. Це одне з основних достоїнств наявності стандартів безпеки, що є частиною загального пула стандартів SOA.

Базові концепції

OASIS прийняла стандарт WS Security у березні 2004 р. як доповнення до протоколу SOAP. До теперішнього часу він визнаний цілком зрілим і придатним до застосування. WS Security не визначає ніяких нових технологій, а опирається на вже існуючі стандарти, приміром, XML Encryption, XML Signature, сертифікати X.509 або різні криптографічні алгоритми. Базова концепція ґрунтується на механізмах повідомлень, тому замість захисту, орієнтованої на транспорт, можливе забезпечення безпеки від краю до краю (End-to-End Security), приміром, за допомогою протоколу SSL. Такий підхід необхідний, щоб уникнути виникнення наскрізних комунікаційних структур у межах SOA, а також забезпечити передачу асинхронних повідомлень або використання проміжних станцій (приміром, сервісної шини підприємства – Enterprise Service Bus, ESB).

Основне завдання WS Security – забезпечення цілісності, конфіденційності й автентичності повідомлення і його відправника при одночасному збереженні відкритості для розширень. Основними елементами стандарту є наступні базові механізми (рисунок 3): токени безпеки, шифрування, підписи й оцінки про час.

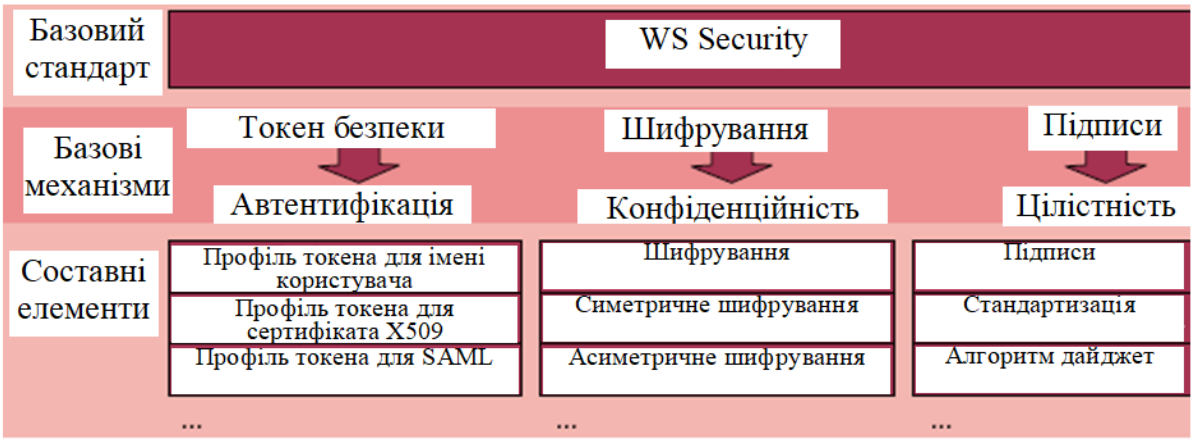


Рисунок 3 – Базові механізми WS Security

Токени безпеки (Security Token). Автентифікація відправника – базова передумова для забезпечення контролю доступу (Access Control) з боку сервісу, а крім того, вона необхідна для організації обліку й контролю. Підтвердження ідентифікації (Credentials), без яких неможлива автентифікація, передаються усередині повідомлення у вигляді токенів. Сама автентифікація не входить до складу WS Security – це самостійний процес провайдеру послуг. Для різних форматів токенів OASIS пропонує окремі специфікації у вигляді профілів WS Security. Так, «Профіль токена з ім'ям користувача» (Username Token Profile) регулює алгоритм широко розповсюдженого методу автентифікації користувача за допомогою ідентифікаційного номера (User ID) і відповідного пароля.

Ідентифікація додатків або ділових процесів звичайно здійснюється за допомогою сертифікатів, і в цьому випадку управляти паролями на стороні клієнта не потрібно. Обіг із сертифікатами для зазначеного методу автентифікації описується в профілі X.509 Certificate

Token Profile. Існують і інші профілі, приміром, для використання токенів мови розмітки тверджень безпеки (Security Assertion Markup Language, SAML) або Kerberos.

Двійкові або базовані на XML токени безпеки потрібні не тільки для автентифікації. Вони виконують ще одну функцію, являючи собою основу для транспорту або прив'язки ключів (Keys), застосовуваних у криптографії.

Шифрування. Щоб забезпечити захист конфіденційних даних, використовується криптографічне шифрування. Оскільки протокол SOAP базується на XML, то WS Security не визначає новий стандарт, а використовує специфікацію XML Encryption з W3C. Зашифровані дані і їхня метаінформація, у свою чергу, включаються в повідомлення у вигляді структур XML. Однак, відповідно до специфікації SOAP, не можна шифрувати елементи «конверт» (Envelope), «заголовок» (Header) і «тіло» (Body), оскільки вони задають структуру повідомлення й повинні бути читаємі завжди.

Принципово розрізняють два механізми шифрування: симетричне й асиметричне. При симетричному шифруванні (метод «секретного ключа» – Secret Key) для шифрування й дешифрування використовується загальний ключ, завжди доступним обом сторонам. При асиметричному шифруванні (алгоритм із відкритими ключами – Public Key) для шифрування й дешифрування застосовуються різні ключі, що істотно скорочує витрати зусиль на їхній розподіл: особистий ключ (Private Key) залишається у власника, а загальний ключ (Public Key) поширюється вільно. Однак у порівнянні із секретними ключами механізм відкритих ключів працює значно повільніше, тому обидва підходи часто поєднують, у результаті чого з'являються нові гібридні варіанти. Клієнт генерує симетричний ключ сеансу (Session Key) і використовує його для симетричного шифрування більших обсягів даних. На закінчення симетричний ключ шифрується за допомогою асиметричного алгоритму, вкладається в повідомлення й надається в розпорядження сервісу.

Підпис (Signature). Для підтвердження цілісності повідомлень застосовуються підписи. Вони дозволяють розпізнати неправомірні модифікації: зміна, видалення або додавання даних. Реалізація цього підходу в рамках WS Security опирається на стандарт XML Digital Signature від W3C. Принцип підписів заснований на створенні контрольних сум за допомогою спеціальних алгоритмів (дайджет). Результати приєднуються до повідомлення й передаються в частково зашифрованому виді. Сервісна сторона формує контрольну суму й порівнює неї зі значенням, присланим клієнтом. Оскільки в XML різні способи написання логічно ідентичні, перед формуванням контрольної суми необхідно зробити нормалізацію даних. Для цього використовуються стандартизовані алгоритми XML Canonicalization, також запозичені з W3C.

Крім того, підпису надають можливість установлення автентичності відправника. Цю інформацію можна використовувати в юридичних цілях для встановлення авторства.

Оцінка про час (Timestamp). Ідея послуг у рамках SOA має на увазі, що сервіси повинні робити визначену дію й у такий спосіб підтримувати взаємодію без обліку стану (Stateless). Однак даний принцип комунікації без установлення сеансу відкриває простір для атак скидання (Replay), що коли атакує повторно відправляє або повідомлення цілком, або окремі їхні частини. Щоб перешкодити таким атакам, необхідно гарантувати унікальність повідомлень, для чого кожне з них одержує свій ідентифікаційний номер (Message ID), що сервіс перевіряє на предмет його унікальності. Тому ідентифікаційні номери вже отриманих повідомлень необхідно зберігати. Термін дії, а виходить, і час зберігання окремих ідентифікаційних номерів повідомлень на стороні сервісу обмежується оцінкою, що втримується в повідомленні, про час.

Крім використання традиційної структури оцінок про час у заголовку безпеки (Security Header), токен з ім'ям користувача пропонує власне керування оцінками про час щоб уникнути несанкціонованого повторного використання даних для автентифікації. Ідентифікаційний номер повідомлення повинен відповідати специфікації WS Addressing. А токени з ім'ям користувача одержують випадкове криптографічне значення (Nonce).

У рамках SOA кожний зі згаданих чотирьох базових механізмів охоплює лише один аспект забезпечення безпеки. Сформувати цілісне рішення можна лише за умови взаємодії всіх компонентів. Цілком традиційна комбінація механізмів безпеки на основі повідомлень (WS Security), орієнтованих на транспорт (SSL). Сценарій, представлений на рисунку 4, докладно роз'ясняє необхідність використання комбінації різних механізмів.

Автентифікація. Будь-який контроль доступу на стороні сервісу припускає автентифікацію клієнта. Сервісній стороні необхідно мати відомості про підтвердження ідентифікації. У випадку методу з користувальницьким ідентифікаційним номером і паролем WS Security надає механізм токенів з ім'ям користувача, де пароль є конфіденційною інформацією, тому необхідно запобігти його зчитуванню в процесі транспорту. Шифрування необхідно, навіть якщо механізм, визначений у специфікації токенів з ім'ям користувача, припускає передачу пароля тільки у вигляді контрольної суми. При використанні контрольної суми, що читається, виникає погроза атаки методом підбора пароля (Brute Force) шляхом перевірки всіх можливих комбінацій, оскільки паролі обмежені по довжині й набору символів.

Крім того, у випадку застосування контрольної суми пароля сервісній стороні знадобиться пароль відкритим текстом. Тому даний підхід у багатьох випадках неприйнятне або адміністрування паролів зажадає додаткових заходів безпеки.

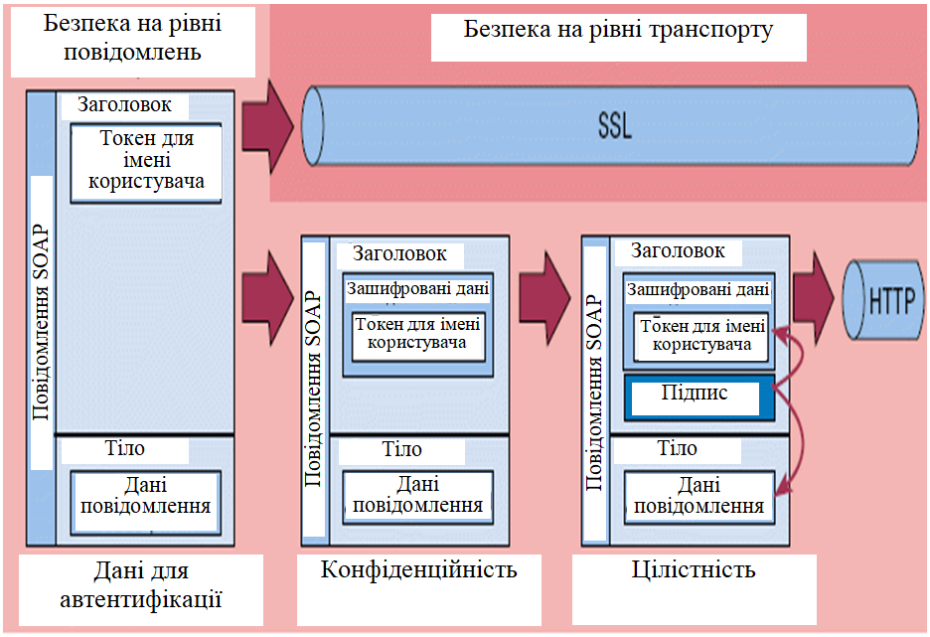


Рисунок 4 – Взаємодія токенів, шифрування й підписів WS Security

Конфіденційність. Запобігти розкраданню пароля під час пересилання повідомлення покликане шифрування токену з ім'ям користувача. У деяких випадках досить застосувати широко розповсюджений протокол SSL. Однак необхідно врахувати, що внаслідок принципу з'єднання двох точок, властивого SSL, використання проміжних вузлів, приміром, сервісної шини підприємства (Enterprise Service Bus, ESB), неможливо, і захист даних після їхньої передачі не забезпечується.

У той же час механізм шифрування WS Security надає метод на основі повідомлень: вихідні дані шифруються й заміняються за допомогою алгоритму шифрування XML. Додатково в повідомлення вкладається метаінформація, приміром, про використані алгоритми або ключі, і тепер воно може передаватися навіть за допомогою незахищених протоколів (приміром, HTTP), а конфіденційність даних не піддається погрозі.

Цілісність. У використаному як приклад сценарії відсутня зв'язок між токеном з ім'ям користувача, що перебуває в заголовку SOAP, і даними в тілі SOAP. У результаті виникає погроза підміни ключових елементів повідомлення. Зокрема, зашифрована інформація про

користувача, зазначена в заголовку, може бути постачена підробленим запитом сервісу. Однак ця проблема легко вирішується за допомогою підписів. Механізми підписів, використовувані в WS Security, «скріплюють» трохи пространственно розділених блоків даних, з яких складається повідомлення, що дозволяє перевірити цілісність усього повідомлення або окремих його частин. У контексті безпеки на базі повідомлень підпису виконують роль елементарних конструктивних компонентів і зачіпають не тільки тему цифрових підписів.

Унікальність повідомлення. Для того щоб запобігти повторному відправленню повідомлення (атака Replay), на сервісній стороні необхідно перевірити унікальність повідомлення. Для цього до повідомлення, представленому в стандартизованому виді, додається ідентифікаційний номер. Стандарт WS Addressing, визначений в W3C, передбачає, серед іншого, завдання ідентифікаційного номера повідомлення, що допомагає встановити його унікальність. Визначена в рамках WS Security структура вказує час створення повідомлення й закінчення строку його дії.

На закінчення потрібно відзначити, що ідентифікаційні номери повідомлень, як і оцінки про час, повинні бути прив'язані до існуючих блоків даних (інформація про користувачів і дані повідомлень). Для цього потрібно розширити діапазон охвату підпису, що дозволяє включити нові елементи при контролі цілісності.

Підписи і їхні завдання

Завдяки своїй інфраструктурі на основі повідомлень, сервіси Web підтримують можливість включення будь-яких проміжних інстанцій (Intermediaries) між кінцевими точками. С допомогою такої хмарної архітектури можна розширити функціональність сервісів Web. Крім того, ця хмарна архітектура стає основою для організації поділу відповідальності за реалізацію властивостей сервісу, особливо вимог, не пов'язаних з функціональністю (якість сервісу – Quality of Services, QoS). При виклику сервісу повідомлення із запитом і відповіддю повинні пройти через проміжні інстанції, причому кожна витягає з повідомлення дані, необхідні для виконання її завдань, і, якщо знадобиться, постачає його додатковою інформацією. Відповідно, необхідно, щоб визначені частини повідомлень були придатні для читання й зміни проміжними інстанціями. Так, за допомогою даних WS Addressing з повідомлення можна управляти функціями маршрутизації в межах ESB.

Для забезпечення конфіденційності й цілісності повідомлення, з одного боку, і читаності й розширюваності, з іншої, до механізмів безпеки пред'являються підвищені вимоги. Приміром, шифрування всього повідомлення за допомогою протоколу SSL перешкоджало б гнучкому використанню проміжних інстанцій. Крім того, стандарт SOAP вимагає, щоб конверт, заголовок і тіло повідомлення представлялися в незашифрованому виді.

Підписи виконують кілька важливих завдань для забезпечення всебічної безпеки в рамках такої вільно зв'язаної хмарної архітектури. Крім загальновідомої ролі цифрового підпису, вони надають механізми для перевірки цілісності й автентичності частин повідомлення. Через основну роль підписів необхідно бути в курсі їхніх базових принципів.

Цілісність даних

Підпису, крім іншого, дозволяють перевірити цілісність окремих блоків даних і розпізнати маніпуляції з повідомленнями (зміна, видалення й додавання даних). Для цього за допомогою спеціального криптографічного алгоритму створення гешу (приміром, SHA1, MD5) розраховуються контрольні суми для важливих блоків даних (Message Digest). Геш-алгоритми – необоротні функції, і відновлення вихідних даних за відомим значенням гешу неможливо. Крім того, його величина для різних даних не повинна збігатися (відсутність колізій). Для перевірки геша приймаюча сторона ще раз розраховує контрольну суму й порівнює отриманий результат із присланим. Якщо обоє значення ідентичні, то цілісність даних дотримана, тоді як в інших випадках велика ймовірність змін повідомлення. Однак цей

механізм не дозволяє встановити, які саме зміни внесені, оскільки підпису повідомляють тільки про вірний або невірний результат.

Принцип підписів базується на формуванні контрольних сум обома сторонами (відправником і одержувачем), тому однакова форма подання даних є обов'язковою умовою. Приміром, для того щоб різне написання XML не привело до різниці контрольних сум, варто ввести проміжний етап – нормалізацію XML (Canonicalization).

Однак одного доказу цілісності окремих блоків даних недостатньо для підтвердження автентичності всього повідомлення – потрібно забезпечити єдність окремих блоків. Для цієї мети створюється загальна контрольна сума шляхом об'єднання значення гешу для всіх блоків даних. У результаті здійснюється криптографічний зв'язок блоків, що не залежить від їхнього положення усередині повідомлення: таким чином, загальна контрольна сума дає можливість перевірити автентичність усього повідомлення. Крім того, так можна розпізнати маніпуляцію зі значеннями гешу окремих блоків даних. У процесі транспорту повідомлення загальна контрольна сума захищається від зміни за допомогою криптографічного механізму шифрування. Для реалізації автентичності повідомлення можна застосовувати симетричне шифрування (приміром, HMAC). При цьому ключ, спільно використовуваний відправником і одержувачем, або передається заздалегідь, або створюється відправником у момент передачі, а потім відправляється в зашифрованому виді разом з повідомленням.

Крім перевірки цілісності даних і автентичності повідомлень, підпису надають можливість автентифікації відправника всього повідомлення або його частин (рисунок 5).

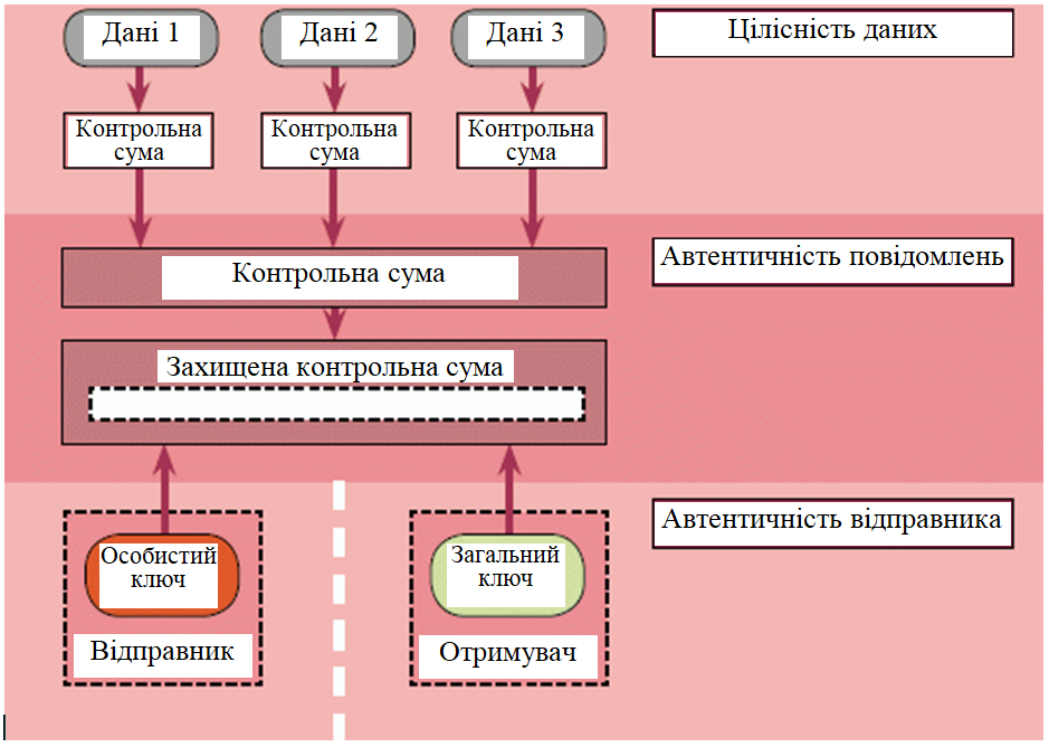


Рисунок 5 – Перевірка цілісності даних і автентичності повідомлень, а також автентифікації відправника всього повідомлення або його частин

Це властивість відомо як цифровий підпис. У цьому випадку застосовується не симетричний алгоритм шифрування загальної контрольної суми, а асиметричний алгоритм із застосуванням відкритих ключів: відправник використовує власний ключ для шифрування значення гешу, а прочитати його можна лише за допомогою відкритого ключа, сертифікат якого надає інформацію про власника особистого ключа. Якщо сертифікат, а виходить, і відкритий ключ, викликає довіру, то з його допомогою можна визначити відправника повідомлення.

На перший погляд, набір стандартів і реалізація хмарної архітектури безпеки в SOA здаються нетривіальними. Але його переваги переважують всі складності опису й

реалізації. До них відносяться:

– Відділення політики безпеки сервісів від самих сервісів дозволяє побудувати універсальні сервіси захисту для всіх бізнес-додатків без необхідності втручання в бізнес-логікові й "прошивання" функцій безпеки в код бізнесів-додатків.

– Чітке розмежування експертизи. Розроблювачі сервісів формують бізнес-логіку, архітектори й адміністратори визначають політику безпеки й керування.

– Єдина точка керування політикою ІБ.

– Зниження витрат на адміністрування, оскільки зміни в політику безпеки вносяться централізовано, а не в кожному Web-сервісі. Крім того, аудит безпеки для всіх сервісів ведеться з єдиної точки адміністрування.

– Спрощення підтримки й внесення змін у середовище керування й забезпечення безпеки Web-сервісів за рахунок використання єдиних сервісів безпеки для всіх Web-сервісних додатків.

Визначенено, що такий значний набір переваг SOA з погляду безпеки послужить достатнім стимулом для співробітників підрозділів ІБ підтримати зусилля своїх колег з IT-підрозділів по побудові Web-сервісної хмарної архітектури.

Розробка структурної схеми

Одним з найважливіших завдань забезпечення інформаційної безпеки в сервіс-орієнтованих хмарних архітектурах (SOA) є захист потоків корпоративних даних, переданих по каналах загального користування, у тому числі й через Internet. Перспективним методом надійного захисту інформації є метод кодування даних.

Для рішення цього завдання необхідно здійснити кодування інформації на виході з локальної мережі й декодування вхідних у неї даних. Ці функції реалізуються спеціальними програмними або програмно-апаратними засобами. Якщо захист сегмента корпоративної мережі вже забезпечений міжмережевим екраном, природно покласти на нього також виконання функцій кодування й декодування.

Для реалізації можливостей кодування/декодування повинне бути виконане попередній (початковий) розподіл ключів. Сучасні технології пропонують для цього цілий ряд методів. Після сертифікації та розподілу ключів з'являється можливість здійснення процесу виробітку спільних секретних ключів, що обслуговують сеанс спілкування абонентів.

У результаті кодування весь обмін даними між територіально-віддаленими локальними мережами є захищеним і для користувачів виглядає як обмін усередині однієї локальної мережі, при цьому від користувачів не потрібно застосування яких-небудь додаткових захисних засобів.

Комплекс кодування міжмережевих потоків

Програмний комплекс кодування міжмережевих потоків (ККМП) реалізує функції кодування міжмережевих інформаційних потоків у мережах передачі даних протоколу TCP/IP для забезпечення обміну інформацією між територіально-віддаленими локальними мережами. Це забезпечується за допомогою організації віртуальних захищених мереж (Virtual Private Networks – VPN).

Комплекс виконує наступні функції:

– **Кодування міжмережевих потоків.** Функції кодування міжмережевих інформаційних потоків у відкритих мережах передачі даних виконуються шляхом організації VPN. Кожна мережа в складі VPN захищена своїм модулем, що кодує, установлюваним у точці її з'єднання із зовнішніми мережами. Інформація, що захищається, кодується на передавальному модулі й декодується на приймаючому, тобто передається у відкритому виді в межах локальних мереж і в кодованому – за їхніми межами. Кодований трафік передається по протоколу IPsec.

– Створення контуру безпеки. Розроблена система дозволяє сформувати контур безпеки, що поєднує IP-адреси всіх абонентів, що мають доступ у віртуальну захищену

мережу. Абонентами VPN можуть бути цілі мережі, підмережі й окремі робітники станції. Крім того, що кодує модуль може бути встановлений на окрему робочу станцію.

– Вибіркове кодування трафіку. Формування контуру безпеки служить для поділу трафіку на кодуємий і некодуємий потоки.

– Модуль, що кодує. Розроблена система робить виділення пакетів, які необхідно кодувати, на підставі IP-адрес відправника пакета й одержувача пакета й, крім того, перевірки інтерфейсу, через який проходить пакет.

– Управління ключовою системою. У розробленій системі реалізована несиметрична ключова система, коли потенційні учасники обміну даними використовують пари довгострокових секретних й відкритих ключів кодування. Кодування здійснюється на основі сеансових ключів, автоматично сформованих за допомогою довгострокових ключів і що мають обмежений час життя. Комплекс здійснює всі необхідні дії по управлінню ключами: генерацію й розподіл довгострокових ключів, виробіток сеансових ключів абонентів, сертифікацію відкритих ключів у довіреному центрі, планову й позаштатну зміну ключів кодування.

– Реєстрація подій, моніторинг і управління міжмережевими потоками. Розроблена система здійснює збір і зберігання статистичної й службової інформації про всі штатні й позаштатні події, що виникають при автентифікації вузлів, передачі кодованої інформації, обмеженні доступу абонентів ЛОМ. Засоби моніторингу проводять збір і аналіз протоколів реєстрації від всіх модулів комплексу по кодованому каналі.

– Захист з'єднань із мобільними клієнтами. До складу віртуальної захищеної мережі можуть входити мобільні користувачі – віддалені комп'ютери, що підключаються по виділеним або каналам зв'язку, що комутуються. Основною відмінністю Мобільного клієнта є динамічно-призначувана IP-адреса. Носієм ключової інформації для них є електронний ключ eToken.

Состав Комплексу

Комплекс складається з наступних компонентів:

1. Набір шлюзів кодування.
2. Центр генерації ключів.
3. Центр сертифікації та розподілу ключів.
4. Центр реєстрації мобільних клієнтів.
5. Центр підготовки електронних ключів мобільних клієнтів.
6. Мобільний клієнт.
7. Центр моніторингу.
8. Програма контролю цілісності.

Шлюз із модулем, що кодує/декодувальним

Шлюз є основним модулем комплексу, що виконує функції маршрутизації, фільтрації й кодування пакетів. Кожний Шлюз призначений для закриття визначеної групи локальних мереж. На комп'ютері-шлюзі встановлюється ядерний модуль с функціями кодування й декодування й запускається програма автентифікації. Функціями шлюзу є:

- Фільтрація трафіку (розподіл на кодуємий/некодуємий потоки).
- Кодування трафіку (кодуємий потік).
- Автентифікація з іншими Шлюзами.
- Реєстрація подій у Центрі моніторингу.
- Забезпечення власного захисту.

Центр сертифікації та розподілу ключів

Центр сертифікації та розподілу ключів здійснює управління контуром безпеки, а також виконує наступні функції:

- Одержання зі змінного носія відкритих ключів Шлюзів.

– Видачу будь-якому Шлюзу відкритих ключів будь-яких інших Шлюзів і інформації про відповідні сегменти структури мережі.

– Розсилання Шлюзам повідомлень про зміни структури закритої мережі.

– Вироблення і виконання процедури зміни сеансових ключів.

– Зберігання інформації про структуру мережі.

Центр реалізований у вигляді програмного комплексу, що виконує функції зберігання й видачі відкритих ключів кодування по мережевому запиті від модулів кодування. Центр сертифікації та розподілу ключів може бути встановлений або на окремому (виділеному) комп'ютері, або разом з одним зі Шлюзів кодування.

Центр генерації ключів

Даний модуль служить для генерації пар комплементарних ключів, а також є репозитарієм всіх відомих системі ключів. У функції Центра генерації ключів входить:

– генерація пар відкритого й секретного ключів модулів, що кодують;

– генерація пари ключів для сертифікації (еталонного завірення) відкритих ключів модулів, що кодують;

– генерація сертифікатів відкритих ключів, підписаних секретним ключем сертифікації;

– приміщення підписаних сертифікатів відкритих ключів на змінні носії;

– зберігання еталонних копій сертифікованих відкритих ключів в архіві.

Центр генерації ключів – програма, що виконується на ізольованому автоматизованому робочому місці.

Рисунок 6 – Структурна схема системи

Центр реєстрації ключів

Центр реєстрації ключів служить репозитарієм всіх відомих системі ключів. У його функції входить:

– Введення зі змінного носія відкритого ключа.

– Введення зі змінного носія закритого ключа Адміністратора безпеки.

– Підпис нового ключа ключем Адміністратора безпеки.

– Приміщення підписаного відкритого ключа в архів довгострокового зберігання й на змінний носій.

– Зберігання еталонних копій сертифікованих (зареєстрованих) відкритих ключів.

Центр реєстрації ключів виконаний у вигляді програми, що виконується на ізольованому автоматизованому робочому місці й призначеної для сертифікації (еталонного завірення) відкритих ключів.

Центр реєстрації мобільних клієнтів і Мобільний клієнт

Для забезпечення доступу до корпоративних даних, які захищаються, мобільних абонентів, не підключених до локальних мереж, які захищаються, використовується Центр реєстрації мобільних клієнтів і програмне забезпечення мобільного клієнта комплексу.

Центр реєстрації мобільних клієнтів являє собою спеціальний модуль, що кодує, для підключення довільної кількості мобільних клієнтів.

Мобільний клієнт являє собою програмний модуль, що працює під управлінням ОС Windows і використовуючий апаратні ключі для автентифікації абонента в VPN.

Центр моніторингу

Центр моніторингу являє собою мережеве автоматизоване робоче місце із установленим на ньому набором програм, що здійснюють збір і аналіз протоколів, що надходять від всіх модулів комплексу.

Програма контролю цілісності

Комплекс містить у собі засобу формування й перевірки контрольних сум файлів. Ці засоби реалізовані у вигляді Програми контролю цілісності, що призначена для визначення й повідомлення системного Адміністратора про зміну, додавання й видалення файлів.

Адміністрування комплексу

Настроювання й адміністрування компонентів комплексу здійснюється централізовано з робочого місця Адміністратора безпеки за допомогою графічного інтерфейсу або командного рядка. Віддалене управління здійснюється по захищеному каналі. Комплекс забезпечує автентифікацію Адміністраторів і розмежування доступу до функцій адміністрування.

Основні особливості комплексу

– Основними особливостями розробленої системи є: Повнофункціональна схема управління ключами, що дозволяє здійснювати динамічний розподіл ключів з використанням довіреного центра сертифікації, перевірку дійсності ключової інформації й оповіщення систем кодування про компрометацію ключів; Висока надійність функціонування, забезпечувана засобами контролю цілісності, протоколювання й аудита, стійкості до збоїв і відновлення у випадку збоїв і відмов; Прозорість кодування переданих даних для абонентів і використовуваного ними програмного забезпечення; Висока продуктивність (робота в мережі 100 Мбіт/с без істотного впливу на пропускну здатність); Забезпечення необхідної якості сервісу (QoS) і підтримка роботи із сервісами, що пред'являють високі вимоги до величин тимчасових затримок (IP-телефонія, відеоконференцзв’язок); Можливість використання в комплексі з міжмережевими екранами, антивірусними рішеннями й засобами контекстного аналізу; Використання відкритих стандартів – протокол тунелювання мережевих пакетів відповідає стандартам IETF IPsec.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів хмарних сервісів з використанням ЦСК.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем хмарних сервісів з використанням ЦСК.

– Досліджена система хмарних сервісів з використанням ЦСК.

– На основі отриманих результатів досліджень створена програмна реалізація системи хмарних сервісів з використанням ЦСК.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання хмарних сервісів з використанням ЦСК.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

Список літератури

1. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
2. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
3. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
4. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
5. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
6. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
7. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
8. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
9. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
10. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
11. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
12. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
13. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
14. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).
15. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019. (Scopus).
16. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629. (Scopus).
17. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 873-884. (Scopus).
18. Smirnov, O., Kuznetsov, A., Prokopovych-Tkachenko, D. «Hiding Data in Images Using a Pseudo-Random Sequence». ISCI'2020: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko, Victor A. Krasnobayev and Alexandr A. Kuznetsov. ASC Academic Publishing, USA, 2020. pp. 46-59. – ISBN: 978-1-7362833-0-1 (Hardback), ISBN: 978-1-7362833-1-8 (Ebook).

19. Smirnov, O., Kuznetsov, A., Shekhanin, K., Chepurko, I. Detecting Hidden Information in FAT. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 412-429. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
20. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

УДК 004

Р.Ковтуненко, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ХМАРНОГО СЕРВІСУ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ У НАВЧАЛЬНОМУ ЗАКЛАДІ

У статті програмне забезпечення, яке призначено для системи хмарного сервісу електронної бібліотеки у навчальному закладі. Метою розробки є дослідження та програмна реалізація системи хмарного сервісу електронної бібліотеки у навчальному закладі. Об'єктом дослідження є процес хмарного сервісу електронної бібліотеки у навчальному закладі. Предметом дослідження є методи хмарного сервісу електронної бібліотеки у навчальному закладі. Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи хмарного сервісу електронної бібліотеки у навчальному закладі. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, хмарний сервіс, електрона бібліотека

Постановка проблеми. Темпи створення й нагромадження інформації, ускладнення й глобалізація знання зробили необхідним пошук інструментів, що дозволяють забезпечити швидкий і ефективний доступ до цього знання, незалежно від різних країн і різних сховищ інформації. Одним з таких інструментів по праву вважаються технології електронних бібліотек (ЕБ), розвиток яких почалося на початку 90-х років XX століття. Електронні бібліотеки як напрямок розвитку електронних ресурсів багато в чому визначає політику бібліотек при плануванні своєї діяльності у формуванні сучасного автоматизованого бібліотечно-інформаційного середовища.

У цей час активно розвиваються електронні бібліотеки ЗВО різних профілів, що у свою чергу створює об'єктивні передумови для підвищення рівня утворення. Ресурси, розміщені в електронних бібліотеках ЗВО, здатні істотно вплинути на інтенсивність процесів навчання й наукових досліджень, а «забезпечення публічного (у тому числі віддаленого) доступу до них стало однієї з першочергових задач обслуговування утворення, науки й культури. Сьогодні загальновизнано, що рішення цієї задачі найбільше ефективно досягається шляхом створення електронних бібліотек» [1]. У цьому змісті університетське середовище є найбільш оптимальним для використання існуючих, створення нових інформаційних ресурсів, розвитку нових інформаційних і комунікаційних технологій, тому що саме у ЗВО одночасно й у різних формах, у навчанні й наукових дослідженнях створюються й використовуються такі інформаційні ресурси й технології. Створювані в університетах інформаційні ресурси мають різну природу – це наукові видання й навчально-методичні посібники, дисертації й автореферати, бібліографічні покажчики й огляди, довідкова література, матеріали теле– і відеоконференцій, електронні журнали й електронні

версії «паперових» наукових видань, електронні підручники, наукові бази даних і ще багато чого іншого.

Таким чином, на сьогоднішній день використання електронних бібліотек дозволяє вирішити проблему обслуговування віддалених і локальних користувачів на глобальному рівні. Але разом з тим стає очевидним протиріччя між наявністю величезних масивів інформації, використовуваних в електронній формі, і відсутністю ефективних інструментів її структурування. Дане протиріччя повною мірою виявилося в значній частині проектів по створенню електронних бібліотек, реалізованих як в Україні, так і за рубежем. У цих проектах використовуються технології засобів комунікацій і забезпечення доступу до різних інформаційних ресурсів, але не акцентується увага на проблемі структурування інформації. У той же час споживач інформації ще не до кінця усвідомлює всіх тих можливостей системи інформаційно-бібліотечного обслуговування, які надають сучасні інформаційні технології.

Тому виникає необхідність створення такого інструмента, що дав би можливість користувачеві самому визначитися в інформаційному поводженні, обробці масивів інформації й ін.

Впровадження різних методів доступу, як нам представляється, повинне йти паралельно з розвитком методів, що дають можливість користувачам засвоювати різні обсяги інформації. Ці тенденції повинні враховуватися при реалізації проектів, пов'язаних з електронними бібліотеками. Однією з першочергових задач побудови моделей електронних бібліотек ЗВО, на наш погляд, повинна бути розробка засобів ідентифікації й класифікації об'єктів ЕБ, інтерфейсу запитів до її ресурсів.

Розглянута нами проблема в останні роки стала активно підніматися науковою громадськістю. Сказане повною мірою підтверджується тим, що сьогодні виникає гостра необхідність використання в наукових і соціально-культурних цілях можливостей, надаваних сучасними інформаційними технологіями й засобами телекомунікацій; у ЗВО з'являється можливість акумулювання унікальної як навчально-методичної, так і наукової інформації, «породженої» у цьому ЗВО. Рішення названих проблем дозволить задовольнити зрослі вимоги до надаваної інформації, а також інформаційні потреби студентів, аспірантів, науковців ЗВО в освітньому процесі й у науковій діяльності.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи хмарного сервісу електронної бібліотеки у навчальному закладі.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи хмарного сервісу електронної бібліотеки у навчальному закладі.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем хмарного сервісу електронної бібліотеки у навчальному закладі.

– Дослідження системи хмарного сервісу електронної бібліотеки у навчальному закладі.

– Програмна реалізація системи хмарного сервісу електронної бібліотеки у навчальному закладі.

Об’єктом дослідження є процес хмарного сервісу електронної бібліотеки у навчальному закладі.

Предметом дослідження є методи хмарного сервісу електронної бібліотеки у навчальному закладі.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Опис вимог до електронної бібліотеки кафедри кібербезпеки та програмного забезпечення

Функціональні вимоги визначають базу для розробки технічного завдання на створення програмного забезпечення для розподіленої системи системи хмарного сервісу електронної бібліотеки у навчальному закладі.

Загальні вимоги до СХСЕБНЗ

СХСЕБНЗ (система хмарного сервісу електронної бібліотеки у навчальному закладі) являє собою систему, що забезпечує кінцевому користувачеві можливість віддаленої роботи з електронними копіями повних текстів документів.

СХСЕБНЗ складається з Локальних Електронних бібліотек (будь-якої кількості, у т.ч. однієї) і центрального вузла електронної бібліотеки.

Електронна бібліотека кафедри кібербезпеки та програмного забезпечення являє собою окрему систему Електронної бібліотеки й може функціонувати в повністю автономному режимі, включаючи власні канали зв'язку й біллінгову систему.

Для забезпечення централізованого доступу кінцевих користувачів Локальні Електронні бібліотеки можуть бути об'єднані (за допомогою центрального вузла СХСЕБНЗ) у єдину розподілену систему СХСЕБНЗ. При цьому, при включенні окремої Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення у розподілену систему СХСЕБНЗ, за нею зберігається і її власна автономність (тобто доступ до Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення може здійснюватися як прямо, так і через центральний вузол розподіленої системи).

Програмне забезпечення Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення складається із програмного забезпечення:

– Електронного фонду.

– Електронного каталогу.

– Віртуального читального залу**.**

Програмне забезпечення Розподіленої системи СХСЕБНЗ складається з:

– Програмного забезпечення Локальної Електронної бібліотеки.

– WEB-інтерфейсу кінцевого користувача для доступу до Розподіленої системи СХСЕБНЗ.

– Адміністративного інтерфейсу розподіленої системи СХСЕБНЗ.

При цьому WEB-інтерфейс кінцевого користувача може бути встановлений, як для доступу до Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення, так і в центральному вузлі розподіленої системи. Після установки програмного забезпечення Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення і WEB-інтерфейсу кінцевого користувача отримана конфігурація повинна забезпечувати повне функціонування Електронної бібліотеки для окремо взятої бібліотеки.

Адміністративний інтерфейс розподіленої системи СХСЕБНЗ установлюється на один сервер і управляє всією розподіленою системою електронної бібліотеки.

Після установки програмного забезпечення Розподіленої системи СХСЕБНЗ будь-яка Електронна бібліотека кафедри кібербезпеки та програмного забезпечення може бути підключена до розподіленої системи СХСЕБНЗ і отримана конфігурація повинна забезпечувати повне функціонування СХСЕБНЗ.

Вимоги до параметрів функціонування системи

При одночасній роботі 100 кінцевих користувачів СХСЕБНЗ час реакції системи при пошуку інформації в Електронному каталозі не повинне перевищувати 2 сек., при доступі до сторінки електронного документа – 1 сек. Під часом реакції розуміється час, що пройшов між натисканням кнопки “Пошук” і одержанням результату на екран без обліку швидкості передачі інформації з мережі. Необхідна для цього конфігурація встаткування й стандартного програмного забезпечення повинна бути визначена при розробці технічного завдання.

Максимальний час відгуку системи при будь-якій кількості користувачів не повинний перевищувати 10 сек.

Вимоги до системи підготовки звітів

Система (модуль) підготовки звітів повинна бути присутнім у кожному Адміністративному інтерфейсі. Перелік стандартних звітів для кожного Адміністративного інтерфейсу повинен бути визначений при розробці технічного завдання. Система підготовки звітів повинна передбачати процедуру додавання нового типу звіту за допомогою використання скриптів MySQL.

Функціональні вимоги до системи Електронного фонду (ЕФ)

Електронний фонд Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення складається з повнотекстових електронних документів. Під електронним документом розуміється файл (або файли), що містять текст закінченого добутку. Крім того, у системі ЕФ зберігаються облікові записи на:

– Електронний документ.

– Колекцію електронних документів.

– Добірку електронних документів.

Під колекцією розуміється об'єднання електронних документів з метою здійснення над ними групових операцій. Наприклад, колекція електронних документів у текстовому форматі TXT. Операція: переклад у формат DOC. Або: колекція електронних дисертацій по програмуванню. Операція: запис на DVD.

Під добіркою розуміється об'єднання електронних документів по певній загальній ознаці (змістовному або формальному). Наприклад, добірка електронних документів, що складає з дисертації, автореферату дисертації, відкликань на дисертацію, документів, записаних на одному носії й т.п. Навіщо потрібна добірка? Наприклад, для проведення дій “усередині” добірки: вибору з автореферату ключових слів, додавання їх до бібліографічного опису дисертації, перезапису декількох CD на DVD і т.п.

Таблиця 1 – Специфікація електронного фонду

N	Найменування	Функціональна вимога
1.	Об'єм електронного документа	Забезпечується зберігання розпізнаних або нерозпізнаних електронних документів при об'ємі одного документа до 10 ГБ. Повний перелік операцій над електронними документами повинен бути визначений при розробці технічного завдання.
2.	Формат електронного документа	Забезпечується підтримка найпоширеніших форматів (графічних, текстових, мультимедійних). Повний перелік форматів електронних документів повинен бути визначений при розробці технічного завдання.
3.	Строк зберігання електронного документа	“Вічно”. Повинне забезпечуватися як завгодно довге зберігання тексту документа, що передбачає конвертування (при зміні програмного забезпечення відтворюючий документ), перезапис (при зміні технічних пристроїв), захист від несанкціонованого видалення, захист від втрати даних при програмних і апаратних збоях.
4.	Фрагментація електронного документа	Забезпечується можливість добування фрагмента електронного документа (абзацу, сторінки, глави) у тому числі й фрагмента неформатованого й/або бінарного тексту для наступної обробки й/або передачі по мережах.
5.	Фіксований URL	Забезпечується незмінність URL-адреси електронного

		документа при фізичному переміщенні електронного документа усередині електронного фонду (наприклад, при переносі на інший носій або сервер)
6.	Цілісність електронного документа	Забезпечується робота з електронним документом, як з єдиним цілим, тобто при зміні фізичного місця положення в електронному фонді одночасно змінюється положення всіх складових частин документа.
7.	Обліковий запис на електронний документ	Кожний електронний документ повинен мати обліковий запис, що містить основні характеристики електронного документа як одиниці інформації в електронному фонді. Обліковий запис – інформація що доповнює бібліографічний запис в електронному каталозі. Обліковий запис повинна містити унікальний номер документа, його URL, дату включення в електронний фонд, дату останньої модифікації й інших атрибутів. Повний перелік інформації в обліковому записі повинен бути визначений при розробці технічного завдання.
8.	Загальний об'єм	Необмежений. Лімітується тільки об'ємом зовнішньої пам'яті.
9.	Фрагментація електронного фонду	Забезпечується можливість об'єднання будь-якої кількості електронних документів у колекцію або добірку. При цьому: один електронний документ може входити в різні колекції, і колекція може містити інші колекції; один електронний документ може входити тільки в одну добірку й добірка не може містити інших добірок.
10.	Колекція електронних документів	Дозволяє проводити групові операції над вхідними в неї електронними документами. Наприклад, переміщення на інший носій, конвертування в інший формат. Повний перелік операцій над колекціями електронних документів повинен бути визначений при розробці технічного завдання.
11.	Добірка електронних документів	Дозволяє проводити операції усередині добірки й здійснювати відновлення даних в Електронном каталозі на базі інформації, що втримується в електронних документах добірки. Повний перелік операцій над добірками повинен бути визначений при розробці технічного завдання.
12.	Права доступу по фрагментах Електронного фонду	Забезпечується призначення прав доступу до наступних фрагментів ЕФ: 1. До всього електронного фонду. 2. До колекції (до будь-якого документа усередині колекції). 3. До добірки електронних документів (до будь-якого документа усередині добірки). 4. До електронного документа.
13.	Права доступу до фрагмента Електронного фонду	Мінімальний набір прав доступу до фрагмента ЕФ для кінцевих користувачів: 1. Дозволено доступ для всіх користувачів. 2. Дозволено доступ для всіх користувачів будь-якого

		віртуального читального залу (ВЧЗ). 3. Дозволено доступ тільки для користувачів з обмеженого списку ВЧЗ. 4. Дозволено доступ тільки для декількох груп з конкретного ВЧЗ. 5. Дозволено доступ тільки конкретному кінцевому користувачеві/користувачам.
14.	Статистика на рівні електронного документа й на рівні користувача	При звертанні до електронного фонду ведеться статистика звертань до кожного електронного документа. Фіксується: користувач, дата й час доступу, операція, об'єм викликаної інформації. Повний перелік показників індивідуальної статистики повинен бути визначений при розробці технічного завдання
15.	Групова статистика	Забезпечується статистика звертань до добірок, колекціям і електронному фонду в цілому. Повний перелік показників групової статистики повинен бути визначений при розробці технічного завдання.
16.	Протоколи взаємодії з ЕК і ВЧЗ	Розробляються протоколи взаємодії системи Електронного фонду із системою Електронного каталогу й системою Віртуального читального залу (ВЧЗ). Повний перелік специфікацій протоколу повинен бути визначений при розробці технічного завдання.
17.	Експорт/Імпорт електронних документів	Імпорт електронних документів в Електронний фонд здійснюється тільки одночасно з бібліографічними записами або при їхній наявності. Експорт електронних документів може здійснюватися як з вивантаженням бібліографічних записів з електронного фонду, так і без них.
18.	Адміністративний інтерфейс до системи Електронного фонду	Забезпечується WEB-інтерфейс адміністратора, що дозволяє виконувати всі функції по керуванню Електронним фондом. Повний перелік адміністраторських функцій повинен бути визначений при розробці технічного завдання.
19.	API до системи Електронного фонду	Забезпечується програмний інтерфейс, що дозволяє програмно виконувати всі функції адміністративного інтерфейсу, а так само розширені функції. Повний перелік розширених функцій повинен бути визначений при розробці технічного завдання.

Для кінцевих користувачів до електронних документів, колекціям і добіркам доступ надається через програмне забезпечення **Віртуального читального залу**.

Програмне забезпечення системи Електронного фонду повинне задовольняти нижче наведеним вимогам.

Електронний документ – обмежений і завершений на конкретний момент часу масив інформації, зафіксований на фізичному носії(ях) у вигляді файлу (набору файлів) з єдиними технічними й загальними змістовними характеристиками.

Функціональні вимоги до системи Електронного каталогу (ЕК)

Таблиця 2 – Специфікація електронного каталогу

N	Найменування	Функціональна вимога
	Бібліографічний запис на електронний документ	Запис у форматі заданому в ЕБ.
	Цілісність зв'язку бібліографічного запису з електронним документом	Жодна операція над бібліографічним записом не повинна приводити до розриву зв'язку з електронним документом.
	Відновлення бібліографічного запису	Забезпечується процедура внесення змін у бібліографічний запис на основі інформації з електронного документа або добірки електронних документів. Повний перелік можливих відновлень повинен бути визначений при розробці технічного завдання.
	Запис на колекцію (опис колекції) електронних документів	Перелік полів повинен бути визначений при розробці технічного завдання.
	Запис на добірку електронних документів	Перелік полів повинен бути визначений при розробці технічного завдання.
	Стандартні операції в системі Електронного каталогу	Система ЕК повинна забезпечувати набір стандартних операцій, характерних для роботи з електронним каталогом: GUI електронного каталогу, OPAC і/або WEB-OPAC. Повний перелік стандартних операцій повинен бути визначений при розробці технічного завдання.
	Експорт/Імпорт бібліографічних записів	При імпорті бібліографічних записів повинна забезпечуватися можливість одночасного додавання електронних документів в Електронний фонд. Експорт бібліографічних записів також може здійснюватися як з вивантаженням даних з електронного фонду, так і без них.
	Конвертування бібліографічної інформації	При виконанні операцій експорту/імпорту бібліографічних записів повинна забезпечуватися можливість підключення конверторів.
	Налаштування конвертерів	Налаштування конвертерів здійснюється на трьох рівнях. 1. На рівні програми, шляхом зміни коду. При цьому необхідна гарна документація на код. 2. На рівні створення скриптів, що підключаються під час роботи програми. При цьому потрібна документація на мову, використовувана для скриптів і методи їхнього підключення. 3. На рівні параметрів, що набудовуються через інтерфейс адміністратора.
	Налаштування конверторів	Забезпечується можливість внесення змін у конвертори без зупинки роботи системи.
	Пошуковий інтерфейс кінцевого користувача	При введенні запиту користувач заповнює дані в одному блоці. Наприклад: Автор: Петров

		Ключові слова: програмування Слова в тексті: функції MySQL У результаті пошуку вертається список бібліографічних записів, у яких у самому бібліографічному записі є автор “Петров” і ключове слово “програмування”, а в тексті електронного документа зустрічається “функції MySQL”.
	Стандартні функції контекстного пошуку	Забезпечується функціональність не нижче пошукових можливостей системи “Yandex”. Повний перелік функцій повинен бути визначений при розробці технічного завдання.
	Сервер Z39.50	Повинен забезпечуватися доступ до Електронного каталогу по протоколу Z39.50. Адміністрування сервера повинне бути доступно через Адміністративний інтерфейс до системи Електронного каталогу.
	Адміністративний інтерфейс до системи Електронного каталогу	Забезпечується WEB-інтерфейс адміністратора, що дозволяє виконувати всі функції по керуванню Електронним каталогом. Повний перелік адміністраторських функцій повинен бути визначений при розробці технічного завдання.
	MySQL до системи Електронного каталогу	Забезпечується програмний інтерфейс, що дозволяє програмно виконувати всі функції адміністративного інтерфейсу, а так само розширені функції. Повний перелік розширених функцій повинен бути визначений при розробці технічного завдання.

Електронний каталог містить бібліографічну інформацію про електронні документи, забезпечує основну функціональність по завантаженню, вивантаженню й відновленню записів, пошуку інформації й доступу до даних. Система Електронного каталогу підтримує можливість двох варіантів пошуку:

–Пошук в Електронному каталозі по елементах бібліографічного опису.

–Пошук в розпізнаних електронних документах.

Реалізація контекстного пошуку може бути виконана в рамках окремої підсистеми, однак пошуковий інтерфейс кінцевого користувача для обох варіантів повинен бути єдиним.

Функціональні вимоги до системи Віртуального читального залу

Таблиця 3 – Специфікація системи Віртуального читального залу

N	Найменування	Функціональна вимога
	Віртуальний читальний зал (ВЧЗ)	Реєстрація віртуального читального залу забезпечує можливість реєстрації кінцевих користувачів і їхніх груп. Одночасно з реєстрацією ВЧЗ реєструється адміністратор ВЧЗ і забезпечується доступ адміністратора ВЧЗ до адміністративних функцій ВЧЗ. Повний перелік полів, що описують ВЧЗ і повний список функцій для роботи із ВЧЗ, повинен бути визначений при розробці технічного завдання.
	Протокол взаємодії	Розробляється протокол взаємодії Web-інтерфейсу із системою Електронного фонду й Віртуального читального залу. Повний перелік специфікацій протоколу повинен бути визначений при розробці технічного завдання.
	Вхід у систему ВЧЗ	Кінцевий користувач реєструється в системі ВЧЗ у момент першого звертання до електронного документа з обмеженим доступом.
	Кінцевий користувач	Користувач (читач) віртуального читального залу. Повний перелік полів, що описують користувача, повинен відповідати

		списку полів прийнятому при розробці технічного завдання. Користувач реєструється адміністратором ВЧЗ із одночасним призначенням прав на доступ до даних Локальної Електронної бібліотеки. Користувачі можуть поєднуватися в групи з однаковими правами. Забезпечується призначення прав для груп користувачів. Повний перелік прав користувачів і їхніх груп повинен бути визначений при розробці технічного завдання.
	Права кінцевого користувача	Система ВЧЗ забезпечує, як мінімум: 1. Доступ тільки на читання. 2. Доступ із правом копіювання частини електронного документа. Повний перелік прав доступу повинен бути визначений при розробці технічного завдання
	Журнал звертань кінцевого користувача	На кожного користувача ведеться журнал звертань до ресурсів Локальної Електронної бібліотеки. Мінімальний запис у журналі повинна містити: дату, час, ID електронного документа, кількість сторінок документа до яких отриманий доступ. Повний перелік атрибутів запису в журналі повинен бути визначений при розробці технічного завдання. Адміністратор ВЧЗ має доступ до журналу звертань кінцевого користувача тільки для читання.
	Групова статистика	Забезпечується збір статистики за встановлюваний період часу по окремому користувачі, групі користувачів, ВЧЗ. Повний перелік показників групової статистики повинен бути визначений при розробці технічного завдання.
	Підсистема білінгу	Забезпечує призначення ціни на доступ до Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення і проведення розрахунків по оплаті доступу.
	Призначення ціни	Підтримується три типи розцінок: 1. За час дії договору на створення ВЧЗ. 2. За отриманий обсяг інформації. 3. За надану послугу.
	Схема розрахунків	Схема розрахунків повинна включати можливість призначення загальної ціни на доступ до розподіленої системи СХСЕБНЗ і подальшого внутрішнього розподілу отриманих засобів між Локальними Електронними бібліотеками пропорційно обсягу інформації, отриманої ВЧЗ від Локальної Електронної бібліотеки. Повний перелік алгоритмів розрахунку повинен бути визначений при розробці технічного завдання.
	Підсистема білінгу ВЧЗ	Підсистема білінгу ВЧЗ забезпечує роботу ВЧЗ по розрахунках з кінцевими користувачами.
	Прозорість підсистем білінгу	Всі дані по розрахунках усередині ВЧЗ повинні бути доступні підсистемі білінгу Локальної Електронної бібліотеки, всі даної підсистеми білінгу Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення повинні бути доступні Центральній системі білінгу розподіленої системи СХСЕБНЗ.
	Підсистема моніторингу	Забезпечує візуальний моніторинг поточного стану доступу до Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення, дозволяє оперативно втручатися в роботу

		окремого ВЧЗ аж до його відключення. Повний перелік показників підсистеми моніторингу й повний перелік керуючих функцій повинен бути визначений при розробці технічного завдання.
	Адміністративний інтерфейс до системи ВЧЗ	Забезпечується WEB-інтерфейс адміністратора, що дозволяє виконувати всі функції по керуванню системою ВЧЗ. Повний перелік адміністраторських функцій повинен бути визначений при розробці технічного завдання.

Система Віртуального читального залу забезпечує:

–Доступ кінцевих користувачів до Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення.

–Призначення прав доступу для кінцевих користувачів.

–Підсистему білінгу для розрахунку вартості доступу.

Специфікації функціональних вимог до системи Віртуального читального залу наведені у таблиці 3.

Адміністративний інтерфейс розподіленої системи СХСЕБНЗ (AI PC СХСЕБНЗ)

Адміністративний інтерфейс складається із трьох частин:

–Системи функцій розподіленою системою, що забезпечує керування, СХСЕБНЗ.

–Системи функцій інтегруючої Адміністративні інтерфейси Електронного фонду, Електронного каталогу, Віртуального читального залу, Web-інтерфейсу кінцевого користувача.

–Центральної білінгової системи.

Розробка структурної схеми

На рисунку 1 зображена структурна схема розроблюємої системи. Зі структурної схеми ми бачимо, що електронна бібліотека кафедри кібербезпеки та програмного забезпечення входить у якості локальної електронної бібліотеки до електронної бібліотеки вищого навчального закладу.

Тому на схемі окрім електронної бібліотеки вищого навчального закладу та електронної бібліотеки кафедри кібербезпеки та програмного забезпечення входять інші локальні електронні бібліотеки.

–Початок роботи з електронною бібліотекою кафедри кібербезпеки та програмного забезпечення починається з звернення до WEB-інтерфейсу кінцевого користувача для доступу до Розподіленої системи електронної бібліотеки вищого навчального закладу. Через цей WEB-інтерфейс з електронною бібліотекою вищого навчального закладу працює

Адміністративний інтерфейс розподіленої системи СХСЕБНЗ установлюється на один сервер і управляє всією розподіленою системою електронної бібліотеки. Після установки програмного забезпечення Розподіленої системи СХСЕБНЗ будь-яка Електронна бібліотека кафедри кібербезпеки та програмного забезпечення може бути підключена до розподіленої системи СХСЕБНЗ і отримана конфігурація повинна забезпечувати повне функціонування СХСЕБНЗ.

Адміністративний інтерфейс електронної бібліотеки кафедри кібербезпеки та програмного забезпечення включає в себе наступні блоки:

– Модуль вибору мови, де відбувається вибір з двох мов: української та англійської.

– Система керування розподіленою системою електронної бібліотеки, призначеною для реалізації мінімального набору функцій, що повинен дозволяти підключити/відключити Локальну Електронну бібліотеку до розподіленої системи й дістати права на керування цією Локальною електронною бібліотекою через систему функцій інтегрального інтерфейсу.

– Модуль підготовки звітів, призначений для підготовки стогових звітів по користувачам, та по бібліотеці. Система (модуль) підготовки звітів повинна бути присутнім у кожному Адміністративному інтерфейсі. Перелік стандартних звітів для кожного

Адміністративного інтерфейсу повинен бути визначений при розробці технічного завдання. Система підготовки звітів повинна передбачати процедуру додавання нового типу звіту за допомогою використання скриптів MySQL.

Перейдемо до розгляду елементів електронної бібліотеки кафедри кібербезпеки та програмного забезпечення, які підключені до адміністративного модулю. Адміністративний інтерфейс електронної бібліотеки кафедри кібербезпеки та програмного забезпечення, який з іншого боку працює с такими елементами:

– Електронний фонд.
– Електронний каталог.
– Віртуальний читальний зал.

Рисунок 1 – Структурна схема

Електронний фонд Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення складається з повнотекстових електронних документів. Під електронним документом розуміється файл (або файли), що містять текст закінченого добутку. Крім того, у системі ЕФ зберігаються облікові записи на:

–Електронний документ.

–Колекцію електронних документів.

–Добірку електронних документів.

Під колекцією розуміється об'єднання електронних документів з метою здійснення над ними групових операцій. Наприклад, колекція електронних документів у текстовому форматі TXT. Операція: переклад у формат DOC. Або: колекція електронних дисертацій по програмуванню. Операція: запис на DVD.

Під добіркою розуміється об'єднання електронних документів по певній загальній ознаці (змістовному або формальному). Наприклад, добірка електронних документів, що складає з дисертації, автореферату дисертації, відкликань на дисертацію, документів, записаних на одному носії й т.п. Навіщо потрібна добірка? Наприклад, для проведення дій "усередині" добірки: вибору з автореферату ключових слів, додавання їх до бібліографічного опису дисертації, перезапису декількох CD на DVD і т.п.

Крім того реалізується модуль реалізації прав доступу, який Забезпечується призначення прав доступу до наступних фрагментів ЕФ:

1. До всього електронного фонду.
2. До колекції (до будь-якого документа усередині колекції).
3. До добірки електронних документів (до будь-якого документа усередині добірки).
4. До електронного документа.

Реалізовано також права доступу до фрагмента Електронного фонду:

1. Дозволено доступ для всіх користувачів.
2. Дозволено доступ для всіх користувачів будь-якого віртуального читального залу (ВЧЗ).
3. Дозволено доступ тільки для користувачів з обмеженого списку ВЧЗ.
4. Дозволено доступ тільки для декількох груп з конкретного ВЧЗ.
5. Дозволено доступ тільки конкретному кінцевому користувачеві/користувачам.

Ще реалізовано модуль статистики на рівні електронного документа й на рівні користувача, який дозволяє реалізувати наступну фнкцію. При звертанні до електронного фонду ведеться статистика звертань до кожного електронного документа. Фіксується: користувач, дата й час доступу, операція, об'єм викликаної інформації.

Другим глобальним модулем електронної бібліотеки кафедри кібербезпеки та програмного забезпечення, є модуль електронного каталогу.

Електронний каталог містить бібліографічну інформацію про електронні документи, забезпечує основну функціональність по завантаженню, вивантаженню й відновленню записів, пошуку інформації й доступу до даних. Система Електронного каталогу включає наступні модулі:

– Модуль пошуку по тексту.

– Модуль пошуку по автору.

– Модуль пошуку по категорії.

– Модуль пошуку по найменуванню.

Призначення цих модулів, виходить з їхньої назви, тому більш детально на них ми зупинятися не будемо.

І останнім структурним модулем електронної бібліотеки кафедри кібербезпеки та програмного забезпечення є Віртуальний читальний зал.

Реєстрація віртуального читального залу забезпечує можливість реєстрації кінцевих користувачів і їхніх груп. Одночасно з реєстрацією ВЧЗ реєструється адміністратор ВЧЗ і забезпечується доступ адміністратора ВЧЗ до адміністративних функцій ВЧЗ. Повний перелік полів, що описують ВЧЗ і повний список функцій для роботи із ВЧЗ, був визначений при розробці технічного завдання.

Система Віртуального читального залу забезпечує:

–Доступ кінцевих користувачів до Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення.

–Призначення прав доступу для кінцевих користувачів.

–Підсистему білінгу для розрахунку вартості доступу.

Система складається з наступних модулів:

– Модуль входу у систему ВЧЗ. Він виконує наступні функції. Користувач за допомогою логіну та паролю, отримує доступ до системи, переходить до модулю реєстрації.

– Модуль реєстрації. Кінцевий користувач реєструється в системі ВЧЗ у момент першого звертання до електронного документа з обмеженим доступом.

– Модуль видачі прав доступу. Система ВЧЗ забезпечує:

1. Доступ тільки на читання.
2. Доступ із правом копіювання частини електронного документа.

– Модуль статистики. Забезпечується збір статистики за встановлюваний період часу по окремому користувачі, групі користувачів, ВЧЗ.

– Модуль білінгу. Забезпечує призначення ціни на доступ до Електронної бібліотеки кафедри кібербезпеки та програмного забезпечення і проведення розрахунків по оплаті доступу, для користувачів які не мають права бесплатного доступу.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів хмарного сервісу електронної бібліотеки у навчальному закладі.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем хмарного сервісу електронної бібліотеки у навчальному закладі.

– Досліджена система хмарного сервісу електронної бібліотеки у навчальному закладі.

– На основі отриманих результатів досліджень створена програмна реалізація системи хмарного сервісу електронної бібліотеки у навчальному закладі.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання хмарного сервісу електронної бібліотеки у навчальному закладі.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Смирнов А.А. Анализ и сравнительное исследование перспективных направлений развития цифровых телекоммуникационных систем и сетей / А.А.Смирнов, В.В.Босько, Е.В.Мелешко // Системи обробки інформації. – Х.: ХУ ПС, 2008. – Вип.7(74). – С.120-123.
2. Смирнов А.А. Усовершенствование метода управления очередями в многопротокольных узлах телекоммуникационной сети / А.А.Смирнов, Е.В.Мелешко // Збірник тез та доповідей другої всеукраїнської науково-практичної конференції «Системний аналіз. Інформатика. Управління». Запоріжжя. Тези доповідей. Запоріжжя: КПУ, 2011.
3. Смирнов С.А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А.К. Дидык, С.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.
4. Смирнов С. А. Сравнительные исследования математических моделей технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, А. В. Коваленко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2014. – Вип. 9(125). – 105-110.
5. Смирнов С. А. Математическая модель интеллектуального узла коммутации с обслуживанием информационных пакетов различного приоритета / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2014. – Вип. 4 (41). – С. 48-52.

6. Смирнов С. А. Исследование показателей качества функционирования интеллектуальных узлов коммутации в телекоммуникационных системах и сетях / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2014. – № 4(17). – С. 90-95.
7. Смирнов С. А. Усовершенствованный алгоритм управления доступом к «облачным» телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2015. –Вип. 1(126). – С. 150-153.
8. Smirnov S.A. Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam, S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.
9. Смирнов С. А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты даннях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2015. – № 3(43). – С. 100-107.
10. Смирнов С. А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2015. –№ 3(20). – С. 134-141.
11. Смирнов С. А. Комплекс gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Безпека інформації: наук. – практ. журн. – К.: НАУ, 2015. – Т. 21, № 3. – С. 251-262.
12. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, А. К. Дидык, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2016. – № 2 (46). – С. 146-149.
13. Смирнов С. А. Модели системы нейросетевых экспертов безопасной маршрутизации в облачных антивирусных системах / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2016. – Вип. 3 (140). – С. 36-39.
14. Смирнов С. А. Метод безопасной маршрутизации на базовом множестве путей передачи метаданных в облачные антивирусные системы / В. Л. Бурячок, С. А. Смирнов // Системи управління, навігації та зв'язку. – Полтава, 2016. – Вип. 4(40). – С. 57-62.
15. Смирнов С. А. Способ контроля линий связи телекоммуникационной системы облачного антивируса / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2016. – № 2 (47). – С. 148-152.
16. Смирнов С. А. Дослідження та реалізація GERT-моделі технології розповсюдження комп'ютерних вірусів для захисту телекомунікаційних систем / В. Л. Бурячок, Мохамад Абу Таам Гани, С. А. Смирнов // Інформаційні технології та комп'ютерна інженерія: зб. тез доп. наук.-практ. конф., м. Кіровоград, 4 грудня 2014 р. – Кіровоград: КНТУ, 2014. – С. 168.
17. Смирнов С. А. Исследование математических моделей технологии распространения компьютерных виру сов / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. праць міжнар. наук.-практ. конф., м. Київ, 25-28 лютого 2015 р. – К.: Європейський університет, 2015. – С. 90-91.
18. Смирнов С. А. Метод управления доступом к «облачным» ресурсам для защиты телекоммуникационных систем / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Всеукраїнська науково-практична конференція «Інформаційна безпека держави, суспільства та особистості», м. Кіровоград, 16 квітня 2015 р.: зб. тез доп. – Кіровоград: КНТУ, 2015. – С. 50-52.
19. Смирнов С. А. Разработка метода управления доступом в интеллектуальных узлах коммутации / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Проблеми і перспективи розвитку ІТ-індустрії: зб. тез VII міжнар. наук.-практ. конф., м. Харків, 17-18 квітня 2015 р. – Х.: ХНЕУ, 2015. – С. 14.
20. Смирнов С.А. Реализация метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Абу Таам Гани, С.А. Смирнов // Збірник тез XVII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 17-18 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 91-92.

УДК 004

О.Лаврусенко, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПРОЗОРОГО ШИФРУВАННЯ ДАНИХ З ЗАСТОСУВАННЯМ ЗАСОБІВ PKI

У статті програмне забезпечення, яке призначено для системи прозорого шифрування даних з застосуванням засобів PKI. Метою розробки є дослідження та програмна реалізація системи прозорого шифрування даних з застосуванням засобів PKI. Об'єктом дослідження є процес прозорого шифрування даних з застосуванням засобів PKI. Предметом дослідження є методи прозорого шифрування даних з застосуванням засобів PKI. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи прозорого шифрування даних з застосуванням засобів PKI. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, шифрування даних, PKI

Постановка проблеми. На сьогоднішній день уже кожний чув повідомлення про те, як особисті або конфіденційні дані були втрачені через крадіжку або втрату портативного комп'ютера. Портативні комп'ютери пропадають постійно. З ростом числа розкрадань особистих даних і при більш ніж будь-коли високою важливістю дотримання нормативних вимог ретельний захист даних на мобільних комп'ютерних системах украй важливий.

Одним з рішень є використання файлової системи EFS (Encrypting File System – шифрована файлова система), що забезпечує убудоване високоефективне шифрування диска. Система EFS працює однаково добре із власними технологіями перевірки дійсності й контролю доступу системи Windows, так що користувачам не потрібно запам'ятовувати для доступу до своїх даних окремі паролі. І, нарешті, система EFS забезпечує зручні варіанти відновлення даних у випадку втрати користувачем доступу до своїх ключів шифрування (наприклад у випадку видалення або ушкодження профілю користувача або у випадку втрати смарт-карти).

Для генерування, зберігання й розгортання ключів для захисту даних у системі EFS використовується технологія відкритих ключів шифрування (PKI). У даному магістерському проекті для шифрування даних на диску у файловій системі EFS використовується алгоритм стандарту DES. Ці симетричні ключі потім захищаються асиметричною парою ключів (RSA). У системі EFS кожний файл шифрується своїм власним ключем DES, потім цей ключ шифрується користувальницьким ключем RSA і результат зберігається у файл.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи прозорого шифрування даних з застосуванням засобів PKI.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи прозорого шифрування даних з застосуванням засобів PKI.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем прозорого шифрування даних з застосуванням засобів PKI.
- Дослідження системи прозорого шифрування даних з застосуванням засобів PKI.

– Програмна реалізація системи прозорого шифрування даних з застосуванням засобів PKI.

Об'єктом дослідження є процес прозорого шифрування даних з застосуванням засобів PKI.

Предметом дослідження є методи прозорого шифрування даних з застосуванням засобів PKI.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу

PKI, або інфраструктура відкритих ключів, охоплює все, що використовується для встановлення та керування шифруванням відкритих ключів. Це включає програмне забезпечення, апаратне забезпечення, політики та процедури, які використовуються для створення, розповсюдження, керування, зберігання та відкликання цифрових сертифікатів.

Цифровий сертифікат криптографічно пов'язує відкритий ключ із пристроєм або користувачем, який ним володіє. Це допомагає автентифікувати користувачів і пристрої та забезпечити безпечний цифровий зв'язок.

PKI є однією з найпоширеніших форм шифрування в Інтернеті, яка використовується для захисту та автентифікації трафіку між веб-браузерами та веб-серверами. Його також можна використовувати для захисту доступу до підключених пристроїв і внутрішніх комунікацій в організації.

Інфраструктура відкритих ключів має довгу історію захисту та автентифікації цифрових комунікацій з двома основними цілями: забезпечити конфіденційність повідомлення, що надсилається, і перевірити, чи відправник є тим, за кого себе видає.

Що таке інфраструктура відкритих ключів (PKI)?

Інфраструктура відкритих ключів є важливим аспектом безпеки в Інтернеті. Це набір технологій і процесів, які складають основу шифрування для захисту та автентифікації цифрових комунікацій.

PKI використовує криптографічні відкриті ключі, пов'язані з цифровим сертифікатом, який автентифікує пристрій або користувача, який надсилає цифрове повідомлення. Цифрові сертифікати видаються надійним джерелом, центром сертифікації (CA), і діють як тип цифрового паспорта, щоб гарантувати, що відправник є тим, за кого себе видає.

Інфраструктура відкритого ключа захищає та перевіряє зв'язок між серверами та користувачами, наприклад між вашим веб-сайтом (розміщеним на вашому веб-сервері) та вашими клієнтами (користувач, який намагається підключитися через свій браузер. Її також можна використовувати для безпечного зв'язку всередині організації, щоб забезпечити що повідомлення бачать лише відправник і одержувач, і вони не були змінені під час передачі.

До основних компонентів інфраструктури відкритих ключів належать:

– Центр сертифікації (ЦС): ЦС є довіреною організацією, яка видає, зберігає та підписує цифровий сертифікат. Центр сертифікації підписує цифровий сертифікат власним закритим ключем, а потім публікує відкритий ключ, до якого можна отримати доступ за запитом.

– Орган реєстрації (RA): RA перевіряє особу користувача або пристрою, що запитує цифровий сертифікат. Це може бути третя сторона, або ЦС також може діяти як RA.

– База даних сертифікатів: у цій базі даних зберігається цифровий сертифікат і його метадані, які включають тривалість дії сертифіката.

– Центральний каталог: це безпечне місце, де індексуються та зберігаються криптографічні ключі.

– Система керування сертифікатами: це система для керування доставкою сертифікатів, а також доступом до них.

– Політика щодо сертифікатів: ця політика описує процедури PKI. Його можуть використовувати сторонні особи для визначення надійності PKI.

Розуміння того, як працює PKI

Інфраструктура відкритих ключів використовує асиметричні методи шифрування, щоб гарантувати, що повідомлення залишаються приватними, а також для автентифікації пристрою або користувача, який надсилає передачу.

Асиметричне шифрування передбачає використання відкритого та закритого ключів. Криптографічний ключ – це довгий рядок бітів, який використовується для шифрування даних.

Відкритий ключ доступний кожному, хто його запитує, і видається довіреним центром сертифікації. Цей відкритий ключ перевіряє та автентифікує відправника зашифрованого повідомлення.

Другим компонентом пари криптографічних ключів, що використовується в інфраструктурі відкритих ключів, є приватний або секретний ключ. Цей ключ зберігається одержувачем зашифрованого повідомлення та використовується для розшифровки передачі.

Складні алгоритми використовуються для шифрування та дешифрування пар відкритих/приватних ключів. Відкритий ключ засвідчує автентичність відправника цифрового повідомлення, тоді як закритий ключ гарантує, що лише одержувач може відкрити та прочитати його.

Сертифікати PKI

Основою інфраструктури відкритих ключів є довіра. Організації-одержувачу важливо безсумнівно знати, що відправником цифрового сертифіката є саме той, за кого вони себе видають.

Довірені сторонні ЦС можуть поручитися за відправника та допомогти довести, що він справді є тим, за кого себе видає. Цифрові сертифікати використовуються для перевірки цифрової ідентифікації.

Цифрові сертифікати також називаються сертифікатами PKI або сертифікатами X.509 . Сертифікат PKI пропонує підтвердження особи суб’єкту, який надіслав запит, який перевіряється третьою стороною та працює як цифровий паспорт або водійське посвідчення.

Сертифікат PKI міститиме наступне:

- Помітне ім'я (DN) власника
- Відкритий ключ власника
- Дата видачі
- Термін придатності
- DN видавця ЦС
- Видача цифрового підпису ЦС

Чому використовується PKI?

Одним із найпоширеніших застосувань PKI є TLS/SSL (рівень безпеки транспортного рівня/рівень захищених сокетів), який захищає зашифрований зв’язок HTTP (протокол передачі гіпертексту).

Власники веб-сайтів отримають цифровий сертифікат від довіреного ЦС. Щоб отримати CA, власник веб-сайту повинен буде довести, що він справді є справжнім власником. Після перевірки власник веб-сайту може придбати сертифікат SSL для встановлення на веб-сервері. Це повідомляє браузеру, що це законний веб-сайт, до якого він намагається отримати доступ.

Протокол TLS/SSL покладається на ланцюжок довіри, де користувач має довіряти органу, що надає кореневий сертифікат. Альтернативною схемою є мережа довіри, яка використовує самопідписані сертифікати, перевірені третьою стороною. Мережа довіри часто використовується в невеликих спільнотах користувачів, наприклад, у самодостатній мережі організації.

Додаткові способи використання PKI включають наступне:

- Шифрування електронної пошти та аутентифікація відправника
- Підписання документів та програмного забезпечення
- Використання серверів баз даних для захисту внутрішніх комунікацій

– Захист веб-комунікацій, наприклад електронної комерції
– Аутентифікація та шифрування документів
– Захист локальних мереж і автентифікація смарт-карт
– Шифрування та дешифрування файлів
– Обмежений доступ до VPN та корпоративних інтрамереж
– Безпечний зв'язок між взаємно довіреними пристроями, такими як пристрої Інтернету речей (Інтернет речей).

Типи відкритих PKI

Інфраструктура відкритого ключа з відкритим кодом є загальнодоступною. Приклади PKI з відкритим кодом:

– EJBCA Enterprise: розроблено на Java як повнофункціональна реалізація CA корпоративного рівня, вона може налаштувати CA як службу або для внутрішнього використання.
– OpenSSL: повнофункціональний інструментарій комерційного рівня, він включений до всіх основних дистрибутивів Linux і розроблений на C. Він може використовувати PKI-додатки та використовуватися для створення простого ЦС.
– CFSSL: це набір інструментів PKI/SSL від Cloudflare для підписання, перевірки та об'єднання сертифікатів TLS і створення спеціальних інструментів TLS PKI
– XiPKI: високопродуктивний і масштабований відповідач CA та OCSP, реалізований на Java з підтримкою SHA-3.
– Система сертифікатів Dogtag: це повнофункціональний ЦС корпоративного класу, який підтримує всі аспекти керування життєвим циклом сертифікатів.

EJBCA® Enterprise

У зв'язаному суспільстві, оскільки потреба в надійних даних зростає, стає все більш очевидним, що безпека та PKI мають вирішальне значення для всіх видів бізнесу та організацій. Багатоцільове програмне забезпечення з відкритим кодом EJBCA Enterprise підтримує багато можливостей інтеграції та автоматизації та видає сертифікати людям, серверам і пристроям Інтернету речей.

EJBCA підтримує широкий спектр варіантів використання, сценаріїв і інтеграцій інфраструктури відкритих ключів (PKI) в інші екосистеми додатків і підтверджено у великих розгортаннях по всьому світу.

Побудований на основі відкритих стандартів і платформи з відкритим кодом, сертифікованої Common Criteria, EJBCA забезпечує прозорість і зобов'язання, необхідні для довгострокового рішення безпеки.

Розгорніть EJBCA відповідно до ваших потреб – як готове програмне чи апаратне забезпечення, або як хмару чи SaaS PKI.

Платформа PKI EJBCA Enterprise пропонує видачу сертифікатів і керування ними, щоб надати вам надійні ідентифікатори та безпечний зв'язок для будь-якого сценарію використання. EJBCA Enterprise є багатокористувачем і підтримує кілька центрів сертифікації (CA) і рівні центрів сертифікації в одному екземплярі програмного забезпечення.

Економічна безпека

Програмне забезпечення EJBCA Enterprise, здатне захистити практично будь-який варіант використання та сферу технології, відповідає всім вашим потребам щодо інфраструктури відкритих ключів (PKI) і надає різні варіанти, які дозволять вам знайти найбільш економічно ефективне рішення. PrimeKey пропонує EJBCA як готове програмне чи апаратне забезпечення, або як хмару чи SaaS PKI.

Масштабований

Гнучкість і надійність EJBCA Enterprise забезпечує можливість обслуговувати як маломасштабні, так і великомасштабні корпоративні впровадження з мільйонами користувачів або пристроїв у середовищах високої доступності, завдяки підтримці різних варіантів розгортання, централізованих операцій і високого рівня автоматизації.

Забезпечує відповідність

EJBCA Enterprise дотримується найкращих практик із детальними, підписаними журналами аудиту та транзакцій, авторизацією на основі ролей і розширеною підтримкою апаратних модулів безпеки. Він сертифікований Common Criteria і вже розгорнутий у численних ETSI/eIDAS- і WebTrust-аудитованих клієнтів і клієнтів ePassport.

Добре інтегрується

Завдяки перевіреній інтеграції в екосистеми додатків, включаючи пристрої IoT та інструменти DevOps, завдяки підтримці багатьох протоколів і форматів, EJBCA Enterprise є на вашому шляху цифровізації.

Параметри розгортання EJBCA

Щоб врахувати унікальні бізнес-завдання вашої організації, включаючи безпеку, бюджет і доступність внутрішніх ресурсів, PrimeKey пропонує комбінацію варіантів розгортання, які відповідають вашим потребам сьогодні та дозволяють вам гнучко розвиватися з часом.

Програмний пристрій

Розгорніть PKI у власному центрі обробки даних, використовуючи власні ресурси віртуалізації. Виберіть HSM і модель приладу, яка найкраще відповідає вашим потребам.

Апаратний пристрій

Виберіть EJBCA Hardware Appliance, якщо ви шукаєте локальне рішення PKI-in-a-box. EJBCA Hardware Appliance – це надійний, високопродуктивний сервер, який постачається з повним апаратним і програмним забезпеченням і HSM.

Хмара EJBCA

Насолоджуйтеся швидким розгортанням за допомогою PKI у загальнодоступній хмарі без необхідності купувати та підтримувати апаратне забезпечення або будь-які попередні витрати на ліцензію на програмне забезпечення. Наші хмарні рішення PKI доступні в AWS і Azure.

Програмне забезпечення EJBCA як послуга

Якщо ви шукаєте повністю розміщене та кероване рішення PKI, то EJBCA SaaS – це ваш вибір. Це допомагає обмежити ризики розгортання та збільшити швидкість виходу на ринок.

Компоненти продукту EJBCA

EJBCA постачається або може використовуватися разом із такими розширеними інструментами для реєстрації та перевірки сертифікатів:

Реєстраційний орган EJBCA

Центр реєстрації EJBCA (RA) є зовнішньою організацією для центру сертифікації (CA) для реєстрації будь-якого типу сертифіката, що забезпечує додатковий рівень безпеки навколо центру сертифікації.

Орган перевірки EJBCA

EJBCA Validation Authority (VA) дозволяє онлайн-перевірку сертифіката за допомогою OCSP або CRL.

Автоматична реєстрація сертифіката

Завдяки функції автоматичної реєстрації сертифіката в EJBCA Enterprise ви можете позбутися будь-якої необхідності використовувати центри сертифікації Microsoft і повністю використовувати повну гнучкість EJBCA Enterprise і Active Directory.

Менеджер повноважень ідентифікації

Обладнання промислового класу PKI Registration Authority (RA), яке можна використовувати разом з EJBCA для видачі сертифікатів продукту безпосередньо на виробництві.

Видання EJBCA eIDAS

З випуском EJBCA eIDAS як апаратним або програмним пристроєм ви отримуєте повний набір функцій для роботи повномасштабної інфраструктури відкритих ключів (PKI), сумісної з eIDAS.

Набір інструментів PKI/TLS CloudFlare

CFSSL – це швейцарський армійський ніж PKI/TLS від CloudFlare. Це як інструмент командного рядка, так і сервер HTTP API для підписання, перевірки та об'єднання сертифікатів TLS. Для створення потрібен Go 1.16+.

Зауважте, що певні дистрибутиви Linux видаляють певні алгоритми (зокрема дистрибутиви на основі RHEL), тому golang з офіційних репозиторіїв не працюватиме. Користувачі цих дистрибутивів повинні інсталювати go вручну, щоб інсталювати CFSSL.

CFSSL складається з:

– набір пакетів, корисних для створення спеціальних інструментів TLS PKI;

– програма cfssl, яка є канонічною утилітою командного рядка, що використовує пакети CFSSL;

– програму multirootca, яка є сервером центру сертифікації, який може використовувати кілька ключів підпису;

– програма mkbundlevикористовується для створення пулів сертифікатів;

– програма cfssljson, яка отримує вихідні дані JSON з cfsslпрограм multirootcai записує сертифікати, ключі, CSR і пакети на диск.

XiPKI

XiPKI (e X tensible s Imple Public Key I nfrastructure) – це високомасштабована та високопродуктивна PKI з відкритим кодом (відповідач CA та OCSP) .

Ліцензія

– Ліцензія на програмне забезпечення Apache, версія 2.0.

Система сертифікації Dogtag

истема сертифікації Dogtag – це центр сертифікації корпоративного класу з відкритим вихідним кодом (CA). Це повнофункціональна система, яка була посилена розгортанням у реальному світі. Він підтримує всі аспекти керування життєвим циклом сертифіката, включаючи архівування ключів, OCSP і керування смарт-картами, а також багато іншого. Систему сертифікатів Dogtag можна завантажити безкоштовно та налаштувати менш ніж за годину.

На цьому сайті є все, що вам потрібно, щоб приєднатися до спільноти Dogtag. Незалежно від того, чи вам потрібна допомога та порада щодо розгортання та використання компонентів Dogtag, чи ви хочете взяти на себе більш активну роль і допомогти сформувати майбутнє PKI, є посилання на документацію, списки розсилки та канали обговорень, які ви можете прочитати або приєднатися:

– Онлайн-документація.
– Посилання на додаткову документацію.
– Списки розсилки.
– Сайти онлайн-чату через канали IRC.

Ключові особливості

Dogtag – це набір технологій, які дозволяють підприємствам розгортати PKI у великих масштабах. Він має такі функції, як:

– Видача, відкликання та відновлення сертифіката.
– Створення та публікація списку відкликаних сертифікатів (CRL).
– Профілі сертифікатів.
– Простий протокол реєстрації сертифікатів (SCEP).
– Місцевий орган реєстрації (LRA) для автентифікації та політики організації.
– Архівація та відновлення ключа шифрування.
– Управління життєвим циклом смарт-карт:

- Профілі токенів.
- Реєстрація маркерів, утримання, відновлення ключа та форматування.
- Особиста реєстрація за допомогою інтерфейсу робочої станції офіцера безпеки.

– Велика документація.

EFS

Перед використанням можливостей шифрування EFS варто визначитися чи буде використовуватися Агент відновлення даних. Агентом відновлення називається користувач, уповноважений розшифровувати дані, зашифруванні іншим користувачем, якщо користувач втратив закриті ключі сертифіката шифрування або обліковий запис користувача віддалений і потрібно відновити зашифруванні дані. Як правило, Агентом відновлення вказується Адміністратор, але може бути призначений і інший користувач. Може бути створене трохи Агентів відновлення. Щоб призначити користувача Агентом відновлення, необхідно спочатку створити сертифікати Агента відновлення.

Шифрувати можна як окремі файли, так і цілі папки, при цьому якщо шифрується папка вже утримуюча файли, тобто можливість вибору, шифрувати тільки папку або папку й вкладені файли. Шифрування папки не означає що інші користувачі не зможуть переглядати вміст папки – вони лише не зможуть відкривати зашифруванні файли. Всі нові файли, збережені в зашифруванній папці або скопійовані в неї будуть автоматично зашифруванні. Шифрувати папки більш зручно, і крім того безпечно, оскільки EFS має схему відновлення після аварійного збою (наприклад якщо під час операції шифрування відбулася критична помилка) яка передбачає створення незашифрованної архівної копії вихідного файлу – при успішному завершенні операції шифрування архівна копія віддаляється, але може бути відновлена спеціальними програмами відновлення віддалених даних, що створює потенційну погрозу інформаційної безпеки. А при збереженні файлу в зашифруванній папці шифрування відбувається без створення такої резервної копії. Якщо все-таки шифрувалися одиночні файли, тобто можливість перезаписати кластери, що залишилися після зміни або видалення файлів на томах NTFS випадковими значеннями – для цього може бути використана команда cipher /w:шлях запущена з командного рядка (докладніше про використання цієї команди дивитеся в довіднику по командному рядку) або програми сторонніх розроблювачів.

Із усього перерахованого вище можна зробити наступні висновки:

−Система EFS надає користувачам можливість зашифровувати каталоги NTFS, використовуючи стійку, засновану на загальних ключах криптографічну схему, при цьому всі файли в закритих каталогах будуть зашифруванні. Шифрування окремих файлів підтримується, але не рекомендується через непередбачене поводження додатків.

−Система EFS також підтримує шифрування віддалених файлів, доступ до яких здійснюється як до спільно використовуваних ресурсів. Якщо мають місце користувальницькі профілі для підключення, використовуються ключі й сертифікати віддалених профілів. В інших випадках генеруються локальні профілі й використовуються локальні ключі.

−Система EFS надає можливість встановити політику відновлення даних таким чином, що зашифруванні дані можуть бути відновлені за допомогою EFS, якщо це буде потрібно.

−Політика відновлення даних убудована в загальну політику безпеки Windows. Контроль за дотриманням політики відновлення може бути делегований уповноваженим на це особам. Для кожного підрозділу організації може бути зконфігурована своя політика відновлення даних.

−Відновлення даних в EFS – закрита операція. У процесі відновлення розшифровуються дані, але не ключ користувача, за допомогою якого ці дані були зашифруванні.

−Робота із зашифруванними файлами в EFS не жадає від користувача яких-небудь спеціальних дій по шифруванню й дешифруванню даних. Дешифрування й шифрування відбуваються непомітно для користувача в процесі зчитування й запису даних на диск.

−Система EFS підтримує резервне копіювання й відновлення зашифруванних файлів без їхньої розшифровки. Програма NtBackup підтримує резервне копіювання зашифруванних файлів.

–Система EFS убудована в операційну систему таким чином, що витік інформації через файли підкачування неможливий, при цьому гарантується, що всі створювані копії будуть зашифруванні

–Передбачено численні запобіжні заходи для забезпечення безпеки відновлення даних, а також захист від витоку й втрати даних у випадку фатальних збоїв системи.

Опис PKI

Розглянемо технологію PKI. Задачею PKI є визначення політики випуску цифрових сертифікатів, видача їх і анулювання, зберігання інформації, необхідної для наступної перевірки правильності сертифікатів. PKI використовується в EFS. Діяльність інфраструктури керування відкритими ключами здійснюється на основі регламенту системи. Інфраструктура відкритих ключів ґрунтується на використанні принципів криптографічної системи з відкритим ключем. Інфраструктура керування відкритими ключами складається із центра сертифікації, кінцевих користувачів, і опціональних компонентів: центра реєстрації й мережного довідника.

Зрозуміло, що PKI оперує в роботі сертифікатами. Але у зв'язку з тим, що в PKI використовуються сертифікати, виникає множина нюансів, без яких будь-яка PKI не буде працювати коректно.

По суті, сертифікат – це ключова пара, що складається із двох ключів – зазвичай перший ключ називається закритим ключем (private key), а другий ключ – відкритим ключем (public key). Ці ключі створюються тільки в парі й мають однаковий електронний відбиток. По електронному відбитку можна визначити, чи відповідає даний відкритий ключ своєму закритому ключу.

Створює ці ключі якийсь центр, що зазвичай називається центром видачі сертифікатів або центром, що засвідчує, по запиті користувача. Користувач робить запит на сертифікат, після чого, після деяких процедур ідентифікації користувача, центр видає йому сертифікат зі своїм підписом (цей підпис свідчить про те, що даний сертифікат виданий саме цим центром видачі сертифікатів і ніким іншим), після чого користувач має в наявності свій закритий ключ і відповідний йому відкритий ключ.

Закритий ключ використовується для підпису даних, відкритий ключ у свою чергу використовується для шифрування даних. Відкритий ключ відомий усім, а закритий ключ зберігається в таємниці. Власник закритого ключа завжди зберігає його в захищеному місці й ні за яких умов не повинен допустити того, щоб цей ключ став відомим зловмисникам або іншим користувачам.

Якщо ж закритий ключ усе таки стане відомий зловмисникам, необхідно терміново сповістити про це колег, щоб запобігти витоку важливої інформації. Тільки власник закритого ключа може підписати дані, а також розшифрувати дані, які були зашифровані відкритим ключем, що відповідає закритому ключу власника.

Тому що закритий ключ використовується для підпису даних – його можна назвати своєрідним ідентифікатором передплатника. Підпис на даних або листі гарантує цілісність отриманої інформації.

Термінологія PKI

Із усього вище сказаного можна виділити деякі пункти, а також додати нові, для того щоб визначити основні терміни, використовувані в PKI.

Отже, в PKI використовуються терміни:

–Сертифікат – це ключова пара (складається з відкритого й закритого ключів), до якої приписаний її унікальний номер, ім'я власника сертифіката, а також ім'я центра видачі, що видав цей сертифікат.

–Закритий ключ – ключ, що зберігається в таємниці, створений з використанням PKI алгоритмів, що має свій унікальний електронний помилок і, що використовується для одержання зашифрованих даних і підпису даних.

–Відкритий ключ – ключ, створений у парі із закритим ключем, що має такий же електронний відбиток, як і закритий ключ, якому він відповідає, використовується для шифрування даних і перевірки підпису

–Підписані дані – дані, підписані за допомогою закритого ключа користувача.

–Зашифровані дані – дані, зашифровані за допомогою відкритого ключа користувача.

Терміни, які необхідні для загального розуміння:

–Мережа довіри – або ланцюжок сертифікацій, необхідна й потрібна для тих випадків, коли є множина різних центрів, що засвідчують, і виникають ситуації, коли один УЦ (удостовірюючий центр), не довіряє якомусь іншому, але при цьому може покластися на те, що загальний дружній УЦ довіряє обом

–Особисті сертифікати – сертифікати які зберігаються в користувача в особистому сховищі сертифікатів.

–Кореневі центри сертифікації – центри сертифікації, яким довіряють споконвічно всі, наприклад після установки ОС. У ці центри сертифікації можна в будь-який час додавати нові центри, яким Ви хочете довіряти

–Довірені центри сертифікації – список центрів сертифікації, яким довіряєте особисто. Щоб зробити любий УЦ довіреним, досить одержати від нього сертифікат і внести його в довірені центри Також важливо знати про поняття центра сертифікації й про кінцевих користувачів.

–Центр сертифікації (удостоверяючий центр) – є основною структурою, що формує цифрові сертифікати підлеглих центрів сертифікації й кінцевих користувачів. Центр сертифікації сам формує власний секретний ключ і сертифікат, що містить відкритий ключ даного центра. Засвідчує автентичність відкритого ключа користувача своїм електронно-цифровим підписом. Формує список відкликаних сертифікатів. Веде бази всіх виготовлених сертифікатів і списків відкликаних сертифікатів. Відкритий ключ, підписаний центром сертифікації, називається сертифікатом відкритого ключа.

–Кінцеві користувачі – є користувачі, додатки або системи, що є власниками сертифіката й використовують інфраструктуру керування відкритими ключами

–Центр Реєстрації – опціональна компонента інфраструктури, призначена для реєстрації кінцевих користувачів і забезпечення їхньої взаємодії із центром сертифікації.

–Мережний довідник – опціональна компонента інфраструктури, що містить сертифікати й списки відкликаних сертифікатів і, що служить для мети поширення цих об'єктів серед користувачів.

Впровадження інфраструктури керування відкритими ключами з урахуванням зниження витрат і строків впровадження здійснюється протягом семи етапів.

– Етап 1. Аналіз вимог до системи.
– Етап 2. Визначення архітектури.
– Етап 3. Визначення регламенту.
– Етап 4. Огляд системи безпеки. Аналіз і мінімізація ризиків.
– Етап 5. Інтеграція.
– Етап 6. Розгортання.
– Етап 7. Експлуатація.

Деякі основні моменти

Двома словами вже було сказано, для чого потрібні закритий і відкритий ключі, що таке сертифікат і центри, що засвідчують. Але тому що це основні компоненти PKI, розберемо докладніше наступні моменти:

– у чому полягає робота УЦ

– як відбувається видача сертифіката, обмін відкритими ключами і як зрозуміти, що відкритий ключ, що перебувати перед моїми очами, не фальшивий

– а також те, які бувають PKI.

УЦ і його робота

Основна робота центра, що засвідчує, полягає в ідентифікації користувачів і їхніх запитів на сертифікати, у видачі користувачам сертифікатів, у перевірках автентичності сертифікатів, у перевірці за сертифікатом, чи не видає користувач сертифіката себе за інший, в анулюванні або відкликанні сертифікатів, у веденні списку відкликаних сертифікатів.

Розробка структурної схеми

На рисунках 1 та 2 показані структурні схеми шифрування та дешифрування EFS.

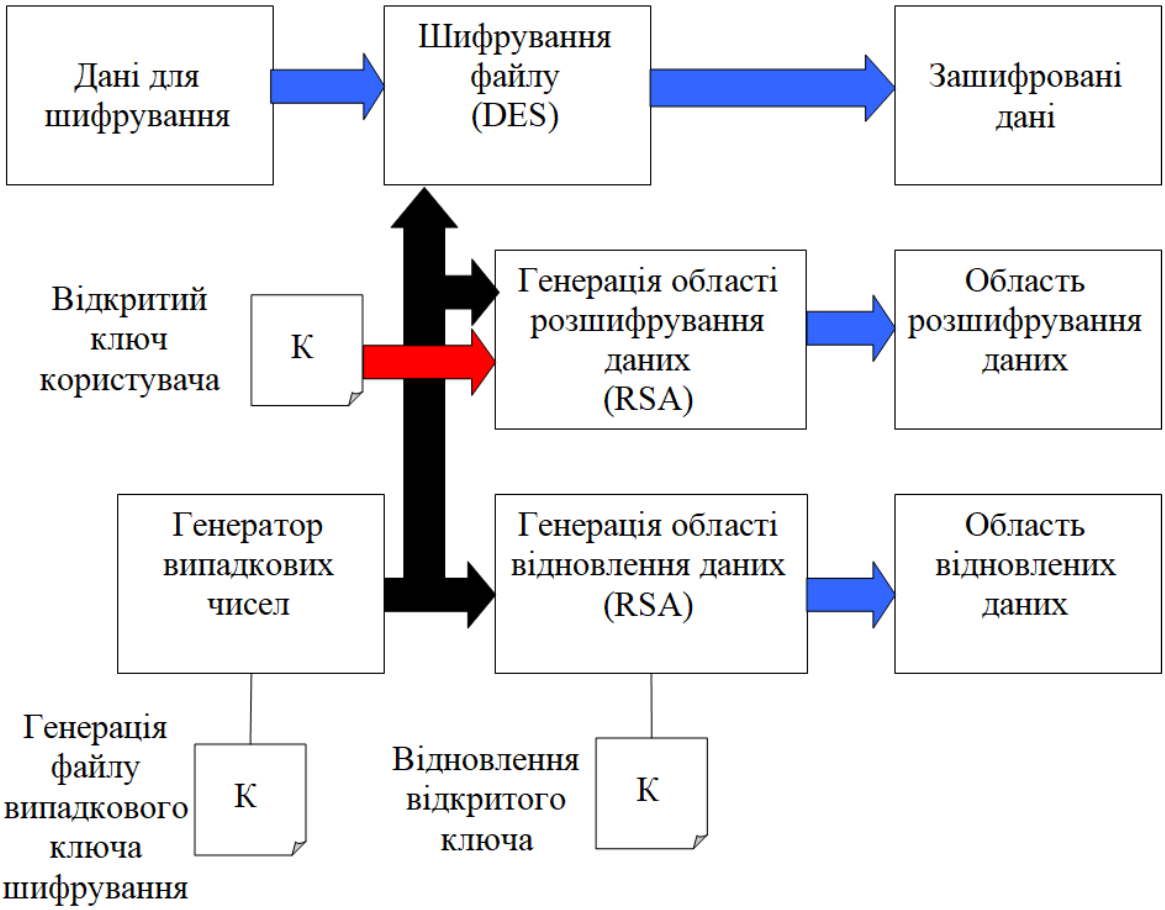


Рисунок 1 – Структурна схема шифрування EFS

Якщо коротко, то можливо сказати, що робота системи відбувається наступним чином. EFS працює, шифруючи кожний файл за допомогою алгоритму симетричного шифрування, що залежить від версії операційної системи й настроювань.

При цьому використовується випадково-згенерований ключ для кожного файлу, називаний **File Encryption Key** (FEK), вибір симетричного шифрування на даному етапі пояснюється його швидкістю й більшою надійністю стосовно асиметричного шифрування. У данному магістерському проекті у якості симетричного алгоритму шифрування вибраний DES, у зв'язку з тим, що він є достатньо стійким та швидким.

FEK (випадковий для кожного файлу ключ симетричного шифрування) захищається шляхом асиметричного шифрування, що використовує відкритий ключ користувача файл, який шифрує, і алгоритм RSA (теоретично можливе використання інших алгоритмів асиметричного шифрування).

RSA обраний тому, що він достаньо стійкий , для цих потреб, та виконується більш швидко, ніж інші алгоритми симметричного шифрування. Зашифрований у такий спосіб ключ FEK зберігається в альтернативному потоці $EFS файлової системи NTFS.

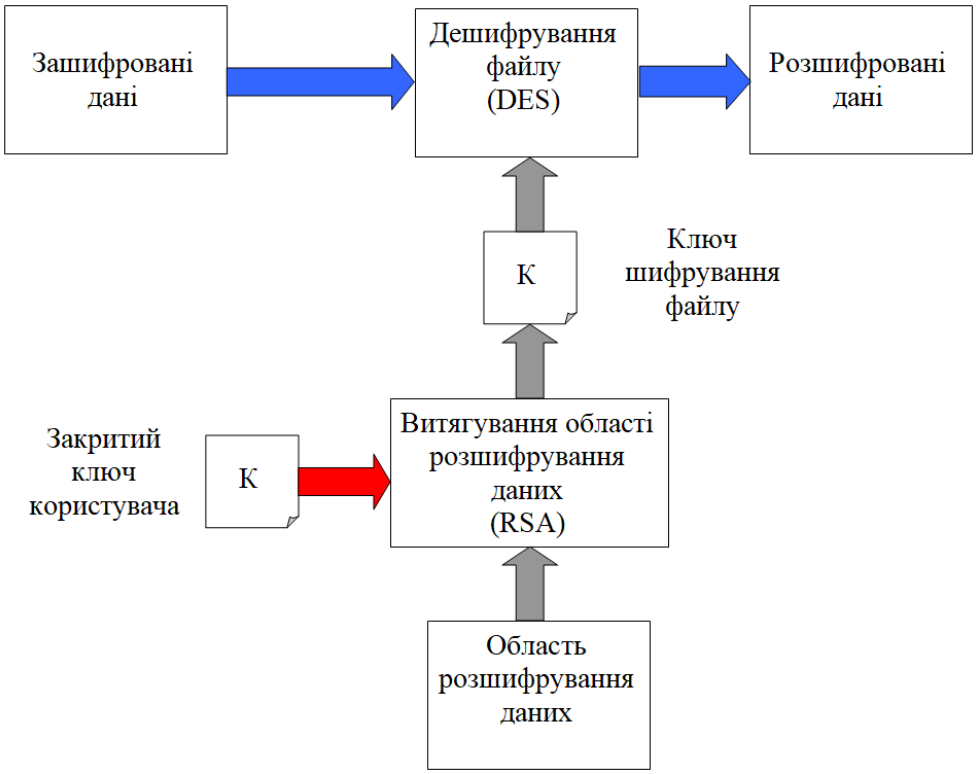


Рисунок 2 – Структурна схема дешифрування EFS

Для розшифрування даних, драйвер шифруванної файлової системи, прозоро для користувача, розшифровує FEK використовуючи закритий ключ користувача, а потім і необхідний файл за допомогою розшифруванного файлового ключа.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів прозорого шифрування даних з застосуванням засобів PKI. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем прозорого шифрування даних з застосуванням засобів PKI. Досліджена система прозорого шифрування даних з застосуванням засобів PKI. На основі отриманих результатів досліджень створена програмна реалізація системи прозорого шифрування даних з застосуванням засобів PKI. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання прозорого шифрування даних з застосуванням засобів PKI. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov S.A. Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam, S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.
2. Смирнов С. А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты даннях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2015. – № 3(43). – С. 100-107.
3. Смирнов С. А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2015. –№ 3(20). – С. 134-141.
4. Смирнов С. А. Комплекс gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Безпека інформації: наук. – практ. журн. – К.: НАУ, 2015. – Т. 21, № 3. – С. 251-262.

5. Смирнов С. А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А. А. Смирнов, А. К. Дидык, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2016. – № 2 (46). – С. 146-149.
6. Смирнов С. А. Модели системы нейросетевых экспертов безопасной маршрутизации в облачных антивирусных системах / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2016. – Вип. 3 (140). – С. 36-39.
7. Смирнов С. А. Метод безопасной маршрутизации на базовом множестве путей передачи метаданных в облачные антивирусные системы / В. Л. Бурячок, С. А. Смирнов // Системи управління, навігації та зв'язку. – Полтава, 2016. – Вип. 4(40). – С. 57-62.
8. Смирнов С. А. Способ контроля линий связи телекоммуникационной системы облачного антивируса / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2016. – № 2 (47). – С. 148-152.
9. Смирнов С. А. Дослідження та реалізація GERT-моделі технології розповсюдження комп'ютерних вірусів для захисту телекомунікаційних систем / В. Л. Бурячок, Мохамад Абу Таам Гани, С. А. Смирнов // Інформаційні технології та комп'ютерна інженерія: зб. тез доп. наук.-практ. конф., м. Кіровоград, 4 грудня 2014 р. – Кіровоград: КНТУ, 2014. – С. 168.
10. Смирнов С. А. Исследование математических моделей технологии распространения компьютерных виру сов / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Актуальні питання забезпечення кібернетичної безпеки та захисту інформації: зб. наук. праць міжнар. наук.-практ. конф., м. Київ, 25-28 лютого 2015 р. – К.: Європейський університет, 2015. – С. 90-91.
11. Смирнов С. А. Метод управления доступом к «облачным» ресурсам для защиты телекоммуникационных систем / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Всеукраїнська науково-практична конференція «Інформаційна безпека держави, суспільства та особистості», м. Кіровоград, 16 квітня 2015 р.: зб. тез доп. – Кіровоград: КНТУ, 2015. – С. 50-52.
12. Смирнов С. А. Разработка метода управления доступом в интеллектуальных узлах коммутации / А. А. Смирнов, Мохамад Абу Таам Гани, С. А. Смирнов // Проблеми і перспективи розвитку ІТ-індустрії: зб. тез VII міжнар. наук.-практ. конф., м. Харків, 17-18 квітня 2015 р. – Х.: ХНЕУ, 2015. – С. 14.
13. Смирнов С.А. Реализация метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Абу Таам Гани, С.А. Смирнов // Збірник тез XVII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 17-18 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 91-92.
14. Смирнов С. А. технология передачи сигнатур в облачные антивирусные системы для обеспечения защищенности телекоммуникационных сетей / А. А. Смирнов, С. А. Смирнов // Збірник тез V міжнародної науково-технічної конференції «ITSEC», Київ, 19-22 травня 2015 р. – К.: НАУ 2015. – С. 12-13.
15. Смирнов С. А. Реализация математической модели интеллектуального узла коммутации для обеспечения защищенности телекоммуникационной сети / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Інформаційна та економічна безпека (INFECO-2015): зб. тез II Міжнар. наук.-практ. Інтернет-конф., м. Харків, 21-22 травня 2015 р. – Х.: ХІБС УБС НБУ, 2015. – С. 20-24.
16. Смирнов С. А. Разработка математической модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Сборник тезисов XI международной конференции «Стратегия качества в промышленности и образовании», г. Варна, Болгария, 01-06 июня 2015 г. – Варна: ТУВ, 2015. – С. 488-491.
17. Смирнов С. А. Метод управления доступом к облачным телекоммуни-кационным ресурсам для обеспечения защиты даннях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Комп'ютерні технології та інформаційна безпека: зб. тез доп. міжнар. наук.-практ. конф., м. Кіровоград, 2-3 липня 2015 р. – Кіровоград: КНТУ, 2015. – С. 4-5.
18. Смирнов С. А. Имитационная модель системы управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Збірник тез першої всеукраїнської науково-практичної конференції «Перспективні напрями захисту інформації» (м. Затока, 7-9 вересня 2015 р.). – Одеса: ОНАЗ, 2015. – С. 90-94.
19. Смирнов С. А. Разработка комплекса gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, С. А. Смирнов, А. К. Дидык // Інформаційні технології та взаємодії» (IT & I): зб. тез II міжнар. наук.-практ. конф., м. Київ, 3-5 листопада 2015 р. – К.: КНУ ім. Тараса Шевченка, 2015. – С. 65-67.
20. Смирнов С. А. Разработка моделей телекоммуникационной системы формирования и обработки метаданных в облачных антивирусных системах / А.А. Смирнов, С.А. Смирнов, А. К. Дидык // Информационные и телекоммуникационные технологии: образование, наука, практика: сб. тезисов II междунар. научно-практ. конф., г. Алматы, Казахстан, 3-4 декабря 2015 г. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – С. 309-313.

УДК 004

І.Науменко, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПОВІДОМЛЕНЬ ЕЛЕКТРОННОЇ ПОШТИ

У статті розроблено програмне забезпечення, яке призначено для системи повідомлень електронної пошти. Метою розробки є дослідження та програмна реалізація системи повідомлень електронної пошти. Об'єктом дослідження є процес повідомлень електронної пошти. Предметом дослідження є методи повідомлень електронної пошти. Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи повідомлень електронної пошти. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, електронна пошта

Постановка проблеми. Адміністратори багатьох організацій намагаються захистити повідомлення електронної пошти співробітників. Secure MIME (S/MIME) – рішення безпеки, реалізоване в більшості сучасних поштових програм, що допоможе зберегти конфіденційність, цілісність поштових повідомлень і перевірити дійсність даних. S/MIME забезпечує наскрізний захист – не тільки в процесі пересилання повідомлень, але й при зберіганні в базі даних поштового сервера. S/MIME забезпечує перевірку дійсності даних, конфіденційність і цілісність повідомлень у форматі MIME. S/MIME – відмінний приклад гібридного рішення шифрування, у якому об'єднані достоїнства симетричного й асиметричного шифрів і функцій гешування. Якщо TLS забезпечує безпеку даних у момент пересилання по незахищеній мережі, такій як Інтернет, то S/MIME забезпечує безпеку даних між кінцевими користувачами: S/MIME повідомлення шифрується відправником (з використанням багатобічного шифрування) і передається на сервер відправника в зашифрованій формі. Та ж зашифрована форма використовується, коли повідомлення передається через мережу, коли воно зберігатися на проміжних серверах, і коли воно міститься в папках одержувачів. Тільки одержувачі, використовуючи свої закриті ключі, можуть розшифрувати повідомлення й в той момент, коли вони фактично читають повідомлення: саме повідомлення залишається зашифрованим у папках одержувачів. Для того, що б кінцеві користувачі могли використовувати S/MIME безпеку, всі вони повинні мати своїми власними PKI-ключі. Кожний користувач повинен мати закритий ключ, що безпечно зберігається в місці, доступному тільки для цього користувача, і відповідний йому відкритий ключ, убудований у сертифікат. Цей Сертифікат повинен бути виданий центром сертифікації (удостовірюючим центром), якому довіряють інші користувачі. S/MIME забезпечує потужні розширення функцій поштової безпеки, які можуть бути дуже корисними користувачам.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи повідомлень електронної пошти.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи повідомлень електронної пошти.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем повідомлень електронної пошти.

– Дослідження системи повідомлень електронної пошти.
– Програмна реалізація системи повідомлень електронної пошти.

Об'єктом дослідження є процес повідомлень електронної пошти.

Предметом дослідження є методи повідомлень електронної пошти.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Опис S/MIME на основі PKI

Специфікація S/MIME визначає два типи MIME-конверта: один для цифрових підписів, іншої – для шифрованих повідомлень. Обидва типи базуються на синтаксисі криптографічних повідомлень стандарту PKCS#7 [6]. Якщо повідомлення повинне бути зашифроване, а шифртексту повинні бути привласнені деякі атрибути, використовуються вкладені конверти. Зовнішній і внутрішній конверт призначаються для захисту цифрових підписів, а проміжний конверт – для захисту шифртексту.

Крім забезпечення цілісності повідомлення під час передачі, S/MIME ідентифікує власника конкретного відкритого ключа за допомогою сертифіката X.509. Цифровий сертифікат засвідчує, що відкритий ключ дійсно належить тому, хто є суб'єктом сертифіката.

S/MIME v3 підтримує наступні важливі властивості, які відсутні в S/MIME v2:

–завірені цифровим підписом квитанції;

–мітки безпеки;

–списки розсилання;

–гнучке керування ключами.

Завірені цифровим підписом **квитанції** дозволяють відправникові повідомлення впевнитися в тому, що воно було отримано адресатами без змін. Одержувач повідомлення не може згенерувати валідну квитанцію доти, поки не перевірить підпис відправника на отриманому повідомленні.

Мітки безпеки дозволяють відправникові задавати керуючі вимоги до змісту повідомлення. Найчастіше мітка безпеки свідчить про включення в зміст повідомлення приватної або конфіденційної інформації.

Підтримка захищеної електронної пошти на основі PKI

Всі сервіси, пропонованими S/MIME (обох версій), покладаються на сертифікати й надійність зв'язування електронної адреси суб'єкта з його відкритим ключем. Адреса електронної пошти (часто називаєма адресою RFC 822 [8]) повинна бути представлена у доповненні сертифіката Subject Alternative Name (альтернативне ім'я суб'єкта). Якщо використовується друга версія S/MIME, то адреса електронної пошти вказується як відмітне ім'я суб'єкта (emailAddress).

Відправник зашифрованого повідомлення повинен бути впевнений, що відкритий ключ належить саме тому одержувачеві, якому він адресує своє повідомлення, у противному випадку доступ до змісту повідомлення може одержати сторонній. Аналогічно, поширюючи ключі шифрування симетричного ключа тільки членам списку розсилання відправника, агент MLA покладається на правильність зв'язування ідентичності одержувача і його відкритого ключа. Відправник і агент MLA порівнюють ідентифікаційну ознаку одержувача, зазначену в сертифікаті, з адресою електронної пошти одержувача, якому посилає своє повідомлення відправник.

Одержувач повідомлення, завіреного цифровим підписом, повинен бути впевнений, що відкритий ключ підпису, що необхідний для верифікації підпису на повідомленні, належить відправникові. Для цього він порівнює адресу електронної пошти, зазначену в полі SENDER (або FROM) отриманого повідомлення, з адресою, представленою в сертифікаті.

Аналогічні дії виконуються при перевірці квитанцій, завірених цифровим підписом, – для цього що перевіряє порівнює адресу електронної пошти із запиту на квитанцію з адресою електронної пошти в сертифікаті особи, що подписали квитанцію.

Засоби безпеки транспортного рівня

Протокол безпеки транспортного рівня Transport Layer Protocol (TLS) [9] забезпечує захист комунікацій між додатками, розробленими в архітектурі "клієнт-сервер", в основному між web-браузером і web-сервером. World Wide Web є найбільш популярний Інтернет-сервісом після електронної пошти. TLS найбільше часто застосовується для захисту web-контента, але може використовуватися з будь-яким протоколом прикладного рівня. Специфікація TLS базується на популярному протоколі Secure Socket Layer (SSL) [10], розробленому корпорацією Netscape. Ці протоколи створювалися для забезпечення автентифікації, цілісності й конфіденційності даних, якими обмінюються взаємодіючі один з одним додатки. Обидва протоколи мають дворівневу організацію: протокол установлення з'єднання (Handshake Protocol) і протокол передачі записів (Record Protocol).

Протокол установлення з'єднання дозволяє серверу й клієнтові виконати взаємну автентифікацію, погодити застосовуваний алгоритм шифрування й криптографічні параметри перед тим, як протокол прикладного рівня почне передачу даних. **Протокол передачі записів** забезпечує захист протоколів більше високого рівня, включаючи протокол установлення з'єднання. Протокол передачі записів залежить від надійності транспортного протоколу, такого як TCP.

Протоколи SSL і TLS незалежні від протоколів прикладного рівня, тому будь-який протокол прикладного рівня може прозоро оперувати поверх SSL і TLS. Протоколи SSL і TLS забезпечують три сервіси безпеки [11]:

−автентифікацію (підтвердження ідентичності з'єднання: протокол установлення з'єднання використовує сертифікати й верифікацію цифрових підписів для підтвердження ідентифікаційних ознак і повноважень вилученого додатка);

−цілісність (захист даних протоколу від несанкціонованої модифікації: протокол передачі записів використовує значення біта контролю цілісності для підтвердження того, що передані дані не змінювалися);

−конфіденційність (забезпечення таємності з'єднання: після узгодження симетричного ключа шифрування на основі протоколу встановлення з'єднання виконується шифрування даних, якими обмінюються сторони під час сеансу зв'язку).

Протоколи SSL і TLS здатні підтримувати взаємну автентифікацію сторін, але зазвичай на базі сертифіката виконується автентифікація сервера клієнтом, а потім клієнт автентифікується іншим способом, наприклад, уводячи по запиту сервера своє ім'я й пароль або номер своєї кредитної карти й дату закінчення її терміну дії.

Протокол установлення з'єднань

Протокол установлення з'єднань Handshake Protocol складається як би із трьох підпротоколів, які дозволяють виконати автентифікацію сторін, погодити алгоритми й параметри безпеки для протоколу передачі записів [11].

Handshake Protocol відповідає за організацію сеансу взаємодії між клієнтом і сервером для протоколу передачі записів, зокрема за узгодження характеристик сеансу:

−ідентифікатора сеансу (Session identifier), тобто довільної послідовності біт, обраної сервером для ідентифікації сеансу;

−сертифіката з'єднання (Peer certificate), що представляє собою сертифікат X.509; цей елемент може бути відсутнім, якщо автентифікація не потрібна;

−методу стиску (Compression method), тобто алгоритму стиску даних перед їхнім шифруванням;

−специфікатора шифрування (Cipher spec), що задає ідентифікатори алгоритму шифрування даних і алгоритму гешування, а також деякі криптографічні атрибути (наприклад, розмір геш-коду);

−головного секрету (Master secret), що представляє собою секретне значення, розділене між клієнтом і сервером;

−ознаки встановлення нового з'єднання (Is resumable) на основі поточного сеансу.

Ці характеристики потім використовуються для установки параметрів безпеки в протоколі передачі записів. Можливість установити кілька захищених з'єднань під час одного сеансу особливо важлива, коли клієнтові й серверу необхідно встановити кілька короткочасних з'єднань.

У результаті роботи протоколу Handshake Protocol формуються криптографічні параметри. Коли клієнт і сервер починають взаємодію, то погоджують версію протоколу, криптографічні алгоритми, автентифікують один одного (за бажанням) і використовують криптографію з відкритими ключами для поділу загального секрету. Робота протоколу Handshake Protocol виконується за шість кроків [12].

1-й крок. Обмін привітальними повідомленнями для узгодження алгоритмів і обміну випадковими числами.

2-й крок. Обмін криптографічними параметрами для узгодження початкового секрету.

3-й крок. Опціональний обмін сертифікатами й криптографічною інформацією для взаємної автентифікації клієнта й сервера.

4-й крок. Генерація головного секрету на основі початкового секрету й обмін випадковими числами.

5-й крок. Формування параметрів безпеки для протоколу передачі записів.

6-й крок. Оповіщення про розрахунок тих же самих параметрів безпеки й коректному завершенні з'єднання.

Коли клієнт бажає встановити нове з'єднання на основі даного сеансу або повторити цей сеанс, то відправляє своє привітальне повідомлення з ідентифікатором сеансу, що повинен бути відновлений. Якщо сервер усе ще зберігає в кеш-пам'яті параметри сеансу й бажає відновити з'єднання, то у відповідь відправляє своє привітальне повідомлення з тим же самим ідентифікатором сеансу. У цей момент клієнт і сервер обмінюються повідомленнями про зміну параметрів шифрування й завершенні формування з'єднання.

При обміні сертифікатами й ключами передаються дані, необхідні для генерації початкового секрету. Якщо використовується алгоритм RSA, то клієнт генерує випадкове початкове число й шифрує його за допомогою відкритого RSA-ключа сервера. Якщо застосовується алгоритм Діффі-Хеллмана, клієнт і сервер обмінюються відкритими ключами, а потім як початковий секрет використовується результат обчислення ключа узгодження ключів по алгоритму Діффі-Хеллмана.

Головний секрет і симетричні ключі генеруються за допомогою псевдовипадкової функції PRF (PseudoRandom Function), отриманої на основі алгоритмів SHA-1 і MD5. Щоб гарантувати безпеку, застосовуються дві різні односпрямовані геш-функции. При першому застосуванні функції PRF генерується головний секрет з початкового секрету й випадкових чисел, отриманих на основі привітальних повідомлень клієнта й сервера. У результаті повторного застосування функції PRF до тих же самим вихідним даним одержують два симетричних ключі, два значення початкового вектора й два значення MAC (коду автентифікації повідомлення) секрету. При поновленні сеансу використовується вже відоме для даного сеансу значення головного секрету, але генеруються нові випадкові числа на основі нових привітальних повідомлень клієнта й сервера, тому в результаті формується новий ключовий матеріал.

Протокол передачі записів використовує один симетричний ключ, одне значення початкового вектора й один MAC секрет для захисту трафіку "клієнт-сервер", а також інші значення перерахованих параметрів для захисту трафіку "сервер-клієнт", – у цілому, для коректної роботи протоколу передачі записів потрібно шість секретних чисел.

Протокол передачі записів

Протокол передачі записів Record Protocol складається з декількох підрівнів. Обробка даних протоколу прикладного рівня полягає в їхній розбивці на керовані блоки, стиску, постачанні імітовставки, шифруванні й наступній передачі результату. Отримані дані обробляються у зворотному порядку: розшифровуються, перевіряються на цілісність, піддаються декомпресії, складанню, а потім доставляються додатку [6].

На підрівні фрагментації інформація розбивається на записи, довжина кожного запису не перевищує 16384 байтів. Допускається агрегування декількох однотипних повідомлень в один запис, а також розбивка одного повідомлення протоколу прикладного рівня на кілька записів. На підрівні стиску виконується стиск або декомпресія всіх фрагментів. Очевидно, що при компресії не повинне бути втрати даних. Потім обчислюється імітовставка, тобто перевіряється цілісність стислого запису, а отримане значення й стислий запис шифруються. Перевірка цілісності здійснюється за допомогою коду автентифікації повідомлення (MAC) або коду автентифікації, отриманого на основі геш-коду повідомлення (HMAC).

Після прийняття даних текст розшифровується, для перевірки його цілісності повторно обчислюється MAC, причому обчислення виконуються з використанням порядкового номера запису з метою виявлення загублених, зайвих або повторно стислих записів.

Підтримка безпеки транспортного рівня на основі PKI

Сертифікати є центральним компонентом всіх сервісів автентифікації й керування ключами, пропонованих як TLS, так і SSL. Ці сервіси покладаються на зв'язування ідентичності суб'єкта з його відкритим ключем. Для ідентифікації web-серверів рекомендується використовувати DNS-імена (типу www.alpha.com) і вказувати їх у доповненні сертифікатів Subject Alternative Name (параметр dNSName). Якщо DNS-ім'я не представлене в сертифікаті, то для ідентифікації використовується відмітне ім'я суб'єкта.

Зазвичай при взаємодії клієнта й сервера сервер пред'являє сертифікат, а клієнт – ні, у результаті чого відбувається однобічна автентифікація сервера клієнтом, що зберігає свою анонімність. Сервер може зажадати від клієнта автентифікації, запитуючи сертифікат по протоколі Handshake Protocol.

У цьому випадку клієнт повинен мати сертифікат і надати його серверу. Зазвичай клієнт пред'являє сертифікат ключа підпису.

У процесі верифікації завіреного цифровим підписом повідомлення, відправленого по протоколу Handshake Protocol, з'ясовується, чи здатний клієнт згенерувати підпис за допомогою свого секретного ключа, що відповідає відкритому ключу підпису, що втримується в сертифікаті.

Сертифікати, використовувані для підтримки безпеки транспортного рівня, також повинні містити доповнення Key Usage, що відбиває відповідне призначення відкритого ключа, що втримується в сертифікаті:

– цифровий підпис, якщо необхідно виконувати верифікацію підписів;

– шифрування ключів, щоб забезпечити RSA-шифрування;

– узгодження ключів, якщо необхідно підтримку узгодження ключів методом Діффі-Хеллмана.

Клієнт повинен бути впевнений, що відкритий ключ належить серверу. Якщо використовується некоректний відкритий ключ, то стороння особа може одержати початковий секрет і можливість згенерувати головний секрет і всі інші секретні параметри, що обчислюються з його допомогою. Сертифікат підтверджує приналежність відкритого ключа серверу.

При автентифікації клієнта сервер покладається на приналежність відкритого ключа клієнтові й ухвалює рішення щодо можливості доступу. Якщо використовується некоректний відкритий ключ, то доступ до сервера може одержати сторонню особу, а не та сторона, який доступ необхідний. Сертифікат забезпечує необхідне зв'язування відкритого ключа з ідентичністю клієнта.

Засоби безпеки IP-рівня

Сукупність механізмів IPsec забезпечує основу для захисту мережного трафіку на IP-рівні, безпеку IP-пакетів, захищеного взаємодії мобільних систем з корпоративною мережею, реалізації віртуальних приватних мереж (Virtual Private Networks – VPN) і т.п. Сімейство специфікацій IPsec представлено серією з 10 документів, розроблених робочою групою IP Security Protocol організації IETF і утримуючі відомості про архітектуру IPsec [14],

формуванні контекстів безпеки, керуванні ключами й базовими протоколами. Ядро IPsec становлять три протоколи: протокол автентифікуючого заголовку (Authentication Header, AH) [14], протокол інкапсулюючий захист вмісту (Encapsulating Security Payload, ESP) [14] і протокол обміну ключами в Інтернеті (Internet Key Exchange, IKE) [14]. Функції по підтримці захищеного каналу передачі даних по мережах IP розподіляються між цими протоколами в такий спосіб:

–протокол AH забезпечує цілісність IP-пакетів, автентифікацію джерела даних, а також захист від відтворення раніше переданих IP-пакетів;

–протокол ESP підтримує конфіденційність, автентифікацію й цілісність IP-пакетів, а також частковий захист від аналізу трафіку;

–протокол IKE дозволяє взаємодіючим сторонам автоматично генерувати й безпечно розподіляти симетричні секретні ключі.

Контексти безпеки

Контексти безпеки (Security Associations) утворюють основу криптографічних сервісів безпеки на базі протоколів IPsec. Для захисту двостороннього зв'язку між вузлами мережі необхідні два контексти безпеки: один – для вхідних потоків, інший – для вихідних. Контексти безпеки містять інформацію про IP-адреси, тип захисного протоколу (AH або ESP), криптографічних алгоритмах, ключах для автентифікації й шифрування й періоді їхньої дії.

Контекст безпеки унікально ідентифікується трьома елементами:

–індексом параметрів безпеки (Security Parameters Index – SPI);

–цільовою IP-адресою;

–ідентифікатором захисного протоколу.

Таблиця 1 – Режими, використовувані для різних типів з'єднань

	Хост	Маршрутизатор або міжмережний екран
Хост	Транспортний режим або тунельний режим	Тунельний режим
Маршрутизатор або міжмережний екран	Тунельний режим	Тунельний режим

Тунельний режим зазвичай реалізують на спеціально виділених захисних шлюзах, у ролі яких можуть виступати маршрутизатори або міжмережні екрани (рисунок 1).

Рисунок 1 – Стеки протоколів у різних режимах

У транспортному режимі заголовок протоколу (AH або ESP) розташовується в стеці протоколів після заголовка вихідного IP-пакету й перед заголовками протоколів більш високого рівня [7]. У тунельному режимі заголовок протоколу (AH або ESP) розташовується в стеці протоколів між двома заголовками: після заголовка зовнішнього IP-пакета й перед заголовком внутрішнього вихідного IP-пакета (рисунок 1).

Протокол автентифікуючого заголовка AH

Протокол автентифікуючого заголовка AH забезпечує:

–цілісність IP-пакетів, даних протоколів більше високого рівня й певних полів IP-заголовків;

–автентифікацію джерела даних (на основі IP-адреси вузла мережі або ім'я кінцевого користувача);

–захист від помилкового відтворення раніше переданих IP-пакетів.

Контроль цілісності базується на перевірці коду автентифікації гешированного повідомлення Hashed Message Authentication Code (HMAC), що обчислюється за допомогою геш-функції MD5 або SHA-1 із секретним симетричним ключем, що відомий обом взаємодіючим сторонам.

Рисунок 2 ілюструє типові поля даних протоколу AH. Протокол містить п'ять полів: Next Header, Length, SPI, Sequence Number і Authentication Data.

Рисунок 2 – Поля даних протоколу AH

Поле Next Header (наступний заголовок) указує, який протокол більш високого рівня інкапсулюється за допомогою AH. У тунельному режимі це поле зазвичай містить IP v4 або IP v6, а в транспортному режимі – TCP, UDP або ICMP.

Поле Length (довжина) задає розмір заголовка протоколу AH. Розмір залежить від типу використовуваної геш-функції, значення HMAC утримується в єдиному полі змінної довжини.

Поле SPI (індекс параметрів безпеки) містить 32-розрядне довільне значення, що ідентифікує контекст безпеки.

Поле Sequence Number (порядковий номер) використовується для завдання значення лічильника IP-пакетів (32-розрядного монотонно зростаючого) і захисту від відтворення пакетів. Відправник пакета повинен задавати це значення, а одержувач пакета може або обробляти його, або ігнорувати.

Поле Authentication Data (автентифікаціонные дані) містить значення HMAC для даного IP-пакета. Це поле має змінну довжину, що повинна бути кратна 32 розрядам.

При передачі пакета його порядковий номер, що вказується в поле Sequence Number, збільшується, а потім поля IP-заголовка й протоколу більш високого рівня гешируются для створення HMAC на основі загального секретного симетричного ключа. Після одержання IP-пакета одержувачем виконується та ж сама послідовність операцій. Якщо обчислене їм значення HMAC не відповідає значенню, отриманому по протоколу AH, то пакет не приймається. Крім того, якщо контекст безпеки містить інформацію про застосування засобу захисту від відтворення пакетів, то значення поля Sequence Number зменшується на одиницю, тобто відновлюється колишнє значення лічильника IP-пакетів.

Протокол інкапсулюючий захист вмісту ESP

Протокол інкапсулюючий захист вмісту ESP підтримує конфіденційність, автентифікацію й цілісність IP-пакетів. Конфіденційність забезпечується шляхом шифрування вмісту IP-пакетів, а також частини заголовка й трейлера (хвостової частини) протоколу ESP; надійність шифрування залежить, насамперед, від використовуваного

алгоритму шифрування. Автентифікація джерела даних і захист цілісності здійснюється на основі HMAC (як і в протоколі AH). Хоча сервіси конфіденційності й автентифікації (який включає цілісність) є опціональними, у кожному контексті безпеки повинен бути заданий, принаймні, один сервіс безпеки.

Як алгоритми шифрування в протоколі ESP використовуються алгоритми DES і Triple-DES, для обчислення HMAC застосовується геш-функція типу MD5 або SHA-1. рисунок 3 ілюструє типові поля даних протоколу ESP. Заголовок ESP містить два поля: SPI і Sequence Number, їхній синтаксис і семантика збігається з однойменними полями протоколу AH. Трейлер ESP складається із чотирьох полів: Padding, Pad Length, Next Header і Authentication Data.

Поле Padding (заповнювач) використовується для того, щоб розмір шифруемых даних був кратний розміру криптографічного блоку.

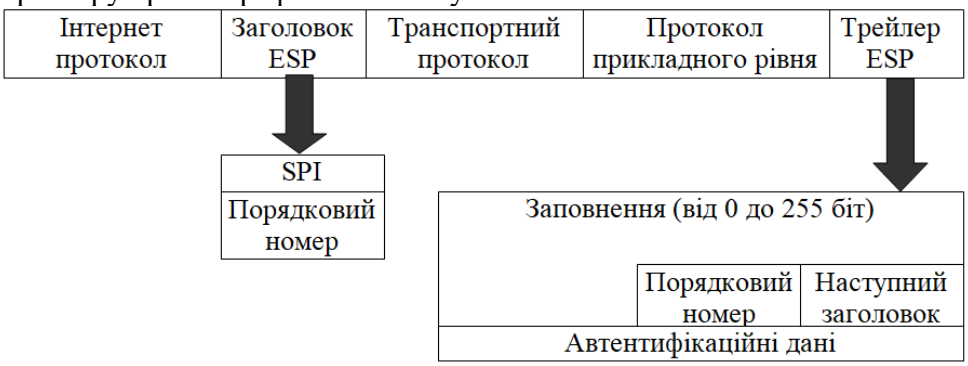


Рисунок 3 – Поля даних протоколу ESP

Поле Pad Length (довжина заповнювача) характеризує розмір заповнювача й залежить від використовуваного алгоритму шифрування й заданого рівня конфіденційності IP-трафіку.

Поле Next Header (наступний заголовок) містить інформацію про те, який протокол більше високого рівня інкапсулюється за допомогою ESP. У тунельному режимі це поле зазвичай містить IP v4 або IP v6, а в транспортному режимі – TCP, UDP або ICMP.

Поле Authentication Data (автентифікаційні дані) містить значення HMAC для даного IP-пакета. Це поле має змінну довжину, що повинна бути кратна 32 розрядам. Якщо автентифікація джерела даних або захист цілісності не потрібна, то це поле відсутнє або має нульову довжину.

При передачі пакета його порядковий номер, що вказується в поле Sequence Number, збільшується, а потім поля заголовка ESP, протоколу більш високого рівня й трейлера ESP гешуються для створення HMAC на основі загального секретного симетричного ключа. Потім поля протоколу більш високого рівня й трейлер ESP (за винятком автентифікаціоних даних) шифруються; якщо необхідно початковий вектор, то він випереджає шифртекст. Після одержання IP-пакета одержувачем виконується розшифрування й розрахунок того ж самого значення HMAC. Якщо обчислене їм значення HMAC не відповідає значенню, отриманому в трейлері ESP, то пакет не приймається. Крім того, якщо контекст безпеки містить інформацію про використання засобу захисту від відтворення пакетів, то значення поля Sequence Number зменшується на одиницю, тобто відновлюється колишнє значення лічильника IP-пакетів.

Протокол обміну ключами IKE

Широке використання IPsec вимагає масштабованого, автоматизованого керування контекстами безпеки. Формування контекстів безпеки по запиту й використання засобів захисту від відтворення пакетів у протоколах AH і ESP неможливо без роботи протоколу обміну ключами в Інтернеті IKE [14]. Протокол IKE розроблений на основі протоколу керування ключами й контекстами безпеки Інтернету – Internet Security Associations and Key Management Protocol (ISAKMP) [14] і протоколу обчислення ключів – OAKLEY Key Determination Protocol (OAKLEY) [14]. Протокол ISAKMP забезпечує незалежну від

криптографічного механізму автентифікацію й задає структуру обміну ключами. Протокол IKE базується на функціональності протоколу ISAKMP, а для формування симетричного ключа використовує можливості протоколу OAKLEY. Як алгоритм узгодження ключів застосовується алгоритм Діффі-Хеллмана.

Робота протоколу IKE виконується за два етапи. На першому етапі встановлюється автенифікований і шифруємий канал зв'язку. Це вимагає формування двох контекстів безпеки – по одному для зв'язку в одному напрямку. Для автентифікації сторін використовуються сертифікати відкритих ключів підпису (як алгоритм цифрового підпису застосовується DSA). На другому етапі формуються контексти безпеки для AH і ESP.

Підтримка безпеки IP-рівня на основі PKI

Сертифікати служать головними компонентами автентифікації на основі IKE. Коли ідентифікується хост або захисний шлюз, найбільш кращою формою ідентифікаційних даних є DNS-імена або IP-адреси, які вказуються в доповненні сертифіката Subject Alternative Name (параметри dNSName і iPAddress відповідно). При ідентифікації користувача рекомендується використовувати адресу електронної пошти або звичайне ім'я. Адреса електронної пошти втримується в доповненні сертифіката Subject Alternative Name, а звичайне ім'я входить до складу відмітного ім'я суб'єкта. Неможливість зв'язати ідентичність суб'єкта з його відкритим ключем робить автентифікацію неможливою. Неправильна автентифікація на основі протоколу IKE може привести до помилкового зв'язування ключа й реалізації IPsec, у результаті неавторизований користувач (або навіть група комп'ютерів) може одержати доступ, наприклад, до віртуальної приватної мережі.

Сертифікати, призначені для захисту трафіку Інтернету, повинні включати доповнення Key Usage, що відбиває відповідне призначення відкритого ключа, що втримується в сертифікаті (наприклад, цифровий підпис, якщо необхідно виконувати верифікацію підписів). Крім того, у доповненні сертифіката Extended Key Usage повинні явно вказуватися ті додатки, для підтримки яких призначений даний відкритий ключ, зокрема додаток IPsec.

Отже, сертифікати є базовим компонентом сервісів безпеки, надаваних S/MIME, TLS і IPsec.

S/MIME за допомогою сертифікатів ідентифікує користувачів. Сервіси автентифікації, цілісності, невідказуємості й конфіденційності захищеної електронної пошти залежать від сертифікатів. Сертифікати підтримують шифрування й завірення цифровим підписом поштових повідомлень.

TLS за допомогою сертифікатів ідентифікує користувачів і сервери. Від сертифікатів залежать сервіси автентифікації, цілісності й конфіденційності. Сертифікати підтримують шифрування потоку, автентифікацію потоку й цілісність потоку.

IPsec за допомогою сертифікатів ідентифікує користувачів, хости й шлюзи безпеки. Від сертифікатів залежать сервіси автентифікації. На базі сертифікатів виконується взаємна автентифікація взаємодіючих сторін при обміні відкритими ключами Діффі-Хеллмана, які, у свою чергу, використовуються для безпечного розподілу симетричних секретних ключів. Секретні ключі забезпечують підтримку автентифікації, цілісності й конфіденційності IP-пакетів.

Типові сценарії використання PKI

Повнофункціональна PKI – це ідеальне подання можливої інфраструктури, поки невідомі PKI-продукти, що реалізують всі перераховані в таблиці 2 функції. Сучасні PKI зазвичай призначені для рішення певної задачі або ряду задач. Конкретні реалізації PKI являють собою деякі підмножини повнофункціональної архітектури.

Таблиця 2 – Повнофункціональна PKI

УЦ	**Репозиторій**	**Анулювання сертифікатів**
Резервне зберігання ключів	Відновлення ключів	Автоматичне відновлення ключів
Керування історіями ключів	Крос-сертифікація	Клієнтське ПЗ
Автентифікація	Цілісність	Конфіденційність
Захищене датування	Нотаризація	Невідказуємість
Захищений архів даних	Розробка повноважень/політики	Перевірка повноважень/політики

Розглянемо чотири розповсюджених на сьогоднішній день сценарія використання PKI [4]. У таблиці 3 представлений Інтернет-PKI, що підтримує звичайну електронну пошту (між знайомими) і навігацію в World Wide Web за допомогою SSL-сервера автентифікації. Такий сценарій вимагає наявності УЦ для випуску сертифікатів відкритих ключів і підтримки основних сервісів автентифікації, цілісності й конфіденційності.

У цьому сценарії не передбачене використання репозитория (сертифікати пересилаються по протоколі зв'язку), не виконується перевірка статусу одержувача електронної пошти (або навіть сертифіката сервера) і керування життєвим циклом ключів і сертифікатів, відсутнє клієнтське програмне забезпечення (як окремий модуль, викликуваний за допомогою браузера), не потрібно ні крос-сертифікація, ні додаткові сервіси, що базуються на PKI.

Таблиця 3 – Інтернет-PKI

УЦ	Репозиторій	Анулювання сертифікатів
Резервне зберігання ключів	Відновлення ключів	Автоматичне відновлення ключів
Керування історіями ключів	Крос-сертифікація	Клієнтське ПЗ
Автентифікація	**Цілісність**	**Конфіденційність**
Захищене датування	Нотаризація	Невідказуємість
Захищений архів даних	Розробка повноважень/політики	Перевірка повноважень/політики

Таблиця 4 ілюструє функції PKI у сценарії, коли для доступу до корпоративної мережі ззовні використовується браузер і виконується SSL-автентифікація клієнтів.

Таблиця 4 – Екстранет-безпека (через SSL-автентифікацію клієнтів)

УЦ	Репозиторій	**Анулювання сертифікатів**
Резервне зберігання ключів	Відновлення ключів	Автоматичне відновлення ключів
Керування історіями ключів	Крос-сертифікація	Клієнтське ПЗ
Автентифікація	**Цілісність**	**Конфіденційність**
Захищене датування	Нотаризація	Невідказуємість
Захищений архів даних	**Розробка повноважень/політики**	**Перевірка повноважень/політики**

У таблиці 5 представлений набір функцій PKI для сценарію захищеної корпоративної електронної пошти. У цьому сценарії може знадобитися керування життєвим циклом ключів і сертифікатів і вбудоване клієнтське програмне забезпечення, тому що стандартні пакети електронної пошти не завжди підтримують безпеку, засновану на PKI. У цьому випадку не потрібні додаткові сервіси, що базуються на PKI, і крос-сертифікація.

Нарешті, у сценарії підтримки міжкорпоративних транзакцій з використанням цифрових підписів можуть знадобитися багато можливостей повнофункціональної PKI, зокрема сильна автентифікація й авторизація, перевірка статусу сертифікатів, розробка й перевірка повноважень і політики, сервіс невідказуємості (підтримка множинних пар ключів, зберігання прийнятих електронних документів із цифровим підписом і т.д.). Якщо корпорації мають свої власні PKI, то необхідно крос-сертифікацію. У даному сценарії можна обійтися без архівування даних, датування й нотаризації.

Таблиця 5 – Захищена корпоративна електронна пошта

УЦ	**Репозиторій**	**Анулювання сертифікатів**
Резервне зберігання ключів	Відновлення ключів	Автоматичне відновлення ключів
Керування історіями ключів	Крос-сертифікація	Клієнтське ПЗ
Автентифікація	Цілісність	Конфіденційність
Захищене датування	Нотаризація	Невідказуємість
Захищений архів даних	Розробка повноважень/політики	Перевірка повноважень/політики

Таблиця 6 – Міжкорпоративні транзакції із цифровим підписом

УЦ	**Репозиторій**	**Анулювання сертифікатів**
Резервне зберігання ключів	Відновлення ключів	Автоматичне відновлення ключів
Керування історіями ключів	Крос-сертифікація	Клієнтське ПЗ
Автентифікація	Цілісність	Конфіденційність
Захищене датування	Нотаризація	Невідказуємість
Захищений архів даних	Розробка повноважень/політики	Перевірка повноважень/політики

Таблиці 3, 4, 5 і 6 підтверджують, що PKI, що реалізують приватні сценарії, є підмножинами повнофункціональної PKI. Технологія PKI продовжує розвиватися, але вже зараз ясно, що багато постачальників програмного й апаратного забезпечення PKI будуть орієнтуватися на реалізацію повнофункціональних систем, а не на PKI-продукти вузького призначення. Очевидно, що в багатьох випадках простіше й економічно більш ефективно адаптувати повнофункціональний продукт для рішення специфічної проблеми, чим розробляти й підтримувати кілька окремих продуктів, кожний з яких призначений для рішення однієї або двох специфічних проблем. У багатьох середовищах PKI відбудеться неминучий перехід від приватних рішень приватних проблем до повнофункціонального PKI, що пропонує універсальне рішення проблем безпеки для широкого кола додатків.

Розробка структурної схеми

На рисунку 4 зображено структурну схему інфраструктури відкритих ключів, та місце S/MIME у ній. Розглянемо компоненти структурної схеми. Почнемо розгляд з точену. Замість використання доступу по паролю до корпоративних ресурсів, для входу в домен і при використанні корпоративної пошти можна використовувати апаратну автентифікацію й захист електронної переписки в мережах, побудованих на базі Windows 10/11. У пропонованому рішенні використовуються вбудовані інструменти безпеки Windows і електронні ідентифікатори **токен** як носії ключової інформації.

Що забезпечується при використанні цього продукту? Авторизація користувачів (вхід у домен при підключенні токена й блокування сесії після його від'єднання). При роботі з поштою – електронний цифровий підпис поштових повідомлень і їхнє шифрування. Доступ

до корпоративних ресурсів по пред'явленні токена й, звичайно, надійне зберігання й використання сертифікатів. Частина з перерахованих задач може бути використана й на домашньому комп'ютері, наприклад, при роботі з поштою, а також для віддаленого підключення до корпоративних ресурсів. Давайте розберемо, що і як необхідно виконати для реалізації перерахованих задач.

Токен як додатковий пристрій не може бути пізнаний операційною системою, оскільки він не є стандартним устаткуванням. Неможлива й установка токена за допомогою inf-файлу, оскільки для коректної установки потрібний вимір деяких параметрів для автоматичної конфігурації драйвера. Тому доводиться для установки використовувати спеціальне програмне забезпечення. І от що ще варто пам'ятати. Не підключайте токен до комп'ютера до того, як ви встановите драйвер. Якщо все-таки підключення буде виконано раніше, припиніть роботу стандартного майстра установки USB-пристроїв, витягніть ключ із порту й установіть драйвер. Крім драйверів у процесі інсталяції на диск будуть скопійовані й утиліти для роботи з токеном (утиліта адміністрування й браузер сертифікатів). Кількість одночасно використовуваних токенів залежить від операційної системи й кількості USB-портів.

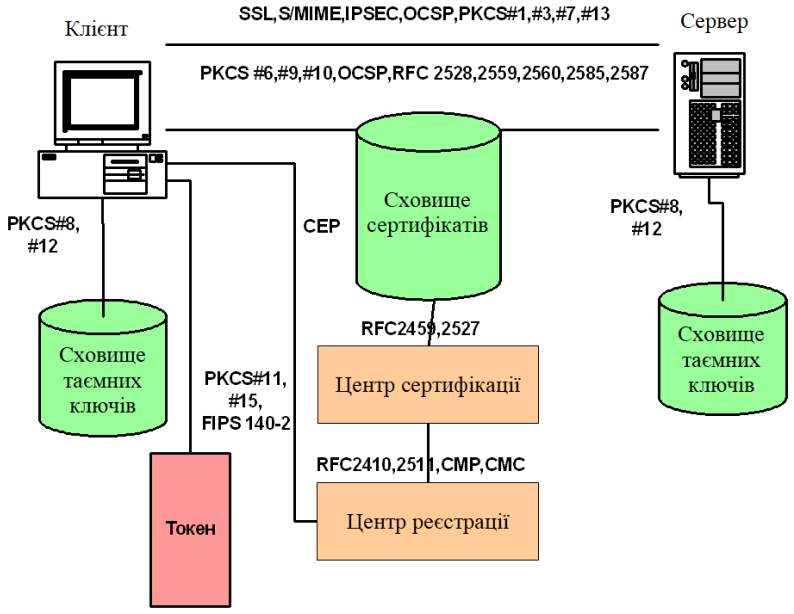


Рисунок 4 – Структурна схема PKI

Центр сертифікації (Удостовірюючий центр) (Certification authority, CA) – це організація, що випускає сертифікати ключів електронного цифрового підпису.

Центр сертифікації – це компонента глобальної служби каталогів, відповідальназа керування криптографічними ключами користувачів. Відкриті ключі й інша інформація про користувачів зберігається центрами сертифікації у вигляді цифрових сертифікатів, що мають наступну структуру:

– серійний номер сертифіката;
– об'єктний ідентифікатор алгоритму електронного підпису;
– ім'я центра, що засвідчує;
– строк придатності;
– ім'я власника сертифіката (ім'я користувача, якому належить сертифікат);
– відкриті ключі власника сертифіката (ключів може бути трохи);
– об'єктні ідентифікатори алгоритмів, асоційованих з відкритими ключами власника сертифіката; електронний підпис, згенерований з використанням секретного ключа центра, що засвідчує (підписується результат гешування всієї інформації, що зберігається в сертифікаті).

Центр реєстрації – опціональна компонента інфраструктури, призначена для реєстрації кінцевих користувачів і забезпечення їхньої взаємодії із центром сертифікації.

CRL – список відозваних сертифікатів. Оснащення «Центру сертифікації» можна використовувати для відкликання сертифіката, для адміністрування публікації списків відкликання сертифікатів (CRL) і для завдання точок поширення списків CRL, які опубліковані в кожному сертифікаті, виданому центром сертифікації (ЦС).

Відкликання сертифікатів

Щоб допомогти в досягненні цілісності інфраструктури відкритого ключа (PKI) організації, адміністратор ЦС відзиває сертифікат, якщо суб'єкт сертифіката залишає організацію, порушений закритий ключ сертифіката або існують інші причини, по яких сертифікат більше не може вважатися дійсним. При відкликанні сертифіката ЦС він додається в список відкликання сертифікатів (CRL) цього ЦС. Це може відбуватися зі створенням нового списку CRL або з використанням різницевого списку CRL, що є невеликим списком сертифікатів, відкликаних з моменту останнього заповнення списку CRL.

Розклад публікації списку відкликання сертифікатів (CRL)

Одна з можливостей служб сертифікації складається в автоматичній публікації оновленого списку CRL по закінченні певного періоду часу, заданого адміністратором ЦС. Цей інтервал часу називається періодом публікації CRL. Після початкової установки ЦС період публікації встановлюється рівним одному тижню (відповідно до часу на локальному комп'ютері, починаючи з дати початкової установки ЦС). Періоди публікації списків CRL і різницевого CRL можуть задаватися незалежно.

LDAP (Lightweight Directory Access Protocol – «полегшений протокол доступу до каталогів») – це мережний протокол для доступу до служби каталогів X.500, розроблений IETF як полегшений варіант розробленого ITU-T протоколу DAP. LDAP – відносно простий протокол, що використовує TCP/IP і дозволяє робити операції автентифікації (bind), пошуку (search) і порівняння (compare), а також операції додавання, зміни або видалення записів. Звичайно LDAP-Сервер приймає вхідні з'єднання на порт 389 по протоколах TCP або UDP. Для LDAP-сеансів, інкапсульованих в SSL, звичайно використовується порт 636.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів повідомлень електронної пошти. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем повідомлень електронної пошти. Досліджена система повідомлень електронної пошти. На основі отриманих результатів досліджень створена програмна реалізація системи повідомлень електронної пошти. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання повідомлень електронної пошти. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).

2. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).

3. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).

4. Смірнова Т.В., Поліщук Л.І., Смірнов О.А., Буравченко К.О., Макевнін А.О., «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

5. Смірнов О.А. Дисперсійний аналіз мережного трафіку для забезпечення інформаційної безпеки телекомунікаційних систем / О.О. Кузнецов, О.А. Смірнов, Д.О. Даниленко // Інформаційна та економічна безпека: сучасний стан та тенденції розвитку : монографія за заг. ред. – Х.: ХІБС УБС НБУ – 2014 – С. 82-100.
6. Смірнов О.А. Дослідження методів виявлення вторгнень в телекомунікаційні системи та мережі / Д.О. Даниленко, О.А. Смірнов, Є.В. Мелешко // Системи озброєння і військова техніка. – Випуск 1(29) – Х.: ХУПС – 2012. – С. 92-100
7. Смирнов А.А. Метод обнаружения вредоносного программного обеспечения. Часть 1. Корреляционный анализ сетевого трафика // А.А.Смирнов, Д.А. Даниленко, Е.В.Мелешко // Науково-технічний журнал «Інформаційно-керуючі системи на залізничному транспорті» – Випуск 4(95). – Х.: УкрДАЗТ – 2012. – С. 8-14.
8. Смирнов А.А. Методы обнаружения вредоносного программного обеспечения в телекоммуникационных системах и сетях / Д.А. Даниленко // Збірник наукових праць "Системи обробки інформації". – Випуск 3(101) том 2. – Х.: ХУПС – 2012. – С. 152-155.
9. Смирнов А.А. Системы обнаружения и предотвращения вторжений для защиты телекоммуникационных сетей от вредоносного программного обеспечения / Д.А. Даниленко, А.А. Смирнов, А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 1 (21) том 2. – Київ: ДП «ЦНДІНУ». – 2012. – С. 183-186.
10. Смирнов А.А. Системы обнаружения и предотвращения вторжений для защиты компьютерных сетей от вредоносного программного обеспечения / Д.А. Даниленко, А.А. Смирнов, И.Г. Кирилов // Збірник тез доповідей науково-практичної конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». м. Харків. 21-22 березня 2012 р. – Харків. АВВ МВС. – 2012. – С. 70-71.
11. Смірнов О.А. Дослідження методів виявлення вторгнень в телекомунікаційні мережі для підвищення інформаційної безпеки // Д.О. Даниленко // Збірник тез науково-практичної конференції «Захист інформації в інформаційно-комунікаційних системах». м. Київ. 24-27 квітня 2012 р. – Київ: НАУ. – 2012. – С. 22-25.
12. Смирнов А.А. Исследование систем обнаружения и предотвращения вторжений для защиты телекоммуникационных сетей от вредоносного программного обеспечения / Д.А. Даниленко // Збірник тез доповідей VIII наукової конференції «Новітні технології – для захисту повітряного простору». Харків. 18-19 квітня 2012 р. – м. Харків. ХУПС. – 2012. – С. 45.
13. Смирнов А.А. Исследование методов сигнатурного обнаружения вредоносного программного обеспечения в телекоммуникационных системах и сетях // Д.А. Даниленко // Збірник тез XIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 13-14 квітня 2012 р. – Кіровоград: КНТУ. – 2012. – С. 43-45.
14. Смирнов А.А. Исследование методов проактивной защиты от вредоносного программного обеспечения в телекоммуникационных системах и сетях / Д.А. Даниленко // Збірник тез V міжнародної науково-практичної конференції «інтегровані інтелектуальні робототехнічні комплекси» (ІПРТК-2012). м. Київ. 15-16 травня 2012 р. – Київ: НАУ. – 2012. – С. 314-315.
15. Смирнов А.А. Метод обнаружения вредоносного программного обеспечения на основе корреляционного анализа сетевого трафика / Д.А. Даниленко // Матеріали XII всеукраїнської наукової інтернет-конференції «Наукові дослідження: зв'язок теорії і практики». м. Тернопіль. 29-30 квітня 2012 р. – Тернопіль: ТНЕУ. – 2012. – С. 9-10.
16. Смирнов А.А. Метод детектирования вредоносного трафика в телекоммуникационных сетях на основе использования bds-тестирования / Д.А. Даниленко // Збірник тез V міжнародної науково-практичної конференції «Комп'ютерні системи та мережні технології» (CSNT-2012). м. Київ. 13-15 червня 2012 р. – Київ: НАУ. – 2012. – С. 121.
17. Смирнов А.А. Обнаружение и предотвращение вторжений в компьютерных сетях на основе статистического анализа сетевого трафика / А.А. Смирнов, Д.А. Даниленко // Збірник тез доповідей науково-практичної конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». м. Харків. 12-13 березня 2014 р. – Харків. АВВ МВС. – 2014. – С. 13-14.
18. Смірнов О.А. дисперсійний аналіз мережного трафіку для забезпечення інформаційної безпеки телекомунікаційних систем та мереж / О.А. Смірнов, Д.О. Даниленко // Збірник тез V Всеукраїнської науково-практичної конференції "Інформатика та системні науки". м. Полтава. 13-15 березня 2014 р. – Полтава: ПУЕТ. – 2014. – С. 289-291.
19. Смирнов А.А. Метод дисперсионного анализа сетевого трафика для обнаружения и предотвращения вторжений в телекоммуникационных системах и сетях/ А.А. Смирнов, Д.А. Даниленко // Збірник тез VI міжнародної науково-практичної конференції "Проблеми і перспективи розвитку ІТ-індустрії". м. Харків. 17-18 квітня 2014 р. – Харків: ХНЕУ. – 2014. – С. 258.
20. Смірнов О.А. метод забезпечення інформаційної безпеки телекомунікаційних систем з використанням дисперсійного аналізу мережного трафіку / О.А. Смірнов, Д.О. Даниленко // Збірник тез міжнародної науково-практичної конференції «Інформаційна та економічна безпека» (INFECO-2014)». м. Харків. 15-16 травня 2014 р. – Харків: ХІБС УБС НБУ. – 2014. – С. 135-139.

УДК 004

О.Підлубний, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ВІДДАЛЕНОГО ДОСТУПУ З ВИКОРИСТАННЯМ WAN-МЕРЕЖ

У статті розроблено програмне забезпечення, яке призначено для системи віддаленого доступу з використанням WAN-мереж. Метою розробки є дослідження та програмна реалізація системи віддаленого доступу з використанням WAN-мереж. Об'єктом дослідження є процес віддаленого доступу з використанням WAN-мереж. Предметом дослідження є методи віддаленого доступу з використанням WAN-мереж. Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи віддаленого доступу з використанням WAN-мереж. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, віддалений доступ, WAN-мережі

Постановка проблеми. Сучасний світ тяжко представити без наявності мереж. У тому або іншому вигляді кожна людина на даний момент стикається з мережами: починаючи від роботи та спілкування в Інтернет й закінчуючи роботою з локальною мережею дома, або на роботі. Ще більше проникнення мереж у життя відбулося з введенням електронного документообігу й електронних платежів (у тому числі й отримання грошей через мережу банкоматів).

Таке глибоке проникнення мережевих технологій приводить до того, що доволі часто виникає потреба у віддаленому управлінні ЕОМ через локальну мережу або Інтернет та контролі того, що відбувається на віддаленій ЕОМ.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи віддаленого доступу з використанням WAN-мереж.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи віддаленого доступу з використанням WAN-мереж.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем віддаленого доступу з використанням WAN-мереж.
– Дослідження системи віддаленого доступу з використанням WAN-мереж.
– Програмна реалізація системи віддаленого доступу з використанням WAN-мереж.

Об'єктом дослідження є процес віддаленого доступу з використанням WAN-мереж.

Предметом дослідження є методи віддаленого доступу з використанням WAN-мереж.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Віддалений доступ дозволяє користувачам отримувати доступ до пристрою або мережі з будь-якого місця, що полегшує керування файлами та даними, що зберігаються на віддаленому пристрої . Це сприяє безперебійній співпраці та продуктивності будь-де. Читайте далі, щоб дізнатися більше про основи віддаленого доступу та про те, як він може принести користь вашій організації.

Що таке віддалений доступ і як він працює?

Віддалений доступ передбачає підключення комп'ютера або мережі в одному місці та пристрою в іншому місці.

Є кілька способів досягти цього. Найбільш поширеним є або через віртуальну приватну мережу (VPN), або через спеціальне програмне забезпечення, наприклад інструмент віддаленого доступу. Для організацій зі складними потребами у віддаленому доступі інструменти віддаленого моніторингу та керування (RMM) часто використовуються для спрощення великомасштабного керування.

Щоб використовувати VPN, обидва пристрої повинні мати підключення до Інтернету. VPN створює безпечний тунель, який забезпечує конфіденційність і плавний потік трафіку. Сервер VPN функціонує як шлюз на межі мережі, спрямовуючи трафік до відповідних хостів у мережі.

Для передачі інформації програмне забезпечення VPN бере трафік і загортає його в захисний шар шифрування. Пакети даних надсилаються через Інтернет різними шляхами залежно від доступності мережі. Досягнувши пункту призначення, шлюз надсилає відповідь, також зашифровану, клієнту VPN, завершуючи процес у зворотному порядку.

Тим часом програмне забезпечення віддаленого доступу – це програмне забезпечення, яке можна завантажити на свій пристрій. Він складається з «агента» та «платформи». Ви встановлюєте агента на свої ноутбуки, ПК та інші пристрої, одночасно розгортаючи платформу в мережі, до якої хочете підключитися. Коли інструмент віддаленого доступу активний, вам більше не потрібен VPN, оскільки платформа автоматично розпізнає агента та дозволяє підключитися.

Хоча обидва ці методи відрізняються за застосуванням, вони дозволяють віддалений доступ і зв'язок між пристроями.

З іншого боку, інструмент віддаленого моніторингу та керування (RMM) дозволяє організаціям виконувати операції в масштабі для всіх своїх віддалених пристроїв. Уявіть, що вам доводиться оновлювати тисячі пристроїв по одному віддалено. За допомогою інструментів RMM організації можуть оптимізувати свої IT-операції, автоматизувати рутинні завдання та скоротити час простою, забезпечуючи оптимальну продуктивність і підвищення продуктивності. IT-команди можуть віддалено розгортати оновлення програмного забезпечення, застосовувати політики безпеки та надавати технічну підтримку кінцевим користувачам, незалежно від місця розташування, усуваючи виснажливу роботу.

Чому віддалений доступ до комп'ютерів важливий для бізнесу?

Віддалений доступ до комп'ютерів дуже важливий для бізнесу в цьому сучасному світі. Ось лише кілька важливих способів зв'язку між організаціями та працівниками віддаленого доступу:

– **Підвищує гнучкість –** гнучкість у робочому місці та графіку може підвищити продуктивність співробітників, дозволяючи їм керувати своїм часом і позбавляючи їх виснажливих поїздок.

– **Зменшує витрати –** коли працівникам не потрібно фізично перебувати в офісі, ви можете зменшити витрати, пов'язані з офісним приміщенням, обладнанням і поїздками. Віддалений доступ також добре працює з політикою Bring Your Own Device (BYOD), яка може позбавити підприємства від значних інвестицій у нові комп'ютери для співробітників.

– **Покращує безпеку.** Ви можете уникнути порушень кібербезпеки, обмеживши доступ до даних і програм на сайті лише тим, хто має певні дозволи на віддалений доступ. Зберігання даних за межами об'єкта також може зменшити ризик втрати даних у катастрофічних ситуаціях.

Дев'ять способів можливого віддаленого доступу:

1. DSL (цифрова абонентська лінія)

DSL (цифрова абонентська лінія) використовує телефонну мережу, DSL-модем і високошвидкісне підключення до Інтернету. DSL-модем підключається до мережі DSL і використовує існуючі телефонні лінії для передачі цифрових даних через Інтернет.

DSL забезпечує швидший і надійніший віддалений доступ, ніж стільниковий Інтернет, але це не завжди можливо без надійної інфраструктури.

2. Широкосмуговий кабель

Ймовірно, один із найпоширеніших методів віддаленого доступу включає кабельний модем і високошвидкісне підключення до Інтернету. Для безпечного підключення до цільового пристрою або мережі також потрібне програмне забезпечення VPN або RMM.

Віддалений доступ є відносно швидким і надійним за допомогою кабельного широкосмугового зв'язку, але він обмежений областями, де доступна кабельна інфраструктура.

3. Стільниковий Інтернет

Віддалений доступ можна забезпечити за допомогою стільникового Інтернету, пристрою з підтримкою стільникового зв'язку, наприклад смартфона або планшета, і тарифного плану передачі даних. Послуга стільникового Інтернету добре працює для забезпечення віддаленого доступу, але покладається на стабільне з'єднання.

4. Супутник

Інший спосіб підключення – через супутник із супутниковим модемом, супутниковою антеною або системою VSAT (термінал із дуже малою апертурою). Супутниковий модем використовується для передачі даних на супутник і з нього, тоді як супутникова антена або система VSAT встановлює зв'язок із супутником.

5. Оптоволоконна широкосмугова мережа

Оптоволоконний широкосмуговий доступ є одним із найкращих методів віддаленого доступу, особливо для роботи, яка потребує швидкої реакції та мінімальної затримки. Допомогло б, якби у вас зазвичай був оптоволоконний модем, високошвидкісне підключення до Інтернету та підключення до VPN або програмне забезпечення для віддаленого робочого столу.

Оптоволоконна широкосмугова мережа не зазнає впливу електромагнітних перешкод або втрати сигналу на великих відстанях – на відміну від мідних кабелів, які використовуються в DSL або кабельній широкосмуговій мережі.

6. VPN/ LAN/ WAN

Щоб отримати віддалений доступ через VPN/LAN/WAN, вам потрібна VPN, локальна мережа (LAN) або глобальна мережа (WAN), залежно від вимог. Це безпечна зашифрована мережа, доступ до якої мають лише ті, хто має дозвіл.

З'єднання LAN (локальна мережа) – це мережа, об'єднана в одному місці, наприклад, в офісі, кампусі або вдома.

Глобальна мережа (WAN) – це мережа, яка охоплює кілька місць, наприклад різні офіси або філії компанії.

7. Спільне використання робочого столу

Цей метод працює лише в парі з іншим рішенням, оскільки для нього потрібне підключення до Інтернету. Ви використовуєте програмне забезпечення для віддаленого робочого столу та налаштовуєте головний комп'ютер, щоб дозволити віддалені підключення. Звідти ви можете поділитися своїм робочим столом. Це чудово підходить для обміну презентаціями або для ознайомлення IT-команди з технічними проблемами.

Однак спільний доступ до робочого столу може становити загрозу безпеці, тому важливо налаштувати з'єднання, щоб воно було безпечним і зашифрованим. Перегляньте наш блог, оскільки ми маємо багато інформації про те, як вибрати безпечний інструмент віддаленого доступу.

8. PAM (керування привілейованим доступом)

PAM – це метод безпеки, який допомагає організаціям керувати та захищати привілейовані облікові записи, які мають доступ до конфіденційних систем і даних. PAM забезпечує безпечні шлюзи доступу, які дозволяють віддаленим користувачам підключатися до привілейованих облікових записів, а також дозволяють вказувати індивідуальні рівні доступу.

9. VPAM (керування привілейованим доступом постачальника)

VPAM – це метод безпеки, який дозволяє організаціям керувати та захищати привілейований доступ сторонніх постачальників, яким потрібен віддалений доступ до своїх систем.

Щоб це було ефективним, найкраще попрацювати зі своєю IT-командою, щоб визначити сторонніх постачальників, яким потрібен віддалений доступ постачальників, і обмежити їх необхідними областями.

Що таке протокол віддаленого доступу?

Протокол віддаленого доступу – це набір правил, які регулюють, як користувач або пристрій можуть віддалено отримувати доступ до комп'ютерної системи чи мережі та спілкуватися з ними.

Ці протоколи визначають методи автентифікації та передачі даних.

Протоколи віддаленого доступу зазвичай використовують механізми шифрування та автентифікації, щоб забезпечити безпеку та автентифікацію віддаленого доступу.

6 Типи протоколів віддаленого доступу

1. Інтернет-протокол послідовної лінії (SLIP)

SLIP – це протокол, який працює на каналі даних і фізичному рівнях моделі OSI і, як правило, забезпечує низькі накладні витрати.

Він може транспортувати TCP/IP через послідовні з'єднання, але не має можливості адресації пакетів і перевірки помилок. SLIP можна використовувати лише в послідовних з'єднаннях.

2. Протокол «точка-точка» (PPP)

PPP дає змогу реалізувати TCP/IP за допомогою зв'язків «точка-точка», виділених виділених ліній та комутованих з'єднань. Він в основному використовується для віддаленого підключення до локальних мереж і провайдерів.

PPP використовує протокол керування зв'язком (LCP) для встановлення зв'язку між клієнтом PPP і хостом.

PPP пов'язаний із високими накладними витратами та може бути несумісним із старішими конфігураціями.

3. Протокол тунелювання точка-точка (PPTP)

Створений корпорацією Майкрософт протокол PPTP – це протокол VPN, який забезпечує безпечний зв'язок між віддаленими клієнтами та приватними мережами через Інтернет.

PPTP також підтримує шифрування та стиснення даних, що передаються, забезпечуючи підвищену безпеку зв'язку.

4. Служби віддаленого доступу Windows (RAS)

RAS – це набір функцій і протоколів в операційних системах Microsoft Windows, які дозволяють користувачам віддалено підключатися до мережі або комп'ютера з іншого місця через Інтернет або приватну мережу.

RAS підтримує такі технології віддаленого доступу, як віртуальні приватні мережі (VPN), мережа віддаленого доступу (DUN) і DirectAccess.

5. Протокол віддаленого робочого стола (RDP)

Розроблений корпорацією Майкрософт протокол RDP дозволяє користувачеві віддалено отримувати доступ до іншого комп'ютера або віртуальної машини та керувати ним через мережеве з'єднання. RDP вбудовано в операційні системи Windows і може підключатися до іншого комп'ютера під керуванням Windows або віртуальної машини, що працює на віддаленому сервері.

6. Обчислення віртуальної мережі (VNC)

Подібно до RDP, VNC дозволяє користувачам віддалено керувати іншим комп'ютером. Однак у цьому випадку на віддаленому комп'ютері інстальовано сервер VNC (яким потрібно керувати), а на пристрої встановлено програму перегляду VNC для керування ним.

Це забезпечує більшу гнучкість, оскільки його можна використовувати на кількох пристроях і операційних системах і має можливості спільного використання екрана.

Підвищте ефективність вашої організації за допомогою інструменту віддаленого доступу RealVNC

RealVNC® виводить ефективність вашої організації на новий рівень. VNC Connect дозволяє дистанційно отримувати доступ до комп'ютера та керувати ним, у той час як основний користувач може з ним взаємодіяти. Це робить його ідеальним для віддаленої роботи, навчання та підтримки.

Ця складна технологія віддаленого доступу, створена з урахуванням безпеки, дозволяє вашій команді працювати з будь-якого місця, яке вона вибере, використовуючи розширені інструменти адміністрування, які надають вам повний контроль.

Завдяки вибору варіантів підключення та безпеки корпоративного рівня ви можете бути впевнені, що інструменти віддаленого доступу RealVNC – це ваше рішення для покращення співпраці та продуктивності у всіх сферах.

Опис загальної технології віддаленого керування комп'ютером через Інтернет. Апаратно-програмні вимоги. Отже, що нам буде потрібно, щоб одержати можливість віддаленого керування нашим домашнім комп'ютером:

–По-перше, це, звичайно ж, доступ домашнього комп'ютера в Інтернет, і необхідно, щоб інтернет-провайдер виділив нам пряму (зовнішню) IP-адресу.

–По-друге, необхідно встановити на комп'ютер спеціальне програмне забезпечення для віддаленого адміністрування.

–По-третє, комп'ютер повинен бути включений, із завантаженою операційною системою й всім комплексом програмного забезпечення, необхідного нам для віддаленого керування. Тримати ПК постійно включеним незручно, але проблема розв'язувана. Якщо залишився старий аналоговий dial-up модем, і BIOS материнської плати підтримує технологію Wake-on-Ring, то комп'ютер може залишатися виключеним. Включеним залишиться тільки модем, що при першому ж вхідному дзвінку «розбудить» комп'ютер, і до нього стане можливим звернутися через Інтернет. Головним мінусом даної технології є саме те, що модем спрацює на будь-який вхідний дзвінок, і, відповідно, це може привести до помилкового включення комп'ютера, але адже його потім можна знову відключити.

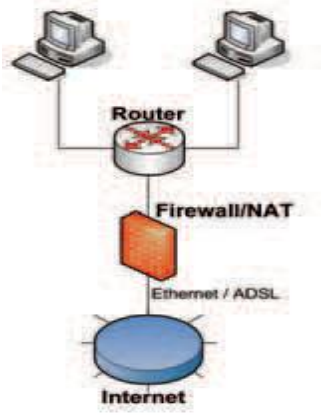


Рисунок 1 – Загальна схема віддаленого керування ПК

Детальніше варто зупинитися на підключенні до Інтернету. Найважливішу роль отут грає IP-адреса. Так, у випадку якщо домашньому комп'ютеру привласнюється пряма IP-адреса, ніяких труднощів виникнути не повинно, а от у випадку підключення через шлюз, і, відповідно, з «сірим» IP усе буде трохи складніше. Якщо підключатися через районну будинкову мережу, то прийдеться домовлятися із провайдером про надання прямої IP-адреси, а от якщо шлюз перебуває в будинку (наприклад, при використанні ADSL-модему в режимі роутера), треба просто забезпечити наскрізне проходження пакетів, адресованих нашому комп'ютеру. Як це зробити?

Для цього існує технологія NAT (Network Address Translation), що транслює запити із внутрішніх IP-адрес у зовнішню мережу, і навпаки. За замовчуванням NAT транслює тільки запити із внутрішньої мережі в зовнішню, а запити, що приходять із зовнішнього миру скидаються, просто тому що подальший маршрут проходження NAT'у невідомий. Для того щоб NAT перенаправляв запити на нашу машину, потрібно жорстко закріпити зовнішні TCP/UDP-порти шлюзу за певним комп'ютером. Для цього NAT'у варто вказати, запити для яких портів необхідно відправляти на адресу нашого ПК.

Підготовка комп'ютера

Отже, необхідно почати підготовку комп'ютера до віддаленого керування. Розглянемо випадок, коли вихід в Інтернет здійснюється через роутер (наприклад, нині популярні роутери CISCO 2801). Тоді схема підключення виходить приблизно така:

Wake-on-Ring

Для початку настроїмо опцію Wake-on-Ring. Якщо в тебе внутрішній модем, то необхідно з'єднати його зі спеціальним рознімання на материнській платі. Із зовнішнім модемом нічого додаткового не потрібно. Заходимо в BIOS материнської плати, у розділ з налаштуваннями живлення, знаходимо щось подібне Resume on Ring або Wake on Ring, і активуємо цю опцію. Тепер після вимикання комп'ютера (у випадку із зовнішнім модемом не забудь залишити його включеним) модем включить комп'ютер при першому ж дзвінку.

Невелике зауваження для тих, у кого будинку АВН. Принцип його функціонування такий, що для визначення номера він знімає трубку (ініціалізує з'єднання з АТМ) і далі дзвінки на телефони/модеми/факси, підключені паралельно АВНу, уже не проходять. Таким чином, у ланцюзі підключення АВН повинен бути першим пристроєм, а всі інші – включатися за ним.

Якщо ти хочеш, щоб комп'ютер автоматично відключався після помилкового дзвінка, необхідно встановити одну із програм, спеціально призначених для цих цілей.

У налаштуваннях програми обов'язково ставимо галочку «автоматично завантажуватися після запуску Windows», створюємо задачу «вимикання живлення», за умови відсутності активності курсору (рухів мишки) протягом десяти мінут (думаю, цього часу сповна вистачить, щоб вийти в Інтернет, зайти RAdmin'ом на комп'ютер і деактивувати цю утиліту) і підтверджуємо додавання задачі. Тепер комп'ютер при надходженні дзвінка ввімкнеться, завантажить Windows, після чого десять мінут буде покірно чекати твоїх вказівок, а після закінчення цього строку знову відключиться до наступного дзвінка.

Wake-on-LAN

Хтось може запитати, а чому б не використовувати для цих цілей технологію включення комп'ютера за допомогою локальної мережі Wake-on-LAN? Вся справа в принципі роботи цієї технології. Wake-on-LAN (а вірніше, її найпоширеніший різновид, Magic Packet) у чомусь схожа на Wake-on-Ring. Тут ключову роль виконує мережева карта, що при включенні функції Wake-on-LAN, продовжує працювати навіть після вимикання комп'ютера, і очікує спеціальний кадр, що будить. Інформація, що перебуває в цьому кадрі, являє собою шість байт синхронізації й шістнадцять разів повторену MAC-адресу мережевої карти-приймача. Послідовність упаковується в UDP, потім у пакет IP із широкомовною адресою, у кадр Ethernet, і адресується приймачу. Як адреса призначення використовується MAC-адреса Ethernet-адаптера, тобто адресація відбувається тільки на канальному рівні моделі OSI. Тому застосовувати технологію можна тільки в локальних мережах, не розділених на сегменти, або усередині одного сегмента. По цій же причині з'являються складності при посилці пакета з послідовністю, що будить, через мережу Internet.

Все це обмежує область застосування даної технології вузьким колом задач. Наприклад, при наявності в будинку декількох комп'ютерів, можно із одного з них включати інші. Для того щоб можна було скористатися цією функцією, її повинні підтримувати й мережева карта, і BIOS материнської плати. Налаштування так само просте, як і налаштування Wake-on-Ring. Заходимо в BIOS материнської плати в той же самий розділ з налаштуваннями живлення й знаходимо щось подібне Resume on LAN або Wake on LAN.

Активуємо цю опцію. Якщо материнська плата має шину PCI специфікації до 2.2 (довідатися, яку специфікацію шини PCI підтримує твоя материнська плати, ти можеш із інструкції до неї), то на ній повинен бути трьохштирьковий роз'єм «Wake On Lan». Аналогічний роз'єм повинен бути на мережевому адаптері. Їх потрібно з'єднати спеціальним кабелем, що входить у комплект поставки мережевого адаптера. Для випадку із шиною PCI 2.2 таке з'єднання вже виконане прямо. Тепер залишається тільки виключити комп'ютер.

Тепер, щоб віддалено включити цей комп'ютер, нам потрібно по мережі послати кадр із послідовністю, що будить. Для цього існує кілька програм, таких як wol.exe або broadc.exe. Всі що нам потрібно знати для запуску програми – це MAC-адреса мережевого адаптера віддаленого комп'ютера. Наприклад, для broadc.exe, що запускається з консолі, вхідний рядок буде така:

broadc.exe (MAC-адреса мережевої карти) 255.255.255.255 67

Допустимо, що MAC-адреса мережевого адаптера – 00:02:B3:D8:B4:E6, тоді рядок прийме вид:

broadc.exe 0002b3d8b4e6 255.255.255.255 67

Інші вхідні параметри змінювати не потрібно. 255.255.255.255 – це широкомовний IP-адреса, завдяки якому сформований кадр пройде через всю мережу, а 67 – номер порту протоколу UDP, у дейтаграмі якого й буде перебувати послідовність, що будить. Використання wol.exe і інших подібних програм повністю аналогічно.

Вибір програмного забезпечення

Наступним кроком є визначення необхідного нам програмного забезпечення, для здійснення функцій віддаленого керування й контролю за системою. Існує множина програм для віддаленого керування комп'ютером.

Принципово можна розділити все це різноманіття на дві групи, що розрізняються по способу керування. Управляти можна через командний рядок/консоль (Telnet, SSH) або за допомогою графічного подання робочого стола віддаленої операційної системи (Remote Desktop, Remote Administrator).

Найбільш зручним і зрозумілим для кінцевого користувача, звичайно, представляється другий спосіб, а найпоширенішою й відомою програмою для такого доступу є Remote Administrator. У якості ftp-сервера можу порекомендувати Bullet Proof або Serv-U – вони досить прості й гнучкі в налаштуванні.

Налаштування NAT

Тепер, перейдемо до підготовки нашого роутера. Отут прийде затурбуватися налаштуванням двох речей: це NAT і вбудований пакетний фільтр (або, як його частіше називають, брандмауер або firewall). І те, і інше зручніше набудовувати за допомогою web-інтерфейсу.

Для початку створимо запис в таблицю статичної NAT-адресації. Залежно від виробника конкретної моделі роутера, ця опція може позначатися по-різному. У роутерах D-link це називається «Virtual Server».

У кожному разі настроюється сам NAT скрізь однаково. Задається зовнішній порт роутера, на який приходить запит, IP-адреса й порт, на який роутер повинен перенаправляти цей запит.

Звичайно номер порту задається в обох випадках однаковий. Усе, що нам треба знати, це те, які порти використовує необхідне нам програмне забезпечення. Так, стандартний порт для Remote Administrator – 4899. З метою безпеки й зменшення ймовірності несанкціонованого доступу, можна замінити в налаштуваннях серверної частини RAdmin'a стандартний порт на будь-який інший.

Для FTP-сервера стандартний порт – 21. Рекомендую також застосовувати нестандартний порт, наприклад 2121. Обумовлено, це тим, що деякі провайдери фільтрують запити, які поступають ззовні, адресовані на стандартний для FTP порт.

Окремо варто поговорити про правильне налаштування FTP. Існують два режими роботи FTP-сервера: пасивний і активний.

Для коректної роботи за NAT'ом, потрібно настроїти сервер на пасивний режим роботи. У такому режимі клієнт, з'єднуючись із сервером, одержує від нього список портів, по яких надалі він повинен ініціювати з'єднання для передачі файлів.

У налаштуваннях самої програми FTP-сервера необхідно вказати зовнішній IP-адресу, на який будуть надсилати запити клієнти, і діапазон портів, по яких їм варто встановлювати з'єднання для передачі даних (наприклад, 47990-48000).

Їх же варто вказати в таблиці статичної NAT-адресації.

Налаштування файрвола

Але настроїти на роутері NAT – недостатньо для повноцінного функціонування цих сервісів. Необхідно ще зконфігурувати пакетний фільтр, реалізований в ADSL-роутері.

Основними правилами завжди повинні бути: пропускати всі запити із внутрішнього інтерфейсу на зовнішній, і скидати всі запити із зовнішнього на внутрішній. Таким чином, ми позбуваємося від зайвого трафіку ззовні, також охороняючи слабкі місця операційної системи від «промацування».

Залишається лише додати правила для пропущення запитів, адресованих FTP-серверу й Remote Administrator'y. Знову таки, налаштування пакетного фільтра схожі в різних виробників роутеров, і завжди містить у собі:

−завдання протоколу передачі інформації (звичайно вибір лежить між TCP, UDP і ICMP; необхідно вказати той або інший набір протоколів, які використовує додаток для передачі свого трафіку);

−завдання маршруту проходження (з якого інтерфейсу/порту на який іде трафік);

−завдання IP-адреси або діапазону IP-адрес відправника (потрібно, тільки якщо ми дозволяємо одержувати ці запити тільки з яких те конкретних IP-адрес);

−завдання номера порту або діапазону портів відправника (для вхідних запитів практичного застосування фактично ні, і в деяких роутерах цей пункт цілком логічно відсутній);

−завдання IP-адреси або діапазону IP-адрес одержувача (потрібно, тільки у випадку наявності в нас декількох зовнішніх IP-адрес, для розмежування функціонального навантаження між ними);

−завдання номера порту або діапазону портів одержувача (цим правилом, ми дозволяємо певним сервісам приймати ззовні запити й обробляти їх);

−завдання розкладу дії правила (другорядний параметр, що рекомендується залишати в значенні за замовчуванням, звичайно – always);

−вибір дії із запитом: пропустити або скинути (вибираємо дія, що буде робити роутер, одержавши пакет, що підходить під ці параметри).

Так, для Remote Administrator'a необхідно дозволити вхідні запити по протоколі TCP, з будь-яких IP-адрес, на будь-які IP-адреси, на порт 4899. Для FTP-сервера варто дозволити вхідні запити по протоколі TCP, з будь-яких IP-адрес, на будь-які IP-адреси, на порт 21 (2121) і на діапазон портів 47990-48000.

Отже, що маємо в підсумку. Ми настроїли комп'ютер, що включається по нашому телефонному дзвінку й відключається через десять мінут, у випадку якщо це був помилковий дзвінок. Ми одержали можливість віддалено включати цей комп'ютер з локальної або домашньої мережі, настроїти на роутері NAT і пакетний фільтр так, що до комп'ютера стало можливим звертатися через Інтернет.

І тепер у нас з'явилася повноцінна можливість діагностувати й віддалено управляти повною мірою домашнім комп'ютером.

Internet Server API фірми Microsoft

Коли застосовується інтерфейс Internet Server API (ISAPI) фірми Microsoft, то взаємодія між сервером і прикладною програмою організується через спеціальну структуру даних, іменовану ECB (Extension Control Block). У ній утримується інформація про те, як прикладній програмі варто обробляти поточний запит клієнта. Спочатку сервер завантажує

прикладний модуль і передає йому ECB-блок. За допомогою функцій GetServerVariable і ReadClient модуль зчитує передані клієнтом вхідні дані.

Обробивши їх, модуль звертається до функції WriteClient і відправляє дані обернено клієнтові. Після цього він викликає функцію ServerSupportFunction, щоб повідомити сервер про закінчення обробки запиту. Інсталяція прикладних ISAPI-модулів виробляється за допомогою Internet Service Manager.

Існують два типи ISAPI-програм для нарощування можливостей сервера: фільтри й прикладні модулі. Фільтри виконують, задають або змінюють запити клієнта до початку їхньої обробки сервером. Звертання до фільтрів відбувається при надходженні будь-якого запиту від клієнта. Відомості про наявні DLL-фільтри втримуються в системному Реєстрі Windows, і сервер звертається до них при своїй ініціалізації. На відміну від фільтрів прикладні модулі повинні запитуватися самим клієнтом.

Крім ISAPI існує ще інтерфейс OLEISAPI, що дозволяє вашої допоміжної API-програмі взаємодіяти із серверами OLE Automation (автоматизація OLE). У результаті можна буде створювати програми в середовищі Visual Basic, що працюють із ISAPI. (Для розробки ISAPI-модулів необхідно використовувати Visual C++ або Delphi.) Специфікація OLEISAPI містить тільки частина ISAPI і офіційно ще не прийнята фірмою Microsoft.

Розробка структурної схеми

Проаналізувавши, всі досліджені технології віддаленого керування комп'ютером, движок розробленого програмного забезпечення виконує наступні дії. Спершу відбувається з'єднання через Інтернет або локальну мережу із клієнтською частиною, що встановлена на комп'ютері який буде віддаленно управлятися. Після цього клієнтом, за допомогою функцій API, грабую весь екран і передаю його серверу, той, у свою чергу, відтворює його у своїй робочій області й там можна рухати мишею й робити довільні дії; все це програма-сервер відслідковує, перехоплює й посилає клієнтові, а той їх відтворює.

Перераховані вище дії утворюють основне ядро програми. На рисунку 2 зображена структурна схема віддаленого управління ЕОМ у загальному випадку. З цієї схеми ми бачимо, що усі віддалені користувачі зв’язуються один з одним за допомогою RAS (серверів віддаленого управління доступом), модемів та маршрутизаторів.

При цьому для доступу у мережу підприємства, сервер віддаленого управління доступом використовує файрвол, для надання взаємного захисту між внутрішньою мережею підприємства, яку можно моніторити за допомогою, розробленого, у результаті виконання магістерського проектування, програмного забезпечення, з однієї сторони та сервером віддаленого управління доступом з іншої сторони.

Схеми віддаленого керування наведені на рисунку 2, відрізняються типом взаємодіючих систем:

(1) – термінал-комп'ютер;

(2) – комп'ютер-комп'ютер;

(3) – комп'ютер-мережа;

(4) – мережа-мережа.

Відповідно при реалізації різних типів взаємодіючих систем використовується різне апаратне обладнання.

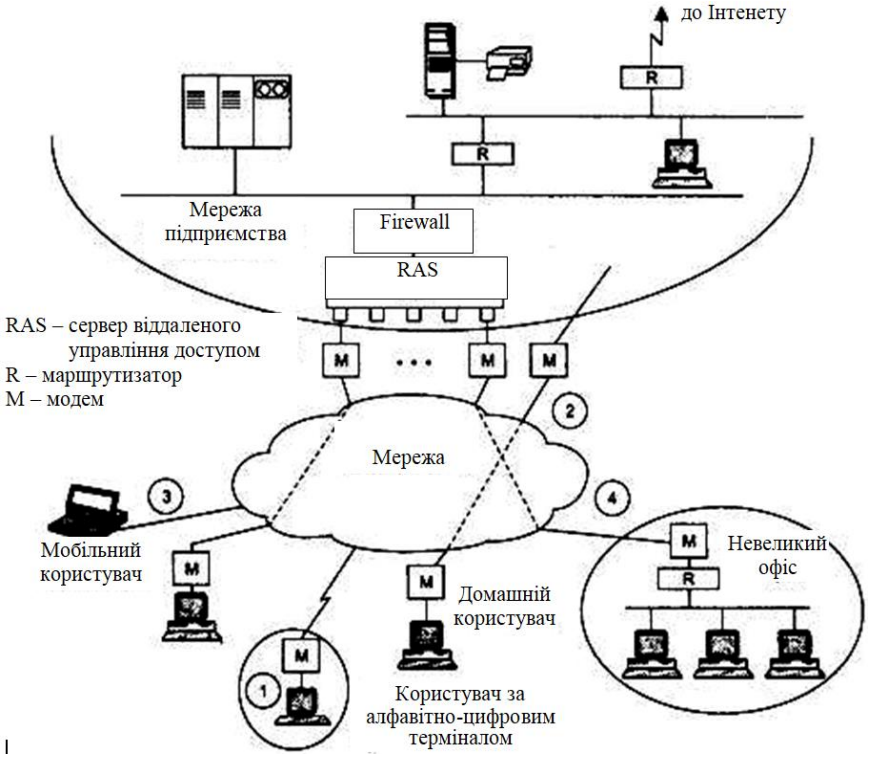


Рисунок 2 – Структурна схема системи

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів віддаленого доступу з використанням WAN-мереж. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем віддаленого доступу з використанням WAN-мереж. Досліджена система віддаленого доступу з використанням WAN-мереж. На основі отриманих результатів досліджень створена програмна реалізація системи віддаленого доступу з використанням WAN-мереж. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання віддаленого доступу з використанням WAN-мереж. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки. Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

Список літератури

1. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
2. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
3. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
4. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).
5. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019. (Scopus).

6. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629. (Scopus).
7. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 873-884. (Scopus).
8. Smirnov, O., Kuznetsov, A., Prokopovych-Tkachenko, D. «Hiding Data in Images Using a Pseudo-Random Sequence». ISCI'2020: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko, Victor A. Krasnobayev and Alexandr A. Kuznetsov. ASC Academic Publishing, USA, 2020. pp. 46-59. – ISBN: 978-1-7362833-0-1 (Hardback), ISBN: 978-1-7362833-1-8 (Ebook).
9. Smirnov, O., Kuznetsov, A., Shekhanin, K., Chepurko, I. Detecting Hidden Information in FAT. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 412-429. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
10. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
11. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
12. Смірнов О.А., Дрєєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.
13. Смирнов А.А., Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений. Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.
14. Смирнов А.А., Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения. Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.
15. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 3(69). С. 93-98. 2022. (Фахове видання. Категорія «Б»)
16. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.
17. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.
18. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95
19. Смирнов А., Кузнецов А., Кузнецова Т. «Шумоподобные дискретные сигналы для асинхронных систем кодового разделения радиоканалов». Радиотехника, № 2(205), 175–183. 2021.
20. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

УДК 004

Д.Тарковський, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ВИЗНАЧЕННЯ РІВНЯ СТІЙКОСТІ СЕРВІСІВ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ НА ОСНОВІ МЕТОДІВ AI

У статті розроблено програмне забезпечення, яке призначено для системи визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI. Метою розробки є дослідження та програмна реалізація системи визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI. Об'єктом дослідження є процес визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI. Предметом дослідження є методи визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI. Методи дослідження базуються на методах захисту інформації та штучного інтелекту (AI), методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерні науки, стійкість, конфіденційність, штучний інтеллект

Постановка проблеми. Питання конфіденційності займають перше місце в онлайн-діяльності, бізнес-діях і державних рішеннях. Це здебільшого у відповідь на зломи, скандали та витік особистих даних, які підірвали довіру до технологій та інформаційних систем.

У звіті Консультативного комітету з питань телекомунікацій національної безпеки про кібербезпеку Moonshot йдеться, що конфіденційність є ключовим компонентом кібербезпеки, і що ми повинні змінити наратив, щоб відновити довіру американців до інформаційних систем. Щоб досягти цього, до 2028 року потрібно «гарантувати», що технологічний прогрес більше не загрожуватиме конфіденційності, а натомість підвищить гарантії конфіденційності завдяки безпеці та безпеці їхніх особистих даних.

Одним з найважливіших елементів у майбутніх технологічних досягненнях і онлайн-безпеці є посилений розвиток штучного інтелекту (AI). Проте принципи конфіденційності необхідно враховувати на ранніх етапах процесу розробки штучного інтелекту, щоб збалансувати технологічні переваги та зберегти конфіденційність.

Більшість відомих експертів-криптоаналітиків вважають що, штучний інтелект, тим більше нейрокомп'ютерні мережі, не кращий інструмент криптоаналізу. Підстав у цієї широко поширеної думки дуже багато, наприклад, низька можливість для навчання.

Недоліками названого підходу є, насамперед те, що для кожної нової криптографічної системи необхідно розробляти нову методику навчання елементів нейромережі.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI.

– Дослідження системи визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI.

– Програмна реалізація системи визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI.

Об'єктом дослідження є процес визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI.

Предметом дослідження є методи визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI.

Методи дослідження базуються на методах захисту інформації та штучного інтелекту (AI), методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Давайте приділимо хвилинку, щоб дослідити наслідки та потенційні наслідки збільшення впровадження штучного інтелекту онлайн. Хоча це здається футуристичним, коли штучний інтелект починає «думати» як люди або навіть замість людей, це може загрожувати трьом основним принципам конфіденційності – точності даних, захисту та контролю:

– Точність даних: щоб штучний інтелект давав точні результати, алгоритми повинні містити великі та репрезентативні набори даних. Недостатня представленість певних груп у наборах даних може призвести до неточних результатів і навіть шкідливих рішень. Це алгоритмічне зміщення часто створюється ненавмисно. Наприклад, дослідники виявили, що розумні мовці не можуть зрозуміти жіночі голоси або голоси меншин, тому що алгоритми побудовані з баз даних, які містять переважно голоси білих чоловіків. З огляду на це, що станеться, якщо ми довіримо AI приймати наші дзвінки в екстрену допомогу?

– Захист даних. Незважаючи на те, що великі набори даних дають більш точні та репрезентативні результати, вони створюють більший ризик конфіденційності, якщо їх порушують. Штучний інтелект може легко деанонімізувати навіть особисті дані, здавалося б, анонімні. Зокрема, дослідники виявили мінімальну анонімність навіть у грубих наборах даних, що призводить до повторної ідентифікації до 95 відсотків. Разом це означає, що ви ризикуєте бути легко ідентифікованим і отримати витік ваших даних, якщо не врахувати міркувань конфіденційності. Використання штучного інтелекту також може призвести до попередження, коли використовується для обробки податків і аналізу права на отримання федеральних пільг.

– Контроль даних: коли штучний інтелект починає бачити та визначати закономірності, він робить висновки та може приймати рішення щодо вас, щоб зробити вашу роботу в Інтернеті легшою або надійнішою. Однак, коли штучний інтелект дає хибні або несприятливі результати, це викликає питання, чи були прийняті рішення справедливими. Наприклад, AI, який використовується для оцінки кредитних ризиків, може ненавмисно скоротити кредитні лінії осіб, які відповідають певним профілям. Ці рішення можуть бути прийняті без вашого відома, згоди чи вибору, особливо якщо дані, які керують цими рішеннями, збираються без вашого відома. Більше того, штучний інтелект може отримати додаткові відомості про вас, наприклад ваші політичні уподобання, расу та релігію, навіть якщо ви ніколи не публікували ці подробиці в Інтернеті.

Суть полягає в тому, що особисті дані можуть використовуватися, а іноді і проти вас, без жодного контролю. Хороша новина полягає в тому, що розробники можуть мінімізувати проблеми конфіденційності на етапі розробки, задовго до виробництва. Таким чином ми все ще можемо реалізувати технологічні переваги AI, не порушуючи конфіденційність людей. Щоб підвищити конфіденційність, ми пропонуємо додати штучний інтелект до стратегії керування даними вашої організації та виділити ресурси не лише на розробку продукту штучного інтелекту, але й на конфіденційність, безпеку та моніторинг.

Інші способи захисту конфіденційності в AI включають:

1. Використовуйте належну гігієну даних. Слід збирати лише ті типи даних, які необхідні для створення AI, а дані мають зберігатися в безпеці та підтримуватися лише стільки часу, скільки необхідно для досягнення мети.

2. Використовуйте хороші набори даних. Розробники повинні створювати AI, використовуючи точні, справедливі та репрезентативні набори даних. Де можливо, розробники повинні створювати алгоритми штучного інтелекту, які будуть перевіряти та забезпечувати якість інших алгоритмів.

3. Надайте користувачам контроль. Користувачі повинні знати, коли використовуються їхні дані, чи використовується AI для прийняття рішень щодо них і чи використовуються їхні дані для створення AI. Їм також слід надати можливість дати згоду на використання таких даних.

4. Зменшити алгоритмічне зміщення. Переконайтеся, що набори даних є широкими та всеосяжними під час «навчання» AI. Алгоритмічні зміщення найчастіше становлять проблеми для жінок, меншин і груп (наприклад, осіб з порушеннями голосу, людей похилого віку), які складають лише невелику частину технологічної робочої сили.

Наш світ переживає інформаційний Великий вибух, під час якого всесвіт даних подвоюється кожні два роки, а щодня генеруються квінтильйони байтів даних.Протягом десятиліть закон Мура про подвоєння обчислювальної потужності кожні 18-24 місяці стимулював розвиток інформаційних технологій. Тепер, коли мільярди смартфонів та інших пристроїв збирають і передають дані через високошвидкісні глобальні мережі, зберігають дані у все більших центрах обробки даних і аналізують їх за допомогою все більш потужного та складного програмного забезпечення, закон Меткалфа вступає в дію. Він розглядає цінність мереж як функцію квадрата кількості вузлів, тобто мережеві ефекти експоненціально посилюють цей історичний ріст інформації. У міру розгортання мереж 5G і, зрештою, квантових обчислень, цей вибух даних зростатиме ще швидше та масштабніше.

Вплив великих даних зазвичай описують у термінах: обсяг, різноманітність і швидкість. Більше даних робить аналіз потужнішим і детальнішим. Різноманітність додає цій потужності та дозволяє робити нові та непередбачені висновки та прогнози. А швидкість полегшує аналіз, а також обмін у реальному часі.

Ймовірно, штучний інтелект прискорить цю тенденцію. Значна частина найбільш У міру розвитку штучного інтелекту він розширює можливості використання особистої інформації у спосіб, який може порушити інтереси конфіденційності, підвищуючи ефективність і швидкість аналізу особистої інформації.

Системи розпізнавання обличчя пропонують попередній перегляд проблем конфіденційності, які виникають. Завдяки багатим базам даних цифрових фотографій, доступних через соціальні мережі, веб-сайти, реєстри водійських прав, камери спостереження та багато інших джерел, машинне розпізнавання облич швидко прогресувало від нечітких зображень котів до швидкого (хоча все ще недосконалого) розпізнавання окремі люди. Системи розпізнавання облич розгортаються в містах і аеропортах Америки. Однак використання Китаєм розпізнавання облич як інструменту авторитарного контролю в Сіньцзяні та інших місцях викликало опозицію до цього розширення та закликає заборонити використання розпізнавання облич. Через занепокоєння щодо розпізнавання облич міста Окленд, Берклі та Сан-Франциско в Каліфорнії, а також Бруклін, Кембридж, Нортгемптон і Сомервіль у Массачусетсі прийняли заборону на цю технологію. У Каліфорнії, Нью-Гемпширі та Орегоні прийнято законодавство, яке забороняє використання розпізнавання обличчя за допомогою поліцейських натільних камер.

У цій аналітичній записці досліджується взаємозв’язок між AI та поточними дебатами щодо конфіденційності. Оскільки Конгрес розглядає всеосяжне законодавство про конфіденційність, щоб заповнити дедалі більші прогалини в поточній таблиці конфіденційності на федеральному рівні та штаті, йому потрібно буде розглянути питання про використання особистої інформації в системах штучного інтелекту. У цьому короткому описі я обговорюю деякі потенційні проблеми щодо штучного інтелекту та

конфіденційності, зокрема дискримінацію, етичне використання та контроль людини, а також варіанти політики, що обговорюються.

Проблеми конфіденційності в AI

Завдання Конгресу полягає в тому, щоб ухвалити законодавство про конфіденційність, яке захищатиме людей від будь-яких несприятливих наслідків використання особистої інформації в штучному інтелекті, але без необґрунтованого обмеження розвитку штучного інтелекту чи втягування законодавства про конфіденційність у складні соціальні та політичні хащі. Обговорення штучного інтелекту в контексті дебатів щодо конфіденційності часто викликає обмеження та помилки систем штучного інтелекту, таких як інтелектуальна поліція, яка може непропорційно вплинути на меншини або невдалий експеримент Amazon із алгоритмом найму, який повторює існуючу непропорційно чоловічу робочу силу компанії. Обидва вони порушують значні проблеми, але законодавство про конфіденційність досить складне навіть без урахування всіх соціальних і політичних проблем, які можуть виникнути в результаті використання інформації. Щоб оцінити вплив штучного інтелекту на конфіденційність, необхідно розрізняти проблеми з даними, які характерні для всіх видів штучного інтелекту, як-от випадки помилкових спрацьовувань і негативних результатів або надмірне пристосування до шаблонів, і ті, які характерні для використання особистої інформації.

Законодавчі пропозиції щодо конфіденційності, які стосуються цих питань, не стосуються штучного інтелекту. Швидше, вони посилаються на «автоматизовані рішення» (запозичені із законодавства ЄС про захист даних) або «алгоритмічні рішення» (використовуються в цьому обговоренні). Ця мова зміщує увагу людей від використання штучного інтелекту як такого до використання особистих даних у штучному інтелекті та до впливу, який таке використання може мати на людей. Ці дебати зосереджуються, зокрема, на алгоритмічній упередженості та потенціалі алгоритмів для незаконної або небажаної дискримінації в рішеннях, яких стосуються алгоритми. Це головне занепокоєння для громадських прав і організацій споживачів, які представляють групи населення, які зазнають неправомірної дискримінації.

Розгляд алгоритмічної дискримінації ставить основні питання щодо сфери застосування законодавства про конфіденційність. По-перше, якою мірою законодавство може або має вирішувати проблеми алгоритмічної упередженості? Дискримінація не є самоочевидною проблемою конфіденційності, оскільки вона представляє широкі соціальні проблеми, які зберігаються навіть без збору та використання особистої інформації та підпадають під дію різних законів про громадянські права. Більше того, надання цих законів для обговорення може фактично відкрити скриньку Пандори через гострі політичні питання, які вони торкаються, і численні комітети Конгресу, які мають юрисдикцію над різними такими питаннями. Незважаючи на це, дискримінація ґрунтується на особистих ознаках, таких як колір шкіри, сексуальна приналежність і національне походження. Використання особистої інформації про ці атрибути, явно або – більш імовірно й менш очевидно – через проксі-сервери, для автоматизованого прийняття рішень, що суперечить інтересам залученої особи, таким чином передбачає інтереси конфіденційності в контролі над тим, як використовується інформація.

Ця шарада згоди зробила очевидним, що помічати і вибирати втрачає сенс. Для багатьох програм штучного інтелекту це стане абсолютно неможливим.

По-друге, захист таких інтересів конфіденційності в контексті AI вимагатиме зміни парадигми регулювання конфіденційності. Більшість існуючих законів про конфіденційність, а також чинні заходи Федеральної торгової комісії проти недобросовісних і оманливих дій ґрунтуються на моделі споживчого вибору на основі «повідомлення та вибору» (також називають «повідомлення та згода»). Ця шарада згоди зробила очевидним, що помічати і вибирати втрачає сенс. Для багатьох застосувань штучного інтелекту (наприклад, інтелектуальні сигнали світлофора та інші датчики, необхідні для підтримки самокерованих автомобілів, як один із яскравих прикладів), це стане абсолютно неможливим.

Хоча майже всі законопроекти на Капітолійському пагорбі все ще певною мірою спираються на модель «повідомлення та вибір», ключові лідери Конгресу, а також зацікавлені сторони конфіденційності висловили бажання змінити цю модель, переклавши тягар захисту конфіденційності особи зі споживачів на компанії. які збирають дані. Замість вибору споживача їхня модель зосереджена на веденні бізнесу, регулюючи обробку компаніями даних – що вони збирають, як вони можуть ними користуватися та ділитися. Розгляд обробки даних, що призводить до будь-якого алгоритмічного розрізнення, може вписатися в цю модель.

Модель, орієнтована на збір і обробку даних, може впливати на AI та алгоритмічну дискримінацію декількома способами:

– Вимоги щодо управління даними, як-от обов'язки чесності чи лояльності, можуть перешкоджати використанню особистої інформації, яке є несприятливим або несправедливим по відношенню до осіб, яких ці дані стосуються.

– Правила прозорості та розкриття даних, а також права осіб на доступ до інформації, що їх стосується, можуть прояснити використання алгоритмічного прийняття рішень.

– Правила керування даними, які передбачають призначення спеціалістів із забезпечення конфіденційності, проведення оцінок впливу на конфіденційність або планування продукту через «конфіденційність за проектом», можуть виявити проблеми, пов'язані з використанням алгоритмів.

– Правила збору та обміну даними можуть зменшити агрегацію даних, що дозволяє робити висновки та прогнози, але можуть передбачати певні компроміси з перевагами великих і різноманітних наборів даних.

На додаток до цих положень загального застосування, які можуть опосередковано впливати на алгоритмічні рішення, ряд пропозицій спеціально стосуються цього предмета.

Варіанти політики AI для захисту конфіденційності

Відповіді на AI, які зараз обговорюються в законодавстві про конфіденційність, мають дві основні форми. Перший спрямований безпосередньо на дискримінацію. Група з 26 громадських правозахисних і споживчих організацій написала спільного листа, в якому закликає заборонити або контролювати використання особистої інформації з дискримінаційним впливом на «кольорових людей, жінок, релігійних меншин, членів ЛГБТК+ спільноти, людей з обмеженими можливостями, осіб, які живуть на l привабливі, іммігранти та інші вразливі верстви населення». Комітет юристів із захисту громадянських прав відповідно до закону та Free Press Action включили цей принцип до типового законодавства, спрямованого на дискримінацію даних, що впливає на економічні можливості, громадські умови або придушення виборців.

Цей підхід до алгоритмічної дискримінації передбачає дебати щодо приватних прав на дії в законодавстві про конфіденційність. Можливість такого індивідуального судового розгляду є ключовим моментом розбіжностей між демократами, які об'єднують інтереси споживачів і конфіденційності, з одного боку, і республіканцями, які об'єднують інтереси бізнесу, з іншого. Перші стверджують, що приватні позови є необхідним примножувачем сили для правоохоронних органів на федеральному рівні та штаті, тоді як другі висловлюють занепокоєння тим, що колективні позови, зокрема, обтяжують бізнес судовими процесами з тривіальних питань. У випадку багатьох видів дискримінації, перелічених у пропозиціях щодо алгоритмічної дискримінації, існуючі федеральні, державні та місцеві закони про громадянські права дозволяють особам подавати позови про дискримінацію. Будь-які федеральні переваги чи обмеження приватних прав на дії у федеральному законодавстві про конфіденційність не повинні порушувати ці закони.

Другий підхід розглядає ризик більш побіжно, із заходами підзвітності, призначеними для виявлення дискримінації під час обробки персональних даних. Численні організації та компанії, а також деякі законодавці пропонують таку підзвітність. Їхні пропозиції мають різні форми:

– Прозорість: це стосується розкриття інформації щодо використання алгоритмічного прийняття рішень. Хоча довгі детальні політики конфіденційності не є корисними для більшості споживачів, вони надають регуляторам та іншим органам захисту конфіденційності орієнтир, за яким можна перевірити обробку даних компанії та притягнути її до відповідальності. Заміна поточної політики конфіденційності на «розкриття конфіденційності», яка вимагає повного опису того, які та як дані збираються, використовуються та захищаються, покращить цю еталонну функцію. У свою чергу, вимога, щоб ці розкриття ідентифікували значні випадки використання особистої інформації для прийняття алгоритмічних рішень, допомогло б спостерігачам і споживачам знати, де слід остерігатися несприятливих результатів.

– Пояснюваність: у той час як прозорість забезпечує завчасне повідомлення про прийняття алгоритмічних рішень, пояснюваність передбачає ретроактивну інформацію про використання алгоритмів у конкретних рішеннях. Це основний підхід, прийнятий у Загальному регламенті захисту даних Європейського Союзу (GDPR). GDPR вимагає, щоб для будь-якого автоматизованого рішення з «юридичними чи подібними значними наслідками», як-от працевлаштування, кредит або страхове покриття, особа, на яку це впливає, мала звернутися до людини, яка може переглянути рішення та пояснити його логіку. Це включає компонент «людини в циклі» та елемент належного процесу, який забезпечує перевірку аномальних або несправедливих результатів.

Почуття справедливості припускає, що такий запобіжний клапан має бути доступним для алгоритмічних рішень, які мають суттєвий вплив на життя людей. Пояснення вимагає (1) ідентифікації алгоритмічних рішень, (2) деконструкції конкретних рішень і (3) встановлення каналу, за яким людина може шукати пояснення. Алгоритми зворотного проектування, засновані на машинному навчанні, можуть бути складними та навіть неможливими, і ця складність зростає, коли машинне навчання стає складнішим. Таким чином, пояснюваність тягне за собою значне нормативне навантаження та обмеження на використання алгоритмічного прийняття рішень і, у цьому світлі, повинна бути зосереджена на його застосуванні, як це зробив ЄС (принаймні в принципі) з його «правовими наслідками або подібними значними наслідками» поріг. У міру того як зростає розуміння порівняльних переваг людських і машинних можливостей, наявність «людини в курсі» рішень, які впливають на життя людей, дає спосіб поєднати потужність машин із людським судженням і співчуттям.

– Оцінка ризику: в Законі про конфіденційність 1974 року оцінки ризиків спочатку були розроблені як «оцінка впливу на конфіденційність» у рамках федерального уряду. З тих пір вони перетворилися на широко використовувані інструменти керування конфіденційністю, щоб заздалегідь оцінювати та зменшувати ризики конфіденційності, і вони вимагаються GDPR для нових технологій або використання даних із високим ризиком.

– Перевірки: перевірки оцінюють практику конфіденційності ретроспективно. Більшість законодавчих пропозицій містять деякі загальні вимоги до підзвітності, щоб переконатися, що компанії дотримуються своїх програм конфіденційності, а деякі включають самоконтроль або аудит третьої сторони. У поєднанні з проактивною оцінкою ризиків аудит результатів алгоритмічного прийняття рішень може допомогти поєднати передбачення з заднім числом; хоча, як і пояснюваність, аудит процедур машинного навчання є складним і все ще розвивається.

Через труднощі передбачення результатів машинного навчання, а також алгоритмічних рішень зворотного проектування, жодна окрема міра не може бути повністю ефективною для уникнення шкідливих ефектів. Таким чином, якщо алгоритмічні рішення є послідовними, має сенс комбінувати заходи для спільної роботи. Попередні заходи, такі як прозорість і оцінка ризиків, у поєднанні з ретроспективними перевірками аудитів і людським переглядом рішень, можуть допомогти виявити та усунути несправедливі результати. Комбінація цих показників може доповнювати один одного і давати більше, ніж сума частин. Оцінка ризиків, прозорість, пояснюваність і перевірки також посилили б існуючі засоби

правового захисту від дискримінації, яка є причиною покарання, шляхом надання документальних доказів, які можна було б використовувати в судовому процесі. Однак не всі алгоритми прийняття рішень є послідовними, тому ці вимоги мають змінюватися відповідно до об'єктивного ризику. Штучний інтелект (AI) вийшов за межі наукової фантастики та став сучасним технологічним рішенням, яке сьогодні використовують багато компаній. Його швидка інтеграція в різні сектори, від охорони здоров'я до фінансів, змінює те, як ми взаємодіємо з даними та приймаємо рішення. У дослідницькому звіті Currents за 2023 рік, в якому опитувалися засновники, керівники та співробітники технологічних компаній, було виявлено, що 49% респондентів використовують інструменти штучного інтелекту та машинного навчання для бізнесу. Однак вагання щодо цих технологій залишаються. На запитання, що заважає їхнім організаціям більше використовувати інструменти AI/ML, 29% згадали про етичні та юридичні проблеми, тоді як 34% відзначили проблеми безпеки.

З цією інновацією виникає нагальна проблема: конфіденційність AI. Оскільки системи AI обробляють величезні обсяги особистої інформації, межа між корисністю та вторгненням стає дедалі розмитішою. Компанії, які використовують бізнес-інструменти штучного інтелекту або розробляють власні, повинні ретельно поєднувати захист конфіденційної інформації з максимальним використанням можливостей технології.

Що таке конфіденційність AI?

Конфіденційність штучного інтелекту – це набір практик і проблем, зосереджених навколо етичного збору, зберігання та використання особистої інформації системами штучного інтелекту. Він відповідає критичній потребі захисту прав особи на дані та збереження конфіденційності, оскільки алгоритми штучного інтелекту обробляють величезну кількість особистих даних і вивчають їх. Забезпечення конфіденційності штучного інтелекту передбачає налагодження балансу між технологічними інноваціями та збереженням особистої конфіденційності в епоху, коли дані є дуже цінним товаром.

Методи збору даних AI та конфіденційність

Системи штучного інтелекту покладаються на велику кількість даних, щоб покращити свої алгоритми та результати, використовуючи ряд методів збору, які можуть становити значні ризики для конфіденційності. Методи, які використовуються для збору цих даних, часто невидимі для осіб (наприклад, клієнтів), від яких збираються дані, що може призвести до порушень конфіденційності, які важко виявити або контролювати.

Ось кілька методів збору даних штучного інтелекту, які мають наслідки для конфіденційності:

– Веб-скрейпінг. AI може накопичувати величезні обсяги інформації, автоматично збираючи дані з веб-сайтів. Хоча деякі з цих даних є загальнодоступними, веб-збирання також може отримувати особисті дані, потенційно без згоди користувача.

– Біометричні дані. Системи штучного інтелекту, які використовують розпізнавання обличчя, зйомку відбитків пальців та інші біометричні технології, можуть втручатися в особисту конфіденційність, збираючи конфіденційні дані, які є унікальними для окремих людей і, якщо їх зламано, незамінні.

– Пристрої IoT. Пристрої, підключені до Інтернету речей (IoT), надають системам AI дані в реальному часі з наших домівок, робочих місць і громадських місць. Ці дані можуть розкривати інтимні подробиці нашого повсякденного життя, створюючи безперервний потік інформації про наші звички та поведінку.

– Моніторинг соціальних медіа. Алгоритми AI можуть аналізувати активність у соціальних мережах, фіксуючи демографічну інформацію, уподобання та навіть емоційний стан, часто без явного відома або згоди користувача.

Наслідки цих методів для конфіденційності є далекосяжними. Вони можуть призвести до несанкціонованого стеження, крадіжки особистих даних і втрати анонімності. Оскільки технології штучного інтелекту стають все більш інтегрованими в повсякденне життя, забезпечення того, щоб збір даних був прозорим і безпечним і щоб люди зберегли контроль над своєю особистою інформацією, стає все більш критичним.

Унікальні виклики конфіденційності AI

Згідно з даними Crunchbase, у 2023 році понад 25% інвестицій в американські стартапи було спрямовано на компанії, що спеціалізуються на AI. Ця хвиля штучного інтелекту розкрила безпрецедентні можливості в обробці даних, аналізі та прогнозному моделюванні. Однак штучний інтелект створює складні та багатогранні проблеми з конфіденційністю, які відрізняються від тих, які створює традиційна обробка даних:

– Обсяг і різноманітність даних. Системи штучного інтелекту можуть переробляти та аналізувати експоненціально більше даних, ніж традиційні системи, що підвищує ризик розкриття особистих даних.

– Прогностична аналітика. Завдяки розпізнаванню образів і прогнозному моделюванню штучний інтелект може визначати особисту поведінку та вподобання, часто без відома чи згоди людини.

– Непрозоре прийняття рішень. Алгоритми штучного інтелекту можуть приймати рішення, що впливають на життя людей, без прозорих міркувань, що ускладнює відстеження або заперечення вторгнень у конфіденційність.

– Безпека даних. Великі набори даних, необхідні AI для ефективного функціонування, є привабливими цілями для кіберзагроз, що збільшує ризик злому, який може поставити під загрозу особисту конфіденційність.

– Вбудоване упередження. Без ретельного нагляду штучний інтелект може зберегти існуючі упередження в даних, які він передає, що призведе до дискримінаційних результатів і порушень конфіденційності.

Ці проблеми підкреслюють необхідність надійних заходів захисту конфіденційності в AI. Збалансування переваг штучного інтелекту з правом на конфіденційність вимагає ретельного проектування, впровадження та управління, щоб запобігти неправомірному використанню персональних даних.

Ключові проблеми конфіденційності AI для компаній

Оскільки компанії все більше інтегрують штучний інтелект у свою діяльність або створюють системи штучного інтелекту для використання своїми клієнтами, вони стикаються з багатьма проблемами конфіденційності, які слід вирішувати завчасно. Ці занепокоєння формують довіру клієнтів і мають значні юридичні та етичні наслідки, до яких компанії повинні обережно орієнтуватися.

Відсутність прозорості в алгоритмах AI

Природа «чорної скриньки» систем AI означає, що їхні процеси прийняття рішень часто непрозорі. Ця невідомість викликає занепокоєння у компаній, користувачів і регуляторів, оскільки вони часто не можуть бачити або розуміти, як алгоритми AI приходять до певних висновків або дій. Відсутність алгоритмічної прозорості також може приховати упередження або недоліки в системах AI, що призведе до результатів, які можуть ненавмисно завдати шкоди певним групам або окремим особам. Без такої прозорості підприємства ризикують підірвати довіру клієнтів і потенційно порушити нормативні вимоги.

Несанкціоноване використання персональних даних

Включення персональних даних у моделі штучного інтелекту без явної згоди створює значні ризики, зокрема юридичні наслідки відповідно до законів про захист даних, як-от GDPR, і потенційні порушення етичних стандартів. Несанкціоноване використання цих даних може призвести до порушення конфіденційності, значних штрафів і шкоди репутації компанії. З етичної точки зору такі дії ставлять під сумнів цілісність бізнесу та підривають довіру клієнтів.

Дискримінаційні результати застосування AI

Упередженість AI, що виникає через спотворені навчальні дані або помилкові алгоритми, може призвести до дискримінаційних результатів. Ці упередження можуть увічнити і навіть посилити існуючу соціальну нерівність, впливаючи на групи, засновані на расі, статі чи соціально-економічному статусі. Наслідки для конфіденційності є серйозними,

оскільки особи можуть бути несправедливо створені та піддані необґрунтованому контролю або виключенню. Для компаній це підриває чесну практику та може призвести до втрати довіри та юридичних наслідків.

Проблеми з авторським правом та інтелектуальною власністю з AI

Системи AI часто вимагають великих наборів даних для навчання, що може призвести до використання захищених авторським правом матеріалів без дозволу. Це порушує закони про авторське право та викликає занепокоєння щодо конфіденційності, коли вміст містить особисті дані. Компанії повинні обережно орієнтуватися в цих викликах, щоб уникнути судових розглядів і потенційних наслідків використання інтелектуальної власності третіх сторін без згоди.

Використання біометричної інформації

Використання біометричних даних у системах штучного інтелекту, таких як технології розпізнавання обличчя, викликає серйозні занепокоєння щодо конфіденційності. Біометрична інформація є особливо чутливою, оскільки вона за своєю суттю особиста і, у більшості випадків, незмінна. Несанкціонований збір, зберігання або використання цих даних може призвести до значного порушення конфіденційності та можливого зловживання. Підприємства, які використовують біометричний штучний інтелект, повинні забезпечити надійний захист конфіденційності, щоб зберегти довіру користувачів і дотримуватися суворих правових стандартів щодо біометричних даних.

Стратегії пом’якшення ризиків конфіденційності AI

Дослідження Deloitte у 2023 році показує, що 56% учасників опитування або не знають, або не впевнені щодо існування етичних принципів щодо генеративного використання AI в їхніх організаціях. Щоб захиститися від інвазивного потенціалу штучного інтелекту, компанії повинні активно приймати стратегії, які гарантують, що конфіденційність не буде порушена. Пом’якшення ризиків конфіденційності штучного інтелекту передбачає поєднання технічних рішень, етичних принципів і надійної політики управління даними.

Вбудуйте конфіденційність у дизайн AI

Щоб зменшити ризики конфіденційності штучного інтелекту, інтегруйте питання конфіденційності на початкових етапах розробки системи AI. Це передбачає прийняття принципів «конфіденційності за проектом», гарантуючи, що захист даних не є запізнілою думкою, а основним компонентом технології, яку створює ваша команда. Завдяки цьому моделі штучного інтелекту побудовані з необхідними засобами захисту, щоб обмежити непотрібний доступ до даних і забезпечити надійний захист із самого початку. Шифрування має бути стандартним для захисту даних у стані спокою та під час передачі, тоді як регулярні перевірки можуть забезпечити постійне дотримання політики конфіденційності.

Анонімізація та зведення даних

Використання методів анонімізації може захистити індивідуальні особи шляхом видалення ідентифікаційної інформації з наборів даних, які використовують системи AI. Цей процес передбачає зміну, шифрування або видалення особистих ідентифікаторів, що гарантує, що дані неможливо відстежити до особи. У поєднанні з анонімізацією агрегація даних об’єднує окремі точки даних у більші набори даних, які можна аналізувати без розкриття особистих даних. Ці стратегії зменшують ризик порушення конфіденційності, запобігаючи асоціації даних із конкретними особами під час аналізу AI.

Обмежте час зберігання даних

Впровадження суворої політики збереження даних мінімізує ризики конфіденційності, пов’язані зі штучним інтелектом. Встановлення чітких обмежень на тривалість зберігання даних запобігає непотрібному тривалому накопиченню особистої інформації, зменшуючи ймовірність її розголошення під час порушення. Ці політики змушують організації регулярно переглядати та видаляти застарілі або нерелевантні дані, оптимізуючи бази даних і мінімізуючи кількість даних, які піддаються ризику.

Збільште прозорість і контроль користувача

Підвищення прозорості в системах штучного інтелекту створює довіру та відповідальність користувачів. Компанії повинні повідомляти, які типи даних збираються, як алгоритми AI їх обробляють і для яких цілей. Надання користувачам контролю над їхніми даними, як-от можливість переглядати, редагувати чи видаляти їхню інформацію, розширює можливості людей і створює відчуття волі над їхнім цифровим слідом. Вони відповідають етичним стандартам і забезпечують відповідність нормам захисту даних, що розвиваються, які все більше вимагають згоди користувачів і управління.

Зрозумійте вплив нормативних актів

Розуміння наслідків GDPR та подібних нормативних актів має важливе значення для зменшення ризиків конфіденційності AI, оскільки ці закони встановлюють суворі стандарти захисту даних і надають особам значний контроль над своєю особистою інформацією. Ці правила зобов'язують організації бути прозорими щодо своєї діяльності з обробки AI та забезпечувати дотримання прав окремих осіб на дані, включаючи право на пояснення алгоритмічних рішень.

Компанії повинні вживати заходів, які гарантують точність, справедливість і підзвітність їхніх систем штучного інтелекту, особливо коли рішення мають юридичний або значний вплив на окремих осіб. Недотримання таких нормативних стандартів може призвести до значних штрафів.

Ось кілька правил і вказівок, на які варто звернути увагу:

– Закон ЄС про штучний інтелект.

– Запропонований Канадою законопроект C-27 (включає Закон про штучний інтелект і дані).

– Рекомендації Федеральної торгової комісії США (FTC) щодо використання штучного інтелекту та алгоритмів.

Розвивайте культуру етичного використання AI

Щоб зменшити ризики конфіденційності AI, установіть етичні принципи використання AI, які надають пріоритет захисту даних і дотриманню прав інтелектуальної власності. Організації повинні проводити регулярне навчання, щоб гарантувати, що всі співробітники розуміють ці вказівки та важливість їх дотримання у своїй повсякденній роботі з технологіями AI. Майте прозорі політики, які регулюють збір, зберігання та використання надзвичайно особистої та конфіденційної інформації. Нарешті, сприяння створенню середовища, де етичні проблеми можна відкрито обговорювати та вирішувати, допоможе підтримувати пильну позицію щодо потенційних порушень конфіденційності.

Майбутнє залежатиме від спільного підходу, коли безперервний діалог між технологами, компаніями, регуляторами та громадськістю формує розробку штучного інтелекту для захисту прав на конфіденційність, одночасно сприяючи технологічному прогресу.

Розробка структурної схеми

Система аналізу криптографічних алгоритмів складається із двох підсистем: підсистеми криптографічного захисту інформації з використанням представників різних класів шифрів і підсистеми криптоаналізу. Структурна схема системи аналізу криптографічних алгоритмів представлена на рисунку 1.

Рисунок 1 – Структурна схема системи криптоаналізу

З нього ми бачимо, що існують чотири структурні блоки які взаємодіють між собою:

– вхідні данні;

– криптоалгоритми, які потребують криптоаналізу;

– власне сама система криптоаналізу, яка складається з використання нейромережі на основі шарів Кохонена та Гроссберга;

– відкритий текст, у якому даються дані про можливість криптоаналізу, того або іншого криптоалгоритму.

Користувач вибирає файл який буде вхідним текстом та відкриває його у програмі.

Потім обирає криптоалгоритм, який хоче протестувати.

Файл, обраний користувачем, шифрується вибраним крптоалгоритмом.

Зашифрований файл підлягає криптоаналізу на базі нейронної мережі та на основі його результатів визначається стійкійкість вибраного алгоритму шифрування.

Висновки. У статті наведені теоретичні узагальнення й рішення наукового завдання дослідження методів визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI. Досліджена система визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI. На основі отриманих результатів досліджень створена програмна реалізація системи визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання визначення рівня стійкості сервісів забезпечення конфіденційності на основі методів AI. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022. (Scopus).

2. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34. (Scopus).

3. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477. (Scopus).
4. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w (Scopus).
5. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
6. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).
7. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
8. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
9. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).
10. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019. (Scopus).
11. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629. (Scopus).
12. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 873-884. (Scopus).
13. Smirnov, O., Kuznetsov, A., Prokopovych-Tkachenko, D. «Hiding Data in Images Using a Pseudo-Random Sequence». ISCI'2020: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko, Victor A. Krasnobayev and Alexandr A. Kuznetsov. ASC Academic Publishing, USA, 2020. pp. 46-59. – ISBN: 978-1-7362833-0-1 (Hardback), ISBN: 978-1-7362833-1-8 (Ebook).
14. Smirnov, O., Kuznetsov, A., Shekhanin, K., Chepurko, I. Detecting Hidden Information in FAT. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 412-429. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
15. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
16. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
17. Смірнов О.А., Дрєєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.
18. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
19. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
20. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).

УДК 004

В.Шевченко, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДОСТУПУ ДО ХМАРНИХ СЕРВІСІВ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ PKI

У статті розроблено програмне забезпечення, яке призначено для системи доступу до хмарних сервісів з використанням технології PKI. Метою розробки є дослідження та програмна реалізація системи доступу до хмарних сервісів з використанням технології PKI. Об'єктом дослідження є процес доступу до хмарних сервісів з використанням технології PKI. Предметом дослідження є методи доступу до хмарних сервісів з використанням технології PKI. Методи дослідження базуються на методах захисту інформації та хмарних обчислень, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи доступу до хмарних сервісів з використанням технології PKI. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, хмарні сервіси, PKI

Постановка проблеми. Сьогодні більшість компаній так чи інакше використовують Інтернет, і із цим зв'язані проблеми захисту віддалених і мобільних користувачів інформаційних систем компанії, захисту корпоративних хмарних сервісів (інтранет-сайту й будь-якого додатка компанії, що працює по http протоколу). Інтернет – це зона підвищеного ризику, відповідно, потрібні спеціальні засоби захисту при роботі віддалених користувачів з WEB-додатками по SSL-протоколу. Таким чином, підсистема захисту WEB-ресурсів вирішує наступні задачі:

- Забезпечення єдиного інтерфейсу до додатків;
- інтегрований контроль доступу до корпоративних хмарних сервісів;
- захист клієнтських браузерів;
- захист хмарних сервісів.

Існує багато технологій захисту хмарних сервісів. У магістерському проекті пропонується система захисту основана на використанні протоколів SSL/TLS, які побудовані з використання інфраструктури відкритих ключів (PKI).

У протоколі SSL/TLS використовується ряд симметричних алгоритмів, асиметричних алгоритмів та геш-функцій. Тому одним із завдань, які потрібно вирішити у даному магістерському проекті є вибір того, або іншого алгоритму, які використовуються на різних етапах побудови системи захисту доступу до хмарних сервісів.

Технологія PKI дозволяє перевіряти й засвідчувати дійсність користувача. PKI забезпечує єдину ідентифікацію, автентифікацію й авторизацію користувачів системи, додатків і процесів і разом із цим гарантує доступність, цілісність і конфіденційність інформації.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи доступу до хмарних сервісів з використанням технології PKI.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи доступу до хмарних сервісів з використанням технології PKI.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем доступу до хмарних сервісів з використанням технології PKI.

– Дослідження системи доступу до хмарних сервісів з використанням технології PKI.

– Програмна реалізація системи доступу до хмарних сервісів з використанням технології PKI.

Об'єктом дослідження є процес доступу до хмарних сервісів з використанням технології PKI.

Предметом дослідження є методи доступу до хмарних сервісів з використанням технології PKI.

Методи дослідження базуються на методах захисту інформації та хмарних обчислень, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Опис SSL/TLS

SSL – криптографічний протокол, що забезпечує безпечну передачу даних по мережі Інтернет. При його використанні створюється захищене з'єднання між клієнтом і сервером. SSL споконвічно розроблений компанією Netscape Communications. Згодом на підставі протоколу SSL 3.0 був розроблений і прийнятий стандарт RFC, що одержав ім'я TLS.

TLS – криптографічний протокол, що забезпечує захищену передачу даних між вузлами в мережі Інтернет. TLS-протокол заснований на Netscape SSL-протоколі версії 3.0 і складається із двох частин – TLS Record Protocol і TLS Handshake Protocol. Розходження між SSL 3.0 і TLS 1.0 незначні, тому далі в тексті термін «SSL» буде відноситися до них обох. TLS Working Group, заснована в 1996 році, продовжує працювати над протоколом.

Опис

TLS надає можливості автентифікації й безпечної передачі даних через Інтернет з використанням криптографічних засобів. Часто відбувається лише автентифікація сервера, у той час як клієнт залишається неавтентифікованим. Для взаємної автентифікації кожна зі сторін повинна підтримувати інфраструктуру відкритого ключа (PKI), що дозволяє захистити клієнт-серверні додатки від перехоплення повідомлень, редагування існуючих повідомлень і створення підроблених.

SSL містить у собі три основних фази:

–Діалог між сторонами, метою якого є вибір алгоритму шифрування.

–Обмін ключами на основі криптосистем з відкритим ключем або автентифікація на основі сертифікатів.

–Передача даних, шифруємих за допомогою симетричних алгоритмів шифрування.

Алгоритм процедури встановлення з'єднання по протоколу TLS handshake

Клієнт і сервер, що працюють по TLS, установлюють з'єднання, використовуючи процедуру handshake ("рукостискання"). Протягом цього handshake, клієнт і сервер приймають угоду щодо параметрів, використовуваних для встановлення захищеного з'єднання. Послідовність дій при встановленні TLS з'єднання:

– клієнт підключається до TLS – підтримуваного сервера й запитує захищене з'єднання;

– клієнт надає список підтримуваних алгоритмів шифрування й геш-функцій;

– сервер вибирає зі списку, наданого клієнтом, найбільш стійкі алгоритми, які також підтримуються сервером, і повідомляє про свій вибір клієнтові;

– сервер відправляє клієнтові цифровий сертифікат для власної ідентифікації. Звичайно цифровий сертифікат містить ім'я сервера, ім'я довіреного центра сертифікації й відкритий ключ сервера;

– клієнт може зв'язатися із сервером довіреного центра сертифікації й підтвердити автентичність переданого сертифіката до початку передачі даних;

– для того щоб згенерувати ключ сесії для захищеного з'єднання, клієнт шифрує випадково згенеровану цифрову послідовність відкритим ключем сервера й посилає результат на сервер. З огляду на специфіку алгоритму асиметричного шифрування,

використовуваного для встановлення з'єднання, тільки сервер може розшифрувати отриману послідовність, використовуючи свій закритий ключ.

Handshake у деталях

Відповідно до протоколу TLS додатки обмінюються записами, інкапсулюючими (що зберігають усередині себе) інформацію, яка повинна бути передана. Кожен із записів може бути стисла, доповнена, зашифрована або ідентифікована MAC залежно від поточного стану з'єднання (стану протоколу). Кожний запис в TLS містить наступні поля: content type (визначає тип умісту запису), поле, що вказує довжину пакета, і поле, що вказує версію протоколу TLS.

Коли з'єднання тільки встановлюється, взаємодія йде по протоколу TLS handshake, content type якого 22.

Нижче описаний простий приклад установлення з'єднання:

1. Клієнт посилає повідомлення **ClientHello**, указуючи найбільш останню версію підтримуваного TLS протоколу, випадкове число й список підтримуваних методів шифрування й стиски, що підходять для роботи з TLS.
2. Сервер відповідає повідомленням ServerHello, що містить: обрану сервером версію протоколу, випадкове число, послане клієнтом, що підходить алгоритм шифрування й стиски зі списку наданого клієнтом.
3. Сервер посилає повідомлення **Certificate**, що містить цифровий сертифікат сервера (залежно від алгоритму шифрування цей етап може бути пропущений)
4. Сервер може запросити сертифікат у клієнта, у такому випадку з'єднання буде взаємно автентифіковано.
5. Сервер відсилає повідомлення **ServerHelloDone**, що ідентифікує закінчення handshake.
6. Клієнт відповідає повідомленням **ClientKeyExchange**, що містить PreMasterSecret відкритий ключ, або нічого (знову ж залежить від алгоритму шифрування).
7. Клієнт і сервер, використовуючи PreMasterSecret ключ і випадково згенеровані числа, обчислюють загальний секретний ключ. Вся інша інформація про ключ буде отримана із загального секретного ключа (і згенерованих клієнтом і сервером випадкових значень).
8. Клієнт посилає **ChangeCipherSpec** повідомлення, що вказує на те, що вся наступна інформація буде зашифрована встановленим у процесі handshake алгоритмом, використовуючи загальний секретний ключ. Це повідомлення рівня записів і тому має тип 20, а не 22.
9. Клієнт посилає повідомлення **Finished**, що містить геш і MAC, згенеровані на основі попередніх повідомлень handshake.
10. Сервер намагається розшифрувати Finished-повідомлення клієнта й перевірити геш і MAC. Якщо процес розшифровки або перевірки не вдається, handshake вважається невдалим і з'єднання повинне бути обірване.
11. Сервер посилає **ChangeCipherSpec** і зашифроване **Finished** повідомлення й у свою чергу клієнт теж виконує розшифровку й перевірку.

Із цього моменту handshake уважається завершеним, протокол установленим. Весь наступний уміст пакетів іде з типом 23, а всі дані будуть зашифровані.

Алгоритми, що використовуються в TLS

У даній поточній версії протоколу доступні наступні алгоритми:

– Для обміну ключами й перевірки їхньої дійсності застосовуються комбінації алгоритмів: RSA (асиметричний шифр), Diffie-Hellman (DH) (безпечний обмін ключами), DSA (алгоритм цифрового підпису) і алгоритми технології Fortezza.

– Для симетричного шифрування: RC2, RC4, IDEA, DES, Triple DES або AES;

– Для геш-функцій: MD5 або SHA.

Алгоритми можуть доповняться залежно від версії протоколу.

На рисунку 1 наведено архітектуру SSL/TLS.

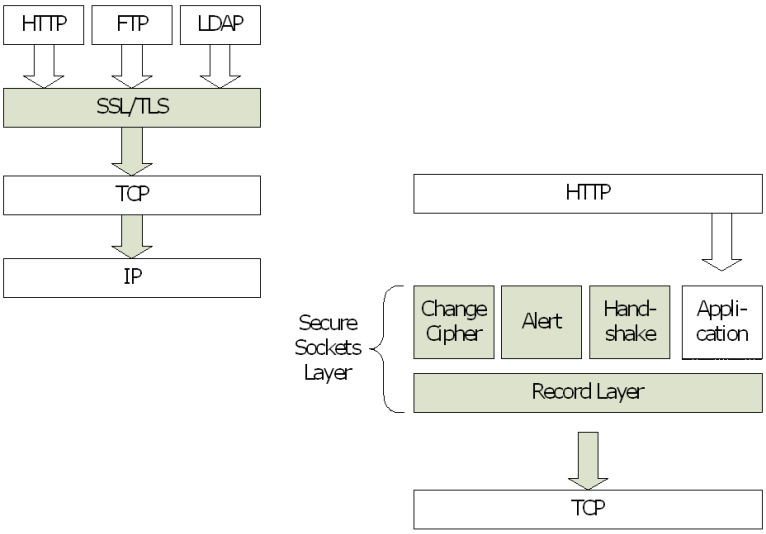


Рисунок 1 – Архітектура SSL/TLS

Як ми бачимо в архітектурі задіяні наступні протоколи:

– HTTP – протокол прикладного рівня передачі даних у першу чергу у вигляді текстових повідомлень. Основою HTTP є технологія «клієнт-сервер», тобто передбачається існування споживачів (клієнтів), які ініціюють з'єднання й надсилають запит, і постачальників (серверів), які очікують з'єднання для одержання запиту, роблять необхідні дії й повертають обернено повідомлення з результатом. HTTP використовується також у якості «транспорту» для інших протоколів прикладного рівня, таких як SOAP. Основним об'єктом маніпуляції в HTTP є ресурс, на який вказує URI (Uniform Resource Identifier) у запиті клієнта. Звичайно такими ресурсами є файли, що зберігаються на сервері, але ними можуть бути логічні об'єкти або щось абстрактне. Особливістю протоколу HTTP є можливість указати в запиті й відповіді спосіб подання того самого ресурсу по різних параметрах: формату, кодуванню, мові й т.д. Саме завдяки можливості вказівки способу кодування повідомлення клієнт і сервер можуть обмінюватися двійковими даними, хоча даний протокол є текстовим. HTTP – протокол прикладного рівня, аналогічними йому є FTP і SMTP. Обмін повідомленнями йде за звичайною схемою «запит-відповідь». Для ідентифікації ресурсів HTTP використовує глобальні URI. На відміну від багатьох інших протоколів, HTTP не зберігає свого стану. Це означає відсутність збереження проміжного стану між парами «запит-відповідь». Компоненти, що використовують HTTP, можуть самостійно здійснювати збереження інформації про стан, пов'язаної з останніми запитами й відповідями. Браузер, що посилає запити, може відслідковувати затримки відповідей. Сервер може зберігати IP-адреси й заголовки запитів останніх клієнтів. Однак сам протокол не обізнаний про попередні запити й відповіді, у ньому не передбачена внутрішня підтримка стану, до нього не пред'являються такі вимоги.

– FTP – протокол, призначений для передачі файлів у комп'ютерних мережах. FTP дозволяє підключатися до серверів FTP, переглядати вміст каталогів і завантажувати файли із сервера або на сервер; крім того, можливий режим передачі файлів між серверами. Протокол FTP відноситься до протоколів прикладного рівня й для передачі даних використовує транспортний протокол TCP. Команди й дані, на відміну від більшості інших протоколів передаються по різних портах. Порт 20 використовується для передачі даних, порт 21 для передачі команд. Протокол не шифрується, при автентифікації передає логін і пароль відкритим текстом. Якщо зловмисник перебуває в одному сегменті мережі з користувачем FTP, те, використовуючи сніффер, він може перехопити логин і пароль користувача, або, при наявності спеціального ПЗ, одержувати передані по FTP файли без авторизації. Щоб запобігти перехопленню трафіку, необхідно використовувати протокол

шифрування даних SSL, що підтримується багатьма сучасними FTP-серверами й деякими FTP-Клієнтами.

– LDAP – це мережний протокол для доступу до служби каталогів X.500, розроблений IETF як полегшений варіант розробленого ITU-T протоколу DAP. LDAP – відносно простий протокол, що використовує TCP/IP і дозволяє робити операції автентифікації (bind), пошуку (search) і порівняння (compare), а також операції додавання, зміни або видалення записів. Звичайно LDAP-сервер приймає вхідні з'єднання на порт 389 по протоколах TCP або UDP. Для LDAP-сеансів, інкапсульованих в SSL, звичайно використовується порт 636. Усякий запис у каталозі LDAP складається з одного або декількох атрибутів і має унікальне ім'я (DN). Унікальне ім'я складається з одного або декількох відносних унікальних імен (RDN), розділених комою. На одному рівні каталогу не може існувати двох записів з однаковими відносними унікальними іменами. У силу такої структури унікального ім'я запису в каталозі LDAP можна легко представити у вигляді дерева. Запис може складатися тільки з тих атрибутів, які визначені в описі класу запису (object class), які, у свою чергу, об'єднані в схеми (schema). У схемі визначено, які атрибути є для даного класу обов'язковими, а які – необов'язковими. Також схема визначає тип і правила порівняння атрибутів. Кожний атрибут запису може зберігати кілька значень.

– TCP – один з основних мережних протоколів Internet, призначений для керування передачею даних у мережах і під мережах TCP/IP. Виконує функції протоколу транспортного рівня спрощеної моделі OSI. IP-ідентифікатор – 6. TCP – це транспортний механізм, що надає потік даних, з попередньою установкою з'єднання, за рахунок цього даючий впевненість у вірогідності одержуваних даних, здійснює повторний запит даних у випадку втрати даних і усуває дублювання при одержанні двох копій одного пакета. На відміну від UDP, гарантує, що додаток одержить дані точно в такій же послідовності, у якій вони були відправлені, і без втрат. Реалізація TCP як правило, убудована в ядро системи, хоча є й реалізації TCP у контексті додатка. TCP часто позначають «TCP/IP». Коли здійснюється передача від комп'ютера до комп'ютера через Internet, TCP працює на верхньому рівні між двома кінцевими системами, наприклад, інтернет-браузер і інтернет-сервер. Також TCP здійснює надійну передачу потоку байт від однієї програми на деякому комп'ютері в іншу програму на іншому комп'ютері. Програми для електронної пошти й обміну файлами використовують TCP. TCP контролює довжину повідомлення, швидкість обміну повідомленням, мережний трафік.

– IP – маршрутизуємий мережний протокол, основа стека протоколів TCP/IP. Протокол IP використовується для негарантованої доставки даних (поділюваних на так звані пакети) від одного вузла мережі до іншого. Це означає, що на рівні цього протоколу (третій рівень мережної моделі OSI) не дається гарантій надійної доставки пакета до адресата. Зокрема, пакети можуть прийти не в тому порядку, у якому були відправлені, продублюватися (коли приходять дві копії одного пакета; у реальності це буває вкрай рідко), виявитися ушкодженими (звичайно ушкоджені пакети знищуються) або не прибути зовсім. Гарантії безпомилкової доставки пакетів дають протоколи більш високого (транспортного) рівня мережної моделі OSI – наприклад, TCP – які IP використовують як транспорт. У сучасній мережі Інтернет використовується IP четвертої версії, також відомий як IPv4. У протоколі IP цієї версії кожному вузлу мережі відноситься у відповідність IP-адреса довжиною 4 октети (іноді говорять «байта», маючи на увазі розповсюджений восьмибітовий мінімальний адресуємий фрагмент пам'яті ЕОМ). При цьому комп'ютери в підмережах поєднуються загальними початковими бітами адреси. Кількість цих біт, загальна для даної підмережі, називається маскою підмережі (раніше використовувалося ділення простору адрес по класах – A, B, C; клас мережі визначався діапазоном значень старшого октету й визначав число адресуємих вузлів у даній мережі, зараз використовується безкласова адресація). У поточний час вводиться в експлуатацію шоста версія протоколу – IPv6, що дозволяє адресувати значно більшу кількість вузлів, чим IPv4. Ця версія відрізняється підвищеною розрядністю адреси, убудованою можливістю шифрування й деяких інших

особливостей. Перехід з IPv4 на IPv6 пов'язаний із трудомісткою роботою операторів зв'язку й виробників програмного забезпечення й не може бути виконаний одномоментно.

Опис алгоритму шифрування AES

Шифрування

Для AES довжина input (блоку вхідних даних) і State (стану) постійна й дорівнює 128 бітів, а довжина шифроключа **K** становить 128, 192, або 256 бітів. Для позначення обраних довжин input, State і Cipher Key у байтах використовується нотація $Nb = 4$ для input і State, $Nk = 4, 6, 8$ для Cipher Key відповідно для різних довжин ключів.

На початку шифрування input копіюється в масив State за правилом $s[r,c] = \text{in}[r + 4c]$, для $0 \le r \le 4$ і $0 \le c < Nb$. Після цього до State застосовується процедура AddRoundKey() і потім State проходить через процедуру трансформації(раунд) 10, 12, або 14 разів (залежно від довжини ключа), при цьому треба врахувати, що останній раунд трохи відрізняється від попередніх. У підсумку, після завершення останнього раунду трансформації, State копіюється в output за правилом $\text{out}[r + 4c] = s[r,c]$, для $0 \le r \le 4$ і $0 \le c < Nb$.

SubBytes()

У процедурі SubBytes, кожний байт в state заміняється відповідним елементом у фіксованої 8-бітній таблиці пошуку, S; $b_{ij} = S(a_{ij})$.

Процедура SubBytes() обробляє кожний байт стану, незалежно роблячи нелінійну заміну байтів використовуючи таблицю замін (S-box). Така операція забезпечує нелінійність алгоритму шифрування. Побудова S-box складається із двох кроків. По-перше, виробляється взяття оберненого числа в $GF(2^8)$. По-друге, до кожного байта b з яких складається S-box застосовується наступна операція:

$$b_s^{'} = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i ,$$

де $0 \le i \le 8$, і де b_i є i-ий біт b, а c_i – i-ий байт c = {63} або {01100011}. У такий спосіб забезпечується захист від атак заснованих на простих алгебраїчних властивостях.

У процедурі ShiftRows, байти в кожному рядку state циклічно зсувається вліво. Розмір зсуву байтів кожного рядка залежить від його номера.

ShiftRows працює з рядками State. При цій трансформації рядка стани циклічно зсувається на r байт по горизонталі, залежно від номера рядка. Для нульового рядка $r = 0$, для першого рядка $r = 1$, і т.д. У такий спосіб кожний стовпчик вихідного стану після застосування процедури ShiftRows складається з байтів з кожного стовпчика початкового стану. Для алгоритму Rijndael паттерн зсуву рядків для 128 і 192-бітних рядків однаковий. Однак для блоку розміром 256 бітів відрізняється від попередніх тим, що 2, 3, і 4-ий рядки зміщаються на 1, 3, і 4 байти відповідно.

MixColumns()

У процедурі MixColumns, кожний стовпчик стану перемножується з фіксованим багаточленом $c(x)$.

У процедурі MixColumns, чотири байти кожного стовпчика State змішуються використовуючи для цього оборотну лінійну трансформацію. MixColumns обробляє стан по стовпчиках, трактуючи кожен з них як поліном четвертого ступеня. Над цими поліномами виробляється множення в $GF(2^8)$ по модулю $x^4 + 1$ на фіксований багаточлен $c(x) = 3x^3 + x^2 + x + 2$. Разом з ShiftRows, MixColumns вносить diffusion у шифр.

AddRoundKey()

У процедурі AddRoundKey, кожний байт стану поєднується з RoundKey використовуючи XORoperation(). У процедурі AddRoundKey, RoundKey кожного раунду поєднується з State. Для кожного раунду Roundkey виходить із CipherKey використовуючи процедуру KeyExpansion; кожний RoundKey такого ж розміру, що й State. Процедура робить побітовий XOR кожного байта State з кожним байтом RoundKey.

KeyExpansion()

AES алгоритм використовуючи процедуру KeyExpansion() і подаючи в неї Cipher Key, K, одержує ключі для всіх раундів. Усього вона одержує $Nb*(Nr + 1)$ слів: с початку для алгоритму потрібен набір з Nb слів, і кожному з Nr раундів потрібно Nb ключових наборів даних. Отриманий масив ключів для раундів позначається як $w[i]$, $0 \le i < Nb*(Nr + 1)$. Алгоритм KeyExpansion() показаний у псевдокоді нижче

Функція SubWord() бере чотирьохбайтне вхідне слово й застосовує *S*-box до кожного із чотирьох байтів, те, що вийшло подається на вихід. На вхід процедури RotWord() подається слово $[a_0,a_1,a_2,a_3]$ яке вона циклічно переставляє й повертає $[a_1,a_2,a_3,a_0]$. Масив слів, слів постійний для даного раунду, $Rcon[i]$, містить значення $[x^{i-1},00,00,00]$, де $x = \{02\}$, а x^{i-1} є ступенем x в $GF(2^8)$ (i починається з 1).

Перші Nk слів розширеного ключа заповнені Cipher Key. У кожне наступне слово, $w[i]$, кладе значення отримане при операції XOR $w[i\text{-}1]$ і $w[i - Nk]$. Для слів, позиція яких кратна Nk, перед XOR'ом до $w[i\text{-}1]$ застосовується трасформація, за якою слідує XOR з константою раунду $Rcon[i]$. Зазначена вище трансформація складається із циклічного зсуву байтів у слові(RotWord()), за якою слідує процедура SubWord() – теж саме, що й SubBytes(), тільки вхідні й вхідні дані будуть розміром у слово.

Важливо помітити, що процедура KeyExpansion() для 256 бітного Cipher Key не набагато відрізняється від тих, які застосовуються для 128 і 192 бітних шифроключів. Якщо $Nk = 8$ й $i - 4$ кратно Nk, то SubWord() застосовується до $w[i\text{-}1]$ до XOR'a.

Опис протоколу обміну ключами Діффі-Хеллмана

Ціль алгоритму полягає в тому, щоб два учасники могли безпечно обмінятися ключем, що надалі може використовуватися в якому-небудь алгоритмі симетричного шифрування. Сам алгоритм Діффі-Хеллмана може застосовуватися тільки для обміну ключами. Алгоритм заснований на труднощі обчислень дискретних логарифмів. Дискретний логарифм визначається в такий спосіб. Уводиться поняття примітивного кореня простого числа Q як числа, чиї ступені створюють всі цілі від 1 до $Q - 1$. Це означає, що якщо A є примітивним коренем простого числа Q, тоді числа: $A \bmod Q$, $A^2 \bmod Q$, . . . , $A^{Q-1} \bmod Q$, є різними й складаються із цілих від 1 до $Q - 1$ з деякими перестановками. У цьому випадку для будь-якого цілого $Y < Q$ і примітивного кореня A простого числа Q можна знайти єдину експоненту X, таку, що:

$$Y = A^X \bmod Q, \text{ де } 0 \le X \le (Q - 1).$$

Експонента X називається дискретним логарифмом, або індексом Y, по підставі $A \bmod Q$. Це позначається як:

$$\text{ind}_{A, Q}(Y).$$

Тепер опишемо алгоритм обміну ключів Діффі-Хеллмана.

Загальновідомі елементи

Q – просте число

A – $A < Q$ і A є примітивним коренем Q

Створення пари ключів клієнтом

Вибір випадкового числа X_i (*закритий ключ*): $X_i < Q$

Обчислення числа Y_i (*відкритий ключ*): $Y_i = A^{Xi} \bmod Q$

Створення відкритого ключа сервером

Вибір випадкового числа X_j (*закритий ключ*): $X_j < Q$

Обчислення випадкового числа Y_j (*відкритий ключ*): $Y_j = A^{Xj} \bmod Q$

Створення спільного секретного ключа клієнтом

$$K = (Y_j)^{Xi} \bmod Q.$$

Створення спільного секретного ключа сервером

$$K = (Y_i)^{Xj} \bmod Q.$$

Розпишемо алгоритм більш докладно. Передбачається, що існують два відомих усім числа: просте число Q і ціле A, що є примітивним коренем Q. Тепер припустимо, що клієнт та сервер хочуть обмінятися ключем для алгоритму симетричного шифрування. Клієнт

вибирає випадкове число $X_i < Q$ і обчислює $Y_i = A^{Xi} \bmod Q$. Аналогічно сервер незалежно вибирає випадкове ціле число $X_j < Q$ і обчислює $Y_j = A^{Xj} \bmod Q$. Кожна сторона тримає значення X у секреті й робить значення Y доступним для іншої сторони. Тепер клієнт обчислює ключ як $K = (Y_j)^{Xi} \bmod Q$, і сервер обчислює ключ як $K = (Y_i)^{Xj} \bmod Q$. У результаті обоє одержать те саме значення:

$$K = (Y_j)^{Xi} \bmod Q = (A^{Xj} \bmod Q)^{Xi} \bmod Q = (A^{Xj})^{Xi} \bmod Q =$$

(за правилами модульної арифметики)

$$= A^{Xj\,Xi} \bmod Q = (A^{Xj})^{Xj} \bmod Q = (A^{Xi} \bmod Q)^{Xj} \bmod Q = (Y_i)^{Xj} \bmod Q$$

Таким чином, дві сторони обмінялися секретним ключем. Так як X_i і X_j є закритими, зловмисник може одержати тільки наступні значення: Q, A, Y_i і Y_j. Тобто дві взаємодіючі сторони по відкритому каналу змогли поширити секретний ключ не розкриваючи його третій стороні.

Для обчислення ключа атакуючий повинен зламати дискретний логарифм, тобто обчислити: $X_j = \mathrm{ind}_{a,\,q}(Y_j)$.

Безпека обміну ключа в алгоритмі Діффі-Хеллмана випливає з того факту, що, хоча відносно легко обчислити експоненти за модулем простого числа, дуже важко обчислити дискретні логарифми. Для великих простих чисел задача вважається нерозв'язною.

Варто помітити, що даний алгоритм уразливий для атак типу "in-the-middle". Якщо зловмисник може здійснити активну атаку, тобто має можливість не тільки перехоплювати повідомлення, але й заміняти їх іншими, він може перехопити відкриті ключі учасників Y_i і Y_j , створити свою пару відкритого й закритого ключа (X_{on}, Y_{on}) і послати кожному з учасників свій відкритий ключ. Після цього кожний учасник обчислить ключ, що буде спільним із зловмисником, а не з іншим учасником. Якщо немає контролю цілісності, то учасники не зможуть виявити подібну підміну.

Розробка структурної схеми

Розроблене програмне забезпечення представляє із себе набір компонентів призначених для забезпечення політики безпеки як у вже існуючих, так і в створюваних мережних інформаційних системах.

Розроблене програмне забезпечення дозволяє забезпечити захист переданої по мережі інформації, строгу взаємну автентифікацію користувачів і серверів, гнучке розмежування доступу. Для реалізації цих функцій у системі використовуються SSL/TLS протоколи й X.509 цифрові сертифікати, тобто універсальні, що стали стандартом де-факто, механізми, підтримувані практично всіма розповсюдженими Веб-агентами.

За допомогою розробленого програмного забезпечення легко забезпечуються вимоги по інформаційній безпеці, запропоновані різними Інтернет додатками, такими як сервера платіжних систем, інтернет-магазини, багатопрофільні корпоративні Веб-сервера, що містять інформацію з різним рівнем конфіденційності, B2B системи, системи захищеного документообігу, обміну електронною поштою й багато які інші.

На рисунку 2 представлена структурна схема розробленої системи. На цій схемі введені наступні позначення:

- ЕЦП – електроний цифровий підпис;
- ЦС – цифровий сертифікат;
- PKI – інфраструктура відкритих ключів;
- DVCS – Data Validation and Certification Server Protocols – протокол підтвердження даних та сертифікації серверу;
- OCSP – Online Certificate Status Protocol – онлайн протокол статусу сертфікату;
- TSP – Time-Stamp Protocol – протокол часових міток;
- TLS – криптографічний протокол, що забезпечує захищену передачу даних між вузлами в мережі Інтернет;
- RFC – документ, у якому описується той або інший стандарт.

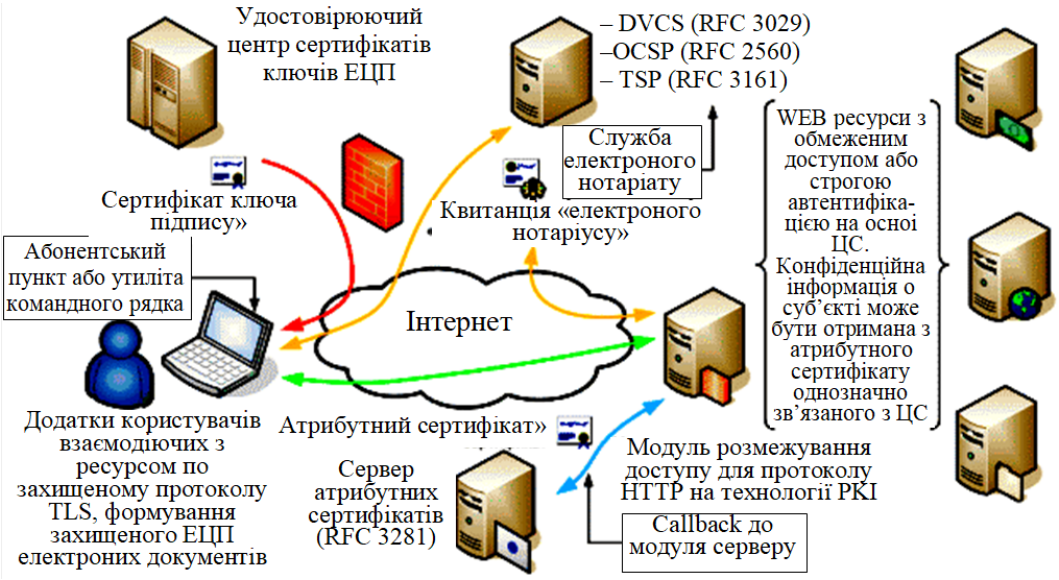


Рисунок 2 – Структурна схема

Розглянемо більш детально складові структурної схеми, та принцип роботи системи.

Центр, що засвідчує, сертифікатів ключів підпису

Центр, що засвідчує, сертифікатів ключів підпису (удостовірюючий центр – УЦ) є повністю вітчизняною розробкою:

– відповідає вимогам Доктрини інформаційної безпеки в плані заміщення імпортних технічних і програмних засобів у українських інформаційних системах;

– завдяки відомості всіх криптографічних процедур в ізольований криптографічний PKSC#11 токен, в УЦ можуть використовуватися як вітчизняні криптографічні алгоритми, так і закордонні;

– використовувані вітчизняні криптографічні механізми й протоколи опираються на інтернет драфти, розроблені вітчизняними компаніями, а отже, рішення сумісні з рішеннями інших вітчизняних виробників;

– технічна реалізація заснована на сучасних керівних документах, стандартах і міжнародних рекомендаціях, що дозволяє:

1) в одному технічному рішенні підтримувати невиразно велике число зовсім ізольованих, у тому числі з різними криптографічними алгоритмами, видавців;

2) підтримуються механізми кросування видавців (аж до рівня мостового УЦ), у тому числі й зовнішніх, для утворення єдиних зон обігу захищених документів;

3) для систем наближених до On-line передбачене поширення відновлень списків відкликаних сертифікатів (delta CRL);

4) компоненти самого УЦ для побудови ланцюжків сертифікації використовують внутрішній сервіс OCSP, що може бути оформлений як корпоративний зі складу служби "електронного нотаріату";

5) до складу системи входить власна служба роздачі міток часу, з механізмами підстроювання під зовнішні еталони;

6) відповідно до RFC 3039 Internet X.509 Public Key Infrastructure. Qualified Certificate Profile у сертифікат може бути введений серійний номер імені, тим самим вирішена "колізія імен" – проблема «однофамільців»;

7) реєстр крім самого сертифіката користувача може містити додаткову інформацію про суб'єкта, включаючи графічні елементи (фотографії, відбитки пальців і т.п.);

8) всі інформаційні блоки при транспортуванні й зберіганні захищені електронними цифровими підписами, що забезпечує цілісність, авторство й невідрікаємість і спрощує процедури розбору конфліктних ситуацій.

Центр, що засвідчує, сертифікатів ключів підпису (УЦ) є основою комп'ютерних систем захищеного документообігу на технології відкритого розподілу ключів (Public Key Infrastructure (PKI)). Технічна реалізація Центра, що засвідчує, відповідає вимогам Закону України "Про електронний цифровий підпис" за умови використання в ЦУ сертифікованих ДССЗЗІ засобів електронного цифрового підпису. УЦ, може виступати як ключовий компонентом для різного типу прикладних захищених систем корпоративного рівня (захищений документообіг, Інтернет-банкінг, білінгові системи, електронна комерція (B2C, B2B), Інтернет процесінг і.т.п.).

Сертифікат X.509 v3

Сертифікат X.509 v3 визначається в такий спосіб. Для обчислення підпису дані, які повинні бути підписані, представляються з використанням ASN.1 однозначних правил подання (DER).

Поля сертифіката. Сертифікат є послідовність трьох обов'язкових полів: tbsCertificate, signatureAlgorithm і signatureValue.

−tbsCertificate. Поле містить імена суб'єкта й випускаючого, відкритий ключ, пов'язаний із суб'єктом, період дійсності й іншу пов'язану із цим сертифікатом інформацію. Поля докладно описані далі; tbsCertificate звичайно включають розширення, які теж будуть описані нижче.

−signatureAlgorithm. Поле signatureAlgorithm містить ідентифікатор криптографічного алгоритму, використовуваного УЦ для підписування даного сертифіката. Існують стандартні алгоритми, які повинні підтримуватися всіма реалізаціями, але конкретна реалізація може підтримувати й інші алгоритми.

Ідентифікатор алгоритму визначається наступної ASN.1-структурою:

```
AlgorithmIdentifier ::= SEQUENCE  {
  algorithm  OBJECT IDENTIFIER,
  parameters ANY DEFINED BY algorithm OPTIONAL
}
```

Ідентифікатор алгоритму використовується для визначення криптографічного алгоритму. Компонент OBJECT IDENTIFIER ідентифікує алгоритм (такий як DSA з SHA-1). Компоненти поля параметрів змінюються відповідно до зазначеного алгоритму. Поле повинне містити той же самий ідентифікатор алгоритму, що й поле підпису в tbsCertificate.

−signatureValue. Поле signatureValue містить цифровий підпис, обчислений для поля tbsCertificate, записаному в DER-поданні ASN.1. Це означає, що поле tbsCertificate, представлене як ASN.1 DER, використовується як вхід у функцію підпису. Отримане значення підпису представлене як BIT STRING і включено в поле підпису. Деталі даного процесу можуть відрізнятися для кожного конкретного алгоритму підпису. Створенням даного підпису УЦ підтверджує дійсність інформації в поле tbsCertificate. Зокрема, УЦ підтверджує зв'язок між матеріалом відкритого ключа й суб'єктом сертифіката.

−TBSCertificate. Послідовність TBSCertificate містить інформацію, пов'язану із суб'єктом сертифіката й УЦ, що випустив сертифікат. Кожний TBSCertificate містить імена суб'єкта й випускаючого, відкритий ключ, пов'язаний із суб'єктом, період дійсності, номер версії й серійний номер сертифіката; деякі поля можуть (але це не обов'язково) містити унікальний ідентифікатор. Розглянемо синтаксис і семантику таких полів. TBSCertificate звичайно включає розширення. Розглянемо також найбільше часто використовувані в Internet розширення.

−Version. Дане поле описує версію подання сертифіката. Якщо використовуються розширення, то версія повинна бути 3 (значення – 2). Якщо розширення не зазначені, але UniqueIdentifier представлений, версія може бути 2 (значення – 1); але версія може бути й 3. Якщо представлені тільки базові поля, версія може бути 1 (значення в сертифікаті опущене як значення за замовчуванням); але версія може бути 2 або 3. Реалізації повинні бути готові приймати будь-яку версію сертифіката. Як мінімум конформні реалізації повинні розпізнавати версію 3 сертифікатів.

–Serial number. Серійний номер повинен бути позитивним цілим, призначуваним УЦ для кожного сертифіката. Він повинен бути унікальним для кожного сертифіката, випущеного даним УЦ. Таким чином, ім'я що выпустили й серійний номер однозначно визначають сертифікат. Cas повинні забезпечувати, щоб серійні номери були ненегативними цілими. Уважається, що серійні номери можуть мати довжину до 20 октетів.

–Signature. Дане поле містить ідентифікатор алгоритму, використовуваного УЦ для підписування сертифіката.

–Дане поле повинне містити той же самий ідентифікатор алгоритму, що й поле signatureAlgorithm в Certificate. Зміст необов'язкового поля параметрів залежить від конкретного алгоритму.

–Issuer. Поле issuer ідентифікує того, хто підписав і випустив сертифікат. Поле issuer повинне містити непусте унікальне ім'я (DN). Ім'я визначається у відповідності з наступної ASN.1 структурою:

```
Name ::= CHOICE { RDNSequence }
RDNSequence ::= SEQUENCE OF
   RelativeDistinguishedName
RelativeDistinguishedName ::=
   SET OF AttributeTypeAndValue
AttributeTypeAndValue ::= SEQUENCE {
   type   AttributeType,
   value  AttributeValue
}
AttributeType ::= OBJECT IDENTIFIER
AttributeValue ::= ANY DEFINED BY
   AttributeType
```

Ім'я описує ієрархічне ім'я, що складається з атрибутів, таких, наприклад, як назва країни, і відповідних значень, таких як RU. Тип компонента AttributeValue визначається значенням AttributeType. Стандарт X.509 не обмежує набір типів атрибутів, які можуть з'явитися в ім'ї. Проте, стандартом рекомендується підтримувати наступні типи атрибутів в іменах випускаючі й суб'єкта:

- Країна.
- Організація.
- Організаційна одиниця.
- Позначення унікального імені.
- Назва штату або регіону.
- Загальноприйняте ім'я (наприклад, Іванов Іван).
- Серійний номер.

Додатково можуть бути присутнім деякі інші типи атрибутів в іменах випускаючі й суб'єкта, наприклад:

- Локалізація.
- Заголовок.
- По батькові.
- Призначене ім'я.
- Ініціали.
- Псевдонім.
- Спеціальна назва (наприклад, "Jr.", " 3-ій" або "IV").

Також може бути присутнім атрибут domainComponent. DNS надає собою ієрархічну систему позначення ресурсів. Даний атрибут надає зручний механізм для організацій, які хочуть використовувати унікальні DN імена паралельно зі своїми DNS-Іменами. Це не заміняє dNSName компонент альтернативного поля ім'я. Стандарт не вимагає конвертувати такі імена в DNS-Імена. Сторона, що перевіряє, повинна обробляти поля унікального ім'я

випускаючого й унікального ім'я суб'єкта для одержання ланцюжка імен при перевірці *дійсності сертифікаційного* шляху. Ланцюжок імен виходить у випадку відповідності унікального ім'я випускаючого в першому сертифікаті ім'я суб'єкта в сертифікаті УЦ.

Служба "Електронного нотаріату"

"Електронний нотаріус" (ЕН) може бути як додатковим сервісом в комп'ютерній системі, що виконує функції Центра, що засвідчує (УЦ) сертифікатів ключів підпису в якості стандартизованогоRFC 3029 Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols (DVCS). RFC 2560 Online Certificate Status Protocol – OCSP. RFC 3161 Time-Stamp Protocol (TSP)) технічного рішення «Про Електронний цифровий підпис», так і як самостійний програмний комплекс, і реалізовувати разову або абонентську послугу з перевірки й сертифікації інформації, перевірки сертифікатів і виробленню квитанції, що містять «штамп» часу. В ЕН зведені служби, технічна реалізація яких стандартизована міжнародними рекомендаціями, по фактах усіляких перевірок, підтверджень і вироблення «штампа часу» для зовнішніх компонентів інфраструктури відкритих ключів.

"Електронний нотаріус" виконує наступні функції:

–Посвідчення факту володіння інформацією з або без її подання сервісу.

–Перевірка дійсності ЕЦП.

–Перевірка дійсності сертифіката відкритого ключа (для компонентів DVCS або OCSP).

–Вироблення квитанції, що містить «штамп» часу (TSP).

Задачі, у рішенні яких, може бути використана Служба "Електронного нотаріату":

1. Створення єдиного домена захищеного електронного документообігу, у тому числі побудованому на несумісних між собою засобах криптографічного захисту інформації, але маючих сертифікат ДССЗЗІ на засоби криптографічного захисту інформації для гетерогенних програмно-апаратних платформ.

2. Одержання штампа «дійсного часу для даної PKI системи» на завіреному електронному документі. Досить важливо (для попередження шахрайських дій або колізій) при завіренні електронного документа коректно вказувати дату підписання, однак проставляння дійсної дати цілком є відповідальністю підписується стороны, що. ЕН у цьому випадку є «третьою» стороною – довіреному арбітром, що фіксує факт наявності дійсної ЕЦП на конкретний момент часу. Даний сервіс іноді називають Time Stamping. Наявність «третьої» незалежної сторони може виявитися корисною щоб зафіксувати певний етап у технологічному ланцюжку документообігу (якийсь документ пройшов яку те стадію свого формування), наприклад, на конкретний момент часу податкова декларація завірена й доставлена від платника податків в інспекцію. У більше широкому змісті ЕН може бути використаний абстрактною прикладною системою як джерело TSP міток «еталонного часу».

3. Тривале архівне зберігання електронних документів. ЕЦП на електронному документі має «строк життя», що, зокрема визначається «строком життя» персонального сертифіката, який бере участь у формуванні ЕЦП. Через багато причин цей час досить обмежений, що не дозволяє будувати повноцінну систему документообігу, включаючи такий важливий компонент як архівне зберігання. Наявність DVC квитанцій по перевірці ЕЦП дозволяє робити висновки про дійсність ЕЦП уже після витікання часу дійсності сертифіката, що брав участь у виробленні даної ЕЦП. Дана властивість пояснюється тим, що сертифікат ЕН (DVCS), яким завірена квитанція, більш тривала й існують механізми пролонгації квитанцій (випуск квитанції на квитанцію).

4. Організація перевірки ЕЦП «третьою» стороною для користувачів дозволяє перевести сам факт перевірки із площини криптографічних обчислень на сертифікованих серверах захисту інформації в площину організації довіреної доставки квитанцій із сервера ЕН, що в багатьох випадках значно технологичніше. Ступінь «доручення» доставки квитанцій не регламентується законом «Про ЕЦП» і цілком визначається специфікою комп'ютерної системи, у якій використовуються завірені документи. Способів доставки досить багато: від доставки квитанції кур'єрською службою, підтвердженням по телефоні,

порівнянням самих файлів – квитанцій отриманих по мережі й з репозиторія ЕН (DVCS), до організації захищених сегментів мережі, на підтвердження що мають, наприклад, атестат ДССЗЗІ.

5. Покладання на сервіс ЕН функцію перевірки дійсності якогось цифрового сертифіката істотно спрощує комп'ютерну систему, у якій циркулюють завірені електронні документи. Сама по собі процедура перевірки сертифіката досить трудомістка, необхідно побудувати ланцюжок перевірки кінцевого сертифіката з перевіркою всіх проміжних кореневих сертифікатів, визначити місце розповсюдження, одержати й обробити списки відкликаних сертифікатів і т.п.

6. Даний сервіс може бути досить корисний для комп'ютерних систем (КС), у яких використовується факт володіння користувачем якоюсь інформацією без її опублікування. Наприклад, КС проведення різних тендерів, регламент яких визначає, що до певного строку ніхто не повинен мати доступ до конкурсного матеріалу (за винятком коротких анотацій) і тільки по настанню часу початку конкурсу «конверти» повинні бути розкриті. Для таких систем учасники представляють квитанції на істинність ЕЦП конкурсного матеріалу без фактичної передачі самого матеріалу до моменту настання конкурсу. Істинність представленого в наслідку матеріалу підтверджено ЕЦП зі складу DVC квитанції. У цьому випадку захист конкурсного матеріалу покладає на самих конкурсантів – самих зацікавлених у захисті даних осіб і повністю знімає ризик шахрайства в системі.

Служба атрибутування (реалізація заснована на RFC 3281) вирішує дві задачі:

– Дозволяє здійснити криптографічний зв'язок сертифіката ключа підпису з додатковою інформацією, захищеної ЕЦП, що визначає роль власника сертифіката в КС, наприклад, для цілей розмежування доступу, розміщення персональної інформації, розміщення інформації уточнюючого повноваження й т.п.

– Дозволяє здійснити криптографічний зв'язок між абстрактним блоком даних і додатковою інформацією (метаданими), наприклад, такий атрибутний контейнер можна асоціювати з електронним документом разом з метаданими при міжсистемному (міжвідомчому) інформаційному обміні. Додатково, використовувана технологія дозволяє (ЕЦП у вигляді CMS або PKCS#7, або «підпис із розширеними даними для перевірки» по ETSI TS 101 733 не може це забезпечити) ввести поняття строку дійсності документа, включаючи механізми екстреного визнання розміщеної в контейнері інформації недійсною. Дана унікальна властивість може бути використана при випуску електронних дозволів, наприклад, імпортно-карантинні дозволи, ліцензії (у тому числі й на програмне забезпечення) і т.п.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів доступу до хмарних сервісів з використанням технології PKI. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем доступу до хмарних сервісів з використанням технології PKI; Досліджена система доступу до хмарних сервісів з використанням технології PKI; На основі отриманих результатів досліджень створена програмна реалізація системи доступу до хмарних сервісів з використанням технології PKI; Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання доступу до хмарних сервісів з використанням технології PKI; Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

2. Смірнов О.А., Дрєєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.
3. Смирнов А.А., Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений. Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.
4. Смирнов А.А., Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения. Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.
5. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 3(69). С. 93-98. 2022.
6. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.
7. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.
8. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95
9. Смирнов А., Кузнецов А., Кузнецова Т. «Шумоподобные дискретные сигналы для асинхронных систем кодового разделения радиоканалов». Радиотехника, № 2(205), 175–183. 2021.
10. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.
11. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12. (Scopus).
12. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022. (Scopus).
13. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34. (Scopus).
14. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477. (Scopus).
15. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w (Scopus).
16. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
17. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).
18. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
19. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
20. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).

УДК 004

О.Дзюбинський, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ВИЯВЛЕННЯ НЕСПРАВНОСТІ ЕЛЕМЕНТІВ ЦИФРОВИХ ПРИСТРОЇВ

У статті розроблено програмне забезпечення, яке призначено для системи виявлення несправності елементів цифрових пристроїв. Метою розробки є дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв. Об'єктом дослідження є процес виявлення несправності елементів цифрових пристроїв. Предметом дослідження є методи виявлення несправності елементів цифрових пристроїв. Методи дослідження базуються на методах системотехніки, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи виявлення несправності елементів цифрових пристроїв. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, виявлення несправності, цифрові пристрої

Постановка проблеми. Цифрові пристрої в даний час активно використовуються для Інтернету речей. Інтернет речей (IoT) отримав значне визнання і став новою парадигмою сенсорної взаємодії з фізичним світом в епоху Індустрії 4.0. Інтернет речей використовується в багатьох різноманітних програмах, які є частиною нашого життя та стають глобальними цифровими нервовими системами. Цілком очевидно, що в найближчому майбутньому сотні мільйонів людей і бізнесів з мільярдами матимуть розумні датчики та передові комунікаційні технології, і ці речі розширять межі існуючих систем.

Це призведе до потенційних змін у тому, як ми працюємо, навчаємося, впроваджуємо інновації, живемо та розважаємося. Гетерогенні розумні датчики в Інтернеті речей є незамінними частинами, які збирають необроблені дані з фізичного світу, будучи першим портом контакту.

Часто датчики в IoT розгортають або встановлюють у суворих умовах. Це неминуче означає, що датчики схильні до виходу з ладу, несправності, швидкого зношення, зловмисних атак, крадіжок і втручання. Усі ці умови змушують датчики в IoT видавати незвичайні та помилкові показання, які часто називають викидами. Значна частина поточних досліджень була проведена для розробки моделей викидів датчиків і виявлення несправностей виключно для бездротових сенсорних мереж (WSN), а адекватних досліджень у контексті IoT досі не проводилося.

Операційна структура бездротової сенсорної мережі значно відрізняється від операційної структури IoT, використання деяких існуючих моделей, розроблених для WSN, не може використовуватися в IoT для виявлення викидів і збоїв. Виявлення несправностей датчиків і викидів є дуже важливим в IoT для виявлення високої ймовірності помилкового зчитування або пошкодження даних, що забезпечує якість даних, зібраних датчиками.

Дані, зібрані датчиками, спочатку попередньо обробляються для перетворення в інформацію, а коли моделі штучного інтелекту (AI), машинного навчання (ML) далі використовуються IoT, інформація далі обробляється в програмах і процесах. Будь-які несправні, помилкові, пошкоджені показання датчиків пошкоджують навчені моделі, що, таким чином, створює ненормальні процеси або викиди, які значно відрізняються від нормальних поведінкових процесів системи.

У даній роботі представляємо вичерпний огляд виявлення несправностей датчиків, аномалій, викидів в Інтернеті речей і проблем. Обговорюються вичерпні рекомендації щодо вибору адекватної моделі виявлення викидів для датчиків у контексті IoT для різних програм.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи виявлення несправності елементів цифрових пристроїв.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем виявлення несправності елементів цифрових пристроїв.
– Дослідження системи виявлення несправності елементів цифрових пристроїв.
– Програмна реалізація системи виявлення несправності елементів цифрових пристроїв.

Об'єктом дослідження є процес виявлення несправності елементів цифрових пристроїв.

Предметом дослідження є методи виявлення несправності елементів цифрових пристроїв.

Методи дослідження базуються на методах системотехніки, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу

Інтернет речей є однією з ключових проривних технологій в епоху Індустрії 4.0 [1]. Існує зростаюча тенденція до використання Інтернету речей (IoT) у наукових та промислових спільнотах [1, 1]. Існує багато визначень, запропонованих для IoT, загалом, IoT можна описати як злиття різних технологій, які надають Інтернет-послуги та програми за допомогою електронних пристроїв, підключених до фізичних речей, з метою збору даних за допомогою різнорідних датчиків., для управління процесами [1].

IoT використовуються в багатьох додатках у різних сферах, від моніторингу навколишнього середовища, охорони здоров'я, сільського господарства та виробничих секторів [5, 6, 7]. Інтернет речей еволюціонував від простого зв'язку та точки зв'язку для отримання даних фізичних об'єктів до комплексних інтелектуальних систем, які здатні збирати величезні обсяги даних і контролювати різні процеси для максимізації прибутку для організацій і окремих осіб.

IoT виявився продуктивною технологією в багатьох секторах, таких як промисловість, громадські та наукові кола. Останнім часом Інтернет речей став головною темою досліджень у більш широкій дослідницькій спільноті. Наразі глобальний ринок Інтернету речей досяг 9,1 мільярда доларів США, і, згідно з [8, 9], зведений річний темп зростання (CAGR) зростатиме на 40% до 2024 року. За оцінками, до року У 2020 році в середньому кожна людина матиме сім комунікаційних пристроїв на основі IoT [10].

В автомобільній промисловості до 2020 року понад 23,6 мільйонів автомобілів матимуть доступ до Інтернету. Згідно зі звітом Verizon [11], очікується, що глобальний ринок Інтернету речей зросте з прогнозованими темпами на 17% і досягне 1,3 трильйона доларів США.

У сфері сільського господарства до 2020 року очікується, що послуги цифрового точного сільського господарства на основі IoT досягнуть 4,5 мільярдів доларів США [12]. Здатність IoT здійснювати моніторинг у режимі реального часу та відносна простота використання відкрили для дослідників цілий новий діапазон використання IoT у багатьох програмах.

Компендіум датчиків на місці, вбудованих у пристрої IoT, є основними компонентами, які збирають цінні необроблені дані. Правильна робота датчиків у пристрої IoT відіграє життєво важливу роль у загальній продуктивності системи та залежних процесів,

додатків [13]. Датчики IoT часто розгортаються в суворих умовах; однак гарантувати правильну роботу датчика та передбачити несправності досить складно. Крім того, датчики в IoT зазвичай є найдешевшими електронними компонентами, які зазвичай схильні до несправностей.

Несправний датчик створює пошкоджені дані або помилкові зчитування або суперечливу інформацію на пристрої IoT [14]. Коли IoT обробляє ці пошкоджені дані датчиків, загальна продуктивність системи IoT знижується, що робить її неточною та ненадійною. Нещодавно зростаюча тенденція автоматизації багатьох процесів, як-от автономне керування транспортними засобами для зменшення кількості аварій, підкреслює важливість правильної роботи датчиків, які працюють у системі.

Оскільки системи IoT працюють безперервно, генеруючи великі обсяги мультимодальних даних, забезпечення точної роботи датчиків в IoT є критично важливим, тому має бути точний процес моніторингу для перевірки поведінки та продуктивності датчиків в IoT. Крім того, цей процес моніторингу датчиків має бути автоматизованим, масштабованим і достатньо гнучким, щоб використовувати його для потокової передачі необроблених даних, створених численними датчиками, вбудованими в пристрій IoT.

Відомо, що цей процес моніторингу, який зазвичай називають виявленням викидів датчика, виявляє будь-яку аномалію чи відхилення в показаннях датчика, і зазвичай це один із ключових процесів, що впливає на якість даних, зібраних датчиком. Останнім часом у дослідницьких спільнотах виявлення викидів викликає великий інтерес [15, 16, 17, 18].

Однак значна частина поточних досліджень щодо виявлення викидів стосується бездротових сенсорних мереж (WSN), а також широко використовується для виявлення шахрайства, порушень безпеки мережі, відстеження цілей, моніторингу навколишнього середовища та здоров'я.

Однак не було проведено адекватних досліджень щодо виявлення викидів датчиків у контексті IoT. Унікальні характеристики IoT у порівнянні з WSN показують, що традиційні методи виявлення викидів не застосовуються безпосередньо до IoT. Зазвичай IoT – це компендіум кількох подібних датчиків, вбудованих як одиниця, здатна виробляти величезні обсяги просторово-часових даних із малою затримкою, на відміну від WSN [19].

Коли є помилкові дані, створені одним типом датчика в межах IoT, спричинені збоєм або несправністю одного датчика з пари, ідентифікація несправного датчика, щоб зробити дані, надані цим датчиком, надлишковими в режимі реального часу, дозволяючи при цьому дані, створені вторинним подібним неушкодженим датчиком, відіграють важливу роль у правильному функціонуванні пристрою IoT.

Методологія дослідження

Методологія дослідження була поділена на три основні етапи. Під час первинного етапу інформацію щодо необхідності виявлення несправностей датчиків, помилок аналізували за допомогою систематичного огляду літератури та консультацій з галуззю та компаніями, які пропонують продукти на основі Інтернету речей на ринку. На другому етапі було проведено критичний аналіз для розуміння етимологічних відмінностей між IoT і WSN, пов'язаних із збоями та помилками датчиків. Після визначення функціональних і робочих відмінностей датчиків між IoT і WSN, на останньому етапі відповідна література була ретельно переглянута, щоб отримати викиди датчиків і методи виявлення несправностей, які підходять для IoT.

Відмінності між IoT та бездротовими сенсорними мережами

У цьому розділі визначено та розглянуто декілька суттєвих відмінностей між IoT та бездротовими сенсорними мережами (WSN). Значна частина досліджень була проведена для розробки моделей викидів датчиків і виявлення несправностей виключно для WSN, однак адекватних досліджень у контексті IoT досі не проводилося. Оскільки бездротові сенсорні мережі та їх операційна структура значно відрізняються від операційної системи IoT, деякі з існуючих моделей, розроблених для WSN, не можна використовувати в IoT для виявлення викидів і збоїв [10, 11, 12].

IoT існує та працює на вищому рівні, ніж WSN. Зобразимо WSN як підмножину IoT, оскільки WSN є технологічною структурою, яка часто використовується в системі IoT для збору даних фізичних явищ у реальних умовах. На відміну від WSN, IoT мають менше проблем, пов'язаних із збоями в мережі, дефіцитом живлення або збоями вузлів тощо. Однак IoT матиме власний набір унікальних проблем (як обговорювалося в розділах вище), де лише кілька типів можна використовувати існуючі методи виявлення викидів і несправностей.

Викиди в контексті IoT

У контексті Інтернету речей викид датчика зазвичай відомий як нерегулярність або розбіжність у поведінці датчика під час процесу каталогізації певних параметрів або подій у порівнянні з його попередньою поведінкою чи показаннями. Не існує стандартного обмеженого визначення викидів датчика.

Джерела викидів датчиків в IoT.

– Внутрішні помилки датчика. Цей тип помилки пов'язаний із неправильними показаннями або вимірюваннями, отриманими від несправного датчика, вбудованого в пристрій IoT. Оскільки датчики є електронними компонентами, вони часто раптово виходять з ладу та припиняють працювати без будь-яких ознак погіршення продуктивності [16, 17, 18]. Цей вид збою датчика передає або відсутність показань, або нульові показання до алгоритму обробки даних у системі IoT [13]. Деяка література визначила цей тип несправності датчика як «бінарну несправність».

– Події датчика: оскільки датчик розгортається для збору даних у сценаріях або подіях реального світу, існує ймовірність безпрецедентної зміни події, спричиненої малоймовірними ситуаціями, які серйозно впливають на датчик, спричиняючи тим самим викиди. Наприклад, система IoT із кількома датчиками, які відстежують рівень температури та вологості на фермі, якщо хробак повзе на один із датчиків, фермер отримуватиме показання про те, наскільки хробак вологий і теплий, ці свідчення будуть неефективними для і перешкоджає роботі всієї системи моніторингу IoT.

– Періодичні помилки датчика: остання категорія несправності датчика – це періодичні помилки, які в основному викликані спорадичними подіями, такими як крадіжка, зловмисна атака та втручання в роботу датчика [19, 10, 11]. Ситуація, коли незакріплений роз'єм у датчику або в іншому місці сенсорного обладнання також може призвести до того, що датчик періодично створює розріджені дані для алгоритмів обробки даних [12].

Несправності датчиків і моделі виявлення викидів для IoT

Виявлення збоїв датчика та ідентифікація викидів у контексті IoT почали привертати значну увагу дослідницького співтовариства. Загалом існує п'ять макрокласів методів автоматичного виявлення викидів датчиків і несправностей, які можна використовувати в контексті IoT.

У цьому розділі обговорюються методи виявлення несправностей датчиків і викидів для IoT на основі таких дисциплін, як методи на основі статистики, методи на основі найближчого сусіда, методи машинного навчання та штучного інтелекту, методи на основі кластеризації та методи на основі класифікації техніки.

Статистичні методи

Методи, засновані на статистиці, були першими алгоритмами, використаними багатьма дослідниками для виявлення несправностей датчика та виявлення викидів. У цій техніці дані від датчиків моделюються за допомогою стохастичного розподілу. Точки даних від датчиків можна визначити як викиди або помилки, коли ймовірність екземпляра даних, згенерованого цією моделлю, дуже низька.

Ця техніка використовує попередні вимірювання датчика для наближення та створення моделі точної поведінки датчика. Однак щоразу, коли реєструється нове вимірювання від того самого датчика, ця точка даних потім порівнюється з моделлю, щоб перевірити, чи нова точка даних статистично несумісна з моделлю. Якщо модель несумісна з показаннями нового датчика, вона позначається як викид або помилкове вимірювання.

Підхід, заснований на статистичному вікні, зазвичай допомагає зменшити кількість

помилкових спрацьовувань помилок і викидів. Фільтр низьких і високих частот є прикладом основного статистичного методу, який класифікує показання датчиків як несправності або аномалії на основі розробки середнього значення попередніх вимірювань і визначення того, наскільки відрізняються нові показання.

Статистичний метод, заснований на просторово-часовій взаємозалежності даних датчиків, запропонований Hida та ін. [10]. Ця техніка здебільшого використовує два статистичні тести для локального виявлення викидів, щоб зробити прості процеси агрегування більш надійними. Статистичні моделі мають відношення до кількісних наборів даних реального значення або, принаймні, є розподілом кількісних даних, який потрібно перетворити на відповідне числове значення для чисельної обробки. Оскільки складність і обсяг даних датчиків зростає (як це зазвичай буває у випадку IoT), ця модель потребує більше часу для обробки, щоб перетворити складні дані.

Методи найближчого сусіда

Техніка на основі найближчого сусіда є широко використовуваною технікою для аналізу точок даних датчиків щодо найближчих сусідів. По суті, метод найближчого сусіда для виявлення несправності датчика та викидів явно спирається на поняття близькості. Техніка найближчого сусіда працює, покладаючись на відстані між вимірюваннями даних датчиків, щоб відрізнити ненормальні показання від правильних. Коефіцієнт локального викиду (LOF) – це відомий алгоритм визначення найближчого сусіда [13], який приписує помилку або оцінку викиду кожному показанню датчика на основі кількості вимірювань навколо його k-найближчих сусідів і кількості вимірювань навколо показання датчика. Показання датчика з вищими балами позначаються як аномалії.

Методи штучної нейронної мережі

Нейронні мережі та нечітка логіка є останніми підходами для виявлення несправностей датчиків і викидів у контексті IoT. Техніка нейронної мережі є логічною моделлю, яка надає комплексну ідею, яка допомагає в процесі прийняття рішень шляхом аналізу всього набору даних датчика [14, 15]. У той час як техніка нечіткої логіки дозволяє перехідні значення (наприклад, правильно/неправильно, так/ні, високий/низький) для розмежування стандартних/правильних показань датчика. В Інтернеті речей підхід нечіткої логіки може бути використаний для покращення прийняття рішень, покращення вибору голови кластеризації, покращення безпеки мережі та агрегації даних, ефективної обробки маршрутизації, протоколів MAC, якості обслуговування та, зрештою, ефективного виявлення несправностей датчиків і викидів.

Кластерні методи

Кластерний аналіз [16] є популярним підходом у спільноті інтелектуального аналізу даних, який групує пов’язані екземпляри даних у кластери подібної поведінки. Шляхом поділу даних на кластери подібних точок даних від датчиків, у яких кожен кластер даних містить точки даних, які схожі одна на одну та відрізняються від точок даних в інших групах кластерів. Цей підхід є підмножиною методів близькості. Початкові показання датчиків спочатку використовуються для створення кластерів, а потім нові вимірювання датчиків, призначені для невеликих і віддалених кластерів даних, або вимірювання датчиків, які знаходяться дуже далеко від центроїда основного кластера, позначаються як ненормальні показання.

Методи, засновані на класифікації

Методи, засновані на класифікації, є важливими точними методами інтелектуального аналізу даних і машинного навчання. Метою методів класифікації є ідентифікація моделі класифікації (названої класифікатором) за допомогою набору визначених точок даних датчиків (точки навчання), а потім класифікація незрозумілих екземплярів даних в одну з вивчених груп (нормальних/викидних).

Цей тип техніки потребує постійного оновлення для адаптації нових даних датчиків, які належать до нормального класу. У випадку IoT ця техніка класифікації адекватно підходить для виявлення несправностей і викидів, оскільки ця техніка прагне працювати за

загального припущення, що класифікатор можна дізнатися з наданої просторової функції для ідентифікації нормальних і викидних класів [17].

Щоб створити цю техніку, її необхідно розділити на два етапи: навчання та експеримент [18]. На етапі навчання методика спрямована на вивчення класифікатора з використанням доступних помічених навчальних даних, після чого слідує фаза експерименту, яка класифікує тестовий екземпляр як звичайний, викид або несправність датчика [19, 10].

Порівняння методів виявлення помилок і викидів для IoT

Незважаючи на те, що деякі дослідники нещодавно спробували досягти високої точності для досягнення високої точності, описані вище методи виявлення несправності сенсора та викидів для IoT, проте кожен із методів має Переваги та недоліки, як обговорюється нижче.

Статистичні методи

Переваги:

– Може ефективно ідентифікувати будь-які несправності датчиків і викиди в IoT після отримання правильної моделі розподілу ймовірностей.

– Несправності датчика та викиди можна виявити за допомогою часової кореляції. Будь-яка непередбачена зміна в розподілі даних негайно зменшує часові кореляції, тим самим виявляючи викиди.

Недоліки:

– Оскільки IoT часто використовуються в умовах реального життя, де часто немає попередніх знань про розподіл даних датчиків, параметричний статистичний підхід не є вигідним.

– Непараметричні статистичні моделі не підходять для IoT з великим об’ємом даних, які працюють у режимі реального часу.

– Часто є високі обчислювальні витрати на керування отриманими багатовимірними даними.

Методи найближчого сусіда

Переваги:

– Дуже просто застосувати до різних типів даних, створених різними датчиками в системі IoT.

– Можуть бути залишені без нагляду та в першу чергу потрібні для визначення відповідної міри відстані для наведених даних.

Недоліки:

– При використанні складних багатовимірних даних, створених IoT, вартість обчислень різко зростає.

– Масштабованість цих типів моделей викликає занепокоєння, особливо в контексті IoT.

– Часто дає високий коефіцієнт хибнонегативних результатів для виявлення несправностей датчика та викидів.

Методи машинного навчання

Переваги:

– Може використовуватися, коли датчики створюють погані, шумні та фрагментарні дані, оскільки властива поведінка цієї моделі узагальнює отримані точки даних.

– Існує обмежена потреба, іноді немає потреби, перенавчати модель, коли додаються нові дані датчика.

Недоліки:

– Модель потребує тонкого налаштування та моделювання, перш ніж її можна буде запустити в реальних умовах.

– Оскільки ця модель часто базується на правилах, якщо кількість змінних даних датчиків збільшується, це також експоненціально збільшуватиме кількість правил.

Кластерні техніки

Переваги:

– Після того, як кластери та нові точки даних буде вставлено в систему та перевірено на наявність помилок датчиків і викидів, модель можна легко адаптувати до інкрементної форми.

– Нагляд не потрібен.

– Дуже добре підходить для виявлення аномалії датчика на основі тимчасових даних IoT.

Недоліки:

– Це дуже дорого обчислювально під час роботи з багатовимірними даними датчиків для виявлення несправностей.

– Через високу обчислювальну вартість моделей він непридатний для датчиків із недостатнім ресурсом.

– Не вдається впоратися з будь-якими змінами в даних IoT.

Методи класифікації

Переваги:

– Ця модель не залежить ні від статистичної моделі, ні від оцінених параметрів даних.

– Забезпечує оптимальну, а іноді й максимальну ідентифікацію несправностей датчика та викидів.

– Може використовуватися на багатовимірних даних для виявлення викидів датчиків і несправностей.

Недоліки:

– Ця модель є обчислювально складною порівняно з кластеризацією та статистичними методами.

– Модель має навчитися отримувати нові точки даних.

Стратегії виявлення та ідентифікації несправності датчика

Автоматичне виявлення несправності датчика та автоматичну ідентифікацію можна виконати за допомогою трьох стратегій: стратегії мережевого рівня, однорідної стратегії та гетерогенної стратегії.

Стратегія мережевого рівня

Використовуючи керування на мережевому рівні та моніторинг мережевих пакетів, цей підхід виявляє будь-які збої датчиків [11, 12]. Датчики в системах IoT ефективно контролюють один одного, щоб виявити будь-які проблемні датчики. Цей підхід переважно використовує моделі Маркова для характеристики нормальної та ненормальної поведінки датчиків [13, 14].

Однорідна стратегія

Гомогенна стратегія використовує кілька датчиків одного типу для ідентифікації та виявлення будь-якого датчика в системах IoT, який має тенденцію демонструвати ненормальну поведінку [15]. Розміщуючи датчики того самого типу, які генерують подібні значення, просторово близько один до одного, цей підхід виявляє будь-яку некорельовану поведінку, таким чином дозволяючи виявити будь-який несправний датчик. Цей тип датчиків в основному використовує модель часових рядів з авторегресійною інтегрованою ковзною середньою (ARIMA) для порівняння прогнозованих вимірювань датчиків із звітними вимірюваннями датчиків [15, 16].

Гетерогенна стратегія

Гетерогенна стратегія має тенденцію поєднувати різні типи точок даних від датчика для виявлення несправності датчика [17, 18]. Стратегія стала популярною останнім часом із розвитком систем IoT із вбудованими датчиками різних типів. Ця стратегія виявляє збій датчика шляхом класифікації виходів датчиків, а потім навчання класифікатора ідентифікувати той самий набір точок даних на основі різних підмножин датчиків в одній системі IoT [19].

На рисунку 1 зображена структурна схема розробленої, в результаті виконання магістерського проектування системи.

Як видно з рисунка структурно система складається з наступних головних блоків:

– інтерфейс користувача;
– блок моделювання;
– блок діагностики;
– бази даних.

Розглянемо ці структурні блоки більш детальніше.

Найбільш великим структурним блоком є блок інтерфейсу користувача. Цей блок є дуже важливим, тому, що багато в чому, від інтерфейсу користувача залежить наскільки легко можна буде навчати користувача діагностиці цифрових схем.

Інтерфейс користувача складається з наступних структурних підблоків:

– меню користувача;
– панель інструментів;
– навігатор;
– графічне зображення елементів та мікросхем;
– візуалізація цифрових схем;
– візуалізація процесу діагностики, показів осцилографа та індикаторів.

Розглянемо блок діагностики.

Структурно він складається з наступних підблоків:

– блок формування та запуску тестуючи послідовностей;
– блок осцилографів, для зняття осцилограм з виходів та входів цифрової схеми;
– блок виявлення помилок та несправностей.

Рисунок 1 – Структурна схема системи

Перейдемо до розгляду блоку моделювання. Він складається з наступних структурних підблоків: побудова цифрових схем, де можно з наявних компонентів побудувати цифрову схему; редагування цифрових схем – для внесення змін у побудовану цифрову схему;

керування цифровою схемою, для відслідковування того, як проходять сигнали по цифровій схемі.

Розглянемо блок роботи з базою даних. Він складається з наступних підблоків: база даних елементів; база даних мікросхем.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів виявлення несправності елементів цифрових пристроїв. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем виявлення несправності елементів цифрових пристроїв; Досліджена система виявлення несправності елементів цифрових пристроїв; На основі отриманих результатів досліджень створена програмна реалізація системи виявлення несправності елементів цифрових пристроїв; Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання виявлення несправності елементів цифрових пристроїв. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Kovalenko A.S. Information model and its element for displaying information on technical condition of objects of integrated information system / A.S. Kovalenko, A.A. Smirnov, A.V. Kovalenko, A.P. Dorensky // International Journal of Computational Engineering Research (IJCER). – India: Delhi, 1016. – Volume 6, Issue 1. – P. 21-27.
2. Кожанова А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. Кожанова, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 1013. – Вип. 6(113). – С. 255-257.
3. Коваленко А.С. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко ., А.А. Смірнов, А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 1014. – Вип. 4(120). – С. 161-164.
4. Коваленко А.С. Підсистема технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А.Смірнов, О.В. Коваленко // Системи озброєння і військова техніка.– Х.: ХУПС, 1014. – № 1(37). – С. 126-129.
5. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 1014. – № 2(38). – С. 106-108.
6. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 1014. – № 2(15). – С.154-157.
7. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 1014. – № 2(15). – С.154-157.
8. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 1014. – Вип. 27. – С. 245-251.
9. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 1014. – № 4(40). – С. 85-88.
10. Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 1015. – Вип. 1(126). – С. 75-79.
11. Коваленко А.С. Обґрунтування набору даних для оцінки технічного стану інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 1015. – Вип. 1(42). – С.39-41.
12. Коваленко А.С. Експертна система технічного діагностування інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 1015. – № 1(41). – С. 106-111.
13. Коваленко А.С. Удосконалення методу технічного обслуговування об’єктів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко, О.П. Доренський // Системи озброєння і військова техніка. – Х.: ХУПС, 1016. – № 2(46). – С. 109-114.
14. Кожанова А.С. Система технічної діагностики інтегрованих інформаційних систем – обґрунтування необхідності створення, визначення понятійного апарату та напрямів досліджень / А.С. Кожанова, О.А. Смірнов,

М.П. Савченко, Д.М. Ізосімов, В.В. Мороз // Створення та модернізація озброєння і військової техніки в сучасних умовах: Тринадцята наук.-техн. конф., 5-6 вер. 2013 р., м. Феодосія: тези доп. – Феодосія: ДНВЦ, 1013. – С. 187-188.
15. Кожанова А.С. Визначення основних напрямків досліджень щодо створення системи технічної діагностики інтегрованих інформаційних систем / А.С. Кожанова, О.А. Смірнов, А.В. Челпанов // Проблемні питання розвитку озброєння та військової техніки Збройних Сил України: IV наук.-техн. конф., 16-20 груд. 2013 р., м. Київ: зб. тез. – Київ: ЦНДІ ОВТ ЗСУ, 1013. – С. 293.
16. Коваленко А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформатика та системні науки : V Всеукр. наук.-практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 1014. – С. 292-294.
17. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 1014. – С. 241.
18. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп'ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 18-31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 1014. – С. 190-193.
19. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 1015. – С. 15.
20. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 1015. – С. 5.

УДК 004

М.Жупило, магістр гр. КІ-21МЗ,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ ДАНИХ ХМАРНИХ СЕРВІСІВ

У статті розроблено програмне забезпечення, яке призначено для системи забезпечення конфіденційності даних хмарних сервісів. Метою розробки є дослідження та програмна реалізація системи забезпечення конфіденційності даних хмарних сервісів. Об'єктом дослідження є процес забезпечення конфіденційності даних хмарних сервісів. Предметом дослідження є методи забезпечення конфіденційності даних хмарних сервісів. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи забезпечення конфіденційності даних хмарних сервісів. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, конфіденційність даних, хмарні сервіси

Постановка проблеми. Незалежно від того, чи виконуєте ви робочі навантаження в загальнодоступній хмарі, приватній хмарі, гібридній інфраструктурі чи мультихмарі з декількома хмарними провайдерами, вам потрібно дотримуватися правил обробки даних і гарантувати безпеку своїх даних.

Недотримання правил щодо даних і наступне порушення може призвести до грошових втрат і шкоди авторитету бренду. Щоб забезпечити захист даних у хмарі, ви можете застосувати різні методи, такі як шифрування, контроль доступу, захист кінцевих точок і моніторинг.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи забезпечення конфіденційності даних хмарних сервісів.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи забезпечення конфіденційності даних хмарних сервісів.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем забезпечення конфіденційності даних хмарних сервісів.
– Дослідження системи забезпечення конфіденційності даних хмарних сервісів.
– Програмна реалізація системи забезпечення конфіденційності даних хмарних сервісів.

Об'єктом дослідження є процес забезпечення конфіденційності даних хмарних сервісів.

Предметом дослідження є методи забезпечення конфіденційності даних хмарних сервісів.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Захист даних означає стратегічні та процедурні кроки, вжиті для захисту конфіденційності, доступності та цілісності конфіденційних даних, і часто взаємозамінно використовується з терміном «безпека даних». Ці захисні заходи, критичні для організацій, які збирають, обробляють або зберігають конфіденційні дані, спрямовані на запобігання пошкодженню, втраті чи пошкодженню даних. В епоху, коли генерація та зберігання даних зростає безпрецедентною швидкістю, важливість надійної стратегії захисту даних є першорядною. Основна мета захисту даних полягає не лише в захисті конфіденційної інформації, а й у забезпеченні її доступності та надійності, таким чином зберігаючи довіру та відповідність операціям, орієнтованим на дані.

Принципи захисту даних

Ось ключові аспекти керування даними, пов'язані із захистом даних:

– Доступність даних.
– Управління життєвим циклом даних.
– Управління життєвим циклом інформації.

Конфіденційність даних

Конфіденційність даних – це вказівка щодо того, як слід збирати та обробляти дані, виходячи з їхньої конфіденційності та важливості. Конфіденційність даних зазвичай застосовується до особистої інформації про здоров'я (PHI) та інформації, що дозволяє ідентифікувати особу (PII). Це включає фінансову інформацію, медичні записи, номери соціального страхування або ідентифікаційні номери, імена, дати народження та контактну інформацію.

Проблеми щодо конфіденційності даних стосуються всієї конфіденційної інформації, яку обробляють організації, включно з інформацією про клієнтів, акціонерів і співробітників. Часто ця інформація відіграє життєво важливу роль у бізнес-операціях, розвитку та фінансах.

Виявлення даних

Перш ніж захистити свої дані, вам потрібно знати, що у вас є та де вони розташовані. Цей процес, відомий як виявлення даних, має вирішальне значення для виявлення конфіденційної інформації та визначення найкращих способів її захисту.

Інвентаризація та класифікація

Щоб почати процес виявлення даних, ви повинні спочатку провести інвентаризацію всіх даних, які є у вашій організації. Це передбачає ідентифікацію різних типів даних, які ви зберігаєте, наприклад інформацію про клієнтів, записи про співробітників, інтелектуальну власність тощо. Отримавши вичерпний список, ви зможете класифікувати кожен тип даних на основі його чутливості та важливості.

Відображення даних

Відображення даних – це наступний крок у виявленні даних, який включає визначення розташування ваших даних і те, як вони проходять у вашій організації. Це допоможе вам зрозуміти взаємозв'язки між різними наборами даних і системами, дозволяючи вам приймати обґрунтовані рішення щодо захисту даних.

Інструменти автоматизованого виявлення

Для подальшого спрощення процесу виявлення даних багато організацій тепер використовують автоматизовані інструменти, які можуть швидко сканувати та ідентифікувати конфіденційні дані. Ці інструменти можуть допомогти вам відстежувати інвентаризацію даних і гарантувати, що ви завжди будете в курсі будь-яких змін або доповнень.

Запобігання втраті даних (DLP)

Запобігання втраті даних (DLP) є критично важливим компонентом захисту даних, призначеним для запобігання несанкціонованому доступу, витоку або крадіжці конфіденційної інформації. Технології DLP складаються з різних інструментів і процесів, які допомагають організаціям контролювати свої дані.

Політики DLP

Створення та впровадження політик DLP є важливим першим кроком у захисті ваших даних. Ці політики окреслюють правила та процедури обробки конфіденційної інформації та мають бути адаптовані до конкретних потреб вашої організації.

Моніторинг і сповіщення

Технології DLP часто включають системи моніторингу та оповіщення, які можуть виявляти потенційні порушення даних або інші інциденти безпеки. Ці системи можуть відстежувати дії користувачів, позначаючи будь-яку підозрілу поведінку або спроби отримати доступ до конфіденційних даних.

Санація

У разі потенційного порушення даних або інциденту безпеки технології DLP також пропонують варіанти виправлення. Це може включати блокування передачі конфіденційних даних, розміщення уражених файлів на карантин або автоматичне скасування доступу до зламаних облікових записів.

Зберігання з вбудованим захистом даних

Вибір правильного рішення для зберігання є важливим для забезпечення безпеки ваших даних. Сучасні технології зберігання тепер оснащені вбудованими функціями захисту даних, які пропонують додаткові рівні безпеки.

Надмірність

Одним із основних способів захисту даних у технологіях зберігання є резервування. Створивши кілька копій своїх даних і зберігаючи їх на окремих дисках або в окремих місцях, ви можете мінімізувати ризик втрати даних через збій обладнання чи інші проблеми.

Виправлення помилок

Вбудована корекція помилок є ще однією особливістю багатьох сучасних систем зберігання. Ця технологія може автоматично виявляти та виправляти пошкодження даних, забезпечуючи цілісність вашої інформації.

Резервне копіювання

Резервне копіювання ваших даних є фундаментальним аспектом захисту даних. Регулярне резервне копіювання гарантує швидке відновлення інформації у разі втрати або пошкодження даних.

Локальні та зовнішні резервні копії

Дуже важливо підтримувати як локальні, так і зовнішні резервні копії ваших даних. Локальні резервні копії забезпечують швидкий доступ до вашої інформації, тоді як зовнішні резервні копії пропонують додатковий захист від катастроф, таких як пожежі чи повені.

Інкрементні та повні резервні копії

Окрім вибору правильного місця резервного копіювання, вам також слід враховувати

тип резервного копіювання, який ви виконуєте. Інкрементне резервне копіювання зберігає лише зміни, внесені з часу останнього резервного копіювання, а повне резервне копіювання створює повну копію ваших даних. Поєднання обох типів може допомогти знайти правильний баланс між простором для зберігання та часом відновлення.

Планування резервного копіювання

Щоб гарантувати, що ваші резервні копії завжди актуальні, важливо встановити регулярний розклад резервного копіювання. Це може включати щоденне, щотижневе або навіть щомісячне резервне копіювання, залежно від потреб вашої організації та конфіденційності ваших даних.

Моментальні знімки

Знімки пропонують додатковий рівень захисту для ваших даних, створюючи копії ваших систем і файлів на певний момент часу. Ці знімки можна використовувати для швидкого відновлення ваших даних у разі інциденту безпеки.

Миттєве відновлення

Однією з головних переваг знімків є їх здатність сприяти миттєвому відновленню. Якщо вашу систему зламано, ви можете швидко повернутися до попереднього знімка, мінімізуючи час простою та втрату даних.

Керування версіями

Знімки також забезпечують форму керування версіями, що дозволяє підтримувати кілька версій ваших даних і систем. Це може бути особливо корисним для відстеження змін і визначення причини інциденту безпеки.

Ефективність зберігання

Завдяки своїй інкрементній природі знімки можуть бути більш ефективними для зберігання, ніж традиційні резервні копії. Це може допомогти вам заощадити місце, зберігаючи комплексну стратегію захисту даних.

Тиражування

Реплікація передбачає створення точної копії ваших даних і збереження її в окремому місці. Це може забезпечити додатковий захист від втрати даних і забезпечити доступність вашої інформації.

Відмовостійкість і відмова

У разі системного збою або іншого збою реплікація дозволяє швидко переключитися на репліковані дані (відмова), забезпечуючи мінімальний час простою. Після того, як проблему буде вирішено, ви зможете повернутися до вихідних даних (відновлення).

Балансування навантаження

Реплікація також може допомогти з балансуванням навантаження, дозволяючи вам розподіляти робоче навантаження між кількома системами або розташуваннями. Це може покращити продуктивність і запобігти перевантаженню системи.

Географічна надмірність

Тиражуючи свої дані в географічно різних місцях, ви можете захистити свою інформацію від регіональних катастроф і зберегти доступ до своїх даних у разі локального збою.

Брандмауери

Брандмауери відіграють вирішальну роль у захисті даних, діючи як бар'єр між внутрішніми системами та зовнішнім світом. Вони можуть допомогти запобігти несанкціонованому доступу та захистити ваші дані від різних загроз.

Виявлення та запобігання вторгненням

Багато сучасних брандмауерів включають функції виявлення та запобігання вторгненням, які можуть ідентифікувати та блокувати потенційні загрози до того, як вони досягнуть ваших систем.

Контроль додатків

Брандмауери також можуть забезпечити контроль програм, дозволяючи обмежувати або дозволяти певним програмам доступ до ваших даних. Це може допомогти запобігти

несанкціонованому доступу та зберегти цілісність вашої інформації.

Моніторинг руху

Нарешті, брандмауери пропонують можливості моніторингу трафіку, дозволяючи відстежувати та аналізувати потік даних у вашу організацію та з неї. Це може допомогти вам виявити потенційні інциденти безпеки та відповідним чином реагувати.

Автентифікація та авторизація

Автентифікація та авторизація є важливими компонентами захисту даних, які гарантують, що лише авторизовані особи можуть отримати доступ до ваших даних. Ці процеси передбачають перевірку ідентичності користувачів і надання їм належного рівня доступу.

Багатофакторна автентифікація

Багатофакторна автентифікація (MFA) додає додатковий рівень безпеки, вимагаючи від користувачів надання двох або більше форм ідентифікації для доступу до ваших даних. Це може включати те, що вони знають (наприклад, пароль), те, що вони мають (наприклад, маркер безпеки), або те, чим вони є (наприклад, відбиток пальця).

Керування ідентифікацією та доступом

Системи керування ідентифікацією та доступом (IAM) призначені для керування ідентифікацією користувачів і правами доступу у вашій організації. Завдяки централізації процесів автентифікації та авторизації IAM може допомогти оптимізувати захист даних і підвищити безпеку.

Шифрування

Шифрування – це процес перетворення даних у код, який можуть прочитати лише авторизовані сторони. Ця технологія є критично важливим компонентом захисту даних, оскільки може допомогти запобігти крадіжці даних або несанкціонованому доступу.

Симетричне шифрування

Симетричне шифрування передбачає використання одного ключа для шифрування та дешифрування даних. Цей метод часто швидший за інші методи шифрування, але вимагає, щоб обидві сторони мали доступ до одного ключа, що є менш безпечним.

Асиметричне шифрування

Асиметричне шифрування, також відоме як шифрування з відкритим ключем, використовує два ключі: один для шифрування даних, а інший – для їх дешифрування. Цей метод повільніший за симетричне шифрування, але забезпечує більший захист, оскільки закритий ключ залишається секретним.

Наскрізне шифрування

Наскрізне шифрування – це метод шифрування, який гарантує, що дані залишаються захищеними з моменту їх надсилання до отримання одержувачем. Ця технологія зазвичай використовується в програмах для обміну повідомленнями та інших комунікаційних платформах.

Захист кінцевої точки

Кінцеві точки, такі як ноутбуки, смартфони та інші мобільні пристрої, часто є вразливими цілями для кібератак. Технології захисту кінцевих точок призначені для захисту цих пристроїв і даних, які вони містять.

Антивірус і захист від шкідливих програм

Антивірусне програмне забезпечення та програмне забезпечення для захисту від зловмисного програмного забезпечення є важливими компонентами захисту кінцевих точок, призначеними для виявлення та видалення шкідливого програмного забезпечення з ваших пристроїв.

Управління пристроєм

Захист кінцевих точок також може включати керування пристроєм, що дозволяє відстежувати та контролювати кінцеві точки з центрального розташування. Це може включати моніторинг активності пристрою, обмеження доступу до певних програм і дистанційне стирання пристроїв у разі крадіжки чи втрати.

Керування виправленнями

Керування виправленнями – це процес підтримки ваших пристроїв в актуальному стані за допомогою останніх виправлень безпеки та оновлень програмного забезпечення. Це може допомогти усунути вразливості та запобігти кібератакам з використанням відомих слабких місць.

Стирання даних

Стирання даних передбачає безпечне й остаточне видалення даних із ваших систем. Цей процес має вирішальне значення для того, щоб конфіденційна інформація не потрапила в чужі руки.

Безпечні методи видалення

Методи безпечного видалення даних передбачають перезапис існуючих даних новими, що унеможливлює відновлення вихідної інформації. Ці методи можуть включати багаторазове перезаписування даних, розмагнічування або фізичне знищення носія даних.

Політика знищення даних

Встановлення політики знищення даних має важливе значення для забезпечення належного видалення конфіденційної інформації, коли вона більше не потрібна. У цих політиках мають бути описані процедури видалення даних і типи даних, які потребують безпечного видалення.

Сертифікація та аудит

Нарешті, сертифікація та аудит можуть допомогти переконатися, що ваші процеси видалення даних є ефективними та відповідають відповідним нормам. Отримавши сертифікацію та проходячи регулярні аудити, ви можете продемонструвати свою відданість захисту даних і гарантувати, що ваші процедури залишаються актуальними.

Аварійного відновлення

Аварійне відновлення передбачає підготовку та реагування на несподівані події, які можуть загрожувати доступності або цілісності ваших даних. Цей процес має важливе значення для забезпечення безперервності роботи та може допомогти мінімізувати вплив катастроф.

Аналіз впливу на бізнес

Перш ніж ви зможете розробити план аварійного відновлення, ви повинні спочатку провести аналіз впливу на бізнес. Цей процес передбачає визначення критичних функцій і систем у вашій організації та визначення потенційного впливу збоїв.

Планування аварійного відновлення

Провівши аналіз впливу на бізнес, ви можете розробити план аварійного відновлення. Цей план має окреслювати процедури реагування на катастрофи та відновлення систем і даних.

Тестування та технічне обслуговування

Щоб забезпечити ефективність вашого плану аварійного відновлення, важливо регулярно тестувати та підтримувати свої процедури. Це може включати проведення настільних навчань або повномасштабного моделювання, а також оновлення вашого плану в міру появи нових технологій або загроз.

Найважливіші найкращі методи забезпечення конфіденційності даних

Створення політики щодо конфіденційності даних може бути складним завданням, але це не неможливо. Наведені нижче практичні поради допоможуть вам переконатися, що створені вами політики є максимально ефективними.

Інвентаризуйте свої дані

Частиною забезпечення конфіденційності даних є розуміння того, які дані у вас є, як вони обробляються та де вони зберігаються. У вашій політиці має бути визначено, як ця інформація збирається та обробляється. Наприклад, вам потрібно визначити, як часто дані скануються та як вони класифікуються після того, як вони знаходяться.

У вашій політиці конфіденційності має бути чітко вказано, які засоби захисту необхідні для різних рівнів конфіденційності ваших даних. Політики також повинні

включати процеси перевірки захисту, щоб переконатися, що рішення застосовуються правильно.

Мінімізуйте збір даних

Переконайтеся, що ваша політика передбачає збір лише необхідних даних. Якщо ви збираєте більше, ніж вам потрібно, ви збільшуєте свою відповідальність і створюєте надмірний тягар для ваших команд безпеки. Мінімізація збирання даних може також допомогти вам заощадити на пропускній здатності та сховищі.

Одним із способів досягнення цього є використання фреймворків «перевіряти, а не зберігати». Ці системи використовують дані третіх сторін для перевірки користувачів і усувають необхідність зберігати або передавати дані користувачів у ваші системи.

Будьте відкритими зі своїми користувачами

Багато користувачів знають про проблеми конфіденційності та, ймовірно, оцінять прозорість, коли мова йде про те, як ви використовуєте та зберігаєте дані. Відображаючи це, GDPR зробив згоду користувача ключовим аспектом використання та збору даних.

Ви можете бути впевнені, що включите користувачів та їхню згоду у ваші процеси, розробивши проблеми конфіденційності у своїх інтерфейсах. Наприклад, наявність чітких сповіщень для користувачів із зазначенням часу збору даних і чому. Ви також повинні включити параметри для користувачів, щоб змінити або відмовитися від збору даних.

Тенденції захисту даних

Портативність даних і суверенітет даних

Портативність даних є важливою вимогою для багатьох сучасних IT-організацій. Це означає можливість переміщення даних між різними середовищами та програмними додатками. Дуже часто портативність даних означає можливість переміщення даних між локальними центрами обробки даних і загальнодоступною хмарою, а також між різними хмарними провайдерами.

Переносимість даних також має юридичні наслідки: коли дані зберігаються в різних країнах, вони підпадають під дію різних законів і правил. Це відомо як суверенітет даних.

Традиційно дані не були переносними, і перенесення великих наборів даних в інше середовище вимагало великих зусиль. Міграція хмарних даних також була надзвичайно складною на початку розвитку хмарних обчислень. Розробляються нові технічні методи, які спрощують міграцію та роблять дані більш переносними.

Пов’язаною проблемою є переносимість даних у хмарах. Постачальники хмарних послуг, як правило, мають власні формати даних, шаблони та механізми зберігання. Це ускладнює переміщення даних з однієї хмари в іншу та створює блокування від постачальника. Все частіше організації шукають стандартизовані способи зберігання та керування даними, щоб зробити їх переносними між хмарами.

Захист мобільних даних

Захист мобільних пристроїв стосується заходів, призначених для захисту конфіденційної інформації, що зберігається на ноутбуках, смартфонах, планшетах, переносних та інших портативних пристроях. Основним аспектом безпеки мобільного пристрою є запобігання доступу неавторизованих користувачів до вашої корпоративної мережі. У сучасному IT-середовищі це критичний аспект безпеки мережі.

Існує багато інструментів безпеки мобільних даних, призначених для захисту мобільних пристроїв і даних шляхом виявлення загроз, створення резервних копій і запобігання загрозам на кінцевій точці від досягнення корпоративної мережі. IT-спеціалісти використовують програмне забезпечення для захисту мобільних даних, щоб забезпечити безпечний мобільний доступ до мереж і систем.

Загальні можливості рішень безпеки мобільних даних включають:

– Забезпечення зв’язку через захищені канали.

– Виконання надійної перевірки особи, щоб переконатися, що пристрої не скомпрометовані.

– Обмеження використання програмного забезпечення сторонніх розробників і

перегляду небезпечних веб-сайтів.

– Шифрування даних на пристрої для захисту від компрометації та крадіжки пристрою.

– Виконуйте регулярні аудити кінцевих точок, щоб виявити загрози та проблеми безпеки.

– Моніторинг загроз на пристрої.

– Налаштування захищених шлюзів, які дозволять віддаленим пристроям безпечно підключатися до мережі.

Програми-вимагачі

Програмне забезпечення-вимагач – це зростаюча загроза кібербезпеці, яка є головним пріоритетом безпеки майже для всіх організацій. Програми-вимагачі – це різновид зловмисного програмного забезпечення, яке шифрує дані користувача та вимагає викуп за його розповсюдження. Нові типи програм-вимагачів надсилають дані зловмисникам перед їх шифруванням, що дозволяє зловмисникам вимагати від організації, погрожуючи оприлюднити її конфіденційну інформацію.

Резервне копіювання є ефективним захистом від програм-вимагачів: якщо організація має свіжу копію своїх даних, вона може відновити її та відновити доступ до даних. Однак програми-вимагачі можуть поширюватися мережею протягом тривалого періоду часу, поки файли не шифруються. На цьому етапі програми-вимагачі можуть заразити будь-яку підключену систему, включно з резервними копіями. Коли програми-вимагачі поширюються на резервні копії, це закінчується для стратегій захисту даних, оскільки відновити зашифровані дані стає неможливо.

Існує кілька стратегій для запобігання програмному забезпеченню-вимагачу та, зокрема, запобіганню його розповсюдженню на резервні копії:

– Найпростіша стратегія полягає в тому, щоб використовувати старе правило резервного копіювання 3-2-1, зберігаючи три копії даних на двох носіях, один з яких знаходиться за межами підприємства.

– Постачальники засобів безпеки мають передові технології, які можуть виявляти програми-вимагачі на ранніх стадіях або, у гіршому випадку, блокувати процеси шифрування, коли вони починаються.

– Постачальники сховищ пропонують незмінне сховище, яке гарантує, що дані не можна буде змінити після їх збереження. Дізнайтеся, як захищене сховище Cloudian може допомогти захистити ваші резервні копії від програм-вимагачів.

Управління копіюванням даних (CDM)

Великі організації мають кілька наборів даних, які зберігаються в різних місцях, і багато з них можуть дублювати дані між собою.

Дублікати даних створюють численні проблеми – це збільшує витрати на зберігання, створює невідповідності та проблеми з роботою, а також може призвести до проблем із безпекою та відповідністю. Як правило, не всі копії даних будуть захищені однаково. Немає сенсу захищати набір даних і гарантувати його відповідність, якщо дані дублюються в іншому невідомому місці.

CDM – це тип рішення, яке виявляє дублікати даних і допомагає керувати ними, порівнюючи подібні дані та дозволяючи адміністраторам видаляти невикористані копії.

Аварійне відновлення як послуга

Аварійне відновлення як послуга (DRaaS) – це керована служба, яка надає організації хмарний віддалений сайт для аварійного відновлення.

Традиційно створення вторинного центру обробки даних було надзвичайно складним і пов’язаним із величезними витратами й актуальним лише для великих підприємств. Завдяки DRaaS організація будь-якого розміру може скопіювати свої локальні системи в хмару та легко відновити роботу в разі аварії.

Сервіси DRaaS використовують загальнодоступну хмарну інфраструктуру, що дає змогу зберігати кілька копій інфраструктури та даних у кількох географічних місцях, щоб

підвищити стійкість.

Захист даних і конфіденційність із Cloudian HyperStore

Для захисту даних потрібна потужна технологія зберігання. Пристрої зберігання Cloudian прості в розгортанні та використанні, дозволяють зберігати дані розміром у петабайт і миттєво отримувати до них доступ. Cloudian підтримує високошвидкісне резервне копіювання та відновлення з паралельною передачею даних (запис 18 ТБ на годину з 16 вузлами).

Cloudian забезпечує довговічність і доступність ваших даних. HyperStore може створювати резервні копії та архівувати ваші дані, забезпечуючи доступні версії для відновлення в разі потреби.

У HyperStore зберігання відбувається за брандмауером, ви можете налаштувати географічні межі для доступу до даних і визначити політики для синхронізації даних між пристроями користувачів. HyperStore дає вам можливості хмарного обміну файлами на локальному пристрої та контроль для захисту ваших даних у будь-якому хмарному середовищі.

Захист даних у хмарі

Захист даних у хмарі – це набір практик, спрямованих на захист даних у хмарному середовищі. Ці методи застосовуються до даних незалежно від того, де вони зберігаються або як ними керують, чи то всередині компанії, чи треті сторони. Хмарні методи захисту даних стали ключовими аспектами безпеки даних, оскільки компанії збільшують обсяг даних, що зберігаються в хмарі.

Якщо вас цікавить захист даних, ви можете дізнатися більше в нашому посібнику: Постійний захист даних.

Чому компаніям потрібен захист даних у хмарі

Багато компаній збирають і зберігають значні обсяги інформації, включно з конфіденційними. Більшість цих даних потрапляє в хмару в певний момент, під час збирання або зберігання.

Частково причиною зростання хмарного сховища даних є те, що організації все частіше працюють через веб-портали або використовують пропозиції програмного забезпечення як послуги (SaaS). Обидва вони потребують доступу до хмари. Крім того, багато компаній вирішують зберігати дані в хмарі навіть для внутрішнього використання.

Оскільки компанії використовують хмарні сервіси, захист даних стає складнішим:

– Компанії можуть не знати, де зберігаються всі програми та дані.

– Сторонній хостинг обмежує видимість доступу до даних і обміну ними.

– Спільні обов’язки щодо безпеки можуть бути неправильно зрозумілі або неправильно застосовані.

– Якщо компанії використовують кілька хмарних провайдерів або гібридну інфраструктуру, безпека може бути непослідовною.

– Дані можуть підпадати під дію нормативних актів щодо захисту даних, наприклад Загального регламенту захисту даних ЄС (GDPR), Каліфорнійського закону про конфіденційність споживачів (CCPA) або Закону США про перенесення та підзвітність медичного страхування (HIPAA).

Проблеми захисту даних у хмарі

Під час налаштування захисту даних у хмарі ваша організація, ймовірно, зіткнеться з кількома з наведених нижче проблем.

– Цілісність – системи мають бути розроблені таким чином, щоб гарантувати надання лише авторизованого доступу. Конфігурації також повинні гарантувати, що дозволи на зміну або видалення даних надаються відповідним користувачам.

– Локальність – правила щодо даних застосовуються відповідно до фізичного розташування даних, місця їх збору та використання. У розподіленій системі це може бути важко визначити та контролювати. Системи мають бути спроектовані таким чином, щоб чітко визначати, де постійно знаходяться дані.

– Конфіденційність – дані мають бути захищені відповідно до рівня конфіденційності. Це вимагає належного обмеження дозволів і застосування шифрування для обмеження читабельності. Так само облікові дані адміністратора та ключі шифрування мають бути захищені, щоб гарантувати дотримання цих обмежень.

– Хмарна інфраструктура зберігання даних повністю контролюється постачальником. Це означає, що компанії повинні покладатися на постачальників, щоб забезпечити безпеку фізичної інфраструктури, мереж і центрів обробки даних.

Найкращі методи безпеки хмарних даних

Щоб забезпечити ефективність створених вами засобів захисту, додайте наведені нижче практичні поради.

Оцініть вбудовану безпеку

Будь-який обраний вами хмарний постачальник повинен мати надійні внутрішні засоби контролю та пропонувати надійні інструменти, які допоможуть вам захистити дані. Шукайте постачальників, які пропонують угоди про рівень обслуговування, які гарантують належний захист систем.

Крім того, обов'язково перевірте, які політики мають постачальники, щоб відповідати вимогам. Якщо постачальники не сертифіковані, можливо, ви не зможете відповідати стандартам відповідності.

Використовуйте шифрування на рівні файлу

Більшість хмарних провайдерів пропонують певну міру шифрування як під час передавання, так і під час спокою. Вам слід увімкнути обидва параметри. Вам також слід розглянути можливість додавання додаткового шифрування на рівні файлу. Найпростіший спосіб зробити це – зашифрувати дані перед перенесенням їх у хмарне сховище.

Якщо ви не можете зашифрувати на рівні файлу, подивіться, чи можете ви «розшардувати» свої дані. Шардинг зберігає частини даних або програм у різних місцях. Це може ускладнити зловмисникам повторно зібрати ваші дані, навіть якщо вони отримають до них доступ.

Обмежте доступ за допомогою надійних облікових даних

Ви повинні застосовувати як сувору політику облікових даних, так і суворі дозволи доступу. Суворі дозволи гарантують, що користувачі та програми можуть отримати доступ лише до тих даних, які їм потрібні. Сувора політика облікових даних гарантує, що зловмисники не зможуть зловживати дозволами, наданими цим користувачам і програмам.

Періодично перевіряйте свої дозволи та встановлюйте життєві цикли паролів. Ви хочете переконатися, що всі облікові дані у вашій системі активно використовуються. Ви також хочете переконатися, що паролі досить важко вгадати, і що користувачі не використовують паролі повторно.

Захистіть пристрої кінцевих користувачів

Кінцеві точки є однією з найбільш вразливих частин вашої системи, особливо якщо кінцевими точками керує користувач. Наприклад, смартфони, підключені до вашої мережі в рамках політики використання власного пристрою (BYOD). Ці пристрої можуть бути проблемою, оскільки групи безпеки зазвичай не мають повного контролю над заходами безпеки, такими як оновлення чи шифрування.

Щоб запобігти зловживанню цими пристроями, вам слід запровадити рішення для захисту кінцевих точок. Ці рішення допомагають відстежувати та обмежувати трафік на периметрі вашої мережі, а також можуть допомогти вам обмежити вихід або надходження даних у ваші системи.

Захист даних за допомогою Cloudian HyperStore

Захист даних у хмарі може бути складним завданням, особливо коли мова йде про розподілену та складну інфраструктуру, як-от мультихмарні та гібридні хмари. Якщо ви користуєтеся кількома постачальниками хмарних технологій або кількома хмарними службами, вам доведеться більше працювати, щоб захистити свої дані.

Захист даних стає набагато легшим, коли ви переміщуєте дані локально. Cloudian

надає локальні пристрої зберігання даних, які прості в розгортанні та використанні, дозволяють зберігати дані розміром у петабайти за низькою ціною та миттєво отримувати до них доступ. Cloudian підтримує високошвидкісне резервне копіювання та відновлення з паралельною передачею даних (запис 18 ТБ на годину з 16 вузлами).

Cloudian HyperStore забезпечує довговічність і доступність ваших даних. Ви можете використовувати HyperStore як пристрій для швидкого та надійного зберігання даних. HyperStore може створювати резервні копії та архівувати ваші дані, забезпечуючи доступні версії для відновлення в разі потреби.

У HyperStore зберігання відбувається в межах брандмауера, ви можете налаштувати географічні межі для доступу до даних і визначити політики для синхронізації даних між пристроями користувачів. HyperStore дає вам можливість хмарного обміну файлами та масштабованість на локальному пристрої.

Розробка структурної схеми

На рисунку 1 зображена структурна схема системи забезпечення конфіденційності даних хмарних сервісів.

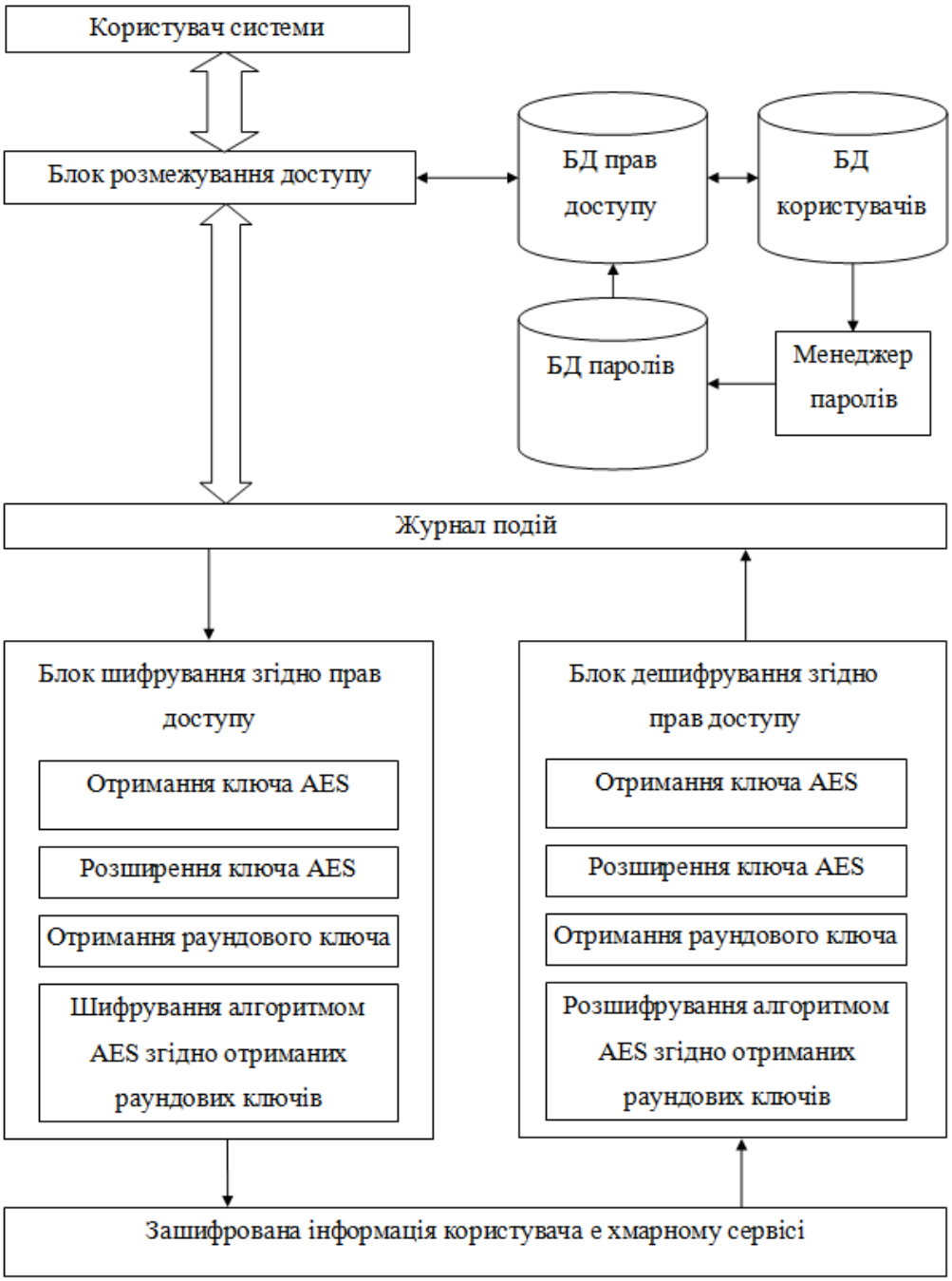


Рисунок 1 – Структурна схема системи

Забезпечення безпеки інформації при зберіганні й обробці більших інформаційних масивів у хмарних сервісах – одна із самих актуальних проблем сучасних інформаційних технологій. Інтенсивний розвиток методів розподіленої обробки даних і різке збільшення обсягів інформації, що накопичується в комп'ютерних системах, привели останнім часом до кардинальної зміни методів довгострокового зберігання даних. Традиційні підходи до організації зберігання великих інформаційних масивів перестали задовольняти зрослим вимогам до ємності носіїв і швидкості доступу до даних. Всі частіше споживач довіряє зберігання своєї власної інформації зовнішнім центрам або мережам зберігання даних (так званий аутсорсинг). Одна з основних сфер застосування мережного зберігання даних – формування банків даних електронних документів, а також електронних архівів і бібліотек. Такі сховища даних можуть бути як публічними, так і обмеженого доступу, залежно від характеру документів, що накопичуються в них.

Нерідко перед приміщенням документів у мережні сховища вони піддаються стиску або іншим спеціальним видам кодування. У зв'язку із цим загострюється необхідність забезпечення керованості, надійності й безпеці зберігання й доступу до електронних документів, а також процедур їхньої передачі між прикладними програмами й пристроями зберігання.

Якщо навіть дані зберігаються локально, виникає інша проблема: адміністратори, що управляють СУБД і персонал так чи інакше звичайно мають права доступу до всієї збереженої інформації. Для захисту від їхніх несанкціонованих дій у деяких випадках доцільне застосування апаратно-програмних засобів шифрування даних перед записом їх на засоби зберігання. Часто шифрувальні модулі вбудовуються, наприклад, у засоби резервного копіювання даних. Однак при зберіганні шифрованих масивів утруднений пошук окремих файлів і оперативний доступ до елементів масиву, необхідним для роботи прикладних програм. Тому що масив зберігається в зашифрованому виді, і серверу, на якому він зберігається (або СУБД), не можуть бути довірені ключі шифрування, користувач (або прикладна програма від його ім'я) змушений завантажувати копії всіх файлів масиву, розшифровувати їх і потім виконувати пошук на локальній машині.

Очевидно, що такий спосіб пошуку дуже неекономічний. У зв'язку із цим вимальовується проблема забезпечення можливості пошуку даних по шифрованим і (або) стислим даним, що може бути конкретизована залежно від застосовуваної в системі моделі шифрування даних.

Для шифрування великих масивів даних, що поміщаються в зовнішні стосовно власника інформації сховища, ефективні лише симетричні схеми шифрування. Можливості їхнього практичного застосування, мабуть, визначаються можливостями організації схеми керування секретними ключами, для яких необхідно забезпечити виконання двох почасти суперечливих вимог: забезпечення високої схоронності ключів (зокрема, за рахунок резервування) і обмеження середовища їхнього поширення тільки тими пристроями, яким довіряє власник інформації.

У зв'язку із цим у деяких випадках більше раціональним виглядає застосування схем відкритого шифрування, що дає можливість невизначеному колу осіб поміщати свої дані в сховище, але доступ до них залишати лише для власників секретного ключа. Така схема може бути корисна, наприклад, для систем електронної пошти або систем планування потоків завдань (workflows), де циркулюють переважно повідомлення невеликої довжини. Для таких схем тим більше необхідні механізми пошуку за шифрованим даними, що операції розшифрування в асиметричних криптосхемах, як відомо, виконуються на кілька порядків повільніше в порівнянні із симетричними. Інша проблема, пов'язана із забезпеченням конфіденційності пошуку в масивах даних, пов'язана з бажанням унеможливити одержання адміністратором СУБД і сторонніми особами відомостей про те, до яких саме записів (або фрагментів) бази даних здійснювався доступ при кожному конкретному запиті. У

закордонній літературі це завдання зветься “Private Information Retrieval” (PIR). Вона особливо актуальна, наприклад, при обробці й зберіганні електронних документів, що містять відомості приватного характеру: фінансові, юридичні, майнові, медичні й інші. Якщо навіть самі поля бази даних зашифровані, характер і частота запитів до них уже можуть дати зловмисникові деяку непряму інформацію, розголошення якої небажано для власника. Ці завдання до визначеної міри аналогічні виникаючої в телекомунікаційних системах завданню маскування інтенсивності трафіка між вузлами, що, як відомо, вирішується шляхом суцільного заповнення каналу псевдовипадковими послідовностями.

Виходячи зі структурної схеми системи зображеної на рисунку 9, система забезпечення конфіденційності даних хмарних сервісів, працює наступним чином. Спершу при вході в систему, користувач звертається до блоку розмежування доступу. Блок розмежування доступу отримує пароль користувача, та звертається до менеджеру паролів, де отримує сеансовий пароль перевірки правильності паролю користувача, та правильності прав доступу користувача, які зберігаються у відповідних зашифрованих базах даних. Розмежування цих баз зроблено з метою підвищення стійкості системи зберігання інформації. Після підтвердження прав доступу, та правильності введеного паролю, користувачеві видається сеансовий ключ AES для роботи з інформацією.

У блоці шифрування згідно прав доступу, з отриманого ключа AES відбувається його розширення, та обирається ключ ітерації, за допомогою яких й відбувається шифрування інформації алгоритмом AES. Процедура дешифрування відбувається аналогічним чином.

Висновки. У статті теоретичне узагальнення й рішення наукового завдання дослідження методів забезпечення конфіденційності даних хмарних сервісів. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем забезпечення конфіденційності даних хмарних сервісів; Досліджена система забезпечення конфіденційності даних хмарних сервісів; На основі отриманих результатів досліджень створена програмна реалізація системи забезпечення конфіденційності даних хмарних сервісів; Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання забезпечення конфіденційності даних хмарних сервісів. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12. (Scopus).
2. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022. (Scopus).
3. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34. (Scopus).
4. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477. (Scopus).
5. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w (Scopus).
6. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
7. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).

8. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
9. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
10. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
11. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
12. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
13. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
14. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
15. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
16. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
17. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
18. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
19. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
20. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).

УДК 004

І.Завірюха, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ SMART HOME З ВИКОРИСТАННЯМ ПРОТОКОЛУ X10

У статті розроблено програмне забезпечення, яке призначено для системи smart home з використанням протоколу X10. Метою розробки є дослідження та програмна реалізація системи smart home з використанням протоколу X10. Об'єктом дослідження є процес smart home з використанням протоколу X10. Предметом дослідження є методи smart home з використанням протоколу X10. Методи дослідження базуються на методах інтернету речей, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи smart home з використанням протоколу X10. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерна інженерія, smart home, X10

Постановка проблеми. У сучасному житлі вже звичайними й навіть обов'язковими стають різні пристрої й пристосування, що забезпечують високу якість життя й комфорт мешканців. І якщо раніше канальні кондиціонери, електроуправляємі водопровідні вентилі, індивідуальне управління кліматом і безліч світлових груп були атрибутами великого

заміського будинку, то зараз ці особливості властиві й міській квартирі. Зрозуміло, при повсякденній експлуатації настільки різноманітного встаткування в користувачів виникає безліч повторюваних завдань, наприклад:

−одночасне вимикання множини світлових груп;

−вимикання великої кількості приладів при відході із квартири й включення їх у потрібний режим при поверненні у квартиру;

−управління кліматом у приміщеннях залежно від часу доби й присутності людей;

−контроль цілісності комунікацій і справності встаткування.

У першу чергу, саме для виконання таких рутинних операцій, причому з великим ступенем надійності й точності, призначені системи домашньої автоматики, тобто системи «Smart home».

Але логічно, що саме все не буде включатися й відключатися у smart home (інтелектуальному домі). Для того, щоб smart home запрацював, по-перше необхідно установити відповідне устаткування, по-друге необхідно з'єднати це устаткування у мережу, й відповідно написати якесь програмне забезпечення, яке повинно керувати устаткуванням.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи smart home з використанням протоколу X10.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи smart home з використанням протоколу X10.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем smart home з використанням протоколу X10.
- Дослідження системи smart home з використанням протоколу X10.
- Програмна реалізація системи smart home з використанням протоколу X10.

Об'єктом дослідження є процес smart home з використанням протоколу X10.

Предметом дослідження є методи smart home з використанням протоколу X10.

Методи дослідження базуються на методах інтернету речей, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Огляд технології X10

Технологія домашньої автоматизації X10 призначена для інтелектуального управління освітленням і побутовими електроприладами з використанням силової електропроводки.

Сигнали управління X10 передаються й приймаються безпосередньо по електричній мережі (Power Line Carrier або PLC). Відсутність додаткових дротів робить інсталяцію систем X10 швидкою й легкою.

Для передачі команд управління в системах X10, крім силової проводки, використовується радіоканал із частотою 433Мгц. Так забезпечується комфорт бездротового способу управління.

Одна з головних переваг технології X10 у порівнянні із традиційною електроінсталяцією полягає в тому, що схема управління може багаторазово мінятися простим переналаштуванням окремих компонентів мережі X10 – контролерів, вимикачів, реле й діммерів. У випадку традиційної електроінсталяції зв'язок «орган управління – об'єкт управління» задаються жорстко раз і назавжди структурою кабельної проводки. Системи X10 легко розширювані. У будь-який момент у діючу інсталяцію можна додати нові компоненти навіть якщо опоряджувальні роботи давно закінчені.

Вартість мінімального комплекту встаткування X10 становить 1-2 тис. грн., що робить технологію доступною й популярною.

Технологія X10 веде свою історію з початку 70-х років минулого століття. У той час інженери з англійської компанії PICO Electronics уперше створили пристрої, що використовували електричну проводку як середовище мережної комунікації. З тих пор

технологія безупинно вдосконалювалася й сьогодні автоматикою X10 оснащені мільйони будинків по усьому світі. Особливе поширення й популярність технологія одержала в країнах Північної Америки.

Управління X10

За допомогою технології X10 вирішуються завдання комфортного управління освітленням, компонентами інженерних систем і побутових електроприладів. Технологія X10 забезпечує різноманітні способи управління домашнім устаткуванням у ручному й автоматичному режимах.

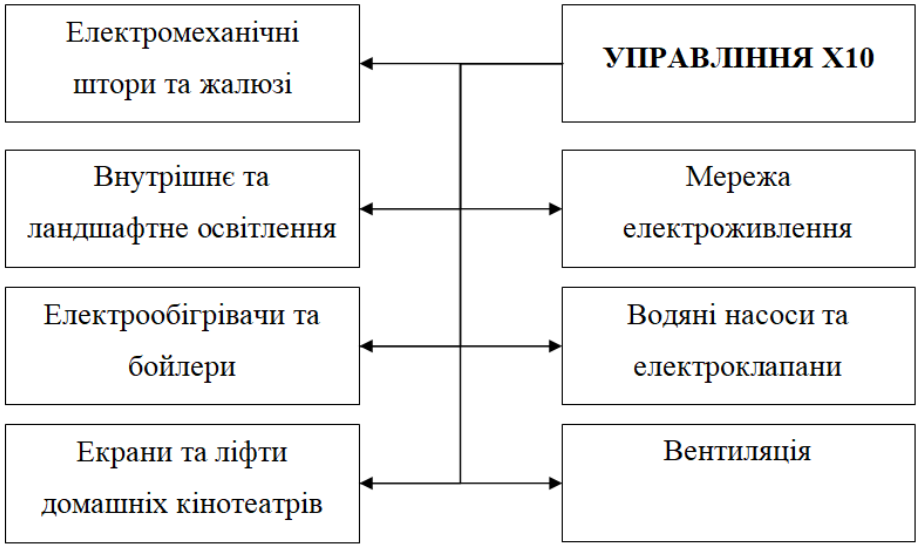


Рисунок 1 – Схема управління технології X10

Додатковий асортименти мультимедійних і охоронних пристроїв розширює функціональні можливості базового комплекту X10 по управлінню освітленням і електроприладами й робить всю лінійку ще більш привабливої для самого широкого кола споживачів.

Функції X10, устаткування X10

Мережа X10 – це сукупність контролерів і виконавчих пристроїв, взаємодіючих один з одним по електричній мережі або радіоканалу. При необхідності додатково використовуються спеціальні системні пристрої, що забезпечують працездатність мережі X10 у цілому. Устаткування X10 може задовольнити найрізноманітнішим вимогам до розміщення.

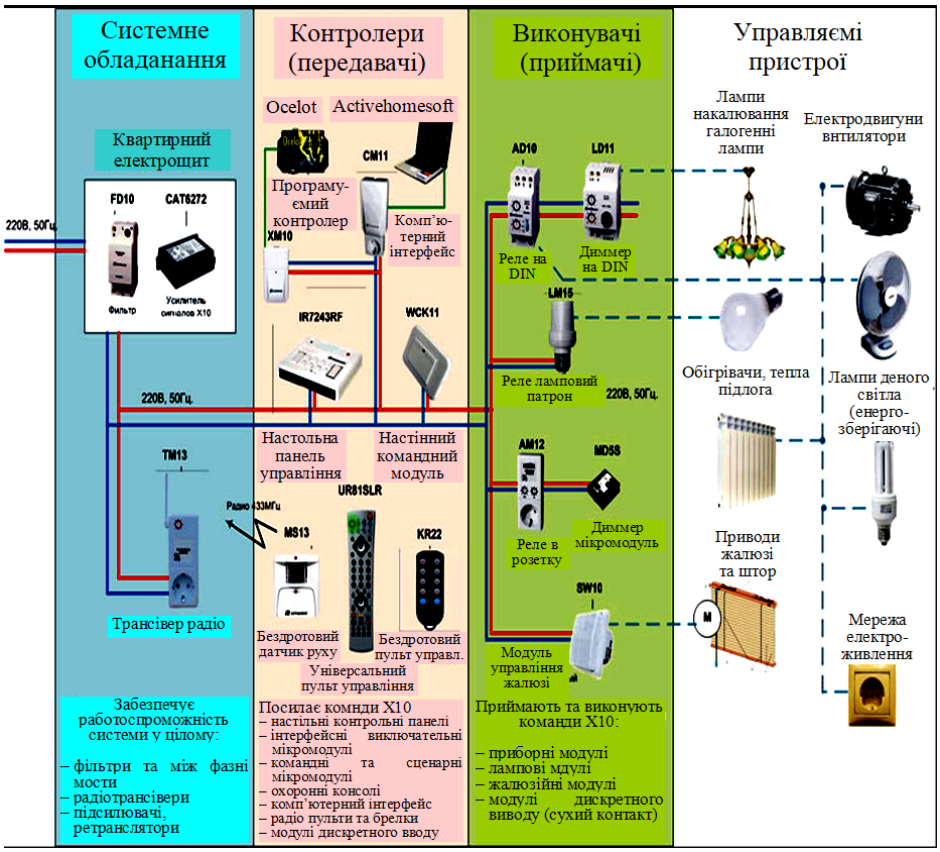


Рисунок 2 – Застосування устаткування X10

Рисунок 3 – Приклади пристроїв X10

Адреси X10

Кожний виконавчий пристрій X10 настроюється на певну адресу, що складається із двох 16-значних полів – Коду будинку (буквені значення A..P) і Коду пристрою (цифрові значення 1..16). Таким чином, усього можливі 256 адресні комбінації. Це максимально можлива кількість керованих груп пристроїв у мережі X10. Виконавчі пристрої X10 настроюються на певну адресу за допомогою двох механічних адресних селекторів-коліщат. Більшість контролерів X10 також вимагають попереднього налаштування адрес.

Рисунок 4 – Виконавчий пристрій X10

Для проходження команд контролер і виконавець повинні бути настроєні на однакові адреси.

Команди X10

У протоколі X10 передбачено шість базових команд:

– Включити (On).
– Виключити (Off).
– Яскравіше (Bright).
– Темніше (Dim).
– Включити все світло (ALL Ligts ON).
– Виключити всі (ALL Units OFF).

Сигнали X10

Технологія X10 використовує цифрове подання сигналів управління. Інформація кодується двійковим кодом і передається по електричній мережі за допомогою високочастотних імпульсів. Кожний переданий імпульс відповідає одному біту інформації c_j значенням "1". Передача чергового імпульсу відбувається в момент часу, коли сіткова напруга приймає нульове значення.

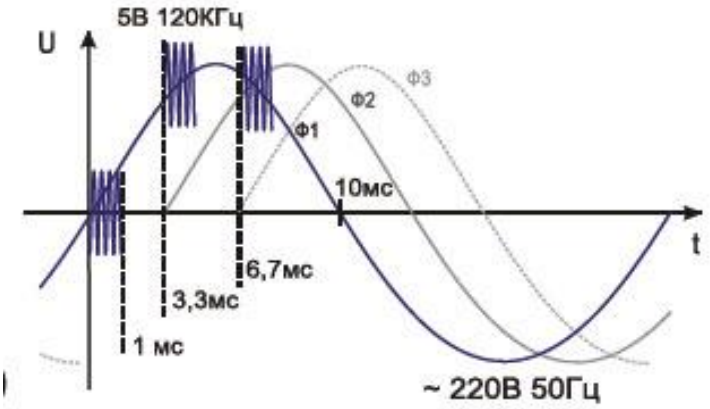


Рисунок 5 – Передача цифрових сигналів

Стандартна команда X10 передається протягом, приблизно, 50 періодів сіткової напруги частоти 50Гц або 1 сек. Більшість переданих по мережі X10 повідомлень містить, принаймні, два інформаційні поля – адреса пристрою, якому ця команда адресована й властиво команду. Підключені до електромережі пристрої X10 приймають передані повідомлення, декодируют поле адреси одержувача й, якщо він збігається з їхньою власною адресою, виконують команду.

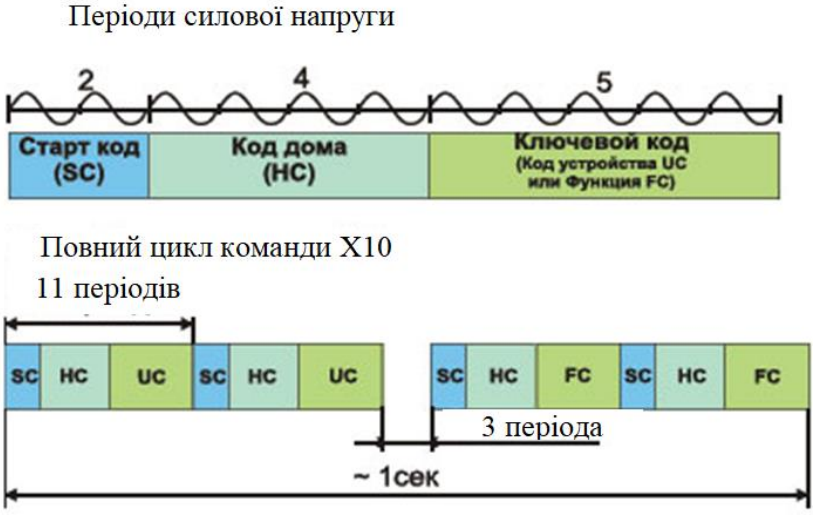


Рисунок 6 – Цикл виконання команди X10

Інтеграція X10

Основним засобом інтеграції мережі X10 із зовнішнім устаткуванням є PLC інтерфейс XM10. Будь-який контролер, що підтримує відкритий протокол обміну з XM10, може відправляти команди X10 в електромережу й, навпаки, одержувати з мережі інформацію про стан пристроїв X10. Так, наприклад, охоронні системи широко відомих у світі брендів – Ademco, DSC, Visonic і деяких інших – підтримують протокол X10 і дозволяють реалізувати інтегровані системи автоматизації й безпеки. Друга можливість апаратної інтеграції – підключення зовнішніх пристроїв по «сухому контакті». У лінійці X10 для цього використовуються модулі UM7206 і SM10.

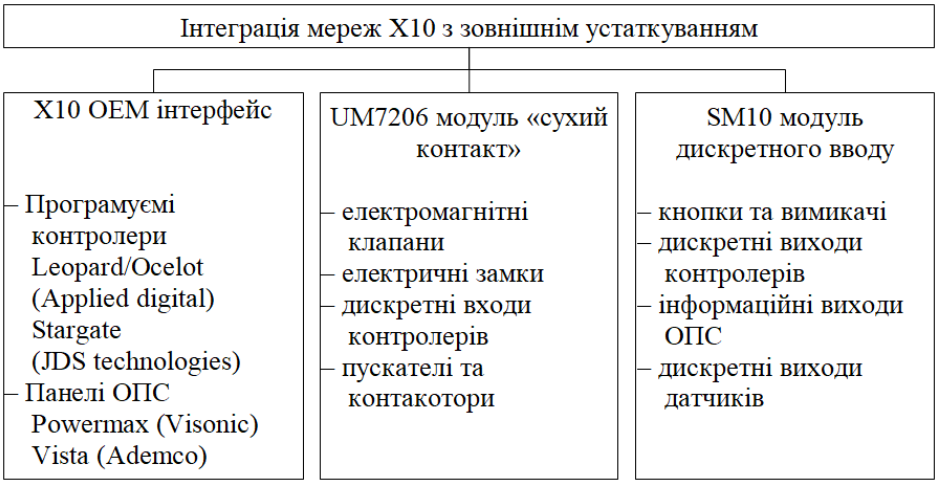


Рисунок 7 – Інтеграція мереж X10

Працездатність мережі X10

При установці автоматики X10 необхідно звертати особливу увагу на:

–перешкодозахищеність системи;

–забезпечення міжфазного проходження сигналу X10 при багатофазній схемі електроживлення;

–захист електричних кіл і пристроїв X10 від перенапруги, перевантаження й струмів короткого замикання.

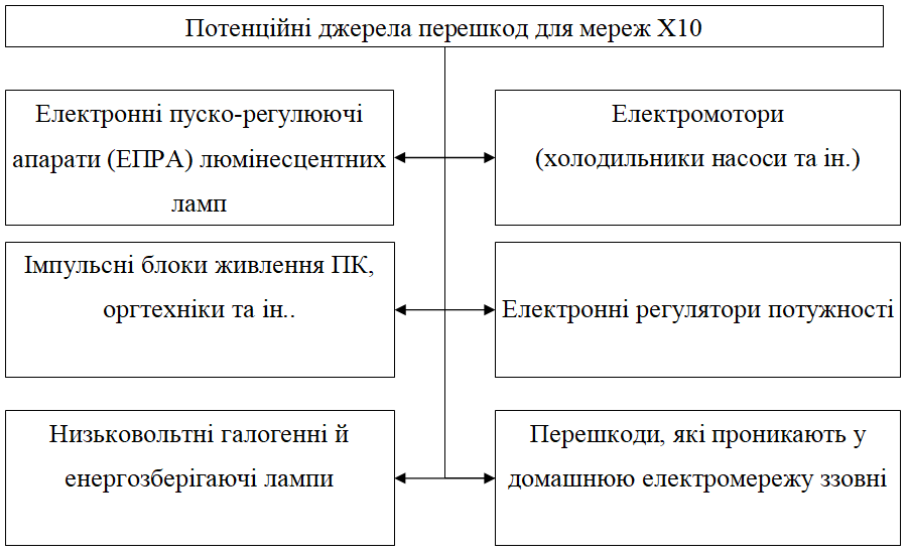


Рисунок 8 – Джерела перешкод для X10

Міри, що рекомендуються, для усунення перешкод у мережах X10:

–використання мережних фільтрів (FM10, FD10, TF678 і ін.) на введенні домашньої електромережі й у місцях підключення до електроживлення приладів-джерел перешкод;

–підключення пристроїв X10 і приладів-джерел перешкод до різних фаз у багатофазній електромережі;

–заміна «шумливих» електроприладів на більш якісні моделі.

У цілому деякі електроприлади стають джерелами перешкод для пристроїв X10, і при дотриманні нескладних правил захисту від високочастотного шуму автоматика X10 працює надійно протягом довгого часу.

Проект X10

Вихідною інформацією для проектів X10, як і для більшості проектів домашньої автоматизації, є:

–плани приміщень

–однолінійна електрична схема

–плани розміщення електроустаткування – світильників, світлових вимикачів, електричних розеток, точок прямого підключення електроустаткування, електричних щитків і шаф, комутаційних вузлів

–перелік автоматизуємих групових і одиночних електричних навантажень, із вказівкою їхніх типів і потужності

–плани існуючої кабельної проводки й кабельний журнал

–відомості про стан об'єкта й можливості проведення кабельних робіт

–вимоги замовника до функціональності проектованої системи управління

При виконанні проекту X10 рекомендується технологічна послідовність робіт зображена на рисунку 10.

Розробка структурної схеми

На рисунку 11 зображена узагальнена структурна схема інтелектуального дому.

Як було відзначено вище, сучасний smart home – це дуже складне інженерне рішення, що складається з наступного набору систем:

–Система безпеки.

–Система комфорту.

–Інформаційна система.

–Система диспетчеризації.

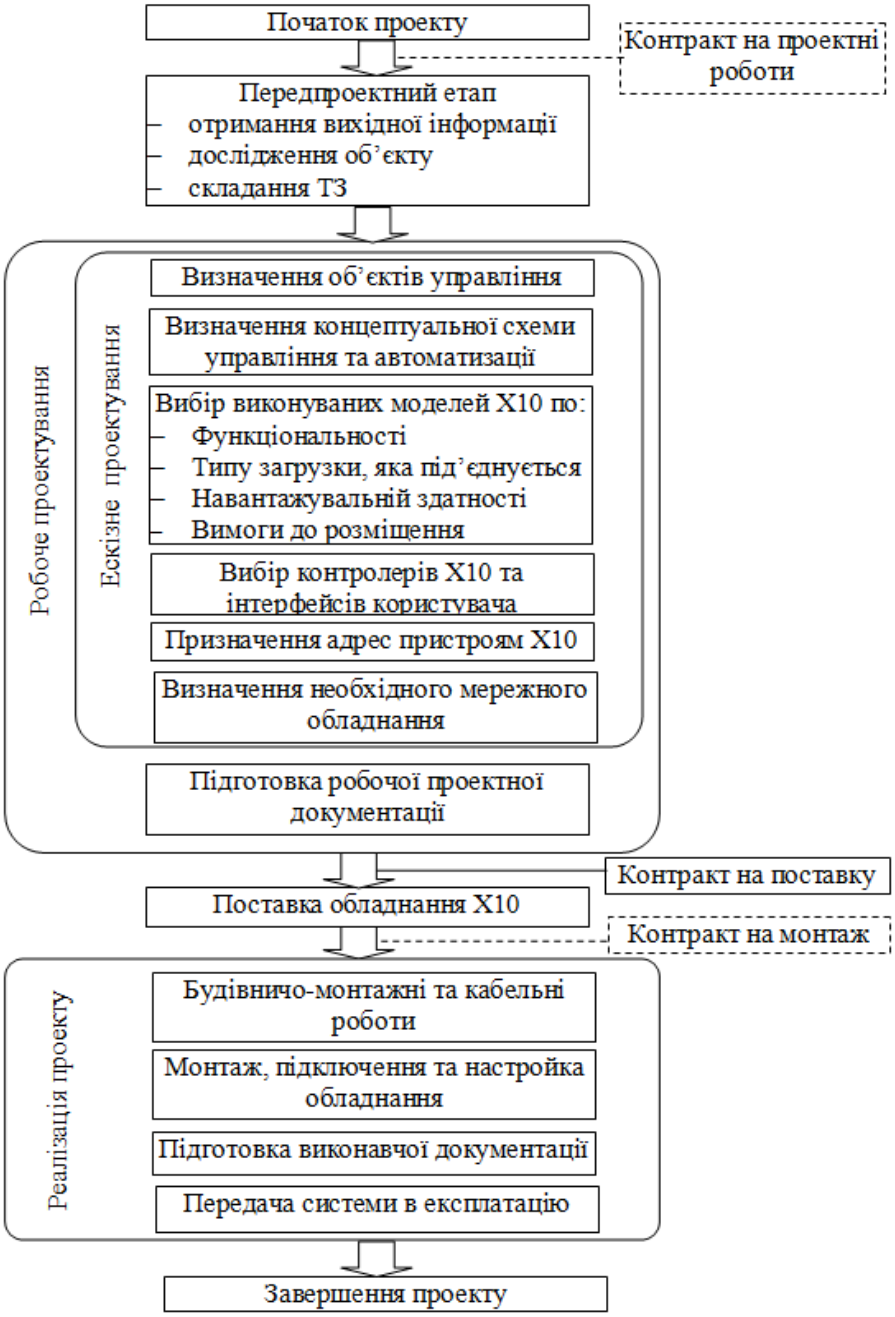


Рисунок 10 – Виконання проекту X10

Розглянемо які підсистеми входять у перераховані вище системи.

Система безпеки:

–Система цифрового відеоспостереження з можливістю одночасного спостереження, перегляду, архівування. Режим віддаленого перегляду й управління через Інтернет.

–Бездротова пожежна й охоронна сигналізація з можливістю обміну інформацією через GSM модуль.

–Система контролю доступу в приміщення (у тому числі віддалене управління гаражними воротами).

Система комфорту:

–Внутрішня телефонна система, голосний зв'язок усередині будинку.

–Система супутникового, ефірного телебачення з можливістю перегляду в будь-якій кімнаті.

–Система "Домашній кінотеатр".

–Система "Мультирум" аудіо– й відео– (система "звук навколо").

–Цифровий розважальний комплекс, оцифровка відео, печать фотографій, створення особистих цифрових фото– і відео– альбомів і т.д.

–Управління світлом у всьому будинку, світлові сцени й сценарії.

–Управління системою вентиляції й кондиціювання.

–Управління системою опалення.

–Управління сауною, басейном.

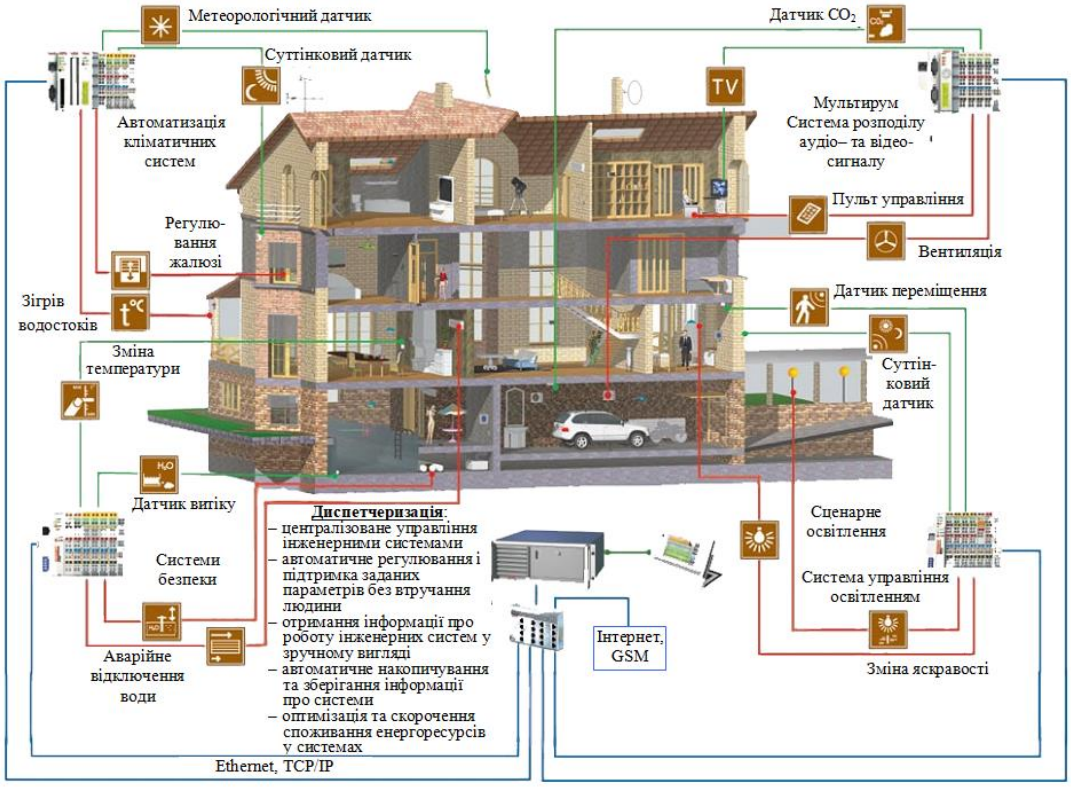


Рисунок 11 – Структурна схема інтелектуального дому

Інформаційна система:

–Установка локальної обчислювальної мережі в будинку, мережна печать, мережні ігри (можлива використання бездротових технологій).

–Вихід в Інтернет з будь-якого комп'ютера в будинку (у тому числі з мобільного).

–Віддалене управління всіма системами будинку через Інтернет.

–"Домашній офіс" з віддаленим підключенням до корпоративної мережі робочого офісу.

Система диспетчеризації:

–Система безперебійного електропостачання

–Управління системою опалення, казаном водонагрівача

–Контроль витоку води, газу.

–Система управління здатна погоджувати роботу інженерних систем, оцінюючи стан сенсорів, датчиків, відпрацьовуючи команди з пультів управління, прив’язуючись до часу доби, пори року й т.п. При цьому виключаються ситуації, коли домашнє устаткування, покликане вирішувати спеціалізовані проблеми, працює в режимах, взаємовиключих один одного.

На рисунку 12 відображено структурну схему взаємодії систем інтелектуального будинку.

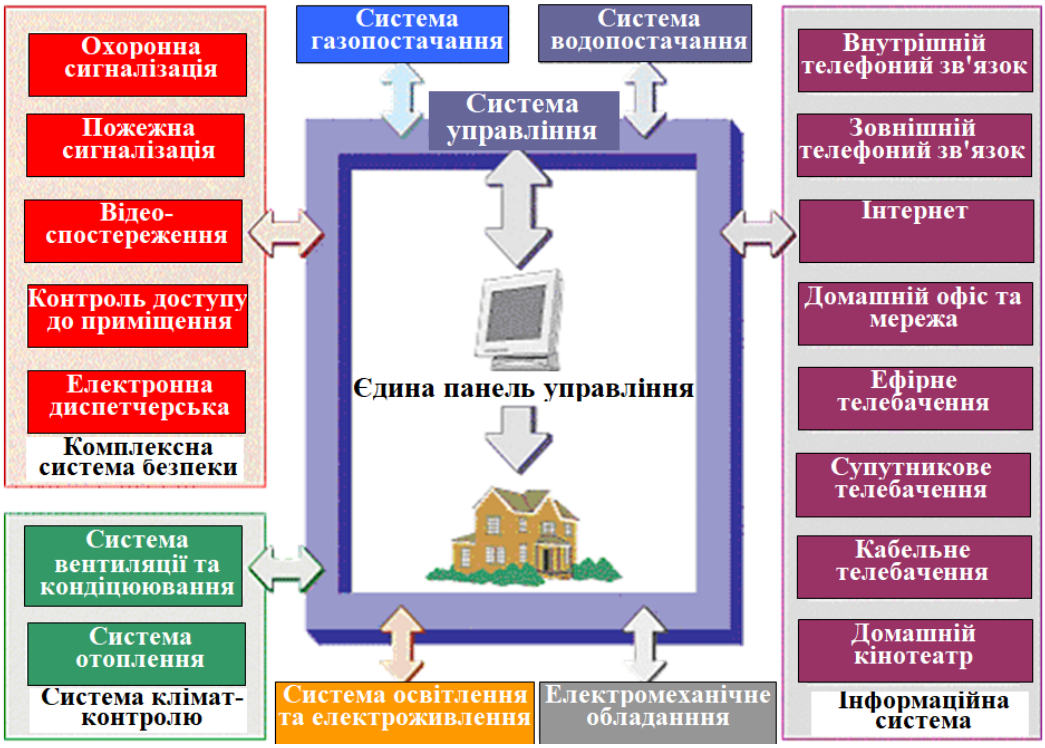


Рисунок 12 – Структурна схема взаємодії систем інтелектуального будинку

Smart home дозволяє замінити всі пульти управління однією або декількома (по кількості зон або кімнат) сенсорними панелями. Вони дозволяють не ламати голову над тим, як підбудувати середовище перебування під необхідні умови.

Використання сучасного устаткування дозволяє створити в будинку єдиний комплекс із систем безпеки (охоронна й пожежна сигналізація, відеоспостереження, контроль доступу), систем зв'язку й комунікації (телефонна й комп'ютерна мережі, оповіщення, екстрений виклик), систем управління опаленням, вентиляцією, освітленням і т.д., що працює по обраному алгоритмі.

Як ілюстрацію можна привести приклад, коли автоматизація й включення в єдиний контур управління освітлювальної системи й систем клімат-контролю (опалення, кондиціювання й вентиляція) будинку (окремої квартири) дозволяють реалізувати автоматичне управління цими системами залежно від пори року й доби, умов навколишнього середовища, присутності людей і інших ,факторів. У результаті досягається істотне зниження витрат на електроенергію й теплопостачання. Досвід показує, що економія експлуатаційних витрат у цьому випадку може досягати 15-20%.

Основні складові інтелектуального будинку

Зв'язок:

–телефонний зв'язок внутрішня (інтерком) і локальні АТС;

–телефонний зв'язок зовнішня провідна, радіорелейна, супутникова;

–інтернет – телефонія;

–системи відеоконференцій.

Мультимедіа:

–наземне й кабельне телебачення;

–супутникове телебачення;

–домашнє відео;

–домашній кінотеатр;

–мульти– аудіо й відео (multiroom).

–один пульт управління для всіх систем

Інформаційні системи:

−локальна комп'ютерна мережа;
−домашній офіс;
−широкополосний доступ у глобальну мережу (Інтернет).
Безпека:
−пожежна сигналізація й система автоматичного пожежогасіння;
−охоронна сигналізація;
−система контролю доступу;
−система зовнішнього й внутрішнього відео спостереження;
−система управління паркінгом;
−система внутрішнього оповіщення (радіомережа);
−аварійний контроль інженерних систем;
−система екологічного контролю.
Водопостачання й газопостачання:
−автоматичне наповнення ванн, басейнів і накопичувальних резервуарів;
−автономне й резервне нагрівання води;
−запобігання витоку водопроводу й витоку газу;
−управління ландшафтними водяними системами (фонтани, водоспади);
−управління температурою води у ваннах і басейнах (термостатування);
−облік витрат й управління споживанням води й/або газу.
Система електроживлення:
−безперебійне й гарантоване електропостачання;
−захист від поразки електричним струмом людей і тварин;
−автоматизація управління електроживленням побутових приладів;
−запобігання перевантажень і короткого замикання в електричній мережі;
−управління якістю електроживлення – моніторинг, стабілізація й фільтрація;
−облік витрат й управління споживанням електроенергії.
Освітлення внутрішнє й зовнішнє:
−аварійне й чергове освітлення;
−автоматизація управління світлом;
−гнучке налаштування світлових груп;
−дистанційне й віддалене управління світлом;
−програмування світлових сцен.
Опалення, вентиляція, кондиціювання:
−автоматизація управління температурою повітря;
−контроль якості повітря;
−узгодження роботи різних кліматичних систем;
−облік витрати й управління споживанням теплової енергії.

Тепер спробуємо створити зі звичайного будинку інтелектуальний. Для цього нам потрібно всі прилади, які виконують перераховані вище функції, об'єднати в одну систему й підключити її до віддаленого сервера. Тим самим буде управління кожним компонентом, та всіма воєдино. І необхідно щоб контроль за станом всіх приладів користувач міг одержати в будь-який момент часу, навіть перебуваючи поза будинком. Для цього треба організувати мережу в будинку, об'єднавши всі прилади й системи над якими треба здійснити контроль, та потім вибрати спосіб підключення до віддаленого сервера.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів smart home з використанням протоколу X10. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем smart home з використанням протоколу X10. Досліджена система smart home з використанням протоколу X10. На основі отриманих результатів досліджень створена програмна реалізація системи smart home з використанням протоколу X10. Розроблені під час виконання випускної

кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання smart home з використанням протоколу X10. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Kovalenko A.S. Information model and its element for displaying information on technical condition of objects of integrated information system / A.S. Kovalenko, A.A. Smirnov, A.V. Kovalenko, A.P. Dorensky // International Journal of Computational Engineering Research (IJCER). – India: Delhi, 2016. – Volume 6, Issue 1. – P. 21-27.
2. Кожанова А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. Кожанова, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2013. – Вип. 6(113). – С. 255-257.
3. Коваленко А.С. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко ., А.А. Смірнов, А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2014. – Вип. 4(120). – С. 161-164.
4. Коваленко А.С. Підсистема технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко , О.А.Смірнов, О.В. Коваленко // Системи озброєння і військова техніка.– Х.: ХУПС, 2014. – № 1(37). – С. 126-129.
5. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.
6. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.
7. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.
8. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.
9. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – С. 85-88.
10. Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 1(126). – С. 75-79.
11. Коваленко А.С. Обґрунтування набору даних для оцінки технічного стану інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2015. – Вип. 1(42). – С.39-41.
12. Коваленко А.С. Експертна система технічного діагностування інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2015. – № 1(41). – С. 106-111.
13. Коваленко А.С. Удосконалення методу технічного обслуговування об’єктів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко, О.П. Доренський // Системи озброєння і військова техніка. – Х.: ХУПС, 2016. – № 2(46). – С. 109-114.
14. Кожанова А.С. Система технічної діагностики інтегрованих інформаційних систем – обґрунтування необхідності створення, визначення понятійного апарату та напрямів досліджень / А.С. Кожанова, О.А. Смірнов, М.П. Савченко, Д.М. Ізосімов, В.В. Мороз // Створення та модернізація озброєння і військової техніки в сучасних умовах: Тринадцята наук.-техн. конф., 5-6 вер. 2013 р., м. Феодосія: тези доп. – Феодосія: ДНВЦ, 2013. – С. 187-188.
15. Кожанова А.С. Визначення основних напрямків досліджень щодо створення системи технічної діагностики інтегрованих інформаційних систем / А.С. Кожанова, О.А. Смірнов, А.В. Челпанов // Проблемні питання розвитку озброєння та військової техніки Збройних Сил України: IV наук.-техн. конф., 16-20 груд. 2013 р., м. Київ: зб. тез. – Київ: ЦНДІ ОВТ ЗСУ, 2013. – С. 293.
16. Коваленко А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформатика та системні науки : V Всеукр. наук.-практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 2014. – С. 292-294.
17. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2014. – С. 241.

18. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп'ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 28-31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 2014. – С. 190-193.
19. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.
20. Коваленко А.С. Метод автоматизованої перевірки результатів вимірювання параметрів об'єкті в інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Стратегія якості у промисловості і освіті: XI міжнар. конф., 1 – 5 черв. 2015 р., м. Варна, Болгарія.: зб. матер. – Варна: ТУВ, 2015. – С. 423-426.

УДК 004

В. Коваленко, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЛЯ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ ІНФОРМАЦІЇ З'ЄМНИХ НОСІЇВ

У статті програмне забезпечення, яке призначено для системи для забезпечення конфіденційності інформації з'ємних носіїв. Метою розробки є дослідження та програмна реалізація системи для забезпечення конфіденційності інформації з'ємних носіїв. Об'єктом дослідження є процес для забезпечення конфіденційності інформації з'ємних носіїв. Предметом дослідження є методи для забезпечення конфіденційності інформації з'ємних носіїв. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи для забезпечення конфіденційності інформації з'ємних носіїв. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, захисту доступу, з'ємних носії

Постановка проблеми. Широке застосування комп'ютерних технологій і постійне збільшення обсягу інформаційних потоків викликає постійний ріст інтересу до криптографії. Останнім часом збільшується роль програмних засобів захисту інформації, просто модернізуємих, не потребуючих великих фінансових витрат у порівнянні з апаратними криптосистемами [1-5]. Сучасні методи шифрування гарантують практично абсолютний захист даних, але завжди залишається проблема надійності їхньої реалізації. Іншою важливою проблемою застосування криптографії є протиріччя між бажанням громадян захистити свою інформацію й прагненням державних спецслужб мати можливість доступу до деякої інформації для припинення незаконної діяльності [6]. Надзвичайно важко знайти незаперечно оптимальне рішення цієї проблеми. Як оцінити співвідношення втрат законослухняних громадян і організацій від незаконного використання їхньої інформації й збитків держави від неможливості одержання доступу до захищеної інформації окремих груп, що приховують свою незаконну діяльність? Чи можна гарантированно не допустити незаконне використання криптоалгоритмів особами, які порушують і інші закони? Крім того, завжди існують способи схованого зберігання й передачі інформації. Хоча стримування відкритих досліджень в області криптографії й криптоаналізу є найпростішим шляхом, але це принесе значний негативний ефект. Застосування ненадійних засобів не захистить користувачів, але викличе поширення комп'ютерних злочинів, навпроти, виявлення своєчасне виявлення помилок у системах захисту інформації дозволить запобігти збитку [5-

8]. У цей час особливо актуальною стала оцінка вже використовуваних криптоалгоритмів. Завдання визначення ефективності засобів захисту найчастіше більше трудомістка, чим їхня розробка, вимагає наявності спеціальних знань і, як правило, більше високої кваліфікації, ніж завдання розробки. Ці обставини приводять до того, що на ринку з'являється безліч засобів криптографічного захисту інформації, про які ніхто не може сказати нічого певного. При цьому розроблювачі тримають криптоалгоритм (як показує практика, часто нестійкий) у секреті. Однак задача точного визначення даного криптоалгоритму не можуть бути гарантированно складною хоча б тому, що він відомий розроблювачам. Крім того, якщо порушник знайшов спосіб подолання захисту, то не в його інтересах про це заявляти. Тому суспільству повинне бути вигідно відкрите обговорення безпеки систем захисту інформації масового застосування, а приховання розроблювачами криптоалгоритму повинне бути неприпустимим.

На сьогоднішній день існують добре відомі й апробовані криптоалгоритми (як із симетричними, так і несиметричними ключами), криптостійкость яких або доведена математично, або заснована на необхідності рішення математично складного завдання (факторизації, дискретного логарифмування й т.п.) [7-9]. З іншого боку, у комп'ютерному світі весь час з'являється інформація про помилки або "діри" у тій або іншій програмі (у т.ч. що застосовує криптоалгоритми), або про те, що вона була зламана. Це створює недовіру, як до конкретних програм, так і до можливості взагалі захистити що-небудь криптографічними методами не тільки від спецслужб, але й від простих хакерів. Тому знання атак і дір у криптосистемах, а також розуміння причин, по яких вони мали місце, є однією з необхідних умов розробки захищених систем і їхнього використання.

У зв'язку з тим, що на даному етапі часто для переносу інформації використовуються пристрої на flash-накопичувачах, дуже актуальним є завдання їхнього захисту. Існує два підходи до захисту інформації на flash-накопичувачах: з використанням доступу до flash-накопичувача за допомогою біопараметричних характеристик і за допомогою шифрування інформації, що зберігається на flash-накопичувачі.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи для забезпечення конфіденційності інформації з’ємних носіїв.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи для забезпечення конфіденційності інформації з’ємних носіїв.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем для забезпечення конфіденційності інформації з’ємних носіїв.

– Дослідження системи для забезпечення конфіденційності інформації з’ємних носіїв.

– Програмна реалізація системи для забезпечення конфіденційності інформації з’ємних носіїв.

Об’єктом дослідження є процес для забезпечення конфіденційності інформації з’ємних носіїв.

Предметом дослідження є методи для забезпечення конфіденційності інформації з’ємних носіїв.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Опис генератора псевдовипадкових чисел

Послідовності випадкових чисел (ВЧ) є невід'ємним інструментом рішення багатьох математичних завдань, і в тому числі завдань захисту інформації. Для потокових шифрів він використовується у ролі гами для накладання на дані, які потрібно захищати. Найчастіше необхідні послідовності випадкових чисел, які рівномірно й рівноймовірно розподілені на

деякому відрізку. Більшість природних процесів є випадковими, і, відповідно до теорії математичної статистики, вони підкоряються різним законам розподілу випадкових величин. Для породження випадкових чисел, рівномірно розподілених на відрізку [0..*M*], досить для деякого випадкового процесу, що підкоряється закону рівномірного розподілу, увести міру випадкової величини. А потім послідовно проводити експерименти, вимірювати значення випадкової величини й після нормування одержувати необхідну рівномірну випадкову послідовність чисел.

Подібні методи дадуть дуже гарні статистичні результати, але зажадають колосального часу для одержання скільки-небудь довгої послідовності. На щастя, математики розробили рекурентні формули одержання псевдовипадкових чисел (ПВЧ). Назва «псевдовипадкові» обумовлена хоча б тим фактом, що якщо відомо деяке і-*е* число, то по формулі однозначно обчислимо ($i + 1$)-ий елемент послідовності. Більше того, з рекурентної природи знайденої формули виходить, що при тому самому x_0 ми одержимо при повторній генерації ту ж саму послідовність. До того ж, всі арифметичні алгоритми генерації ПВЧ періодичні, тобто існує деяке $p \in N$, для якого виконується $x_i = x_{i+pn}$ при будь-якому натуральному n. Тому будь-який ГПВЧ повинен бути ініціалізований випадковою величиною, якимось зовнішнім джерелом випадкових значень. Такими «апаратними» у ПК генераторами можуть виступати, наприклад, електричні шуми в напівпровідникових пристроях, а також поточна кількість виконаних процесором тактів або поточне значення часу.

Недоліки ГПВЧ

- порівняно короткий період генеруємої послідовності.
- залежність між сусідніми послідовними значеннями.
- нерівномірність розподілу значень, у тому числі через різний ступінь.

Ініціалізація ГПВЧ

Читання поточного значення мілісекунд

Як уже було сказано вище, рекурентну формулу обчислення і-*го* члена ПВЧ при реалізації на ПК можна ініціалізувати значенням мілісекунд поточного часу в момент запуску програми. Для цього можна викликати функцію «2Ch» переривання 21h операційної системи DOS і одержати «майже» випадкове число на відрізку [0..99].

Читання значення лічильника тактів процесора

Починаючи з лінійки процесорів Pentium в архітектурі x86 з'явилася інструкція, що дозволяє прочитати лічильник тактів процесора з моменту останнього скидання. Для інструкції *rdtsc* (Read Time Stamp Counter) заданий машинний код 0F 31. 64-бітне значення лічильника вертається в парі регістрів <EDX:EAX>. Якщо для ініціалізації генератора досить 32-бітного значення, то необхідно використовувати найбільше «чутливі» молодші 32 біта лічильника тактів. При використанні старого компілятора мови асемблера для звертання до 32-бітного регістра EAX знадобиться вручну проставити префікс із кодом 66*h*.

Лінійний конгруентний метод

Рекурентна формула виглядає в такий спосіб:

$$x_n = (ax_{n-1} + c) \bmod m. \qquad (1)$$

Період породжуваної послідовності не перевищує m. Природно, що від вибору параметрів a, c, m і значення першого члена x_0 істотно залежать основні властивості породжуваної послідовності. Довжина періоду буде максимальною (рівна m) тільки в тому випадку, коли:

- НЗД(c, m) = 1 (тобто c і m взаємно прості)
- $a - 1$ кратно всім простим дільникам m
- якщо m кратно 4, то й $(a - 1) \bmod 4 = 0$

Алгоритм Блюма-блюма-Шуба

Алгоритм ГПВЧ, стійкий до зворотних перетворень. Основна рекурентна формула алгоритму:

$$x_n = (x_{n-1})^2 \bmod pq, \qquad z_n = parity\ (x_n), \qquad (2)$$

де p і q – два великих простих числа. Для підвищення якості одержуваної послідовності на черговому кроці вибираються не всі біти x_n, а тільки молодші, або навіть тільки біт парності. З отриманих «випадкових біт» формуються двійкові ПВЧ довільної розрядності. Однієї з особливостей обчислювальної формули є наскрізна можливість обчислити x_n без генерації всіх попередніх членів послідовності.

$$x_n = (x_0)^{2n \bmod (p-1)(q-1)} \bmod pq, \quad (3)$$

Даний алгоритм більше вимогливий до обчислювальних ресурсів, але, з іншого боку, має гарні статистичні характеристики.

Алгоритм xor-shift

Професором університету Флориди Джорджем Марсаглією був розроблений дуже швидкий генератор, що був названий « xor-shift».

$$x = 3456789,\ y = 62436069,\ z = 1288629,\ w = 675123 \quad (4)$$

$$t = (x \text{ xor } (x \text{ shl } 11)),\quad x = y,\quad y = z,\quad z = w, \quad (5)$$

$$\text{xor128} = w = (w \text{ xor } (w \text{ shr } 19)) \text{ xor } (t \text{ xor } (t \text{ shr } 8))) \quad (6)$$

Існує комбінація даного алгоритму з лінійним конгруентним алгоритмом і алгоритмом Фібоначчі із запізнюваннями.

$$x = 123456789,\quad y = 362436000,\quad z = 521288629,\quad c = 7654321, \quad (7)$$

$$x = \text{int64}(69069)\cdot x + 12345,\quad y = y \text{ xor } (y \text{ shl } 13), \quad (8)$$

$$y = y \text{ xor } (y \text{ shr } 17),\quad y = y \text{ xor } (y \text{ shl } 5), \quad (9)$$

$$t = \text{int64}(698769069)\cdot z + c,\quad c = t \text{ shr } 32, \quad (10)$$

$$z = t,\quad \text{Random} = x + y + z, \quad (11)$$

Опис алгоритму шифрування

У якості криптоалгоритму спрямованого на захист інформації, яка утримується в flash-пристрої візьмемо алгоритм RC4. Розглянутий нами криптоалгоритм RC4 відноситься до класу потокових шифрів, які останнім часом стали популярними завдяки високій швидкості роботи. Потокові шифри перетворять відкритий текст у шифротекст по одному біті за операцію. Генератор потоку ключів (іноді називаний генератором із ключем, що біжить) видає потік біт: k_1, k_2, k_3, ..., k_i. Цей потік ключів і потік біт відкритого тексту, p_1, p_2, p_3, ..., p_i, піддаються операції “або, що виключає", і в результаті виходить потік біт шифротексту.

$$c_i = p_i \oplus k_i \quad (12)$$

При дешифруванні операція XOR виконується над бітами шифротексту й тим же самим потоком ключів для відновлення біт відкритого тексту.

$$p_i = c_i \oplus k_i \quad (13)$$

Безпека системи повністю залежить від властивостей генератора потоку ключів. Генератор потоку ключів створює бітовий потік, що схожий на випадковий, але в дійсності детермінований і може бути безпомилково відтворений при дешифруванні. Чим ближче вихід генератора потоку ключів до випадкового, тим більше часу буде потрібно для взлому шифру.

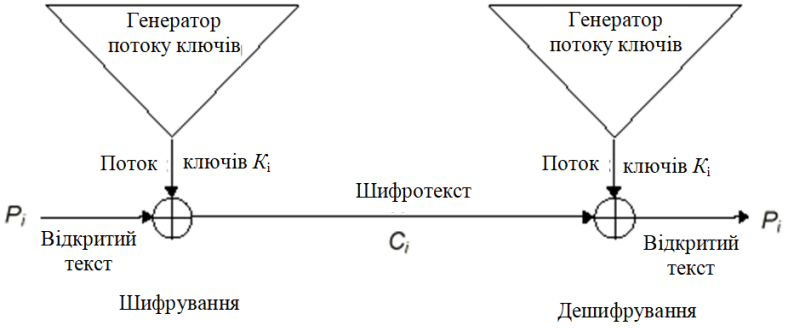


Рисунок 1 – Потоковий шифр

Для всіх потокових шифрів використовуються ключі. Вихід генератора потоку ключів є функцією ключа. Тепер, якщо одержати пару відкритий текст/шифротекст, то можна читати тільки ті повідомлення, які зашифровані тим же ключем.

Потокові шифри особливо корисні для шифрування нескінченних потоків комунікаційного трафіку, наприклад, при записі данних на flash-пам'ять.

Генератор потоку ключів складається із трьох основних частин:

– Внутрішній стан описує поточний стан генератора потоку ключів.

– Два генератори потоку ключів, з однаковим ключем і однаковим внутрішнім станом, видають однакові потоки ключів.

– Функція виходу по внутрішньому стану генерує біт потоку ключів.

– Функція наступного стану по внутрішньому стану генерує новий внутрішній стан.

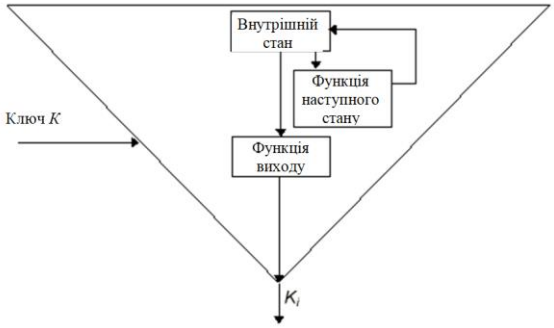


Рисунок 2 – Пристрій генератора потоку ключів

Криптоалгоритм RC4 відноситься до так званих шифрів, що самосинхронізуються. У потокових шифрах, що самосинхронізуються, кожний біт потоку ключів є функцією фіксованого числа попередніх біт шифротексту. Військові називають цей шифр автоключом шифротексту.

Потоковий шифр, що самосинхронізується, показаний на рисунку 3. Внутрішній стан є функцією попередніх n біт шифротексту. Криптографічно складною є вихідна функція, що використовує внутрішній стан для генерації біта потоку ключів.

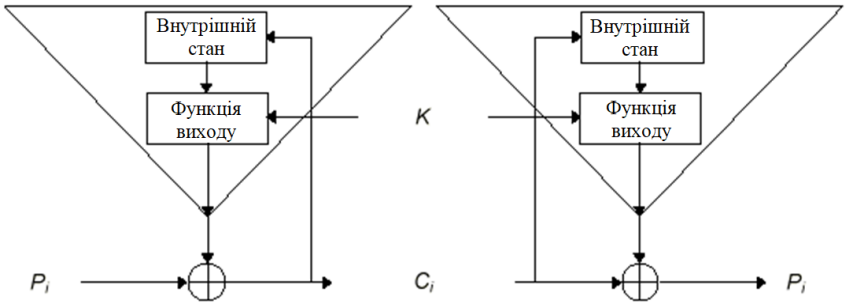


Рисунок 3 – Генератор потоку ключів, що самосинхронізується

Так як внутрішній стан повністю залежить від попередніх *n* шифротексту, дешифруючий генератор потоку ключів автоматично синхронізується з генератором, що шифрує, потоку ключів, прийнявши *n* біт шифротексту. В інтелектуальних реалізаціях цього режиму кожне повідомлення починається випадковим заголовком довжиною *n* біт.

Цей заголовок шифрується, передається й потім розшифровується. Розшифровка буде неправильною, але після цих *n* біт обидва генератори потоку ключів будуть синхронізовані.

Слабкою стороною потокового шифру, що самосинхронізується, є поширення помилки. Для кожного біта шифротексту, зіпсованого при передачі, дешифруючий генератор потоку ключів видає n неправильних біт потоку ключів. Отже, кожному неправильному біту шифротексту відповідають n помилок у відкритому тексті, поки зіпсований біт не перестане впливати на внутрішній стан.

Алгоритм RC4 і його криптоаналіз

Істотне підвищення продуктивності мікропроцесорів в 80-і роки викликало в криптографії посилення інтересу до програмних методів реалізації криптоалгоритмів як можливій альтернативі апаратним схемам на регістрах зрушення.

Одним з найперших подібних криптоалгоритмів, що получили широке поширення, став RC4. Алгоритм RC4 – це потоковий шифр зі змінною довжиною ключа.

Він володіє наступними властивостями:

- адаптивністю для апаратних засобів і програмного забезпечення, що означає використання в ньому тільки примітивних обчислювальних операцій, звичайно присутніх на типових мікропроцесорах;
- алгоритм швидкий, тобто в базисних обчислювальних операціях оператори працюють на повних словах даних;
- адаптивністю на процесори різних довжин слова;
- компактністю в термінах розміру коду, і особливо зручний для процесорів з побайтно-орієнтованою обробкою;
- низькою вимогою до пам'яті, що дозволяє реалізовувати алгоритм на пристроях з обмеженою пам'яттю;
- використанням циклічних зрушень, залежних від даних, з "змінним" числом;
- простотою й легкістю виконання.

У цей час алгоритм RC4 реалізований у десятках комерційних криптографічних продуктів, включаючи Lotus Notes, Apple Computer's AOCE, Oracle Secure SQL, а також є частиною специфікації стандарту стільникового зв'язка CDPD.

Криптогенератор функціонує незалежно від відкритого тексту. Генератор має підстановочну таблицю (S-бокс 8 x 8): S_0, S_1, . . ., S_{255}. Входами генератора є замінені по підстановці числа від 0 до 255, і ця підстановка є функцією від ключа змінюваної довжини. Генератор має два лічильники i і j, ініціалізуємих нульовим значенням.

Для генерації випадкового байта гами виконуються наступні операції:

$$i = (i+1) \bmod 256 \quad (14)$$

$$j = (j+S_i) \bmod 256 \quad (15)$$

$$\text{swap}\ (S_i, S_j) \quad (16)$$

$$t = (S_i+S_j) \bmod 256 \quad (17)$$

$$K = S_t \quad (18)$$

Байт K складається операцією XOR з відкритим текстом для виробітку шифротексту, або із шифротекстом для одержання байта відкритого тексту. Шифрування відбувається досить швидко – приблизно в 10 разів швидше DES-Алгоритму. Ініціалізація S-боксу настільки ж проста. На першому кроці він заповнюється лінійно: S0 = 0, S1 = 1, . . ., S255 = 255.

Потім ще один 256-байтний масив повністю заповнюється ключем, для чого ключ повторюється відповідне число раз залежно від довжини: K0, K1,...,K255. Індекс j обнуляется. Потім:

```
for (i=0; i<= 255; i++)
{
j = (j+Si+Ki) mod 256;
swap (Si , Sj);
}
```

Схема показує, що RC4 може приймати приблизно 2^{1700} ($256! * 256^2$) можливих станів. S-бох повільно змінюється в процесі роботи: параметр i забезпечує зміну кожного елемента, а j відповідає за те, щоб ці елементи змінювалися випадковим образом.

Фактично, RC4 являє собою сімейство алгоритмів, що задаються параметром n, що є позитивним цілим з рекомендованим типовим значенням $n = 8$.

Внутрішній стан генератора RC4 у момент часу t складається з таблиці $S_t = (S_t(L))_{t=0}^{n2\ -1}$, що містить $2^{n\ n-6}$ітних слів і із двох n-бітних слів-покажчиків i_t і j_t. Таким

чином, розмір внутрішньої пам'яті становить $M = n2^n + 2n$ біт. Нехай вихідне n-бітне слово генератора в момент t позначається як Z_t.

Нехай початкові значення $i_0 = j_0 = 0$. Тоді функція наступного стану й функція виходу RC4 для кожного $t > 1$ задається наступними співвідношеннями:

$$i_t = i_{t-1} + 1 \quad (19)$$

$$j_t = j_{t-1} + S_{t-1}(i_t) \quad (20)$$

$$S_t(i_t) = S_{t-1}(j_t) \quad (21)$$

$$S_t(j_t) = S_{t-1}(i_t) \quad (22)$$

$$Z_t = S_t(S_t(i_t) + S_t(j_t)), \quad (23)$$

де всі додавання виконуються по модул. 2^n. Мається на увазі, що всі слова, крім тих, які піддаються перестановці, залишаються тими ж самими. Вихідна послідовність n-бітних слів позначається як $Z_t = (Z_t)_{t=1}^{\infty}$.Початкова таблиця S_0 задається в термінах ключової послідовності:

$$K = (K_L)_{t=0}^{2n-1} \quad (24)$$

з використанням тієї ж самої функції наступного стану, починаючи від таблиці одиничної підстановки $(L)^{2n}{}_{L=0}{}^{2n-1}$. Більш строго, нехай $j_0 = 0$ і для кожного $1 \le t \le 2^n$ обчислюється $j_t = (j_t - 1 + S_{t-1}(t\text{-}1) + K_{t-1}) \bmod 2n$, а потім переставляються місцями $S_{t-1}(t\text{-}1)$ і $S_{t-1}(j_t)$.

На останньому кроці породжується таблиця, що представляє S_0. Ключова послідовність K складається із секретного ключа, що можливо повторюється, і випадкового ключа, переданого у відкритому виді з метою ресинхронизації.

До останнього часу у відкритій літературі практично не було публікацій по криптоаналізу алгоритму RC4. Компанія RSA Data Security оголосила, що шифр має імунітет до методів лінійного й диференціального криптоаналізу, високо не лінієн і не схоже, щоб у нього були короткі цикли.

Відзначається, що для послідовностей, генеруємих RC4, не підходять методи статистичного аналізу. Але, з іншого боку, для блоків, розмір яких перевищує M (розмір внутрішньої пам'яті генератора), завжди існує лінійна статистична слабість або так звана "лінійна модель". Таку модель можна ефективно визначати за допомогою методу апроксимації лінійною послідовною схемою. Лінійна статистична слабість – це лінійне співвідношення між бітами гами, що виконується з імовірністю, що відрізняється від 1/2.

За допомогою методу АЛПС були виведені лінійні моделі для RC4. Метод АЛПС полягає в знаходженні й рішенні послідовної лінійної схеми, що апроксимує генератор гами й приводить до лінійних моделей з відносно великим кореляційним коефіцієнтом c, де ймовірність відповідного лінійного співвідношення між бітами гами становить $(1 + c)/2$. При аналізі використовувалася техніка двійкових похідних. Нехай $Z = (Z_t)_{t=1}^{\infty}$ позначає послідовність самих молодших біт слів виходу RC4, і нехай $Z' = (Z'_t = Z_t + Z_{t+1})_{t=1}^{\infty}$ і $Z'' = (Z''_t = Z_t + Z_{t+2})_{t=1}^{\infty}$ позначають її перші й другу двійкові похідні, відповідно. Показано, що Z' не корелює ні з 1, ні з 0, але Z'' корелює з 1 з кореляційним коефіцієнтом, близьким до $15^* 2\text{-}3^n$ при великих $2n$, де n – довжина ключа. Оскільки довжина вихідної послідовності, необхідна для виявлення статистичної слабості з кореляційним коефіцієнтом c, становить O(c-2), те ця довжина дорівнює приблизно $64^n /225$. Наприклад, якщо $n = 8$, як рекомендується в більшості додатків, те необхідна довжина близька до 2^{40}.

Результати комп'ютерних експериментів погодяться з теоретичними пророкуваннями. Оскільки результуючий коефіцієнт кореляції істотно перевищує величину $2^{M/2}$, те встановлену лінійну модель варто розглядати як статистичну слабість генератора, принаймні в теоретичному аспекті.

Був проведений криптоаналіз узагальненої схеми вузла, що комбінує, з довільним розміром пам'яті. Досліджено кореляційні властивості таких вузлів, обґрунтовані нові конструктивні критерії, пропоновані до схем подібного типу.

Розроблено ефективний метод апроксимації лінійною послідовною схемою для побудови лінійних функцій від входу й виходу з порівняно більшим коефіцієнтом взаємної

кореляції. Це практична процедура, що дозволяє з високою ймовірністю знаходити пари взаємно коррельованих лінійних функцій (від якнайбільше $M + 1$ послідовних вихідних біт і якнайбільше $M + 1$ послідовних векторів входу) з порівняно більшими коефіцієнтами кореляції. Метод АЛПС складається в завданні й рішенні лінійної послідовної схеми (ЛПС), що апроксимує вузол, що комбінує, з пам'яттю. Ця ЛПС має додаткові незбалансовані входи й заснована на лінійних апроксимаціях функції виходу й всіх компонентів функції наступного стану. Лінійна апроксимація булевої функції – це будь-яка лінійна функція, з якою задана булева функція скорельована. Описаний метод застосуємо до довільних вузлів, що комбінують, з пам'яттю без обмежень на функції виходу й наступний стан.

Спочатку відшукуються лінійні апроксимації функції виходу f і кожної з функцій-компонентів функції наступного стану F. Це еквівалентно вираженню кожної із цих $M + 1$ функцій у вигляді суми лінійної функції й незбалансованої функції. Якщо підлягаючої декомпозиції функція вже несбалансована, то можна вибрати константно-нульову лінійну функцію. Якщо підлягаюча декомпозиції функція статистично незалежна від деякої підмножини змінних, то кожна лінійна апроксимація з необхідністю повинна задіяти принаймні одну зі змінних цієї підмножини. Основна вимога – щоб відповідні кореляційні коефіцієнти відрізнялися від нуля. Також бажано, щоб вибиралися лінійні апроксимації з кореляційними коефіцієнтами, абсолютні значення яких близькі до максимального. Кореляційні коефіцієнти можна визначати за допомогою техніки перетворення Уолша.

На наступному кроці, одержавши лінійні апроксимації, у матричній формі записують базові рівняння вузла, що комбінує, з пам'яттю

$$S_{t+1} = A \cdot S_t + B \cdot X_t + \Delta(X_t, S_t),\ t \geq 0, \tag{25}$$

$$y_t = C \cdot S_t + D \cdot X_t + \varepsilon(X_t, S_t),\ t \geq 0, \tag{26}$$

де вектори розглядаються як матриці-стовпці; A, B, C, D – двійкові матриці; а ε і кожний компонент в $D = (d_1, \ldots, d)$ – незбалансовані булеві функції, іменовані функціями шуму. Основна ідея полягає в тому, щоб розглядати $\{\varepsilon(X_t, S_t)\}_{t=0}^{\infty}$ і $\{\delta(X_t, S_t)\}_{t=0}^{\infty}$, $1 \leq i \leq M$, як вхідні послідовності, так що останні рівняння виявляються задаючими неавтономну лінійну машину з кінцевим числом станів або ЛПС, іменовану АЛПС вузла, що комбінує, з пам'яттю. Тоді можна вирішувати цю ЛПС із використанням техніки виробляючих функцій (D-перетворень). Зокрема, нехай S, X, Δ, ε, у позначають виробляючі функції від змінної z для послідовностей $\{S_t\}$, $\{X_t\}$, $\Delta(X_t, S_t)\}$, $\varepsilon(X_t, S_t)\}$, $y_t\}$, відповідно. Тоді рівняння зводяться до виду:

$$y = \left(D - \frac{C \cdot adj(zA - I)B}{\det(zA - I)} \right) X - \frac{C \cdot adj(zA - I)}{\det(zA - I)} (z\Delta + S_0) + \varepsilon \tag{27}$$

де I – одинична матриця, $\det(z - I) = \phi(z)$, $\phi(0) = 1$, – багаточлен, зворотний до характеристичного багаточлена матриці переходів A ступеня, що не перевищує ранг A ($\leq M$); а елементи (приєднаної) матриці $\mathrm{adj}(z - I)$ – це поліноми від z ступеня не більше M-1. Обчислювальна складність для відшукання такого рішення становить $O(M^3(N+1))$. В іншому виді рішення можна переписати як

$$y = \frac{1}{\varphi(z)} \sum_{i=1}^{N} g_i(z) x_i + \frac{1}{\varphi(z)} \sum_{j=1}^{M} h_j(z)(z\delta_j + s_{j0}) + \varepsilon \tag{28}$$

де x_i і δ_j позначають виробляючі функції для $\{x_{it}\}$ і $\{\delta_j(X_t, S_t)\}$, а ступеня поліномів $g_i(z)$ і $h_j(z)$ якнайбільше рівні M і M-1, $1 \leq i \leq N$, $1 \leq j \leq M$, відповідно. Взявши

$$\varphi(z) = \sum_{k=0}^{M} \varphi_k z^k, \quad g_i(z) = \sum_{k=0}^{M} g_{ik} z^k, \quad h_j(z) = \sum_{k=0}^{M-1} h_{jk} z^k \tag{29}$$

рішення можна перетворити до виду:

$$\sum_{k=0}^{M} \varphi_k y_{l-k} = \sum_{i=1}^{N} \sum_{k=0}^{M} g_{ik} x_{i,l-k} + e\left(X_i^{M+1}, S_{l-M}\right), \qquad t \geq M, \tag{30}$$

$$e\left(X_i^{M+1}, S_{l-M}\right) = \sum_{j=1}^{N}\sum_{k=0}^{M-1} h_{jk}\delta_j\left(X_{t-1-k}, S_{t-1-k}\right) + \sum_{k=0}^{M-1}\varphi_k\varepsilon\left(X_{t-k}, S_{t-k}\right), \qquad t \geq M, \qquad (31)$$

де мається на увазі, що вектор стану S_{t-k} – це функція від (X_{t-k-1}^{M-k}, S_{t-M}) для кожного $0 \leq k \leq M$-1. Лінійні функції входу й виходу в (30) скоррельовані тоді й тільки тоді, коли функція шуму *e* незбалансована. Коефіцієнт кореляції не залежить від часу, якщо функція наступного стану збалансована. Якщо ця умова не задовольняється, то кореляційний коефіцієнт може залежати від часу, оскільки від S_t більше не потрібна збалансованість для кожного $t \geq 0$. Функція шуму *e* в (31) визначена як сума індивідуальних шумових функцій, які незбалансовані за умови, що збалансовано функцію наступного стану. Оскільки від індивідуальних шумових функцій не потрібно бути незалежними, у принципі не можна виключати можливість, що коефіцієнт кореляції *e* з константною нульовою функцією дорівнює нулю або дуже близький до цього значення.

У розглянутому випадку індивідуальні шумові функції можна трактувати як булеві функції від $n = MN + N + M$ змінних в ($X^{M+1}{}_t$, S_{t-M}). Отже, за винятком деяких особливих випадків, у загальному випадку можна з високою ймовірністю очікувати, що загальний кореляційний коефіцієнт дуже близький до добутку індивідуальних і, таким чином, відрізняється від нуля. Відповідно, метод АПЛС не тільки з високою ймовірністю дає взаємно коррельовані лінійні функції від входу й виходу, але також дозволяє оцінити значення відповідного кореляційного коефіцієнта, використовуючи незалежність або інші імовірнісні припущення. Оскільки в ідеальному випадку хотілося б одержати такі АЛПС, у яких кореляційні коефіцієнти за абсолютним значенням близькі до максимуму, те індивідуальні кореляційні коефіцієнти повинні бути великими по величині, а кількість шумових членів в (31) повинне бути маленьким.

Звичайно, ці вимоги можуть суперечити один одному. Тому гарним підходом буде повторення процедури АЛПС кілька разів, починаючи з найкращих лінійних апроксимацій для функції виходу й компонент функції наступного стану. Ця процедура може також виконуватися для всіх можливих лінійних апроксимацій, що представляється єдиним систематичним способом перевірити всі кореляції, виявлені в процесі застосування методу АЛПС. У загальному випадку є якнайбільше $(M+1)2^{M+N}$ таких лінійних апроксимацій. Однак, у принципі завжди можна перевірити всі можливі лінійні апроксимації навіть при великому *M*, оскільки в практичних реалізаціях функції виходу й наступного стану залежать від порівняно невеликої кількості змінних або ж складені з таких булевих функцій.

Із практичної точки зору дана лінійна модель може бути використана для виділення по шифротексту генератора RC4 серед інших криптосистем, а також для відновлення параметра *n*. В 2000 році була опублікована стаття присвячена статистичному аналізу потокового генератора RC4, у якій були використані результати роботи для знаходження значення компонент *S*-боксу. Приблизний час роботи цього методу становить 2^{6n}, де *n* – порція біт у вихідному потоці, довжина вихідної послідовності, необхідна для виявлення статистичної слабості, близька до 2^{30}. Отриманий результат указує на істотну слабість генератора й можливість відновити параметри *i* і *n*. *S*-бокс може приймати 2^{nk}, де n_k – число біт ключа.

Розробка структурної схеми

На рисунку 4 зображена структурна схема роботи системи. Схема розділена на три основних компоненти:

- Flash накопичувач;
- розроблена програма;
- користувач.

Коли користувач вставляє в персональний комп'ютер Flash накопичувач, відбувається розпізнання операційною системою типу пристрою й виводу меню вироблених дій.

Розроблена програма перехоплює системне повідомлення операційній системі про виклик меню вироблених дій над Flash накопичувачем і активізує власний інтерфейс програми.

Розроблена програма складається з декількох частин:

- інтерфейсу програми;
- модуля потокового шифрування RC4;
- модуля потокового дешифрування RC4;
- налаштовування програми й захисту програми.

Розроблена програма управляє процесом обміну інформацією між Flash накопичувачем і персональним комп'ютером, використовуючи потоковий алгоритм шифрування інформації RC4. Завдяки такому підходу, можливо використовувати всі існуючі на даний момент Flash накопичувачі не зупиняючись на окремих реалізаціях з підвищеними вимогами захищеності Flash накопичувача (рисунок 4).

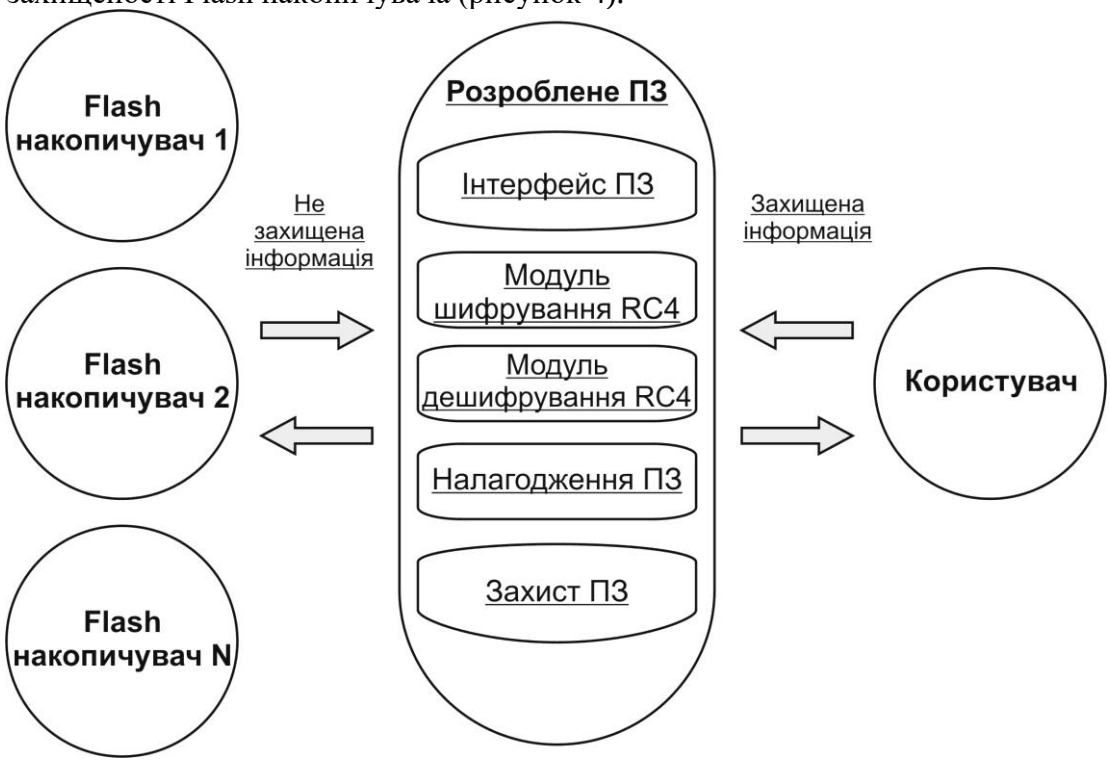


Рисунок 4 – Структурна схема роботи системи

Висновки. У статті теоретичне узагальнення й рішення наукового завдання дослідження методів для забезпечення конфіденційності інформації з’ємних носіїв.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем для забезпечення конфіденційності інформації з’ємних носіїв.

– Досліджена система для забезпечення конфіденційності інформації з’ємних носіїв.

– На основі отриманих результатів досліджень створена програмна реалізація системи для забезпечення конфіденційності інформації з’ємних носіїв.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання для забезпечення конфіденційності інформації з’ємних носіїв.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).
2. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
3. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
4. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
5. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
6. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
7. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
8. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
9. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
10. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
11. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
12. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
13. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
14. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
15. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
16. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).
17. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019. (Scopus).
18. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629. (Scopus).
19. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 873-884. (Scopus).
20. Smirnov, O., Kuznetsov, A., Prokopovych-Tkachenko, D. «Hiding Data in Images Using a Pseudo-Random Sequence». ISCI'2020: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko, Victor A. Krasnobayev and Alexandr A. Kuznetsov. ASC Academic Publishing, USA, 2020. pp. 46-59. – ISBN: 978-1-7362833-0-1 (Hardback), ISBN: 978-1-7362833-1-8 (Ebook).

УДК 004

Є. Кузьменко, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПЕРЕГЛЯДУ ХМАРНИХ СЕРВІСІВ

У статті розроблено програмне забезпечення, яке призначено для системи перегляду хмарних сервісів. Метою розробки є дослідження та програмна реалізація системи перегляду хмарних сервісів. Об'єктом дослідження є процес перегляду хмарних сервісів. Предметом дослідження є методи перегляду хмарних сервісів. Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи перегляду хмарних сервісів. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, хмарні сервіси

Постановка проблеми. Сучасний світ неможливо представити без глобальної мережі Інтернет. Через Інтернет зараз відбувається доступ до різного виду інформації (починаючи від фільмів та музики й закінчуючи новинами та науковими статтями), спілкування між людьми, які розташовані на величезних відстанях один від іншого, покупки в Інтернет-магазинах та ще багато-багато інших функцій. Інтернет надає доступ до різного роду хмарних сервісів.

Але для того, щоб працювати з інтернетом необхідна програма доступу до веб-сторінок, з яких складається Інтернет. Такою програмою є веб-браузер, або, якщо казати по іншому – Інтернет-браузер, як складова системи перегляду хмарних сервісів. Ве́б-оглядач, або браузер, як складова системи перегляду хмарних сервісів – це програмне забезпечення для пошуку, перегляду веб-сайтів, тобто для запиту веб-сторінок (переважно з Інтернету), для їхньої обробки, висновку й переходу від однієї сторінки до іншої. Більшість браузерів також наділені здатностями до перегляду змісту FTP-серверів. Браузери постійно розвивалися із часів зародження Інтернету, і з його ростом ставали усе більше важливою програмою типового персонального комп'ютера. Нині браузер, як складова системи перегляду хмарних сервісів – комплексний додаток для обробки й виведення різних складових веб-сторінки, і для надання інтерфейсу між веб-сайтом і його відвідувачем. Практично всі популярні браузери поширюються безкоштовно або «у комплекті» з іншим додатком: Microsoft Edge (як невід'ємна частина Microsoft Windows), Mozilla Firefox, Opera, Safari (разом з Mac OS або безкоштовно для Windows). Крім цих браузерів існує ще велика кількість інших браузерів, які мають ті або інші функцію, крім основної, перегляду веб-сторінок. Але не дивлячись на таке розмаїття браузерів, немає вітчизняного продукту, який міг би створити конкуренцію цим браузерам. У цьому магістерському проекті зроблена спроба створити такий браузер, як складова системи перегляду хмарних сервісів.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи перегляду хмарних сервісів.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи перегляду хмарних сервісів.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем перегляду хмарних сервісів.
– Дослідження системи перегляду хмарних сервісів.

– Програмна реалізація системи перегляду хмарних сервісів.

Об'єктом дослідження є процес перегляду хмарних сервісів.

Предметом дослідження є методи перегляду хмарних сервісів.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Вимоги висунуті до браузера

Базова ідея Web-технологій полягає в тім, що будь-який користувач будь-якого браузера може підключитися до будь-якому Web-додатку. Однак на перевірку ця потенційно приваблива ідея є фікцією. Універсальний доступ до інформації став можливим лише завдяки титанічним зусиллям розроблювачів додатків для Web, яким довелося пройти дуже складний шлях, перш ніж їхні програми «навчилися» розпізнавати різні версії браузерів і динамічно адаптувати інтерфейс до особливостей цих браузерів. Це кропітка й стомлююча робота, яку доводиться пробробляти щораз після появи нового браузера або чергової його версії. Але ж всі ці зусилля можуть виявитися даремними, коли повсюдне поширення загальноприйнятих стандартів зведе їх нанівець.

Відповідальність за твердження стандартів, що стосуються браузерів, покладена на Консорціум World Wide Web і Європейську асоціацію виробників комп'ютерної техніки (European Computer Manufacturer's Association – ECMA). Незважаючи на те що представники компаній, що займаються розробкою браузерів, беруть участь у діяльності комісій, вони не можуть скільки-небудь істотно вплинути на прийняття остаточних рішень. Адже тут перетинаються інтереси самих різних груп. Розроблювачам браузерів хотілося б робити враження затятих апологетів стандартів – і в той же час мати цілком лояльних клієнтів. Microsoft і Netscape протягом тривалого часу вбудовували у свої продукти всі нові й нові специфічні можливості, намагаючись у такий спосіб залучити розроблювачів. Користувачі ж віддають перевагу найбільш зручному для себе браузеру й вимагають, щоб додатка були сумісні з обраним ними варіантом. Природно, все це жадає від розроблювачів додаткових зусиль, а код Web-додатків непомірно роздувається, оскільки в ньому повинні враховуватися всі розходження між представленими на ринку браузерами.

З випуском кожної нової версії виробники пропонують своє оригінальне трактування стандартів. Якби всі браузери дійсно відповідали стандартам, розроблювачеві досить було б написати всього один інтерфейс свого Web-додатка, і цей інтерфейс виглядав би абсолютно однаково в середовищі будь-якого браузера. Захвати, що супроводжували появу кожної нової версії, через деякий час перемінялися розчаруванням, оскільки користувачі починали зауважувати, що прірва, яка розділяє продукти різних виробників, стає усе ширше.

На щастя, часи міняються. Зокрема, останні версії браузерів компаній Microsoft, Mozilla і Opera Software повністю відповідають вимогам існуючих стандартів. Те ж саме можна сказати й про Konqueror – складову частиниу відомого середовища KDE (K Desktop Environment). Звичайно, повністю гарантувати те, що всі браузери будуть точно відповідати специфікаціям стандартів, ще не можна, але розроблювачі вже переходять до створення Web-додатків, орієнтованих на стандарти, а не на специфічні характеристики конкретних браузерів.

Призначення стандартів

Стандарти браузерів охоплюють два основних аспекти Web-технологій у частині інтерфейсу: його зовнішній вигляд і функціональна реалізація тих або інших можливостей. До стандартів зовнішнього подання екранного інтерфейсу відносяться затверджені W3C специфікації HTML 5 і CSS (Cascading Style Sheets) рівня 3. Мова HTML визначає базову структуру й форматування Web-документа, а каскадні таблиці стилів дозволяють додати до неї точний контроль за атрибутами екранного інтерфейсу: шрифтами, квітами й схемами розташування. Стандарти функціональної реалізації ECMA-262 (або ECMAScript) і запропонована W3C об'єктна модель документа CSS DOM (Document Object Model) описують клієнтські сценарії і їхню можливість динамічно змінювати інформаційне наповнення Web-сторінки.

Специфікації HTML 5, затверджені W3C в 1999 році, описують лише дещицю базових конструкцій, що лежать в основі Web-додатків. Однак незважаючи на розходження, що існують між браузерами різних виробників, всі вони підтримують HTML – якщо й не версії 4.01, то 4.0. У користувача, що має у своєму розпорядженні останню редакцію браузера Opera, Firefox, Microsoft Edge або Konqueror, не повинне виникати ніяких труднощів с переглядом HTML-сторінок. Їхній код не містить специфічних розширень типу керуючих елементів Active або модулів Netscape. Будь-який розроблювач, що бажає встановити, чи відповідає його Web-сторінка вимогам стандартів, має можливість перевірити це за допомогою служби HTML Validator на сайті www.w3c.org. Цей програмний інструментарій ідентифікує всі нестандартні конструкції HTML, які можуть бути оброблені браузерами неправильно.

Формування стилів

Стандарт W3C CSS1 (CSS Level 1) з'явився ще в 1996 році, але виробники браузерів не поспішали підтримувати його у своїх продуктах. У браузерах Netscape до появи шостої версії було дуже багато помилок у підтримці цього стандарту, розроблювачі не забезпечили повної сумісності з CSS1. Корпорація Microsoft реалізувала функції CSS1 тільки у версії Microsoft Edge 6.0. Сьогодні обоє розроблювача повністю підтримують стандарт CSS1. Недавно до браузерів шостого покоління Microsoft і Netscape приєдналися продукти Opera 5 і Konqueror 2.2, розроблювачі яких також заявили про повну сумісність їхнього програмного забезпечення з CSS1. Через п'ять років після твердження специфікацій стандарту нарешті з'явилася повсюдна підтримка CSS1.

Але що це дає розроблювачам Web-додатків? Кожному елементу HTML-коду, у тому числі й відображуваним на екрані (наприклад, заголовкам і гіперпосиланням), ставиться у відповідність певний стиль CSS1. Стандарт CSS1 і модель DOM дозволяють розроблювачам динамічно змінювати властивості поточної сторінки. Підтримка стандарту CSS1 всіма виробниками браузерів стала великою перемогою розроблювачів додатків для Web.

В 1998 році W3C затвердив специфікації стандарту CSS2 (CSS Level 2), у якому підтримка таблиць стилів була поліпшена за рахунок функцій розміщення текстових і графічних елементів у довільному місці сторінки. На щастя, всі виробники браузерів уже оголосили про сумісність майбутніх версій своїх продуктів з CSS2.

Підтримка сценаріїв

Мова Netscape JavaScript, доповнена технологією Microsoft JScript, перетворилася згодом у стандарт ECMA-262. У грудні 1999 року ECMA затвердила третю редакцію стандарту ECMA-262, що розроблялася під найменуванням ECMAScript. Сьогодні всі популярні браузери підтримують ECMAScript, а Microsoft і Netscape оголосили про повну сумісність своїх продуктів із третьою редакцією ECMAScript. Проте розроблювачі як і раніше продовжують використовувати клієнтський код, у якому присутні нестандартні функції JavaScript і JScript. Виробникам варто ретельно вивчити свої клієнтські сценарії й внести зміни в ті з них, де невиправдано використовуються специфічні можливості браузера. Конструкції JavaScript і JScript можна трансформувати в код ECMAScript при мінімальних зусиллях.

Додати до браузера базовий рівень підтримки сценаріїв досить просто. Організувати керування зв'язками між кодом сценаріїв і документом Web складніше, тут браузери найбільшою мірою відрізняються один від іншого. Всі популярні браузери підтримують сьогодні сукупність стандартів DOM1 (DOM Level 1) і DOM2 (DOM Level 2), надаючи за допомогою сценаріїв простий доступ до елементів HTML і властивостям таблиць CSS. В останніх версіях Microsoft Edge, Navigator і Konqueror забезпечена повна підтримка DOM1. А от Opera має тут дуже серйозні обмеження, особливо це стосується можливості додавання й видалення елементів HTML на відображуваній сторінці.

Гарантії безпеки

Відповідність Web-браузерів вимогам стандартів залишає поки бажати кращого, однак уже сьогодні розроблювачі додатків для Web можуть використовувати набір базових

функцій, що буде підтримуватися всіма популярними браузерами. Здається, що першим кроком у будь-якому новому проекті створення додатків для Web (а також у будь-якому проекті, ціль якого полягає в адаптації вже існуючого коду до особливостей стандартів) повинне стати надання клієнтам гарантій технічної підтримки й відновлення ПЗ в майбутньому. Практика показує, що не можна змусити користувачів Navigator перейти на Microsoft Edge, але при цьому більшість охоче встановлюють у себе останню версію улюбленого браузера.

Попередньо необхідно все вивчити й протестувати, щоб виявити області, у яких функції часткової підтримки браузерами специфікацій DOM2 і CSS2 перетинаються.

В ідеалі всі ці стандарти повинні позбавити нас від додаткових зусиль по розпізнаванню браузерів і адаптації до їхніх особливостей. На практиці ж завжди залишаться користувачі, у яких будуть установлені застарілі версії або не настільки широко розповсюджені програми браузерів. Якщо виявиться, що клієнтський браузер, як складова системи перегляду хмарних сервісів не підтримує якісь функції вашого додатка, переадресуйте його на більше простий сайт і повідомите про переваги, які принесе установка оновленого варіанта браузера.

GUI інтерфейс браузера

У сучасному браузері повинні бути реалізовані наступні елементи інтерфейсу GUI**:**

−Адресний рядок.

−Автопошук.

−Вибір протоколу за замовчуванням в адресному рядку.

−Механізм автопідстановки адреси сайту.

−Префікс і суфікс за замовчуванням. За замовчуванням стандартним префіксом є префікс www. а стандартним суфіксом – .com.

−Механізм автозаповнення – браузер, як складова системи перегляду хмарних сервісів перевіряє рядок, що ви вводите, з рядками з галузі реєстру, і якщо знаходить збіги, тоді пропонує вам вибрати їх у списку, що розкривається.

−Панель пошуку.

−Зміна пошукової машини за замовчуванням.

−Меню Виправлення.

−Меню Вид.

−Меню Вибране.

−Меню Сервіс.

−Меню Довідка.

−Спливаючі панелі.

−Вкладка Додатково.

−Кнопка режиму передперегляду.

−Панель команд.

−Контекстне меню браузера.

−Зміна масштабу.

Стандартні функції браузеру

До стандартних функцій сучасного браузеру відносяться наступні:

−перегляд із вкладками;

−здатність зберігати множина вкладок під однією закладкою;

−download менеджер;

−блокувальник спливаючих вікон;

−фішинг-фільтр;

−вбудований RSS-агрегатор;

−підтримку інтернаціональних доменних імен;

−підтримку засобів групової політики;

–підтримку HTML 5;

–підтримку CSS Level 1;

–підтримку XML 1.0;

–підтримку DOM Level 1 і частково CSS Level 2 і DOM Level 2;

–підтримку підключення розширень, що реалізується через об'єктну модель компонентів (COM);

–hotclick;

–переклад тексту на іншу мову;

–перевірка орфографії;

–розпізнавання мишачих жестів;

–функцію блокування рекламних баннеров і флеш-роликів;

–меню швидкого перемикання прокси-серверів зі списку;

–опцію коректування змінної User Agent String, відповідальної за ідентифікацію браузера веб-серверами;

–конвертор сторінок у графічні зображення форматів JPG, BMP, GIF, PNG або TIFF;

–механізм підключення користувальницьких сценаріїв для маніпулювання HTML-контентом на стороні браузера;

–функцію відновлення всіх відкритих вкладок, що збереглися з попередньої робочої сесії браузера;

–опцію Super Drag&Drop, що дозволяє банальним перетаскуванням об'єктів на веб-сторінці виконувати звичні речі: робити швидкий пошук виділеного тексту в заданому за замовчуванням пошуковому сервісі, ефектно зберігати зображення й документи, а також пробляти інші речі спритним рухом мишки;

–кілька готових надбудов і скриптів для скачивания відеороликів з Google Video, Youtube і інших онлайнових сервісів.

Панелі

Функція, пов'язана з переглядом електронної пошти й іншого. Крім того, користувач може завантажити додаткові панелі або створити свою власну.

–Контакти – служить як адресна книга;

–Історія – надає журнал з усіма сторінками, які були відвідані, в хронологічному порядку, починаючи із самої останньої;

–Посилання – перераховані всі посилання, які користувач використовував у період перегляду сайтів;

–Примітки – дозволяє користувачеві копіювати й вставляти вміст зі сторінок в убудований текстовий редактор для наступного перегляду й редагування;

–Інформація – показує інформацію про сторінку, у тому числі її тип MIME, розмір і кодування;

–Вікна – приводиться коротка інформація із всіх вкладок і вікон, відкритим у цей момент.

Перспективи розвитку функцій браузера

До перспективних функцій браузера за результатами дослідження OpenAjax Alliance відносяться наступні:

–Більш повна й послідовна підтримка 2D і векторної графіки, спрямована на забезпечення платформної й вендоро-незалежності.

–Скасування обмеженя у два одночасних підключення до сервера. Воно проявляється, якщо браузер, як складова системи перегляду хмарних сервісів запитує більше 2 сторінок одночасно й серйозно ускладнює роботу Ajax web 2.0 додатків. Зараз браузери, що встановили більше 2 сесій спеціально блокують наступні з'єднання до завершення попередніх.

–Швидкодія HTML DOM операцій. У частині споживання ресурсів, це самі не економічні із всіх JavaScript операцій, що вириваються «уперед» на тисячі відсотків. І тому що вони є одними з найважливіших, необхідно починати роботу з їхньої оптимізації.

Опис розроблюємого Інтернет–браузера

При написанні різних браузерів спостерігається таке явище: основна кількість браузерів роблять на якихось певних движках.

Движок – виділена частина програмного коду для реалізації конкретної прикладної задачі – програма, частина програми, комплекс програм або бібліотека, залежно від задачі й реалізації. Як правило, прикладна частина виділяється із програми для використання в декількох проектах і/або роздільної розробки/тестування.

Використання готового движка при розробці програми, сайту або іншого продукту скорочує час розробки, дозволяє приділити більше часу розробці інших підсистем, наприклад користувальницькому інтерфейсу (або інформаційної наповненості сайту).

Разом з тим продукти, зроблені з використанням движків, успадковують їхні помилки й проблеми безпеки.

Найбільше браузерів зроблено на движку Microsoft Edge, це й не дивно, тому що браузер, як складова системи перегляду хмарних сервісів древній як динозавр, і ще не швидко він вимре, тому що його розвитком і підтримкою займається Майкрософт. На розробку браузерного движку потрібні чималі ресурси, тому програмістам легше робити різні оригінальні примочки й доповнення до движка, іменованому Microsoft Edge, і називати цей браузер, як складова системи перегляду хмарних сервісів уже своїм, оригінальним ім'ям, хоча це все той же Microsoft Edge тільки модифікований як у кращу, так і в гіршу (що буває досить часто) сторону.

У магістерському проекті за основу браузера взято движок Trident. Trident (також відомий як MSHTML) – назва браузерного движка для Windows– версії Microsoft Microsoft Edge. Уперше Trident був реалізований у четвертій версії Microsoft Edge, і з тих пор піддавався постійним поліпшенням і переробкам.

Trident був розроблений як програмний компонент, що дозволяв розроблювачам програмного забезпечення легко додавати можливість перегляду веб-сторінок у їхні власні додатки. Він використовує технологію COM (*компонентну модель об'єктів*) для перегляду й виправлення веб-сторінок у будь-якому оточенні, що підтримує інтерфейс цієї моделі – наприклад, в Delphi, C++ або .NET. Відповідний елемент керування може бути доданий, скажемо, у програму, написану на Delphi – і Trident буде використовуватися для доступу до заздалегідь заданого веб-сторінці, для читання або зміни інформації, що перебуває на ній. Події елемента керування будуть перехоплюватися й передаватися в основну програму. Функції ядра Trident стануть доступні при додаванні бібліотеки mshtml.dll до програмного проекту.

Розробка структурної схеми. Структурна схема розробленої системи зображена на рисунку 1.

Веб-сервер

Веб-сервер – це програмне забезпечення, установлене на комп'ютері, за допомогою якого по протоколу HTTP надається доступ до веб-сторінок. Найпоширенішими веб-серверами є Apache, що працює на платформі Unix/Linux і IIS (Internet Information Service), що працює під керуванням Windows. Також веб-сервером називається й сам комп'ютер, на якому встановлене це програмне забезпечення й зберігаються файли веб-сайтів. Для захисту сервера від атак і взлому звичайно встановлюється мережевий екран і выдповыдне програмне забезпечення.

Функції сервера:

- на сервері зберігаються різні ресурси (наприклад веб-сторінки);
- для кожного ресурсу встановлюються права доступу;
- сервер обробляє запити клієнтів, яким необхідний той або інший ресурс;
- сервер здійснює обмін інформацією із клієнтами й іншими серверами.

Internet-браузер, як складова системи перегляду хмарних сервісів (клієнт)

Розроблений Internet-браузер, як складова системи перегляду хмарних сервісів являться програмою клієнтом.

Функції клієнта:

– браузер, як складова системи перегляду хмарних сервісів здійснює запит необхідного ресурсу;

– браузер, як складова системи перегляду хмарних сервісів обробляє отриманий ресурс;

– у рядку адреси браузера набирається адреса сайту, на який хоче потрапити користувач (наприклад http://www.eucp.com);

– браузер, як складова системи перегляду хмарних сервісів відправляє запит спеціальному комп'ютеру, який зветься DNS-сервер (Domain Name System);

– DNS-сервер перетворить набрану адресу в числову (IP-адреса) адресу сервера, на якому розташований сайт (наприклад 212.147.139.162) і поверне його браузеру;

– браузер, як складова системи перегляду хмарних сервісів відправляє запит на отриману адресу й у відповідь одержує запитуваний ресурс.

– Після того, як ресурс переданий, з'єднання між клієнтом і сервером розривається.

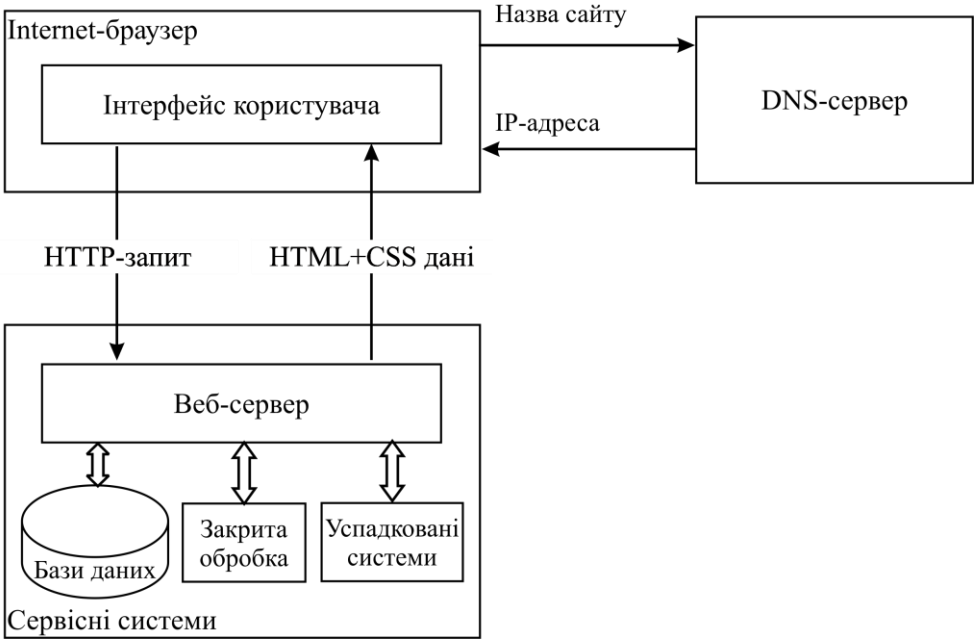


Рисунок 1 – Структурна схема системи

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів перегляду хмарних сервісів. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем перегляду хмарних сервісів; Досліджена система перегляду хмарних сервісів; На основі отриманих результатів досліджень створена програмна реалізація системи перегляду хмарних сервісів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання перегляду хмарних сервісів. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створені програмного забезпечення було використано об’єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Список літератури

1. Смірнова Т.В., Поліщук Л.І., «дослідження хмарних технологій як сервісів для системи інженерних розрахунків» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. С. 106-121.
2. Smirnova, T., Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., «Biometric authentication using convolutional neural networks». Lecture Notes in Networks and Systems, 2021, vol. 152, pp. 85–98. (Scopus).
3. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184. (Scopus).
4. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». CEUR Workshop Proceedings Volume 3156, 2022, Pages 390-399. (Scopus).
5. Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Smirnov O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022. (Scopus)
6. Смірнова Т.В., Дрєєв О.М., Смірнов О.А., «Експертна система оптимізації процесу відновлення та зміцнення поверхонь деталей типу «вал» електродуговим напиленням», Системи управління, навігації та зв’язку, № 2 (54). с. 149-154, 2019.
7. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дрєєв О.М., «Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей», Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.
8. Смірнова Т.В., Дрєєв О.М., Смірнов О.А., Солових Є.К., «Методи оптимізації технологічних процесів відновлення сталевих покриттів», Shipbuilding & marine infrastructure / Суднобудування і морська інфраструктура № 1 (11). с. 48-57, 2019.
9. Smirnova T., Ageev M., Lopata L., Dudan A., Lopata A., «Of combined electric arc coatings», Problems of Tribologyy, Vol. 24 № 3/93. P. 51-61, 2019.
10. Смірнова Т.В., Поліщук Л.І., Смірнов О.А., Буравченко К.О., Макевнін А.О., «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.
11. Смірнова Т.В., «Формалізація та реалізація структури технологічного процесу електродугового напилення для оптимізаційної експертної системи», Технічні науки та технології. № 1(19). С. 104-113. 2020.
12. Smirnova T., Smirnov I., Lopata A., Lopata L., «Improvement of functional properties of gas-thermal coatings by electro-contact treatment», Problems of Tribology, Vol. 25, № 1/95, P. 41-48. 2020.
13. Смірнова Т.В. «формування евристичних правил, бази знань та формалізація структури й правил технологічного процесу для оптимізаційної хмарної інформаційної системи», Системи управління, навігації та зв’язку, № 2 (60). с. 101-104, 2020.
14. Смірнова, Т.В., Смірнов, С.А., Минайленко, Р.М., Доренський, О.П., Сисоєнко С.В., «Хмарна автоматизована система інтелектуальної підтримки прийняття рішень для технологічних процесів». Вісник Черкаського державного технологічного університету. Технічні науки. №4, 2020, С. 84-92.
15. Смірнова Т.В., Пархоменко Д.О., Голубець Р.О., Щербань А.В., Багдасарян Е.К., «Формалізація проблеми підтримки технологічних процесів у хмарних сервісах». Системи озброєння і військова техніка. 2021. № 3(67). С. 105-112.
16. Смірнова Т.В., Столяренко М.П., Янков М.О., Грудік В.В., Моторін Ю.Ю., «Модель реалізації структури технологічного процесу у хмарному сервісі». Збірник наукових праць Харківського національного університету Повітряних Сил. 2021. № 4(70). С. 132-142.
17. Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., Смірнов О.А., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95
18. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Бурмак Ю.А., Оспанова Д.М., «Удосконалений модуль криптографічного захисту інформації в сучасних інформаційно-комунікаційних системах та мережах». Кібербезпека: освіта, наука, техніка. 2021. № 2(14). С. 176-185.
19. Смірнова Т.В., Бурмак Ю.А., Улічев О.С., Усік П.С., Доренський О.П., «Стійка функція шифрування удосконаленого модуля криптографічного захисту інформації в інформаційно-комунікаційних системах» Кібербезпека: освіта, наука, техніка. 2021. № 1(13). С. 183-201.
20. Смірнова Т.В., Моторін Ю.Ю., Буравченко К.О., Бочуля Т.В., Коваленко О.В. «Вибір оптимальної технології побудови хмарної інформаційно-комунікаційної системи автоматизації виробничих процесів». Вимірювальна та обчислювальна техніка в технологічних процесах, № 1 (2022). С. 15-26. 2022.

УДК 004

О.Кульчицький, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ РЕАГУВАННЯ НА ВИНИКНЕННЯ НАДЗВИЧАЙНИХ СИТУАЦІЙ З ЗАСТОСУВАННЯМ ХМАРНИХ ТЕХНОЛОГІЙ

У статті розроблено програмне забезпечення, яке призначено для системи реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій. Метою розробки є дослідження та програмна реалізація системи реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій. Об'єктом дослідження є процес реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій. Предметом дослідження є методи реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій. Методи дослідження базуються на методах хмарних обчислень, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, надзвичайні ситуації, хмарні технології

Постановка проблеми. Сучасне суспільство, на жаль, доволі часто являється свідком різного виду та типу аварійних та надзвичайних ситуацій. Особливості керування в надзвичайних ситуаціях природного й техногенного характеру передбачені відповідними законами й положеннями.

Для того, щоб мати можливість вести облік, накоплювати інформацію, наводити статистичні дані, і врешті решт, надавати аналітичні звіти та прогнози, необхідно, щоб було якесь сховище даних, де зберігалися би дані про аварійні ситуації. Вочевидь, що така система повинна бути розгалуженою, і будуватися на основі локальних та глобальних комп'ютерних мереж. На даних час дуже широко розповсюджена мережа Інтернет, яка є глобальною, та за допомогою якої дуже зручно збирати, зберігати та візуалізувати інформацію різного плану та типу. Відповідно у магістерському проекті необхідно реалізувати систему зберігання інформації у мережі Інтернет, призначену для роботи з даними про аварійні ситуації.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

– Дослідження системи реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

– Програмна реалізація системи реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

Об'єктом дослідження є процес реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

Предметом дослідження є методи реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

Методи дослідження базуються на методах хмарних обчислень, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Швидке та ефективне реагування на надзвичайні ситуації – це виклик, з яким громади завжди стикалися. Керівники громадської охорони здоров'я можуть створювати інноваційні стратегії управління надзвичайними ситуаціями, які включають передові технології, щоб підвищити шанси на ефективну та ефективну відповідь, незалежно від кризи.

Технологія може спростити для служб реагування оцінку загроз, обмін інформацією та планування реагування на надзвичайні ситуації. Деякі з останніх інновацій у реагуванні на надзвичайні ситуації готові революціонізувати спосіб, у який групи реагування аналізують події та координують свою діяльність, тоді як інші кардинально змінюють те, як звичайні громадяни реагують на надзвичайні ситуації.

Федеральне агентство з управління надзвичайними ситуаціями (FEMA) визначає управління надзвичайними ситуаціями як «управлінську функцію, покликану створити структуру, в рамках якої громади зменшують вразливість до небезпек і справляються з катастрофами».

Управління надзвичайними ситуаціями є аспектом громадського здоров'я. Хоча нею може керувати федеральний уряд, агенції на рівні округу та міста також відіграють ключову роль. Згідно з FEMA, управління в надзвичайних ситуаціях складається з чотирьох окремих етапів:

– Пом'якшення наслідків – пом'якшення наслідків включає проактивні дії для запобігання або зменшення впливу та наслідків надзвичайних ситуацій. Це можуть бути фізичні дії, як-от запобігання потенційному збитку від води шляхом копання каналів чи спорудження дамб, або це можуть бути фінансові чи юридичні дії, як-от придбання страхових полісів.

– Готовність – бути підготовленим означає заздалегідь планувати всі типи надзвичайних ситуацій і навчати громади та окремих людей щодо того, що робити, якщо така станеться. Цей крок може включати проведення тренувань і розробку планів із переліком кроків, які необхідно вжити під час катастрофи.

– Реагування – Етап реагування включає всі кроки, які вживаються відразу після надзвичайної ситуації або катастрофічної події. Це може включати впровадження планів реагування, застосування тактики пошуку та порятунку та вжиття захисних заходів.

– Відновлення – відновлення – це процес відновлення нормального функціонування після катастрофи чи надзвичайної ситуації. Цей крок, який часто може розтягнутися на довгі періоди часу, може включати мінімізацію фінансових проблем, відновлення пошкоджених структур і усунення вразливостей, щоб уникнути майбутніх катастроф.

Метою управління надзвичайними ситуаціями є виявлення потенційної вразливості громади до стихійних лих і розробка стратегій, які забезпечують захист громади та зміцнюють її стійкість. Досягнення цієї мети може допомогти громадам мінімізувати збитки та максимізувати функціональність у разі надзвичайної ситуації.

Приклади надзвичайних ситуацій та технологічні рішення

Хоча всі форми управління надзвичайними ситуаціями мають головну мету захистити громаду, самі надзвичайні ситуації відрізняються за масштабами та впливом. Це означає, що керівники громадської охорони здоров'я повинні розуміти, як реагувати на кожен тип надзвичайної ситуації, щоб оптимізувати свій вплив після катастрофи. Приклади надзвичайних ситуацій включають:

– Стихійні лиха – це повені, урагани, торнадо, лісові пожежі та землетруси.

– Кризи громадського здоров'я – це епідемії, пандемії та інші спалахи захворювань, починаючи від нових штамів грипу і закінчуючи новими потенційно смертельними

захворюваннями, такими як COVID-19.

– Терористичні акти – вони можуть варіюватися від погроз вибуху до погроз або використання біологічно небезпечних матеріалів .

– Радіаційні та хімічні надзвичайні ситуації – вони включають випадковий викид радіоактивних матеріалів або інших токсичних речовин, які можуть завдати великої шкоди у визначеній території.

Види стихійних лих

Стихійні лиха – це потужні й часто непередбачувані сили природи, які можуть завдати шкоди громадам, економікам та екосистемам. Ці події можуть завдати величезної шкоди та мати далекосяжні наслідки, впливаючи на різні аспекти суспільства та критичну інфраструктуру:

Землетруси: землетруси є результатом руху тектонічних плит під поверхнею Землі. Вони можуть завдати значної шкоди будівлям, мостам і транспортним мережам. Критична інфраструктура, така як електромережі, системи водопостачання та комунікаційні мережі, може бути серйозно порушена внаслідок сейсмічної активності.

Урагани та тайфуни: ці потужні тропічні шторми можуть спричинити руйнівні вітри, сильні опади та штормові хвилі, що призведе до широкомасштабних повеней та руйнування вітром. Прибережні регіони особливо вразливі, оскільки така основна інфраструктура, як електростанції та каналізаційні системи, вразлива до повеней і штормів.

Повені: повені можуть статися внаслідок сильних опадів, штормових хвиль або розливу річок і озер. Критична інфраструктура, така як дамби, дамби та очисні споруди, може бути перевантажена, що призведе до забруднення водопостачання та повсюдних збоїв.

Лісові пожежі: лісові пожежі часто посилюються через посуху та можуть охопити величезні площі землі. Вони становлять значну загрозу для ліній електропередач, веж зв'язку та транспортних мереж, перешкоджаючи зусиллям з реагування на надзвичайні ситуації.

Торнадо: торнадо – це сильні шторми, які можуть зруйнувати будівлі, вивести з ладу лінії електропередач і порушити транспортні системи. Їх особливо складно передбачити та підготуватися до них через їх швидке формування.

Виверження вулканів: виверження вулканів можуть призвести до викиду попелу, лави та пірокластичних потоків. Вони можуть пошкодити інфраструктуру, як-от дороги, аеропорти та комунальні послуги, а також порушити авіасполучення.

Вплив на критичну інфраструктуру

Стихійні лиха можуть завдати серйозної шкоди різним критичним компонентам інфраструктури, порушуючи основні послуги та створюючи загрозу громадській безпеці. Ці інфраструктурні системи, які є основою сучасного суспільства, охоплюють широкий спектр секторів і включають наступне:

Електричні мережі

Електромережі є джерелом життя сучасного суспільства, вони забезпечують електроенергією будинки, підприємства та основні послуги. Коли відбуваються стихійні лиха, вони можуть порушити роботу електромереж, що призведе до масових відключень. Ці відключення не тільки створюють незручності для населення, але й мають серйозні наслідки для екстрених служб, які покладаються на постійне електропостачання. Забезпечення стійкості енергетичної інфраструктури за допомогою таких заходів, як посилення ліній електропередач і резервних генераторів, має вирішальне значення для ефективного реагування на катастрофи та швидкого відновлення.

Водопостачання та очищення

Стихійні лиха, такі як повені, можуть поставити під загрозу системи водопостачання, забруднюючи воду та роблячи її непридатною для пиття. Цей дефіцит чистої води може швидко перерости в гуманітарну кризу, що призведе до поширення захворювань, що передаються через воду, і погіршить надзвичайну ситуацію. Для пом'якшення цих ризиків необхідна надійна інфраструктура водопостачання та очищення. Це включає стійкі до повеней водні споруди, резервні системи очищення та запаси чистої води для задоволення

потреб постраждалих громад.

Транспортні мережі

Транспортні мережі, включно з дорогами, мостами, залізницями та аеропортами, мають вирішальне значення для переміщення товарів, служби екстреного реагування та евакуації постраждалого населення під час стихійних лих. Пошкодження цих мереж може серйозно перешкодити зусиллям з реагування на катастрофи та сповільнити відновлення. Щоб підвищити стійкість транспортної інфраструктури, такі заходи, як покращені дренажні системи, зміцнені мости та заздалегідь розміщене обладнання для екстреної допомоги, можуть допомогти забезпечити доступність основних маршрутів під час та після стихійних лих.

Системи зв'язку

Ефективна комунікація є наріжним каменем реагування на катастрофи та відновлення. Порушені комунікаційні мережі можуть перешкоджати координації між екстреними службами, ускладнюючи розгортання ресурсів і реагування на ситуації, що змінюються. Крім того, постраждалим людям може бути важко звернутися за допомогою або отримати доступ до важливої інформації. Резервні системи зв’язку, резервні джерела живлення та стійка до катастроф інфраструктура необхідні для підтримки ефективного зв’язку під час криз.

Заклади охорони здоров'я

Лікарні та медичні заклади відіграють важливу роль у реагуванні на катастрофи, надаючи медичну допомогу тим, хто її потребує. Однак вони не застраховані від стихійних лих. Під час таких подій заклади можуть зазнати пошкоджень або відключень електроенергії, що обмежить їхню здатність надавати основні медичні послуги. Готовність закладів охорони здоров’я до стихійних лих включає такі заходи, як структурне посилення, резервні системи живлення та чітко визначені плани евакуації, щоб забезпечити безперервність надання допомоги під час надзвичайних ситуацій.

Критичні промислові об'єкти

Критичні промислові об’єкти, включаючи фабрики, нафтопереробні заводи та хімічні заводи, можуть становити значну небезпеку для навколишнього середовища та здоров’я людей, якщо вони пошкоджені або скомпрометовані під час стихійних лих. Розливи хімікатів, вибухи та інші промислові аварії можуть ще більше ускладнити реагування на катастрофи та відновлення. Забезпечення стійкості цих об’єктів передбачає суворі протоколи безпеки, оцінку небезпеки та заходи стримування, щоб запобігти забрудненню навколишнього середовища та зменшити ризики для навколишніх громад.

Створення стійкості до катастроф і пом’якшення наслідків

Профілактичні заходи та стратегічне планування є нашим найсильнішим захистом від непередбачуваних сил природи. Давайте розглянемо низку стратегій, які можуть допомогти захистити нашу життєво важливу інфраструктуру та забезпечити більш стійку реакцію, коли стається лихо.

Плани стійкості до катастроф, готовності та реагування

Розробка комплексних планів готовності до стихійних лих і реагування на них має важливе значення для мінімізації впливу стихійних лих на критичну інфраструктуру. Це включає встановлення шляхів евакуації, накопичення необхідних запасів і проведення регулярних навчань.

Такі ініціативи, як EARTHEX , міжгалузеві глобальні навчання зі стійкості до катастроф, відіграють важливу роль у досягненні цієї мети. EARTHEX об’єднує світових лідерів, експертів і зацікавлених сторін з різних секторів, щоб заохочувати міжсекторальне співробітництво та розробку інноваційних рішень, які забезпечать майбутнє наших критичних інфраструктур.

EARTHEX служить платформою для обміну знаннями та найкращими практиками, сприяння міжнародній співпраці та підвищення стійкості до катастроф у глобальному масштабі. Моделюючи сценарії катастроф і тестуючи стратегії реагування, EARTHEX не

тільки підвищує обізнаність, але й допомагає виявити прогалини в готовності та можливостях реагування. Цей спільний підхід гарантує, що ми зможемо адаптувати та зміцнити наші критично важливі інфраструктурні системи, щоб протистояти викликам, створеним гнівом матері-природи. Роблячи це, ми можемо покращити нашу здатність захищати життя, майно та основні послуги, на які покладаються наші громади.

Стійкість інфраструктури

Створення та підтримка критичної інфраструктури з урахуванням стійкості може допомогти їй протистояти стихійним лихам. Це включає використання матеріалів і конструкцій, стійких до землетрусів, повеней та інших небезпек.

Системи резервування та резервування

Впровадження резервування в критично важливу інфраструктуру, таку як резервне електропостачання та системи зв'язку, може допомогти підтримувати основні служби під час катастроф.

Системи раннього попередження

Інвестиції в системи раннього попередження обіцяють забезпечити громади та операторів критичної інфраструктури необхідним часом для ретельної підготовки та ефективного реагування на різні стихійні лиха. Однак у цьому підході є суттєве застереження – властива непередбачуваність стихійних лих, що може зробити системи раннього попередження менш надійними.

Тим не менш, є срібна підкладка. Інноваційна технологія стійкості до катастроф відкриває двері для підвищення ефективності наших заходів із реагування та відновлення, навіть незважаючи на цю непередбачуваність.

Саме тут модель оптимізації мережі глобальної інфраструктури (GINOM) виходить на перший план. Серед хаосу після катастрофи GINOM стоїть як маяк надії. Він може похвалитися можливостями «цифрового двійника» в режимі реального часу та прогнозним моделюванням, надаючи безцінні вказівки операторам, що охоплюють багато секторів, щодо того, як ефективно орієнтуватися в складних збоях інфраструктури.

Особливий підхід GINOM робить сильний акцент на функціональному взаємозв'язку, надаючи користувачам важливу ситуаційну обізнаність і підтримку прийняття рішень, необхідну для виконання критично важливих дій з точністю та ефективністю.

Ретельний розвиток міст і планування землекористування

Належне планування землекористування може допомогти уникнути будівництва в зонах високого ризику, схильних до стихійних лих, зменшивши вплив критичної інфраструктури.

Інформування та освіта громадськості

Щоб звести до мінімуму жертви та пошкодження критично важливої інфраструктури під час катастроф, освіта громадськості про заходи реагування на катастрофи має вирішальне значення. Крім цього, не менш важливо розпалити в нашій молоді пристрасть до стійкості до стихійних лих . Таким чином ми забезпечуємо передачу наших знань майбутнім поколінням. Не кажучи вже про те, що ми використовуємо їх енергію, інновації та відданість, щоб побудувати більш стійке та стійке майбутнє для всіх.

Незалежно від типу надзвичайної ситуації, посадові особи охорони здоров'я можуть підготуватися до них, створивши інноваційні стратегії управління надзвичайними ситуаціями, які включають технології, які допомагають ефективніше розгортати ресурси та полегшують людям звертатися за допомогою.

Доступ зі смартфона

Набрати номер має бути простим, і це те, що діти повинні навчитися робити якомога раніше у разі надзвичайної ситуації. Незважаючи на те, що смартфони та технології покращили та спростили багато типів реагування на надзвичайні ситуації, вони, можливо, ускладнили ситуацію, коли йдеться про ці три числа.

Не всі інновації такі елегантні та прості, як можна було б очікувати. Яскравим прикладом є смартфони. Хоча вони еволюціонували, щоб включити розширені функції

безпеки, і тепер вони зобов'язані за законом надавати доступ навіть без SIM-карти, вони також встановили деякі додаткові бар'єри для дітей, яким потрібен доступ до екстрених служб. Переконайтеся, що всі у вашій родині знають, як отримати доступ до екстреної клавіатури на вашому смартфоні. Якщо у вас є пароль на телефоні, ви можете використовувати екстрені налаштування, щоб обійти його та зробити критично важливі дзвінки, включно з викликами. Покажіть своїм дітям, як працює ця технологія, щоб ця інновація не сповільнювала їх роботу.

Автоматизовані системи захисту від лісових пожеж

За даними Національного міжвідомчого координаційного центру (NICC), у 2021 році 58 985 лісових пожеж спалили понад 7,1 мільйона акрів землі по всій країні. Ці жахливі цифри залишаються незмінними протягом останніх 10 років. Зростаюча загроза зміни клімату робить боротьбу з цими катастрофами – і роботу над пом'якшенням їхнього впливу до того, як вони виникнуть – критично важливою.

Для боротьби з лісовими пожежами використовується кілька технологій. Потенційні рішення включають використання штучного інтелекту для аналізу метеорологічних супутникових зображень, щоб виявити джерело пожежі до того, як її стане важко контролювати. Інші системи використовують прогнозну аналітику, щоб визначити, куди може поширитися пожежа, що загорілася, що дозволяє пожежникам ефективніше її локалізувати. Безпілотники можна запускати в небезпечні зони для збору даних про пожежі, не наражаючи людей на небезпеку. Безпілотники також можна використовувати для запуску «прописаних опіків»: невеликих вогнів, навмисно розміщених на шляху вогню, що вторгається, щоб уповільнити його розвиток.

Геоінформаційні системи

Геоінформаційні системи (ГІС) є надзвичайно цінними для служб реагування на надзвичайні ситуації. Ця технологія аналітичної картографії допомагає їм зрозуміти, де розташовані небезпеки, скільки людей постраждали та які дії потрібні. Відповідно до FEMA, технологію ГІС можна застосовувати на всіх чотирьох етапах управління надзвичайними ситуаціями. Можливості глибокого моделювання ГІС можуть допомогти керівникам охорони здоров'я створювати прогнозні та оціночні дані, що може призвести до більш ефективних цілеспрямованих стратегій готовності до надзвичайних ситуацій і відновлення.

Інтелектуальні вуличні ліхтарі

Хоча вуличні ліхтарі можуть здатися малоймовірним інструментом для управління надзвичайними ситуаціями та реагування на них, інтелектуальне вуличне освітлення надає, здавалося б, звичайному освітленню нову потужну функцію. Ці вуличні ліхтарі оснащені датчиками навколишнього середовища, які виявляють такі небезпеки, як підвищення рівня води, сильний вітер, високі температури та смертоносний газ. Вони також можуть бути оснащені 180-градусними камерами, які пропонують у режимі реального часу спостерігати за рухом пішоходів і ситуаціями, що розвиваються.

За допомогою розумних динаміків вуличних ліхтарів і кольорових ліхтарів команди управління надзвичайними ситуаціями можуть передавати важливу інформацію громадянам у цьому районі, наприклад безпечні шляхи евакуації.

Завдяки цій інновації в управлінні надзвичайними ситуаціями служби реагування можуть збирати важливу інформацію про загрози та спілкуватися з громадянами на вуличному рівні, перш ніж вони зможуть прибути на місце події.

Програми екстреного зв'язку

Широке поширення смартфонів робить ці пристрої ідеальним засобом для передачі інформації про заходи реагування на надзвичайні ситуації широкому загалу. Бачачи потенціал цієї тенденції, FEMA розробило додаток, який допомагає передавати важливу інформацію групам ризику.

За допомогою програми FEMA користувачі можуть отримувати звіти про погоду, знаходити притулки для надзвичайних ситуацій і навіть реєструватися для допомоги в разі лиха. Додаток пропонує поради щодо дій у понад 20 типах надзвичайних ситуацій і

катастроф.

Надзвичайні ситуації та катастрофи зазвичай трапляються без попередження. Ключем до ефективного реагування на ці надзвичайні ситуації у сфері охорони здоров'я є створення узгодженої стратегії, яка зосереджена на пом'якшенні наслідків, підготовці, реагуванні та відновленні рівною мірою. Технології можуть відіграти важливу роль у забезпеченні безпеки людей.

Розробка структурної схеми

На рисунку 1 зображена структурна схема системи.

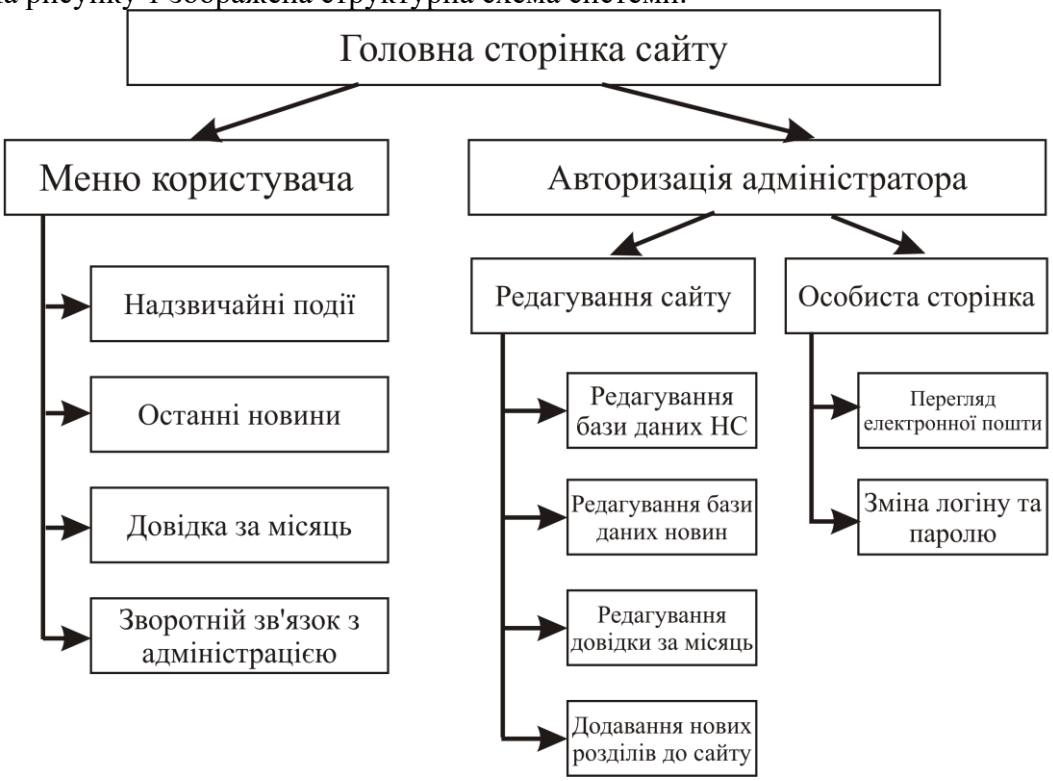


Рисунок 1 – Структурна схема системи

З рисунку видно, що сайт має наступну структуру: з головної сторінки можна зайти у меню користувача, або авторизуватися як адміністратор.

Меню користувача має наступні розділи:

- Надзвичайні події.
- Останні новини.
- Довідка за місяць.
- Зворотній зв'язок з адміністрацією.

Якщо відбулася авторизація адміністратора, то на екрані з'являється вхід до редагування сайту та вхід на особисту сторінку.

Редагування сайту включає в себе наступні розділи:

- Редагування бази даних надзвичайних подій.
- Редагування бази даних новин.
- Редагування довідки за місяць.
- Додавання нових розділів до сайту.
- Особиста сторінка містить електронну пошту адміністратора, та вхід на сторінку зміни логіну та паролю.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

– Досліджена система реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

– На основі отриманих результатів досліджень створена програмна реалізація системи реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання реагування на виникнення надзвичайних ситуацій з застосуванням хмарних технологій.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
2. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
3. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
4. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
5. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
6. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
7. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
8. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
9. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
10. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
11. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
12. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).
13. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019. (Scopus).
14. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629. (Scopus).
15. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 873-884. (Scopus).
16. Smirnov, O., Kuznetsov, A., Prokopovych-Tkachenko, D. «Hiding Data in Images Using a Pseudo-Random Sequence». ISCI'2020: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko, Victor A. Krasnobayev and Alexandr A. Kuznetsov. ASC Academic Publishing, USA, 2020. pp. 46-59. – ISBN: 978-1-7362833-0-1 (Hardback), ISBN: 978-1-7362833-1-8 (Ebook).

17. Smirnov, O., Kuznetsov, A., Shekhanin, K., Chepurko, I. Detecting Hidden Information in FAT. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 412-429. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
18. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
19. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х.: ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
20. Смірнов О.А., Дрєєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х.: Вид. Рожко С.Г. 2019. С. 123-139.

УДК 004

І.Мицак, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ДАНИХ У ХМАРНИХ СЕРВІСАХ

У статті розроблено програмне забезпечення, яке призначено для системи забезпечення цілісності даних у хмарних сервісах. Метою розробки є дослідження та програмна реалізація системи забезпечення цілісності даних у хмарних сервісах. Об'єктом дослідження є процес забезпечення цілісності даних у хмарних сервісах. Предметом дослідження є методи забезпечення цілісності даних у хмарних сервісах. Методи дослідження базуються на методах забезпечення цілісності даних, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи забезпечення цілісності даних у хмарних сервісах. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, цілісність даних, хмарні сервіси

Постановка проблеми. Сучасні хмарні сервіси – це ефективний засіб зберігання й передачі даних (у них найбільш низька удільна вартість зберігання інформації), потреби ринку дуже великі, і в продаж найчастіше надходять хмарні сервіси, що не відповідають закладеним у формат стандартам. Виробники допускають таке зниження якості лише тому, що коригувальна здатність, закладена в хмарних сервісах досить велика, і навіть не дуже якісний хмарний сервіс, швидше за все, буде й записуватися й зчитуватися, нехай і на знижених швидкостях.

Але основною проблемою такої "другосортності" є те, що "запас міцності" системи корекції помилок хмарних сервісів при роботі з ними вкрай низький, і будь-яка, навіть незначна подряпина на диску серверу хмарного сервісу може ушкодити дані. Імовірність збою збереженого файлу прямо пропорційна кількості займаних їм секторів, і якщо файл – це архів, що займає весь хмарний сервіс, для надійного архівного зберігання інформації потрібно дуже якісний носій.

Існує цілий ряд методів, які дозволяють вирішити цю задачу. Але посеред цього ряду особливо стоять методи завадостійкого кодування, які дозволяють, навідь при досить сильному ушкоджені тексту відтворити його.

Посеред методів завадостійкого кодування в останній час дуже активно розвиваються методи, які засновані на використанні циклічних кодів. Беззаперечною перевагою цих методів є дуже велика здатність виправляти помилки в ушкодженому тексті. Саме до таких

методів й відноситься метод Ріда-Соломона. Вони, у цей час широко використовуються в системах відновлення даних з компакт-дисків, при створенні архівів з інформацією для відновлення у випадку ушкоджень, у завадостійкому кодуванні.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи забезпечення цілісності даних у хмарних сервісах.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи забезпечення цілісності даних у хмарних сервісах.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем забезпечення цілісності даних у хмарних сервісах.
– Дослідження системи забезпечення цілісності даних у хмарних сервісах.
– Програмна реалізація системи забезпечення цілісності даних у хмарних сервісах.

Об'єктом дослідження є процес забезпечення цілісності даних у хмарних сервісах.

Предметом дослідження є методи забезпечення цілісності даних у хмарних сервісах.

Методи дослідження базуються на методах забезпечення цілісності даних, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Опис коду Ріда-Соломона

Коди Ріда-Соломона – недвійкові циклічні коди, що дозволяють виправляти помилки в блоках даних. Елементами кодового вектора є не біти, а групи біт (блоки). Дуже поширені коди Ріда-Соломона, що працюють із байтами (октетами).

У цей час широко використовується в системах відновлення даних з компакт-дисків, при створенні архівів з інформацією для відновлення у випадку ушкоджень, у завадостійкому кодуванні.

Код Ріда-Соломона був винайдений в 1960 році співробітниками лабораторії Лінкольна Массачуссетского технологічного інституту Ірвином Рідом і Густавом Соломоном. Ідея використання цього коду була представлена в статті «Polynomial Codes over Certain Finite Fields». Перше застосування код Ріда-Соломона одержав в 1982 році в серійному випуску компакт-дисків. Ефективний алгоритм декодування був запропонований в 1969 році Елвином Берлекемпом і Джеймсом Мессі.

Формальний опис

Коди Ріда-Соломона є важливим частковим випадком БЧХ-коду, корінь полінома, що породжує, якого лежать у тім же полі, над яким і будується код ($m = 1$).

Коди Боуза-Чоудхури-Хоквингхема (БЧХ коди) – у теорії кодування це широкий клас циклічних кодів, застосовуваних для захисту інформації від помилок. Відрізняється можливістю побудови коду із заздалегідь певними коригувальними властивостями, а саме, мінімальною кодовою відстанню.

Поліном, що породжує

Визначення поліномом, що породжує, циклічного (n,k) *коду* C називається такий ненульовий $g(x) = \sum_{i=0}^{r} g_i x^i$ поліном з C, ступінь якого найменша й коефіцієнт при старшому *ступені* $g_r = 1$.

Теорема 1

Якщо C – циклічний (n,k) код і $g(x)$ – його поліном, що породжує, тоді *ступінь* $g(x)$ *дорівнює* $r = n - k$ і кожне кодове слово може бути єдиним чином представлено у вигляді $c(x) = m(x)g(x)$, де ступінь $m(x)$ менше або дорівнює $k - 1$.

Теорема 2

$g(x)$ – поліном, що породжує, циклічного (n,k) коду є дільником *двочлена* $x^n - 1$

Наслідок: у такий спосіб як поліном, що породжує, можна вибирати будь-який поліном, дільник $x^n - 1$. Ступінь обраного полінома буде визначати кількість перевірочних символів r, число інформаційних символів $k = n - r$.

Матриця, що породжує

Поліноми $g(x), xg(x), x^2g(x), \ldots, x^{k-1}g(x)$ лінійно незалежні, інакше $m(x)g(x) = 0$ при ненульовому $m(x)$, що неможливо.

Значить кодові слова можна записувати, як і для лінійних кодів, наступним чином:

$$\overline{m}G = (m_0, m_1, ..., m_{k-1}) \begin{bmatrix} g(x) \\ xg(x) \\ ... \\ x^{k-1}g(x) \end{bmatrix} = m(x)g(x), \quad (1)$$

де G є *матрицею, що* породжує, $m(x)$ – інформаційним поліномом.

Матрицю G можна записати в символьній формі:

$$G = \begin{bmatrix} g_0 & g_1 & \ldots & g_{r-1} & g_r & 0 & \ldots & 0 \\ 0 & g_0 & \ldots & g_{r-2} & g_{r-1} & g_r & \ldots & 0 \\ . & . & \ldots & . & . & . & \ldots & . \\ 0 & 0 & \ldots & 0 & g_0 & g_1 & \ldots & g_r \end{bmatrix}$$

Перевірочна матриця

Для кожного кодового слова циклічного коду справедливо $c(x) = 0 \mod g(x)$. Тому *перевірочну матрицю* можна записати як $H = \begin{bmatrix} 1 & x & x^2 & \ldots & x^{n-2} & x^{n-1} \end{bmatrix} \mod g(x)$.

Тоді:

$$\overline{c}H^T = \sum_{i=0}^{n-1} c_i x^i \bmod g(x). \quad (2)$$

Нехай α – елемент поля $GF(q)$ порядку n. Якщо α – *примітивний* елемент, то його порядок дорівнює q-*1*, т.е. $\alpha^{q-1}=1$, $\alpha^i \neq 1$, $0 < i < q - 1$.

Тоді нормований поліном $g(x)$ мінімального ступеня над полем $GF(q)$, кореннями якого є $d - 1$ підряд, що йдуть ступенів, $\alpha^{l_0}, \alpha^{l_0+1}, ..., \alpha^{l_0+d-1}$ елемента α, є поліномом, що породжує, коду над *полем* GF(q) $g(x) = (x - \alpha^{l_0})(x - \alpha^{l_0+1}) \ldots (x - \alpha^{l_0+d-1})$., де l_0 – деяке ціле число (у тому числі 0 і 1), за допомогою якого іноді вдається спростити кодер. Звичайно покладається $l_0 = 1$. Ступінь багаточлена $g(x)$ дорівнює $d - 1$.

Довжина отриманого коду n, мінімальна відстань d (мінімальна відстань d лінійного коду є мінімальним із всіх відстаней Хеммінга всіх пар кодових слів). Код містить $r = d - 1 = \deg(g(x))$ перевірочний символ, де deg() позначає ступінь полінома; число інформаційних символів $k = n - r = n - d + 1$. У такий спосіб $d = n - k - 1$ і код Ріда-Соломона є *роздільним кодом з максимальною відстанню* (є оптимальним у змісті границі Синглтона).

Кодовий поліном $c(x)$ може бути отриманий з інформаційного полінома $m(x)$, $\deg m(x) \leq k - 1$, шляхом перемножування $m(x)$ і $g(x)$: $c(x) = m(x)g(x)$

Властивості

Код Ріда-Соломона над $GF(q^m)$, що виправляє t помилок, вимагає $2t$ перевірочних символів і з його допомогою виправляються довільні пакети довжиною t і менше. Відповідно до теореми про границю Рейгера, коди Ріда-Соломона є оптимальними з погляду співвідношення довжини пакета й можливості виправлення помилок – використовуючи $2t$ додаткових перевірочних символів виправляються t помилок (і менш).

Теорема (границя Рейгера). Кожний лінійний блоковий код, що виправляє всі пакети довжиною t і менш, повинен містити щонайменше $2t$ перевірочних символів.

Виправлення багаторазових помилок

Код Ріда-Соломона є одним з найбільш потужних кодів, що виправляють багаторазові пакети помилок. Застосовується в каналах, де пакети помилок можуть утворюватися настільки часто, що їх уже не можна виправляти за допомогою кодів, що виправляють одиночні помилки. $(q^m - 1, q^m - 1 - 2t)$-код Ріда-Соломона над полем $GF(q^m)$ з кодовою відстанню $d = 2t + 1$ можна розглядати як $((q^m - 1)m, (q^m - 1 - 2t)m)$-код над полем $GF(q)$, що

може виправляти будь-яку комбінацію помилок, зосереджену в t або меншому числі блоків з m символів.

Найбільше число блоків довжини m, які може торкнути пакет довжини l_i, де $l_i <= mt_i - (m - 1)$, не перевершує t_i, тому код, що може виправити t блоків помилок, завжди може виправити й будь-яку комбінацію з p пакетів загальної довжини l, якщо $l + (m - 1) <= mt$.

Практична реалізація

Кодування за допомогою коду Ріда-Соломона може бути реалізовано двома способами: систематичним і несистематичним.

При несистематичному кодуванні інформаційне слово множиться на якийсь полином, що неприводиться, у полі Галуа. Отримане закодоване слово повністю відрізняється від вихідного й для добування інформаційного слова потрібно виконати операцію декодування й уже потім можна перевірити дані на зміст помилок. Таке кодування вимагає більші витрати ресурсів тільки на добування інформаційних даних, при цьому вони можуть бути без помилок.

При систематичному кодуванні до інформаційного блоку з k символів приписуються $2t$ перевірочних символів, при обчисленні кожного перевірочного символу використовуються всі k символів вихідного блоку.

У цьому випадку немає витрат ресурсів при добуванні вихідного блоку, якщо інформаційне слово не містить помилок, але кодер/декодер повинен виконати $k(n - k)$ операцій додавання й множення для генерації перевірочних символів. Крім того, тому що всі операції проводяться в поле Галуа, те самі операції кодування/декодування вимагають багато ресурсів і часу. Швидкий алгоритм декодування, заснований на швидкому перетворенні Фур'є, виконується за час порядку lnn^2.

Кодування

При операції кодування інформаційний поліном множиться на багаточлен, що породжує. Множення вихідного слова S довжини k на не приводиться полином, що, при систематичному кодуванні можна виконати в такий спосіб:

–До вихідного слова приписуються $2t$ нулів, виходить поліном $T = Sx^{2t}$.

–Цей поліном ділиться на поліном, що породжує, G, перебуває залишок R, $Sx^{2t} = QG + R$, де Q – частка.

–Цей залишок й буде коригувальним кодом Ріда-Соломона, він приписується до вихідного блоку символів. Отримане кодове слово $C = Sx^{2t} + R$.

Кодер будується зі регістрів зсуву, суматорів і перемножувачів. Регістр зсуву складається з комірок пам'яті, у кожній з яких перебуває один елемент поля Галуа.

Наведений як приклад кодер Ріда-Соломона генерує 16 коригувальних байт, що дозволяє виправляти до 8 і виявляти до 16 помилок у кадрі даних.

Перемножувачі на константи $GF(0)...GF(15)$ у полі Галуа реалізуються в такий спосіб: спочатку вихідне число й константа перетворяться в індексну форму, потім складаються в межах байта без обліку переносу.

Результатом операції є результат додавання, перетворена обернено в поліноміальну форму.

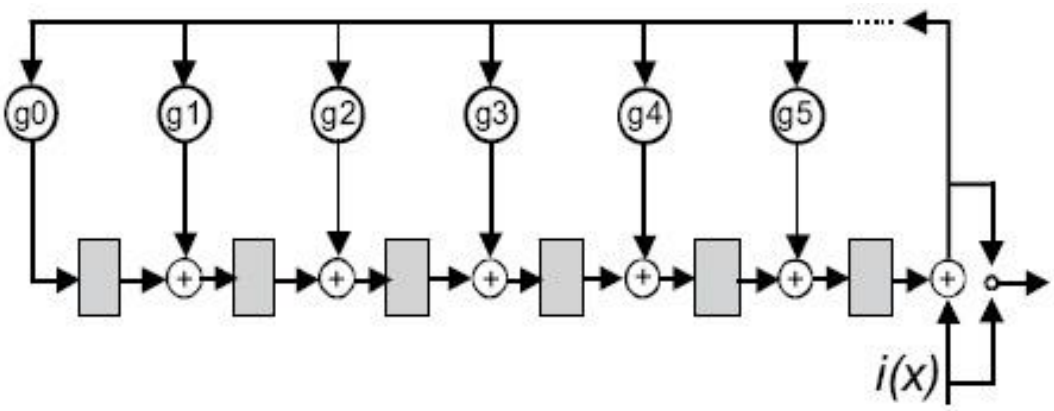


Рисунок 1 – Функціональна схема кодера Ріда-Соломона

При переході від однієї форми подання даних до інший доцільно використовувати таблицю істинності розміром 256 байт, що становить ємність одного EAB (Embedded Array Block – блок зосередженої пам'яті). Для реалізації кодеру потрібно 16 таких перемножувачів, при цьому те саме число множиться на різні константи, що дозволяє використовувати для його перекладу в індексну форму один EAB.

Для перекладу результатів у поліноміальну форму потрібно вже 16 таких таблиць, що вимагає застосування IMC FPGA дуже великої ємності. У запропонованій схемі використовується тактування кодера із частотою, в 8 разів перевищуючу частоту надходження байт даних.

Це дає можливість використовувати дві пари «суматор – EAB", мультиплексуючи константи на входах суматорів і дозволяючи роботу регістрів-накопичувачів у моменти появи відповідних даних на виходах засувок EAB.

На структурній схемі кодера (рисунок 2) символу «C» відповідають дві константи $GF(n)$.

Символ «L» у логіці регістрів-накопичувачів відповідає наступний: вихід компаратора нуля (символи CMP0 і SYNC) дозволяє роботу схеми «АБО що виключає», на входи якої подаються вихід попереднього регістра й EAB. Якщо ж вектор зворотного зв'язку дорівнює "0", схема пропускає дані з виходу попереднього регістра-накопичувача на вхід наступного.

У результаті кодер з урахуванням схеми синхронізації (на рисунку не показана) займає 255 LE (Logic Element – логічний елемент) і 3 EAB, що дозволяє розмістити його в IMC EPF10K10. Після оптимізації розміщення схеми на кристалі FPGA швидкодія схеми досяглася 11,57 Мгц (частота надходження байт даних, далі – байтова частота).

При використанні IMC EPF10K20, у складі якої 6 EAB, використовуючи 4 пари "суматор – EAB", можна тактувати кодер із частотами, що перевищують байтову частоту не в 8, а в 4 рази, що дозволить підняти її до 25...30 Мгц.

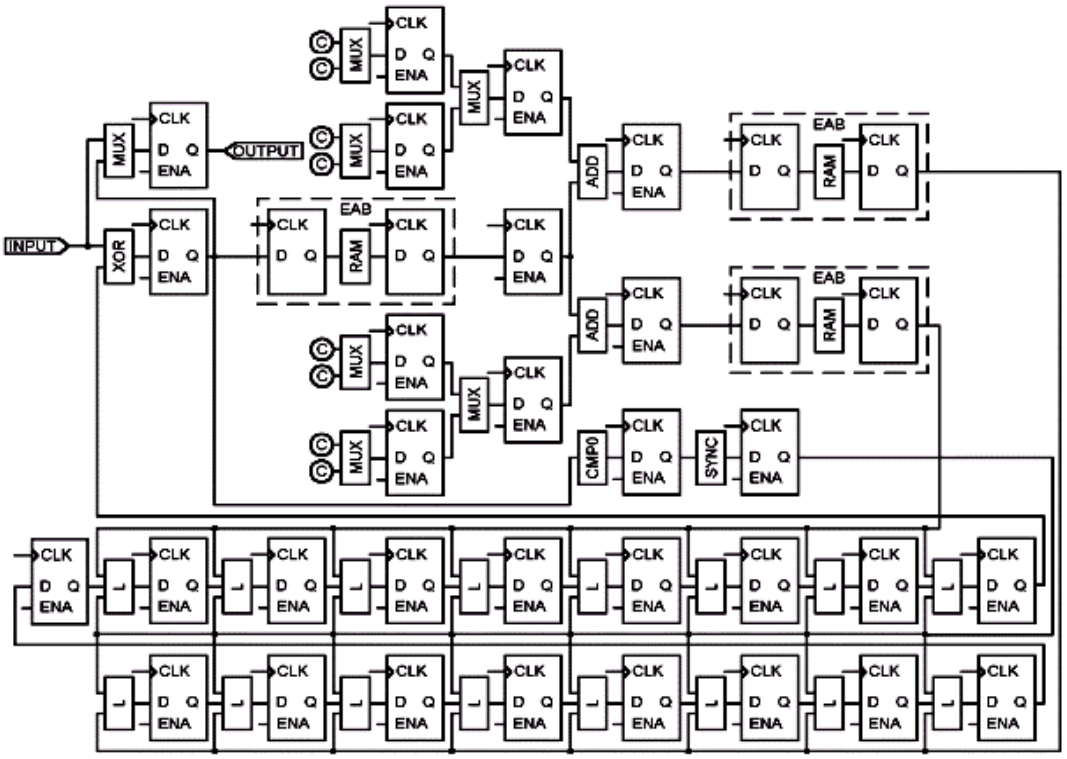


Рисунок 2 – Структура кодера РІда-Соломона

Декодування

Декодер, що працює по авторегресивному спектральному методі декодування, послідовно виконує наступні дії:

– Обчислює синдром помилки.

– Будує поліном помилки.

– Знаходить корінь даного полінома.

– Визначає характер помилки.

– Виправляє помилки.

Обчислення синдрому помилки

Обчислення синдрому помилки виконується синдромним декодером, що ділить кодове слово на багаточлен, що породжує. Якщо при діленні виникає остача, то в слові є помилка. Остача від ділення є синдромом помилки.

Побудова полінома помилки

Обчислений синдром помилки не вказує на положення помилок. Ступінь полінома синдрому дорівнює $2t$, що багато менше ступеня кодового слова n. Для одержання відповідності між помилкою і її положенням у повідомленні будується поліном помилок.

Поліном помилок реалізується за допомогою алгоритму Берлекемпа-Месси, або за допомогою алгоритму Евкліда. Алгоритм Евкліда має просту реалізацію, але вимагає більших витрат ресурсів. Тому частіше застосовується більше складний, але менш затратоємний алгоритм Берлекемпа-Месси. Коефіцієнти знайденого полінома безпосередньо відповідають коефіцієнтам помилкових символів у кодовому слові.

Алгоритм Евкліда виконується наступним чином.

Нехай a і b суть цілі числа, не рівні одночасно нулю, і послідовність чисел $a, b, r_1 > r_2 > r_3 > r_4 > \cdots > r_n$ визначена тим, що кожне r_k це остача від ділення перед-попереднього числа на попереднє, а передостаннє ділиться на останнє нацело, тобто

$$a = bq_0 + r_1$$
$$b = r_1q_1 + r_2$$
$$r_1 = r_2q_2 + r_3 \qquad (3)$$
$$\cdots$$

$$r_{n-1} = r_n q_n$$

Тоді (a,b), найбільший загальний дільник a і b, дорівнює r_n, останньому ненульовому члену цієї послідовності.

Існування таких $r_1,r_2,...$, тобто можливість ділення з остачею m на n для будь-якого цілого m і цілого $n \neq 0$, доводиться індукцією по m.

Коректність цього алгоритму випливає з наступних двох тверджень:

–Нехай $a = bq + r$, тоді $(a,b) = (b,r)$.

–$(0,r) = r$. для будь-якого ненульового r.

Знаходження корня

На цьому етапі шукаються коріння полінома помилки, що визначають положення перекручених символів у кодовому слові. Реалізується за допомогою процедури Ченя, рівносильній повному перебору. У поліном помилок послідовно підставляються всі можливі значення, коли поліном звертається в нуль – коріння знайдені.

Визначення характеру помилки і її виправлення

По синдрому помилки й знайдених корінь полінома за допомогою алгоритму Форни визначається характер помилки й будується маска перекручених символів. Ця маска накладається на кодове слово за допомогою операції XOR і перекручені символи відновлюються. Після цього відкидаються перевірочні символи й виходить відновлене інформаційне слово.

Застосування

У даний момент коди Ріда-Соломона мають дуже широку область застосування завдяки їхній здатності знаходити й виправляти багаторазові пакети помилок.

Запис і зберігання інформації

Код Ріда-Соломона використовується при записі й читанні в контролерах оперативної пам'яті, при архівуванні даних, запису інформації на жорсткі хмарні сервіси (ECC), запису у хмарних сервісах хмарні сервіси.

Навіть якщо ушкоджено значний обсяг інформації, зіпсовано кілька секторів дискового носія, то коди Ріда-Соломона дозволяють відновити більшу частину загубленої інформації. Також використовується при записі на такі носії, як магнітні стрічки й штрихкоди.

Запис у хмарних сервісах-ROM

Можливі помилки при читанні з диска з'являються вже на етапі виробництва диска, тому що зробити ідеальний хмарний сервіс при сучасних технологіях неможливо. Так само помилки можуть бути викликані подряпинами на поверхні диска, пилом і т.д.

Тому при виготовленні компакт-диску серверу хмарного сервісу, що читається, використовується система корекції CIRC (Cross Interleaved Reed Solomon Code). Ця корекція реалізована у всіх пристроях, що дозволяють зчитувати дані з CD дисків, у вигляді чипа із прошиванням firmware. Знаходження й корекція помилок заснована надмірності й перемежені (redundancy & interleaving). Надмірність приблизно 25% від вихідної інформації.

При записі на цифрові аудіокомпакт-хмарні сервіси (Compact Disc Digital Audio – CD-DA) використовується стандарт Red Book. Корекція помилок відбувається на двох рівнях C1 і C2. При кодуванні на першому етапі відбувається додавання перевірочних символів до вихідних даних, на другому етапі інформація знову кодується.

Крім кодування здійснюється також перемішування (перемеження) байтів, щоб при корекції блоки помилок розпалися на окремі біти, які легше виправляються. На першому рівні виявляються й виправляються помилкові блоки довжиною один і два байти (один і два помилкових символи відповідно).

Помилкові блоки довжиною три байти виявляються й передаються на наступний рівень. На другому рівні виявляються й виправляються помилкові блоки, що виникли в C2, довжиною 1 і 2 байти. Виявлення трьох помилкових символу є фатальною помилкою, не можуть бути виправлені.

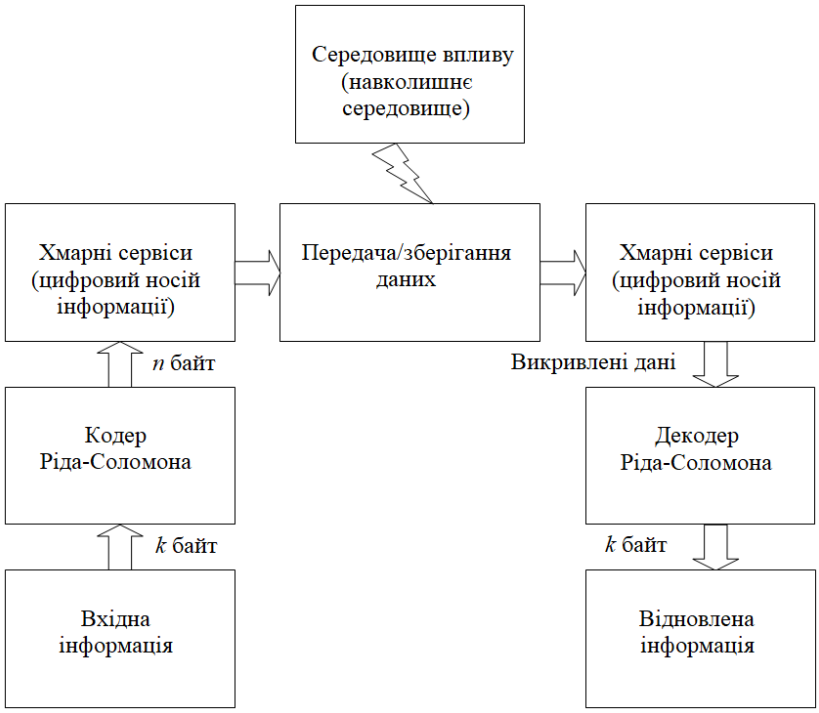


Рисунок 3 – Структурна схема системи

Бездротовий і мобільний зв'язок

Цей алгоритм кодування використовується при передачі даних по мережах WiMAX, в оптичних лініях зв'язку, у супутниковому й радіорелейному зв'язку. Метод прямої корекції помилок у минаючому трафике (Forward Error Correction, FEC) ґрунтується на кодах Ріда-Соломона.

Розробка структурної схеми

Структурна схема системи забезпечення цілісності даних у хмарних сервісах зображена на рисунку 3.

З цієї схеми ми бачимо, що над вхідними даними, перед записом у хмарний сервіс, відбуваються перетворення кодеком Ріда-Соломона. Після кодування дані записуються на носій. У якості носія окрім хмарні сервіси може використовуватися любий інший носій інформації. Після цього інформація зберігається на носієві. При зчитуванні інформації, розроблене програмне забезпечення декодує інформацію, яка зберігається на носієві, і якщо потрібно, після проведення відповідних перевірок, проводить відновлення втраченої інформації. Якщо таке відновлення неможливе, то програма видає відповідне повідомлення.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів забезпечення цілісності даних у хмарних сервісах.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем забезпечення цілісності даних у хмарних сервісах.

– Досліджена система забезпечення цілісності даних у хмарних сервісах.

– На основі отриманих результатів досліджень створена програмна реалізація системи забезпечення цілісності даних у хмарних сервісах.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання забезпечення цілісності даних у хмарних сервісах.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
2. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
3. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
4. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
5. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
6. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
7. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
8. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
9. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
10. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
11. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
12. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).
13. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019. (Scopus).
14. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629. (Scopus).
15. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 873-884. (Scopus).
16. Smirnov, O., Kuznetsov, A., Prokopovych-Tkachenko, D. «Hiding Data in Images Using a Pseudo-Random Sequence». ISCI'2020: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko, Victor A. Krasnobayev and Alexandr A. Kuznetsov. ASC Academic Publishing, USA, 2020. pp. 46-59. – ISBN: 978-1-7362833-0-1 (Hardback), ISBN: 978-1-7362833-1-8 (Ebook).
17. Smirnov, O., Kuznetsov, A., Shekhanin, K., Chepurko, I. Detecting Hidden Information in FAT. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 412-429. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
18. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
19. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
20. Смірнов О.А., Дрєєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.

УДК 004

А. Олійник, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ CAN-МЕРЕЖІ НА ОСНОВІ ТЕХНОЛОГІЇ CSDN

У статті розроблено програмне забезпечення, яке призначено для системи CAN-мережі на основі технології CSDN. Метою розробки є дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN. Об'єктом дослідження є процес CAN-мережі на основі технології CSDN. Предметом дослідження є методи CAN-мережі на основі технології CSDN. Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи CAN-мережі на основі технології CSDN. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерна інженерія, CAN-мережі, CSDN

Постановка проблеми. Новий етап, що наступив у розвитку обміну інформацією, який характеризується інтенсивним впровадженням сучасних інформаційних технологій, широким поширенням локальних, корпоративних і глобальних мереж, створює нові можливості і якість інформаційного обміну.

Корпоративні інформаційні системи (CAN) стають сьогодні одним з головних інструментів управління бізнесом, найважливішим засобом виробництва сучасного підприємства, вони використовуються в банківських, фінансової сферах, у сфері державного управління. CAN містить у собі інфраструктуру й інформаційні сервіси. Інфраструктура CAN (мережі, сервери, робочі станції, додатки) є географічно розподілені, її структурна одиниця – сегмент CAN (СГ CAN).

Однак застосування інформаційних технологій немислимо без підвищеної уваги до питань інформаційної (комп'ютерної) безпеки через наявність погроз захищеності інформації.

Для сучасного етапу розвитку теорії й, особливо, практики забезпечення захисту інформації (ЗІ) характерна парадоксальна ситуація: з одного боку, посилене увага до безпеки інформаційних об'єктів, істотне підвищення вимог по ЗІ, прийняття міжнародних стандартів в області інформаційної безпеки (ІБ), постійно зростаючі витрати на забезпечення захисту, з іншого боку – настільки ж неухильно зростаючий збиток, заподіюваний власникам і власникам інформаційних ресурсів, про що свідчать публікуємі регулярно дані про збиток світовій економіці від комп'ютерних атак.

Очевидно, що сучасні підходи до організації ЗІ не повною мірою забезпечують виконання вимог по захисту інформації. Основні недоліки СЗІ визначаються сформованими твердими принципами побудови архітектури й застосуванням в основному оборонної стратегії захисту від відомих погроз. Критична ситуація в сфері ІБ збільшується у зв'язку з використанням глобальної мережі для зовнішніх і внутрішніх електронних транзакцій підприємства й появою невідомих раніше типів деструктивних інформаційних впливів.

Тому для успішного використання сучасних інформаційних технологій необхідно ефективно управляти не тільки мережею, але й СЗІ, при цьому на рівні СГ CAN автономно повинна працювати система, що реалізує управління складом подій інформаційної безпеки, планування модульного состава СЗІ й аудит. Оскільки об'єкт управління – СЗІ є досить

складною організаційно-технічною системою, що функціонує в умовах невизначеності, суперечливості й неповноти знань про стан інформаційного середовища, управління такою системою повинне бути засноване на застосуванні системного аналізу, методів теорії прийняття рішень і необхідної інтелектуальної підтримки.

Разом з тим в області розробки методів і систем захисту інформації в цей час практично відсутні дослідження, спрямовані на забезпечення автоматизованої підтримки управління ЗІ для рішення проблеми забезпечення необхідного рівня захищеності інформації протягом усього періоду функціонування СЗІ.

Одним з варіантів рішення даної проблеми, розглянутим у магістерській роботі, є використання методів інтелектуальної підтримки управління ЗІ в сегменті корпоративної інформаційної системи, що у свою чергу, вимагає розробки на основі принципів системного аналізу й загальнонаукових підходів методологічних основ управління захистом інформації, що відповідають моделей, методів, алгоритмів і програмного забезпечення.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи CAN-мережі на основі технології CSDN.

Мета й завдання дослідження. роботи є дослідження та програмна реалізація системи CAN-мережі на основі технології CSDN.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем CAN-мережі на основі технології CSDN.
- Дослідження системи CAN-мережі на основі технології CSDN.
- Програмна реалізація системи CAN-мережі на основі технології CSDN.

Об'єктом дослідження є процес CAN-мережі на основі технології CSDN.

Предметом дослідження є методи CAN-мережі на основі технології CSDN.

Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Опис архітектури Cisco Self-Defending Network

Частково завдяки діяльності Cisco, що представляє стратегію мережі, яка самозахищається Cisco (Self-Defending Network) (CSDN), багато хто починає усвідомлювати необхідність інтегрованих засобів мережного захисту.

Механізми забезпечення мережної безпеки еволюціонували від незалежно використовуваних «точкових» продуктів, таких як міжмережні екрани або засоби виявлення вторгнень, в область інтегрованих і цілісних рішень. Cisco Systems є провідною компанією по розробці технології, що дозволяє зробити мережі, що самозахищаються, реальністю.

Ідея рішення досить просте: призначення IT-інфраструктури полягає в створенні систем, що надають можливість виявлення порушень безпеки й захисту від несанкціонованого доступу з одночасним наданям оперативного доступу легальним користувачам. Проста відмова в доступі вже не є підходящою реакцією на атаку – сучасні мережі повинні реагувати на атаки, зберігаючи свою доступність, надійність і працездатність. У багатьох відносинах, метою забезпечення безпеки стає підвищення ступеня відказостійкості мереж. Замість того, щоб ставати жертвами, мережі повинні стати здатними «поглинати» атаки й зберігати працездатність, подібно імунній системі людини, що дозволяє організму функціонувати при наявності в ньому вірусів і бактеріальних інфекцій.

Розвиток ситуації в сфері безпеки

За останні три роки технології забезпечення безпеки змінилися більше, ніж за все попереднє десятиліття. Обсяг і темп цих змін ускладнили покладену на IT-Фахівців завдання підтримки належного рівня захищеності. Перед тим, як продовжити розповідь про Cisco SDN, необхідно одержати поданя про суть цих змін:

– Захист периметра мережі. Мабуть, найбільш істотним фактором, що вплинув на зміну підходу до забезпечення безпеки мереж, стала зміна самої сутності мережі. Після того, як корпорації стали консолідувати центри обробки даних, використовувати конверговані внутрішні мережі й активно використовувати мережу Інтернет, уже не можна забезпечити безпеку мережі тільки за рахунок організації захисту її периметра. Середовище, що раніше вважалося ізольованим і контрольованим, тепер є напіввідчиненим за рахунок, наприклад, мереж "екстранет", підключень пунктів роздрібного продажу, надомних працівників та ін. Розширення корпоративної мережі, таким чином, приводить до необхідності взаємодії через ненадійні проміжні мережі й неконтрольовані середовища. Пристрої, що підключаються до корпоративної мережі через ці проміжні мережі, найчастіше не відповідають вимогам корпоративних політик безпеки. Пристрої, їм відповідні, часто використовуються для доступу до інших неконтрольованих мереж до з'єднання з корпоративною мережею. У результаті, пристрої, підключені до зовнішніх мереж, можуть стати «перевалочним пунктом» для атак і пов'язаних з ними несанкціонованих дій.

– Бездротові мережі й мережі мобільного зв'язку. Прив'язані до поняття периметра захисту бездротові мережі й мережі мобільного зв'язку підприємств тепер забезпечують підтримку ноутбуків, кишенькових комп'ютерів (PDA) і мобільних телефонів, які підключені до декількох мереж. Ці пристрої з декількома мережними інтерфейсами підтримують можливість установлення однорангових бездротових з'єднань для роботи в мережі "точка-точка". Крім того, пакети можуть ефективно передаватися між пристроями на прикладному рівні. У результаті поняття границь мережі стає усе більше розмитим і для забезпечення безпеки компаніям необхідно мати можливість управління такими мобільними пристроями.

– Електронна комерція, мережі "екстранет" і проведення ділових операцій у глобальній мережі. Поява загальних прикладних інтерфейсів на основі протоколів передачі повідомлень, таких як XML і SOAP, зробило благотворний вплив на електронну комерцію й продуктивність роботи підприємств. Але, як і в більшості випадків появи нових технологій, їхня поява привела до виникнення зовсім нових уразливістей і джерел атак, з якими доводиться боротися. Дані, які раніше передавалися за допомогою множини мережних протоколів і проходили фільтрацію на міжмережних екранах, тепер передаються за допомогою декількох або всього одного транспортного протоколу (наприклад, HTTP з використанням порту 80 TCP). У результаті, більша частина даних, що раніше містилася в заголовках пакетів, тепер розташовується в тілі пакетів. Це істотно полегшує зловмисникові завдання обходу класичної системи захисту мережі. Більше того, для забезпечення конфіденційності й цілісності корпоративних даних всі частіше використовується шифрування трафіка прикладного рівня за допомогою протоколів SSL/TLS і HTTPS. При цьому виникає побічний ефект, пов'язаний з ускладненням контролю доступу на границі мережі через неможливість перевірки пакетів у зашифрованих потоках даних.

– Віруси, Інтернет-хробаки й швидкість їхнього поширення. Кількість і різноманіття вірусів, що з'явилися за останні три роки, і Інтернет-хробаків саме по собі є застрашливим. Дивовижний вплив цих Інтернет-хробаків і вірусів на мережі підприємств і їхня продуктивність було обумовлено наявністю двох факторів: короткого проміжку часу між виявленням уразливості й появою атаки з її використанням, а також швидкості, з якою більшість атак поширювалося по мережі. При цьому число порушень роботи мереж досягало неприпустимого рівня, а для усунення наслідків доводилося йти на незаплановані витрати людських, тимчасових і матеріальних ресурсів.

– Дотримання встановлених норм. Факти, що одержали широкий розголос, порушень і неправомірні дії усередині корпорацій підштовхнули керуючі органи багатьох галузей до створення норм по регулюванню ризиків відносно корпоративної інформації. У США ці норми, найбільш відомими з яких є закон Сарбейнса-Окслі, закон Гремма-Ліча-Блілі й закон про дотримання конфіденційності інформації про охорону здоров'я й особистих даних пацієнтів (HIPAA), привели до корінних змін способів організації корпоративних мереж, серверів, баз даних і хостів. Аналогічна тенденція спостерігається й в Україні.

Хоча багато організацій думають, що дотримання норм забезпечує більш надійний захист їхньої інфраструктури, дане думка найчастіше є помилковим. Сам процес проходження встановленим нормам може привести до виникнення нових уразливістей. Наприклад, Інтернет-хробаки й віруси можуть більш ефективно поширюватися в мережі, що підтримує наскрізні VPN-з'єднання, у зв'язку з тим, що минаючі по них потоки даних є невидимими для проміжних вузлів. Такі потоки даних можуть переносити Інтернет-хробаків на критично важливі корпоративні сервери за допомогою надійно зашифрованих пакетів. Крім того, що на виявлення такої атаки йде багато часу, наскрізні VPN-з'єднання ускладнюють процес усунення її наслідків.

Принципи побудови сучасних безпечних мереж

Корпорації не можуть нескінченно додержуватися напрямків в області безпеки, що змініються. В ідеалі, удосконалювання системи безпеки повинне впливати на існуючу інфраструктуру маршрутизації й комутації, методи розмежування й контролю доступу й суміжних організаційних структур, що забезпечують підтримку цих систем. У цьому розділі ми опишемо основні елементи мережі, що самозахищається, **Cisco Self-Defending Network**:

– Присутність. Фундаментальним поняттям захищеної системи є поняття контрольних точок, що ми визначимо як присутність. Подібно імунній системі людини, що заснована на розосереджені по всьому тілу людини й виконуючі функції виявлення інфекції й виконання відповідних дій клітки, мережа покладається на наявність певних можливостей в окремих вузлів. До таких можливостей відносяться класичні методи ідентифікації, контролю доступу, перевірки даних і захисту взаємодії, а також нові можливості аналізу дій клієнтів файлообмінних мереж, web-сервісів, голосових сервісів і сервісів передачі динамічного контенту по мобільних мережах.

– Контекст. При вході користувача в систему мережа запитує й одержує доступ до набору реквізитів доступу користувача й хоста, що представляють собою кінцеву сутність. Повноваження можуть змінюватися із часом залежно від дій підключеного до мережі хоста. Сукупність цих даних і являє собою контекст. На відміну від існуючих систем мережної безпеки, у яких велика увага приділяється тільки перевірці повноважень користувача при вході в мережу, мережа, що самозахищається Cisco Self-Defending Network ухвалює рішення щодо наданя або скасування повноважень на основі змін поводження й відповідного йому контексту за увесь час з'єднання користувача з мережею. Наприклад, якщо мережа виявляє, що хост заражено вірусом (при цьому користувач може мати всі повноваження на доступ), вона ізолює цей хост у карантинний сегмент мережі. Оскільки дані можуть бути підмінені, у процесі забезпечення безпеки системи може знадобитися одержання контексту від інших систем для точного й своєчасного визначення прав хоста й привілеїв у конкретний момент часу.

– Взаємозв'язок. Взаємозв'язки між окремими пристроями дозволяють обмінюватися контекстом і створювати «систему». Традиційно, взаємозв'язки між пристроями мережі встановлювалися за допомогою протоколів маршрутизації, такими як протокол BGP. Для того щоб протистояти найсучаснішим видам погроз і несанкціонованих дій, тепер необхідно розширювати ці взаємозв'язки по всьому маршруті від джерела до одержувача мережного трафіка. Крім того, через зростаюче число мобільних пристроїв, взаємозв'язки вийшли за межі границь, які донедавна розглядалися як зовнішні границі мереж у традиційному розумінні. Привілеї, які пристрій одержує при доступі до мережі й характер їхньої зміни в процесі сеансу роботи визначаються на основі контексту цього пристрою і його взаємозв'язків у мережі.

– Довіра. Безпека системи визначається безпекою вступної у неї інформації; система функціонує набагато ефективніше, якщо в ній присутні довірчі відносини. Раніше ступінь довіри визначалася головним чином на основі ідентифікації пристрою або користувача. Результати останніх досліджень показали, що в концепцію захищених систем повинні бути включені поняття стану й місця розташування пристрою.

По багатьом параметрам дії, виконувані користувачами або пристроями в мережі, можна зрівняти з управлінням автомобілем. Подібно тому, як людина дістає водійські права, що дозволяють йому управляти певним класом транспортних засобів, користувачі повинні мати деяку ідентифікаційну інформацію для входу в мережу. Крім того, у кожного автомобіля є ідентифікаційний номер, що повинен бути зареєстрований у місцевих органах управління – мережі й кінцеві вузли незабаром будуть мати цифрові сертифікати, створювані під час випуску й потребуючі виконання певного типу реєстрації при використанні в рамках компанії. Але оскільки пристроям не завжди вдається вчасно надавати ідентифікаційні дані, мережі, що самозахищаються Cisco Self-Defending Network використовує передові методи непрямої довіри й максимальних зусиль для автентифікації й авторизації сутностей. Мережа, що самозахищається, Cisco Self-Defending Network повинна як мінімум уміти запитувати ідентифікаційні дані кожного пристрою й користувача, виконувати аналіз стану пристрою й установлювати місце розташування пристрою в мережі. Технологія, що дозволяє реалізувати ці можливості, буде повсюдно поширена й задіяна за допомогою чітко певних стандартних форматів повідомлень і протоколів, таких як протокол 802.1x і протокол автентифікації EAP.

Саме по собі кожне із цих понять не дуже примітно. Але вони здобувають силу при об'єднанні в мережі, що самозахищається, Cisco Self-Defending Network. У частині, що залишилася, даного огляду описуються деякі способи використання цих понять у рамках мережі, що самозахищається Cisco Self-Defending Network.

Елементи й побудова мережі

Оскільки одночасна перебудова всіх підсистем без порушення цілісності IT-сервисів може виявитися складним завданням, більшість споживачів не зможе впровадити всі компоненти стратегії Cisco SDN одночасно. Крім того, деякі споживачі можуть баритися з передачею функцій контролю безпеки автоматизованій системі доти, поки вони не переконаються в надійності роботи рішення. Стратегія мережі, що самозахищається, Cisco Self-Defending Network дозволяє таким компаніям здійснювати поступовий перехід до Cisco SDN за рахунок наданя продуктів, які можуть використовуватися незалежно друг від друга. Таким чином, має сенс розглянути наступні основні етапи проектування мережі, що самозахищається, Cisco Self-Defending Network.

Захист кінцевих вузлів. Віруси й Інтернет-хробаки, що заражають кінцеві вузли, часто приводять і до побічного ефекту – перевантаженню мережі, що є наслідком їхнього швидкого поширення.

Cisco пропонує засіб запобігання вторгнень на кінцеві вузли Cisco Security Agent, що дозволяє вирішити обидві проблеми. Використовувані в Cisco Security Agent передові методи захисту на основі аналізу поводження дозволяють виявляти віруси й Інтернет-хробаки, а також запобігати їхнє проникнення на кінцеві системи й поширення по мережі. Фактично, Cisco Security Agent є першою лінією оборони для запобігання поширення вірусів і Інтернет-хробаків.

Другим очевидним аргументом на користь застосування Cisco Security Agent є те, що він використовується на кінцевих вузлах і дозволяє створити ланцюг відповідної реакції між кінцевим вузлом і мережею. У результаті виходить мережа, здатна швидко адаптуватися до виникаючих погроз.

Контроль доступу. Однією з найбільш важливих можливостей мережі, що самозахищається, Cisco Self-Defending Network є механізм контролю доступу до мережі Cisco Network Admission Control (NAC).

NAC дозволяє вирішити, який рівень доступу варто надати кінцевому вузлу, виходячи з відповідності стану вузла політиці безпеки компанії, обумовленого шляхом аналізу стану безпеки операційної системи й установлених додатків. На додаток до функцій контролю й розмежування доступу NAC надає IT-адміністраторам можливість автоматичного перекладу в карантин і лікування кінцевих вузлів, що не пройшли перевірку відповідності. Перевірка відповідності є ефективною другою лінією оборони для запобігання поширення вірусів і

Інтернет-хробаків. NAC можна також розглядати як інструментальний засіб аналізу уразливістей і управління установкою «латок» на вимогу.

Відмінною рисою NAC є наданя як клієнтського, так і адміністративного інтерфейсу AAA, що дозволяють споживачам додатково встановлювати продукти великої кількості розроблювачів засобів захисту.

У цей час більше 250 лідируючих на ринку розроблювачів інтенсивно впроваджують або вже впровадили у свої продукти механізми NAC.

Важливо надати можливість використання NAC у системах малих і середніх підприємств. Для цього Cisco кілька років назад придбала корпорацію Perfigo, областю діяльності якої є розробка комплексних рішень контролю доступу до мережі. Основними функціями рішень є аналіз політик кінцевих вузлів, перевірка відповідності стану вузлів установленим вимогам і забезпечення працездатності засобів контролю й розмежування доступу. Тепер у рамках ініціативи Network Admission Control компанія Cisco пропонує рішення за назвою Cisco NAC Appliance (Cisco Clean Access).

Обмеження області зараження. Посилені політики доступу не є панацеєю й не усувають необхідність моніторингу пристроїв після їхнього входу в мережу. Кваліфіковані зловмисники в стані обійти практично будь-яку перевірку прав доступу, а мережі не можуть постійно покладатися на заражений елемент або довіряти йому. Пристрої, що пройшли перевірку відповідності, також можуть бути інфіковані за допомогою різноманітних джерел зараження після входу в мережу – наприклад, зараження з USB-накопичувача.

Мережа, що самозахищається Cisco Self-Defending Network спроектована для виконання перевірок безпеки не тільки під час одержання вузлом доступу до мережі, але й протягом усього сеансу з'єднання. Крім того, мережа, що самозахищається Cisco Self-Defending Network може покладатися на інші елементи мережі, включаючи кінцеві вузли для визначення компрометації інших вузлів, за аналогією з тим, як поліція контролює рівень злочинності шляхом аналізу дзвінків на номер 911. Cisco розглядає засоби обмеження області зараження як третю лінію оборони для запобігання поширення вірусів і Інтернет-хробаків.

На жаль, протоколи автентифікації, що існують, не розроблялися для роботи після початкового обміну інформацією. Таким чином, мережа, що самозахищається Cisco Self-Defending Network повинна забезпечувати нові способи обміну інформацією про стан пристроїв (контекст), а також способи оцінки вірогідності цієї інформації на основі як непрямого, так і прямої довіри. Наприклад, адміністратор повинен мати можливість створювати правило, відповідно до якого повідомлення, отримане від кінцевого вузла із установленим агентом Cisco Security Agent, заслуговує більшої довіри, чим повідомлення, що прийшло від незахищеного кінцевого вузла. У результаті компанія Cisco почала розробку нових механізмів кореляційного аналізу й відповідної реакції на основі непрямих атрибутів.

Інтелектуальні засоби кореляційного аналізу й реагування на інциденти

Для забезпечення ефективної роботи методів відповідної реакції, швидкої оцінки впливу, вибору конкретної дії й визначення найкращого засобу захисту необхідно, щоб мережа, що самозахищається, Cisco Self-Defending Network надавала сервіси кореляційного аналізу подій у сфері безпеки в режимі реального часу.

Для рішення цього завдання компанія Cisco придбала компанію Protego Networks, що розробила сімейство продуктів MARS, що надають методи зв'язування відповідної реакції від різних мережних пристроїв (міжмережні екрани, системи виявлення вторгнень, маршрутизатори, комутатори й хости) з контекстом, одержуваним у результаті вивчення топології мережі на рівні 2 і 3. Це дозволяє групі реакції на порушення в сфері безпеки швидко визначити місце появи атак у мережі.

Інтегровані системи виявлення вторгнень і механізми виявлення аномалій. Проектування ефективних систем виявлення мережних вторгнень (NIDS) завжди було важливим напрямком в області постійно, що ведуться досліджень, і розробок Cisco. Одним з

перших нововведень Cisco у цій області було впровадження NIDS у маршрутизатори й комутатори.

Але для того щоб система NIDS мала повну функціональність, її необхідно перетворити в систему запобігання вторгнень (IPS) з убудованими можливостями фільтрації трафіка, що дозволяє відкидати непотрібні пакети за допомогою підсистем, що набудовуються тонко, класифікації трафіка.

На жаль, більшість NIDS видають занадто багато помилкових спрацьовувань і не можуть надійно виконувати завдання запобігання атак при установці системи на проміжному пристрої. Почасти проблема полягає в необхідності збору й обробки великого обсягу інформації (контексту) протягом досить короткого проміжку часу.

Особливо це, до речі, стосується додатків, які дуже чутливі до затримок передачі (наприклад, IP-телефонія). Для рішення цього завдання Cisco розробляє кілька методів, що забезпечують високоякісну й ефективну обробку й класифікацію контрольованого трафіка.

Багато легальних дій можуть бути помилково сприйняті мережею як аномальні; головним чином, це стосується мереж зі значним числом змінних факторів.

У результаті компанія Cisco стала випливати консервативному поетапному підходу до виявлення аномалій, починаючи з Cisco Security Agent, оскільки було встановлено, що операційні системи моделювати простіше, ніж мережні середовища. Після цього компанією Cisco була придбана ефективна система запобігання вторгнень Riverhead, що характеризується низьким числом помилкових спрацьовувань за рахунок чіткого поділу дій, спрямованих на проведення атак типу «відмова в обслуговуванні», і іншої мережної активності.

Безпека додатків і захист від шкідливих програм (Anti-X). За останні кілька років з'явилися нові мережні додатки, що забезпечують захист від нових видів погроз, включаючи віруси, Інтернет-хробаків, спам, шпигунські програми, зловмисне використання web-сервісів і засобів IP-телефонії, а також несанкціоноване використання клієнтів файлообмінних мереж, – захист від яких не забезпечувалася повною мірою класичними міжмережними екранами й продуктами NIDS.

З метою захисту від цих погроз фахівцями Cisco були розроблені сервіси захисту нового покоління, що виконують перевірку заголовків пакетів і їхнього вмісту. Це дозволяє забезпечити ретельну перевірку трафіка в критично важливих точках мережі й обробляти зловмисний трафік до влучення в корпоративну мережу.

Об'єднання цих сервісів у багатофункціональні платформи дозволяє розширити можливості розроблювачів, а також знизити сукупну вартість володіння для споживача. Крім того, інтеграція цих механізмів дозволить розширити можливості мережі, що самозахищається, Cisco Self-Defending Network по контролі додатків.

Якщо в додатках використовується наскрізне шифрування, мережа, що самозахищається Cisco Self-Defending Network може збирати інформацію з кінцевих вузлів, компенсуючи втрати, пов'язані з неможливістю контролю даних на границі мережі.

Структурна схема мережі, що самозахищається, Cisco Self-Defending Network наведена на рисунку 1.

Розробка структурної схеми

Ґрунтуючись на принципах системного аналізу, що являє собою теорію й практику поліпшуючого втручання в проблемну ситуацію, пропонується варіант декомпозиції проблеми дозволу наявних протиріч в області забезпечення безпеки інформації.

На підставі системного підходу видно, що модель проблемної ситуації в області захисту інформації містить сукупність трьох взаємодіючих систем:

– проблемоутримуючої СЗІ,

– проблемодозволяючої керуючої системи, що розробляється для того, щоб проблема зникла або ослабнула, що оточує;

– істотного середовища, з якої взаємодіє СЗІ, під якою розуміється безліч потенційна можливих погроз інформаційної безпеки.

Вимога постійно наростаючої деталізації приводить до побудови моделі состава проблемоутримуючої системи, моделі об'єкта захисту й моделі погроз.

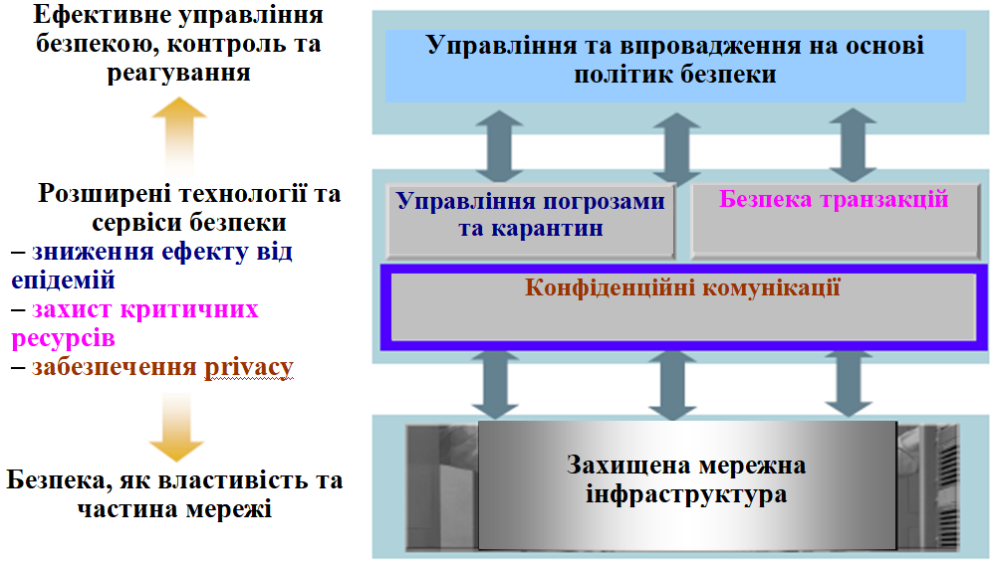


Рисунок 1 – Структурна схема мережі, що самозахищається, Cisco Self-Defending Network

Відзначається, що основною проблемою при побудові керуючої системи є розробка моделі погроз, що зв'язано зі специфічністю взаємодії об'єкта управління – СЗІ з навколишнім середовищем. У зв'язку із цим пропонується концепція побудови моделі погроз безпеки інформації, що базується на розроблювальній класифікаційній схемі навмисних цілеспрямованих погроз інформаційному середовищу корпоративної інформаційної системи. Показано доцільність побудови сукупності моделей:

– функціональної, на основі опису послідовності дій зловмисника (порушника) за допомогою дерев погроз;

– просторової графової, систематизованих у форматі інтегральної структурної моделі каналів несанкціонованого доступу, витоку й деструктивних впливів, що дозволяє провести всебічний аналіз реальних погроз, підвищити адекватність моделі погроз для конкретного об'єкта захисту.

На основі аналізу принципів управління в умовах невизначеності пропонується узагальнена архітектура системи управління захистом інформації в сегменті корпоративної інформаційної системи. Проаналізуємо основні функції управління, обґрунтовується доцільність варіанта побудови системи, що включає дві функціональні підсистеми:

– підсистему організаційно-технічного управління;

– і підсистему оперативного управління в реальному масштабі часу.

Відповідно до вимоги кількісної оцінки характеристик систем, висунутим системотехнікою, у якості керованої змінної введемо показник – рівень захищеності, необхідна значення якого залежить від максимального рівня критичності оброблюваної в даний період часу інформації.

У контурі організаційно-технічного управління створюються механізми управління захистом інформації при зміні інфраструктури, бізнес-додатків, планів обробки інформації й відповідних їм вимог до рівня захищеності інформації. Контур включає: систему інтелектуальної підтримки прийняття рішень на вибір стратегії захисту, систему оцінки рівня захищеності (ризику), що управляє вплив реалізується співробітниками відділу інформаційної безпеки. Командна інформація формується в ході планування – цілеспрямованого вибору раціонального комплексу засобів захисту.

У контурі оперативного управління формується оперативна командна інформація, що доводить до об'єкта управління адміністратором безпеки або автоматично за допомогою засобів реалізації керуючих впливів на убудовані в засоби захисту керуючі модулі.

У системі управління, що має таку архітектурну побудову, ефективні рішення вибираються й приймаються як на основі відомостей про технічні характеристики засобів захисту, так і на основі аналізу контрольованого простору.

Структурна схема системи управління захистом інформації в сегменті корпоративної інформаційної системи показана на рисунку 2.

На основі аналізу можливостей удосконалювання управління захистом інформації за рахунок застосування нових методів рішення завдань управління й скорочення тривалості циклу управління розробляється функціональна модель системи управління в стандарті IDEF0, що дозволяє наочно й ефективно відобразити механізм управління загрозами, виявити процеси, для реалізації яких необхідна розробка автоматизованої системи інтелектуальної підтримки управління.

Рисунок 2 – Структурна схема системи управління захистом інформації в сегменті корпоративної інформаційної системи

У структурній схемі застосовуються наступні скорочення:

- АБ – адміністратор безпеки;
- ОІБ – співробітники відділу інформаційної безпеки;
- Обу (СЗІ) – об'єкт управління;
- ГПБ, ЛПБ – глобальна, локальні політики безпеки;
- НСДУВ – несанкціонований доступ, витік, деструктивний вплив;
- ЗЗ – засоби захисту;
- $\eta_{тр}$ – необхідне значення рівня захищеності.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів CAN-мережі на основі технології CSDN. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем CAN-мережі на основі технології CSDN. Досліджена система CAN-мережі на основі технології CSDN. На основі отриманих результатів досліджень створена програмна реалізація системи CAN-мережі на основі технології CSDN. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання CAN-мережі на основі технології CSDN. Проведено аналіз предметної

галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w (Scopus).
2. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
3. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).
4. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
5. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
6. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
7. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
8. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
9. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
10. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
11. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
12. Смирнов А.А. Анализ и сравнительное исследование перспективных направлений развития цифровых телекоммуникационных систем и сетей / А.А.Смирнов, В.В.Босько, Е.В.Мелешко // Системи обробки інформації. – Х.: ХУ ПС, 2008. – Вип.7(74). – С.120-123.
13. Смирнов А.А. Усовершенствование метода управления очередями в многопротокольных узлах телекоммуникационной сети / А.А.Смирнов, Е.В.Мелешко // Збірник тез та доповідей другої всеукраїнської науково-практичної конференції «Системний аналіз. Інформатика. Управління». Запоріжжя. Тези доповідей. Запоріжжя: КПУ, 2011.
14. Смирнов С.А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А.К. Дидык, С.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.
15. Смирнов С. А. Сравнительные исследования математических моделей технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, А. В. Коваленко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2014. – Вип. 9(125). – 105-110.
16. Смирнов С. А. Математическая модель интеллектуального узла коммутации с обслуживанием информационных пакетов различного приоритета / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2014. – Вип. 4 (41). – С. 48-52.
17. Смирнов С. А. Исследование показателей качества функционирования интеллектуальных узлов коммутации в телекоммуникационных системах и сетях / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2014. – № 4(17). – С. 90-95.
18. Смирнов С. А. Усовершенствованный алгоритм управления доступом к «облачным» телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, Н. С. Якименко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2015. –Вип. 1(126). – С. 150-153.

19. Smirnov S.A. Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam, S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.
20. Смирнов С. А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты даннях / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Системи озброєння і військова техніка: наук. журн. – Х.: ХУПС, 2015. – № 3(43). – С. 100-107.
21. Смирнов С. А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Абу Таам Гани, А. А. Смирнов, С. А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України: наук. журн. –Х.: ХУПС, 2015. –№ 3(20). – С. 134-141.
22. Смирнов С. А. Комплекс gert-моделей технологии облачной антивирусной защиты телекоммуникационной системы / А. А. Смирнов, А. К. Дидык, А. Н. Дреев, С. А. Смирнов // Безпека інформації: наук. - практ. журн. – К.: НАУ, 2015. – Т. 21, № 3. – С. 251-262.

УДК 004

А.Пилипенко, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ BIG DATA НАУКОВИХ ДОСЛІДЖЕНЬ

У роботі розроблено програмне забезпечення, яке призначено для системи big data наукових досліджень. Метою розробки є дослідження та програмна реалізація системи big data наукових досліджень. Об’єктом дослідження є процес big data наукових досліджень. Предметом дослідження є методи big data наукових досліджень. Методи дослідження базуються на методах big data, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи big data наукових досліджень. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп’ютерні науки, big data, наукові дослідження

Постановка проблеми.

Великі дані (big data) обіцяють революціонізувати виробництво знань у науці та за її межами, забезпечивши нові, високоефективні способи планування, проведення, поширення та оцінки досліджень. Останні кілька десятиліть стали свідками створення нових способів виробництва, зберігання та аналізу даних, кульмінацією яких стала поява галузі даних, яка об’єднує обчислювальні, алгоритмічні, статистичні та математичні методи для екстраполяції знань із великих даних. У той же час рух *відкритих даних,* що виник на основі таких політичних тенденцій, як поштовх до відкритого уряду та відкритої науки, заохочував обмін і взаємозв’язок різнорідних дослідницьких даних через великі цифрові інфраструктури. Наявність величезних обсягів даних у машиночитаних форматах створює стимул для створення ефективних процедур збору, організації, візуалізації та моделювання цих даних. Ці інфраструктури, у свою чергу, служать платформами для розвитку штучного інтелекту з метою підвищення надійності, швидкості та прозорості процесів створення знань. Дослідники з усіх дисциплін бачать, що нова здатність зв’язувати та перехресно посилатися на дані з різних джерел покращує точність і прогностичну силу наукових висновків і допомагає визначити майбутні напрямки дослідження, таким чином, зрештою, забезпечуючи нову відправну точку для емпіричного дослідження. Як свідчить зростання цільового фінансування, навчальних програм і місць публікацій, великі дані широко розглядаються як започаткування нового способу проведення досліджень і кидання виклику існуючому розумінню того, що вважається науковим знанням.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи big data наукових досліджень.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи big data наукових досліджень.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем big data наукових досліджень.
– Дослідження системи big data наукових досліджень.
– Програмна реалізація системи big data наукових досліджень.

Об'єктом дослідження є процес big data наукових досліджень.

Предметом дослідження є методи big data наукових досліджень.

Методи дослідження базуються на методах big data, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

У цій роботі досліджуються ці твердження щодо використання великих даних у наукових дослідженнях і з наголосом на філософських питаннях, які виникають у результаті такого використання. З цією метою у роботі обговорюється, як поява великих даних – і пов'язаних з ними технологій, інститутів і норм – інформує про аналіз наступних тем:

– як статистика, формальні та обчислювальні моделі допомагають екстраполювати закономірності з даних і з якими наслідками;
– роль критичного аналізу (людського інтелекту) у машинному навчанні та його зв'язок із зрозумілістю дослідницьких процесів;
– характер даних як компонентів дослідження;
– зв'язок між даними та доказами, а також роль даних як джерела емпіричного розуміння;
– погляд на знання як на теорії;
– розуміння зв'язку між прогнозом і причинністю;
– поділ факту і цінності; і
– ризики та етика науки про дані.

Це сфери, де увага до дослідницьких практик, що обертаються навколо великих даних, може принести користь філософії, і особливо роботі в епістемології та методології науки. Цей запис не охоплює величезну науку в історії та соціальних дослідженнях науки, яка виникла в останні роки на цю тему, хоча посилання на деякі з цієї літератури можна знайти, якщо це концептуально доречно. Доповнюючи історичну та соціальну наукову роботу в дослідженнях даних, філософський аналіз практик обробки даних також може викликати значні проблеми для ажіотажу навколо науки про дані та сприяти критичному розумінню ролі штучного інтелекту, що базується на даних, у дослідженнях.

Великі дані (big data)

Діяльність людини та взаємодія з навколишнім середовищем відстежуються та реєструються все ефективніше, де розробляються дедалі складніші обчислювальні інструменти для отримання знань із таких даних. Одним із прикладів є використання різних даних, отриманих від хворих на рак, включаючи геномні послідовності, фізіологічні вимірювання та індивідуальні реакції на лікування, для покращення діагностики та лікування. Іншим прикладом є інтеграція даних про транспортні потоки, навколишні та географічні умови, а також поведінку людини для розробки заходів безпеки для безпілотних транспортних засобів, щоб у разі непередбачуваних подій (наприклад, дитина раптово вибігла на вулицю в дуже холодний день), дані можна швидко проаналізувати, щоб ідентифікувати та сформувати відповідну реакцію (автомобіль повертає достатньо, щоб уникнути дитини, а також мінімізує ризик заносу на льоду та пошкодження інших транспортних засобів). Ще одним прикладом є розуміння харчового статусу та потреб конкретної групи населення, яке можна витягти з об'єднання даних про споживання їжі, отриманих комерційними службами (наприклад, супермаркетами, соціальними мережами та ресторанами), з даними, що надходять від громадської охорони здоров'я та соціальних служб, наприклад результати аналізів крові та госпіталізації, пов'язані з недоїданням. У

кожному з цих випадків доступність даних і відповідних аналітичних інструментів створює нові можливості для дослідження та розробки нових форм дослідження, які широко сприймаються як такі, що мають трансформаційний вплив на науку в цілому.

Корисною відправною точкою для роздумів про значення таких випадків для філософського розуміння дослідження є розгляд того, що насправді означає термін «великі дані» в сучасному науковому дискурсі. Існує кілька способів визначення великих даних (Kitchin 2014, Kitchin & McArdle 2016). Мабуть, найпростішою характеристикою є *великі* набори даних, які створюються в *цифровій* формі та можуть аналізуватися за допомогою *обчислювальних* інструментів. Отже, дві характеристики, які найчастіше асоціюються з великими даними, – це обсяг і швидкість. *Обсяг* означає розмір файлів, які використовуються для архівування та поширення даних. *Швидкість* означає швидкість натискання, з якою дані генеруються та обробляються. Обсяг цифрових даних, створених дослідженнями, зростає шаленою швидкістю та такими способами, які, мабуть, неможливо осягнути людською когнітивною системою, і тому вимагають певної форми автоматизованого аналізу.

Альтернативою є визначення великих даних не через посилання на їхні фізичні атрибути, а скоріше через те, що можна і що не можна з ними робити. З цієї точки зору, великі дані – це різнорідний ансамбль даних, зібраних із різних джерел, як правило (але не завжди) у цифрових форматах, придатних для алгоритмічної обробки, з метою створення нових знань. Наприклад, Бойд і Кроуфорд (2012: 663) ототожнюють великі дані зі «здатністю шукати, агрегувати та перехресно посилатися на великі набори даних», тоді як О'Меллі та Соєр (2012) зосереджуються на здатності опитувати та взаємозв’язувати різні типи даних., щоб мати можливість ознайомитися з ними як з єдиною сукупністю доказів. Приклади трансформаційних «досліджень великих даних», наведені вище, легко вписуються в цю точку зору: це не просто факт, що доступно багато даних, що робить відмінність у цих випадках, а скоріше той факт, що багато даних можна мобілізувати з різноманітних джерел (медичні записи, дослідження навколишнього середовища, вимірювання погоди, поведінка споживачів).

Таке розуміння великих даних сягає корінням у довгу історію дослідників, які стикаються з великими та складними наборами даних, прикладами яких є астрономія, метеорологія, таксономія та демографія (див. колекції, зібрані Daston 2017; Anorova та ін. 2017; Porter & Chaderavian). 2018; також Анорова та ін., Сепкоскі 2013, Стівенс 2019). Подібним чином біомедичні дослідження – і особливо такі підгалузі, як епідеміологія, фармакологія та громадське здоров’я – мають широку традицію роботи з даними великого обсягу, швидкості, різноманітності та мінливості, достовірність, достовірність і цінність яких регулярно обговорюється та оскаржується пацієнтами, урядами., спонсори, фармацевтичні компанії, страхові компанії та державні установи (Bauer 2008). Протягом двадцятого століття ці зусилля стимулювали розвиток методів, установ та інструментів для збору, упорядкування, візуалізації та аналізу даних, таких як: стандартні системи та формати класифікації; вказівки, інструменти та законодавство щодо управління та безпеки конфіденційних даних; та інфраструктури для інтеграції та підтримки зборів даних протягом тривалих періодів часу (Daston 2017).

Кульмінацією цієї роботи стало застосування обчислювальних технологій, інструментів моделювання та статистичних методів до великих даних (Porter 1995; Humphreys 2004; Edwards 2010), що все більше розширює межі аналітики даних завдяки керованому навчанню, підгонці моделі, глибоким нейронним мережам, пошуку та методи оптимізації, складні візуалізації даних та різноманітні інші інструменти, пов’язані зі штучним інтелектом. Багато з цих інструментів базуються на алгоритмах, функціонування та результати яких перевіряються на конкретних зразках даних (цей процес називається «навчанням»). Ці алгоритми запрограмовані на «навчання» з кожної взаємодії з новими даними: іншими словами, вони мають здатність змінювати себе у відповідь на нову інформацію, що вводиться в систему, таким чином стаючи більш налаштованими на явища,

які вони аналізують, і покращуючи свої здатність передбачати майбутню поведінку. Обсяг і ступінь таких змін визначаються припущеннями, які використовуються для побудови алгоритмів, а також здатністю відповідного програмного та апаратного забезпечення ідентифікувати, отримувати доступ і обробляти інформацію, яка має відношення до відповідного навчання. Однак існує певний ступінь непередбачуваності та непрозорості цих систем, які можуть розвиватися до такого рівня, що кидає виклик людському розумінню (докладніше про це нижче).

Також з'явилися нові інституції, комунікаційні платформи та нормативні рамки для збирання, підготовки та підтримки даних для такого використання (Kitchin 2014), наприклад, різні форми інфраструктури цифрових даних, організації, які прагнуть координувати та вдосконалювати глобальний ландшафт даних (наприклад, Research Data Alliance), а також нові заходи щодо захисту даних, як-от Загальний регламент захисту даних, прийнятий у 2017 році Європейським Союзом. Разом ці методи та інституції дають можливість збирати та інтерпретувати дані в набагато ширшому масштабі, а також обіцяють забезпечити вищий рівень деталізації в аналізі даних. [1] Вони розширюють масштаби будь-якого дослідження, даючи дослідникам можливість пов'язувати власні висновки з висновками незліченної кількості інших у всьому світі, як у академічній сфері, так і за її межами. Підвищуючи мобільність даних, вони полегшують їх перепрофілювання для різноманітних цілей, які могли бути непередбачуваними під час початкового створення даних. Трансформуючи роль даних у дослідженнях, вони самі по собі підвищують свій статус цінних результатів дослідження. Ці технологічні та методологічні розробки мають значні наслідки для філософської концептуалізації даних, процесів висновків і наукових знань, а також для того, як дослідження проводяться, організовуються, керуються та оцінюються. Саме до цих філософських проблем я зараз звертаюся.

Екстраполяція шаблонів даних: роль статистики та програмного забезпечення

Великі дані часто асоціюють з ідеєю дослідження, *керованого даними*, де навчання відбувається через накопичення даних і застосування методів для вилучення значущих моделей із цих даних. Очікується, що в рамках дослідження, керованого даними, дослідники використовуватимуть дані як відправну точку для індуктивного висновку, не покладаючись на теоретичні упередження – ситуацію, яку прихильники описують як «кінець теорії», на відміну від підходів, керованих теорією, де дослідження складається з перевірка гіпотези (Anderson 2008, Hey et al. 2009). Принаймні в принципі, великі дані становлять найбільший пул даних, який коли-небудь збирався, і, отже, сильну відправну точку для пошуку кореляцій (Mayer-Schönberger & Cukier 2013). Вирішальним для достовірності підходу, керованого даними, є ефективність методів, що використовуються для екстраполяції шаблонів із даних і оцінки того, чи є такі шаблони значущими чи ні, і яке «значення» може включати в себе в першу чергу. Тому деякі філософи та дослідники даних стверджують це

– найважливішою та відмінною характеристикою великих даних є використання статистичних методів і обчислювальних засобів аналізу (Symons & Alvarado 2016: 4).

– наприклад, інструменти машинного навчання, глибокі нейронні мережі та інші «інтелектуальні» практики обробки даних.

Акцент на статистиці як ключовому критерії достовірності та надійності моделей, отриманих із даних, не є новим. Прихильники логічного емпіризму шукали логічно надійні методи для забезпечення та виправдання висновків на основі даних, і їхні зусилля з розробки теорії ймовірності йшли паралельно з укоріненням статистичних міркувань у науках у першій половині двадцятого століття (Romeijn 2017). . На початку 1960-х років Патрік Суппес запропонував фундаментальний зв'язок між статистичними методами та філософією науки завдяки своїй роботі над створенням та інтерпретацією моделей даних. Як філософ, глибоко вкорінений в експериментальній практиці, Суппес цікавився засобами та мотивацією ключових статистичних процедур для аналізу даних, таких як редукція даних і підгонка кривої. Він стверджував, що як тільки дані належним чином *підготовлені* для статистичного моделювання, усі проблеми та вибір, які мотивували обробку даних, стають

неактуальними для їх аналізу та інтерпретації. Це надихнуло його розрізнити моделі теорії, моделі експерименту та моделі даних, зазначивши, що такі різні компоненти дослідження керуються різною логікою і не можуть порівнюватися прямолінійним способом. Наприклад, точне визначення моделей даних для будь-якого даного експерименту вимагає наявності теорії даних у сенсі експериментальної процедури, а також у звичайному розумінні емпіричної теорії явищ, що вивчаються. (Suppes 1962: 253)

Суппес розглядав моделі даних як обов'язково статистичні, тобто як об'єкти призначений для включення всієї інформації про експеримент, яка може бути використана в статистичних перевірках адекватності теорії. (Suppes 1962: 258)

Що повинно входити в моделі даних? Основним обмеженням є потреба в моделях даних, які дозволяють статистичну оцінку відповідності (між прогнозом і фактичними даними); (Mayo 1996: 136) і Бас ван Фраассен, який також прийняв ідею моделей даних як «узагальнення відносних частот, знайдених у даних» (Van Fraassen 2008: 167). Тісно пов'язаний наголос на статистиці як на засобі виявлення помилок у наборах даних щодо конкретних гіпотез, найбільш помітно схвалених статистичним підходом до висновку про помилки, який відстоюють Мейо та Аріс Спанос (Mayo & Spanos 2009a). Цей підхід узгоджується з наголосом на обчислювальних методах для аналізу даних у рамках дослідження великих даних і підтримує ідею про те, що чим кращі інструменти та методи висновків, тим більше шансів витягти надійні знання з даних.

Проте, коли справа доходить до вирішення методологічних проблем, пов'язаних з обчислювальним аналізом великих даних, статистичний досвід має бути доповнений обчислювальною кмітливістю в навчанні та застосуванні алгоритмів, пов'язаних зі штучним інтелектом, включаючи машинне навчання, а також інші математичні процедури для роботи з даних (Bringsjord & Govindarajulu 2018). Розглянемо, наприклад, проблему переобладнання, тобто помилкову ідентифікацію шаблонів у наборі даних, яка може бути значно посилена методами навчання, що використовуються алгоритмами машинного навчання. Немає жодної гарантії, що алгоритм, навчений для успішної екстраполяції шаблонів із даного набору даних, буде таким же успішним, коли його застосувати до інших даних. Загальні підходи до цієї проблеми передбачають перевпорядкування та розділення як даних, так і методів навчання, щоб можна було порівняти застосування одних і тих самих алгоритмів до різних підмножин даних («перехресна перевірка»), об'єднати передбачення, що виникають із по-іншому навчених алгоритмів («ансамблювання») або використовувати гіперпараметри (параметри, значення яких встановлюються до навчання даних), щоб підготувати дані для аналізу.

Вирішення цих проблем, у свою чергу, вимагає знайомство з математичними операціями, про які йде мова, їх реалізацією в коді та апаратною архітектурою, що лежить в основі таких реалізацій. (Лоурі 2017: 3)

Наприклад, машинне навчання націлене на створення програм, які розробляють власні аналітичні або описові підходи до сукупності даних, а не використовують готові рішення, такі як дедукція на основі правил або регресії більш традиційної статистики. (Лоурі 2017: 4)

Іншими словами, статистика та математика мають бути доповнені досвідом у програмуванні та комп'ютерній інженерії. Сукупність навичок, витлумачених таким чином, призводить до специфічного епістемологічного підходу до дослідження, який загалом характеризується наголосом на засобах дослідження як найважливішому рушії дослідницьких цілей і результатів. Цей підхід, який Сабіна Леонеллі охарактеризувала як *орієнтований на дані*, передбачає «більше зосередження на процесах, за допомогою яких здійснюється дослідження, ніж на його кінцевих результатах» (Leonelli 2016: 170). З цієї точки зору, процедури, техніки, методи, програмне забезпечення та апаратне забезпечення є основними двигунами дослідження та головним впливом на його результати. Зосереджуючись більш конкретно на обчислювальних системах, Джон Саймонс і Джек Хорнер стверджували, що більша частина дослідження великих даних складається з *наукових досліджень, які інтенсивно займаються програмним забезпеченням,* а не досліджень,

керованих даними: тобто наука, яка залежить від програмного забезпечення для свого проектування, розробки, розгортання та використання, і, таким чином, охоплює процедури, типи міркувань і помилки, які є унікальними для програмного забезпечення, наприклад, проблеми, породжені спробами відобразити величини реального світу на дискретних автоматах або наближення числових операцій (Symons & Horner 2014: 473). Наука, яка інтенсивно займається програмним забезпеченням, мабуть, підтримується *алгоритмічною раціональністю,* зосередженою на здійсненності, практичності та ефективності алгоритмів, які зазвичай оцінюються на основі конкретних ситуацій дослідження (Lowrie 2017).

Людський і штучний інтелект

Алгоритми надзвичайно різноманітні за своїми математичними структурами та концептуальними зобов'язаннями, тому необхідно провести більше філософської роботи над специфікою обчислювальних інструментів і програмного забезпечення, що використовується в науці про дані та пов'язаних із ними додатках. Нові роботи з філософії інформатики пропонують чудовий спосіб вперед (Turner & Angius 2019). Тим не менш, зрозуміло, що те, чи буде певний алгоритм успішно застосовуватися до наявних даних, залежить від факторів, які неможливо контролювати за допомогою статистичних чи навіть обчислювальних методів: наприклад, розмір, структура та формат даних, природа класифікатори, які використовуються для поділу даних, складність меж прийняття рішень і самі цілі дослідження.

У сильній критиці, заснованій на філософії математики, Крістіан Калуде та Джузеппе Лонго стверджували, що існує фундаментальна проблема з припущенням, що більше даних обов'язково дасть більше інформації:

дуже великі бази даних повинні містити довільні кореляції. Ці кореляції виникають лише через розмір, а не через природу даних. (Calude & Longo 2017: 595)

Вони прийшли до висновку, що аналіз великих даних за визначенням не здатний відрізнити хибні кореляції від значущих і тому становить загрозу для наукових досліджень. Пов'язане занепокоєння, яке інколи називають «прокляттям розмірності» дослідниками обробки даних, стосується того, наскільки аналіз певного набору даних можна збільшити за складністю та за кількістю змінних, що розглядаються. Добре відомо, що чим більше вимірів враховується при класифікації вибірок, наприклад, тим більший набір даних, на якому такі виміри можна точно узагальнити. Це демонструє постійну, тісну залежність між обсягом і якістю даних, з одного боку, і типом і широтою дослідницьких питань, для яких дані повинні служити доказами, з іншого боку.

Визначення відповідності між методами логічного висновку та даними вимагає високого рівня знань і контекстуального судження (ситуація, відома в машинному навчанні як «теорема про відсутність безкоштовного обіду»). Дійсно, надмірна залежність від програмного забезпечення для висновків і моделювання даних може призвести до дуже проблематичних результатів. Саймонс і Хорнер відзначають, що використання складного програмного забезпечення в аналізі великих даних робить межі похибки невизначеними, оскільки немає чіткого способу їх статистичного тестування (Саймонс і Хорнер 2014: 473). Складність шляху програм із високою умовністю накладає обмеження на стандартні методи виправлення помилок. Як наслідок, не існує ефективного методу для характеристики розподілу помилок у програмному забезпеченні, окрім тестування всіх шляхів у коді, що є нереалістичним і важкорозв'язаним у переважній більшості випадків через складність коду.

Замість того, щоб діяти як заміна, ефективне та відповідальне використання інструментів штучного інтелекту в аналізі великих даних вимагає стратегічного вправляння людського інтелекту, але для цього системи штучного інтелекту, які застосовуються до великих даних, повинні бути доступними для перевірки та модифікації. Так це чи ні, і хто найкраще кваліфікований для здійснення такого контролю, залишається предметом суперечок. Томас Ніклс стверджував, що все більш складні та розподілені алгоритми, які використовуються для аналізу даних, йдуть слідами давніх наукових спроб вийти за межі людського пізнання. Отримані в результаті епістемічні системи можуть більше не бути

зрозумілими для людей: «інопланетний інтелект», у межах якого «людські здібності більше не є основним критерієм епістемічного успіху» (Ніклз готовий до публікації). Таке необмежене пізнання обіцяє можливість потужного логічного висновку на основі раніше немислимих обсягів даних. Однак труднощі в контекстуалізації та ретельному аналізі таких міркувань ставлять під сумнів надійність результатів. Не тільки алгоритми машинного навчання стають дедалі недоступнішими для оцінювання: крім складності програмного коду, аналіз обчислювальних даних вимагає цілої екосистеми класифікацій, моделей, мереж і інструментів логічного висновку, які зазвичай мають різну історію та цілі, і які пов'язані з один одного – і ефекти, коли вони використовуються разом – далекі від розуміння і цілком можуть бути непростеженими.

Це ставить питання про те, чи знання, створені такими аналітичними системами даних, взагалі зрозумілі людям, і якщо так, то які форми зрозумілості вони дають. Безумовно, отримання знань із великих даних може не передбачати підвищення людського розуміння, особливо якщо розуміння розуміти як епістемічну навичку (de Regt 2017). Це може не бути проблемою для тих, хто чекає на появу нового виду розумних машин, які можуть оволодіти новими когнітивними інструментами так, як не можуть люди. Але, як зазначали Ніклс, Ніколас Решер (1984), Вернер Каллебаут (2012) та інші, навіть у такому випадку «ми б не досягли науки без перспективи» (Ніклз буде випущено). Хоча людські історії та припущення, вплетені в ці системи, може бути важко роз'єднати, вони все одно впливають на їхні результати; і незалежно від того, чи є ці процеси дослідження відкритими для критичного розгляду, їх телос, наслідки та значення для життя на планеті, мабуть, мають бути такими. Як стверджував Ден МакКвіллан (2018), зростаюча автоматизація аналітики великих даних може сприяти прийняттю неоплатонічної *машинної метафізики*, у рамках якої математичні структури, «розкриті» штучним інтелектом, переважатимуть будь-яке звернення до людського досвіду. Лучано Флоріді повторює цю інтуїцію у своєму аналізі того, що він називає *інфосферою* :

Великі можливості, які пропонують інформаційно-комунікаційні технології, супроводжуються величезною інтелектуальною відповідальністю зрозуміти їх і правильно скористатися ними. (2014: vii)

Ці міркування відповідають давній критиці Пола Хемфріса комп'ютерного моделювання як *епістемічно непрозорого* (Humphreys 2004, 2009) – і зокрема його визначення того, що він називає *суттєвою* епістемічною непрозорістю:

Процес по суті є епістемічно непрозорим для *X* тоді і тільки тоді, *коли* для *X неможливо* знати всі епістемічно релевантні елементи процесу. (Хамфріс 2009: 618)

Різні аспекти загальної проблеми епістемічної непрозорості наголошуються в широкому філософському дослідженні про роль моделювання, обчислення та моделювання в науці: наслідки відсутності експериментального доступу до конкретних частин світу, що моделюється, наприклад (Morgan 2005). ; Паркер 2009; Раддер 2009); труднощі у перевірці надійності обчислювальних методів, що використовуються в рамках моделювання (Winsberg 2010; Morrison 2015); зв'язок між непрозорістю та виправданням (Durán & Formanek 2018); форми чорного ящика, пов'язані з механістичними міркуваннями, реалізованими в обчислювальному аналізі (Craver and Darden 2013; Bechtel 2016); і дебати щодо внутрішніх обмежень обчислювальних підходів і відповідного досвіду (Коллінз 1990; Дрейфус 1992). Роман Фрігг і Джуліан Райс стверджували, що такі проблеми не є фундаментальними проблемами для природи дослідження та моделювання, а фактично існують у континуумі з традиційними методологічними проблемами, добре відомими в науці (Frigg & Reiss 2009). Незалежно від того, погоджується хтось із цією позицією чи ні (Humphreys 2009; Beisbart 2012), аналіз великих даних явно розширює можливості обчислювальних і статистичних методів, таким чином підкреслюючи межі того, що навіть технологічно вдосконалені люди здатні знати та розуміти.

Природа (великих) даних

Таким чином, дослідження аналізу великих даних проливає світло на елементи дослідницького процесу, які неможливо повністю контролювати, раціоналізувати чи навіть розглянути за допомогою офіційних інструментів.

Одним із таких елементів є робота, необхідна для представлення емпіричних даних у машиночитаному форматі, сумісному з наявним програмним забезпеченням та аналітичними інструментами. Дані потрібно відібрати, очистити та підготувати для статистичного та обчислювального аналізу. Процеси, пов'язані з відокремленням даних від шуму, кластеризацією даних, щоб їх можна було простежити, та інтеграцією даних різних форматів виявилися дуже складними та теоретично структурованими, як продемонстрували, наприклад, Джеймс Макаллістер (1997, 2007, 2011) та Ульяна Фіст. (2011) робота над моделями даних, порівняння Марселем Бумансом і Леонеллі принципів кластеризації в різних галузях (готується до публікації), а також аналізи особливостей наборів даних Джеймсом Ґріземером (готується до друку) і Мері Морган (готується до друку). Суппес був настільки стурбований тим, що він назвав «дивовижною складністю» виробництва та обробки даних, що він хвилювався, що філософи не оцінять способи, якими статистика може і допомагає вченим абстрагувати дані від такої складності. Він описав велику групу дослідницьких компонентів і заходів, які використовуються для підготовки даних для моделювання, як «прагматичні аспекти», що охоплюють «кожне інтуїтивне розгляд експериментального плану, що не передбачає формальної статистики» (Suppes 1962: 258), і позиціонує їх як найнижчий рівень його ієрархія моделей – на протилежному кінці її вершини, якою є моделі теорії. Незважаючи на нещодавні спроби реабілітації методології індуктивно-статистичного моделювання та логічного висновку (Mayo & Spanos 2009b), цей підхід поділяється багатьма філософами, які вважають процеси виробництва та обробки даних настільки хаотичними, що не піддаються систематичному аналізу. Це пояснює, чому дані отримали так мало уваги у філософії науки порівняно з моделями та теорією.

Однак питання про те, як дані визначаються та ідентифікуються, є вирішальним для розуміння ролі великих даних у наукових дослідженнях. Давайте тепер розглянемо дві філософські точки зору – *репрезентативну* та *реляційну* – обидва сумісні з появою великих даних, але при цьому акцентуємо увагу на різних аспектах цього явища, що має значні наслідки для розуміння ролі даних у висновках. і, як ми побачимо в наступному розділі, як доказ. Репрезентативний *погляд* тлумачить дані як надійні уявлення про реальність, створені через взаємодію між людьми та світом. Взаємодії, які генерують дані, можуть відбуватися в будь-якому соціальному середовищі незалежно від цілей дослідження. Приклади варіюються від біолога, який вимірює окружність клітини в лабораторії та записує результат у файл Excel, до вчителя, який підраховує кількість учнів у своєму класі та записує це в класний журнал. Даними в цих взаємодіях вважаються об'єкти, створені в процесі опису та/або вимірювання світу. Ці об'єкти можуть бути цифровими (файл Excel) або фізичними (реєстр класів) і формувати відбиток певної взаємодії з природним світом. Цей відбиток – «слід» або «мітка», за словами Яна Хекінга (1992) і Ханса-Йорга Рейнбергера (2011), відповідно, є важливою точкою відліку для аналітичного дослідження та для отримання нових ідей. Ось чому дані формують законну основу для емпіричного знання: виробництво даних еквівалентно «захопленню» особливостей світу, які можна використовувати для систематичного вивчення. Відповідно до репрезентативного підходу, дані – це об'єкти з фіксованим і незмінним змістом, значення яких через те, що вони представляють реальність, необхідно досліджувати та розкривати крок за кроком за допомогою адекватних методів висновку. Дані, що документують форму клітини, можна моделювати, щоб перевірити відповідність форми еластичності, проникності та стійкості клітин, створюючи доказову базу для розуміння передачі сигналів між клітинами та розвитку. Отримані дані підрахунку учнів у класі можна об'єднати з аналогічними даними, зібраними в інших школах, створюючи доказову базу для оцінки щільності учнів у цьому районі та частоти відвідування ними школи.

Це відображає інтуїцію про те, що дані, особливо коли вони надходять у формі числових вимірювань або зображень, таких як фотографії, якимось чином віддзеркалюють явища, для документування яких вони створені, створюючи таким чином моментальний знімок цих явищ, який можна вивчати в контрольованих умовах. досліджень. Це також відображає ідею даних як «необроблених» продуктів дослідження, які максимально наближені до безпосереднього знання реальності. Це має сенс істинного значення, яке іноді приписують даним як неспростовним джерелам доказів – ідея Поппера про те, що якщо знайдено дані, що підтверджують дане твердження, то це твердження підтверджується як істинне принаймні до тих пір, поки не знайдено інших даних, спростувати це. У цій точці зору дані являють собою об'єктивну основу для отримання знань, і саме ця об'єктивність – здатність отримувати знання з людського досвіду, виходячи за його межі – робить знання емпіричними. Ця позиція добре узгоджується з ідеєю, що великі дані є цінними для науки, оскільки вони сприяють індуктивному накопиченню знань (у широкому розумінні): збір даних, зібраних за допомогою надійних методів, створює гору фактів, готових до аналізу, і чим більше фактів створені та пов'язані один з одним, тим більше знань можна отримати.

Філософи давно визнали, що дані не говорять самі за себе, а різні типи даних вимагають різних інструментів для аналізу та підготовки для інтерпретації (Bogen 2009 [2013]). Згідно з репрезентативною точкою зору, існують правильні та неправильні способи інтерпретації даних, які особи, відповідальні за аналіз даних, повинні розкрити. Але що таке «правильна» інтерпретація у сфері великих даних, де дані послідовно розглядаються як мобільні об'єкти, які, принаймні в принципі, можна повторно використовувати незліченною кількістю способів і для досягнення різних цілей? Можливо, більше, ніж будь-коли в історії науки, нинішня мобілізація та повторне використання великих даних підкреслює ступінь, до якого інтерпретація даних – а разом з цим і будь-які дані, які представляють – можуть відрізнятися залежно від концептуальних, матеріальних і соціальні умови розслідування. Аналіз того, як великі дані переміщуються між контекстами, показує, що очікування та здібності тих, хто бере участь, визначають не лише спосіб інтерпретації даних, але й те, що в першу чергу вважається «даними» (Леонеллі та Темпіні, готові до публікації). Представницький погляд на дані як на об'єкти з фіксованим і контекстуально незалежним значенням суперечить цим спостереженням.

Альтернативний підхід полягає в тому, щоб прийняти ці висновки та повністю відмовитися від ідеї даних як фіксованого відображення реальності. У рамках *реляційного погляду* дані – це об'єкти, які розглядаються як потенційні чи фактичні докази наукових тверджень у спосіб, який, принаймні в принципі, можна ретельно перевірити та врахувати (Leonelli 2016). Значення, яке надається даним, залежить від їхнього походження, фізичних особливостей і того, що ці характеристики представляють, а також мотивів та інструментів, які використовуються для їх візуалізації та захисту конкретних інтерпретацій. Таким чином, надійність даних залежить від достовірності та чіткості процесів, які використовуються для їх отримання та аналізу. Подання даних; спосіб їх визначення, відбору та включення (або виключення) до баз даних; та інформація, яка надається користувачам для їх реконтекстуалізації, є фундаментальною для отримання знань і значно впливають на їх зміст. Наприклад, зміни у форматі даних – що, очевидно, пов'язано з процедурами оцифрування, стиснення даних або архівування – можуть мати значний вплив на те, де, коли та хто використовує дані як джерело знань.

Цей фреймворк визнає, що будь-який об'єкт можна використовувати як даний або припинити використовувати як такий, залежно від обставин – міркування, знайоме аналітикам великих даних, які звикли вибирати та змішувати дані, що надходять із великої різноманітності джерел. Реляційний погляд також пояснює, як, залежно від дослідницької перспективи, що його інтерпретує, один і той самий набір даних може використовуватися для представлення різних аспектів світу («явища», як їх знаменито охарактеризували Джеймс Боген і Джеймс Вудворд, 1988). Розглядаючи повний цикл наукового дослідження з точки зору виробництва та аналізу даних, саме на етапі *моделювання* даних даним надається певна

репрезентативна цінність (Leonelli 2019b).

Реляційний погляд на дані спонукає звернути увагу на історію даних, висвітлюючи їх постійну еволюцію, а іноді й радикальні зміни, а також вплив цієї функції на здатність даних підтверджувати чи спростовувати гіпотези. Це пояснює критичну важливість документування процесів управління даними та перетворення, особливо з великими даними, які передаються далеко й широко цифровими каналами та групуються та інтерпретуються різними способами та форматами. Це також пояснює зростаюче визнання досвіду тих, хто створює, курує та аналізує дані, як необхідного для ефективної інтерпретації великих даних у межах науки та за її межами; і нерозривний зв'язок між соціальними та етичними проблемами щодо потенційного впливу обміну даними та науковими проблемами щодо якості, дійсності та безпеки даних (boyd & Crawford 2012; Tempini & Leonelli, 2018).

Залежно від погляду на дані, очікування щодо того, що великі дані можуть зробити для науки, різко відрізнятимуться. Репрезентативний погляд враховує ідею великих даних як забезпечення найповнішої, надійної та генеративної бази знань, яку будь-коли бачили в історії науки, завдяки її величезному розміру та неоднорідності. Реляційний погляд не бере на себе таких зобов'язань, зосереджуючись замість цього на тому, які висновки робляться з таких даних у будь-який момент, як і чому.

Великі дані та докази

Єдине, в чому погоджуються репрезентативні та реляційні погляди, – це ключова епістемічна роль даних як емпіричних доказів претензій щодо знань або втручань. Хоча існує велика кількість філософської літератури про природу доказів (наприклад, Achinstein 2001; Reiss 2015; Kelly 2016), проте зв'язку між даними та доказами приділено менше уваги. Можливо, це пов'язано з неявним прийняттям багатьма філософами репрезентативного погляду на дані. У рамках репрезентативного погляду ідентифікація того, що вважається даними, передує дослідженню того, для чого ці дані можуть бути доказами: іншими словами, дані є «даними», як вказує етимологія слова, а методи висновку відповідають за визначення того, чи і як дані, доступні слідчим, можуть бути використані як докази та для чого. Таким чином, фокус філософської уваги зосереджений на формальних методах виділення помилок і оманливих інтерпретацій, а також на ймовірнісному та/або пояснювальному відношенні між тим, що без проблем вважається набором доказів, і даною гіпотезою. Тому велика частина обширних філософських робіт про докази взагалі уникає терміна «дані». Основна робота Пітера Ахінштейна є яскравим прикладом: у ній обговорюються спостережувані факти та експериментальні результати, а також те, чи будуть у вчених підстави вірити таким фактам і за яких умов, але в ній не згадується дані та відповідна практика обробки (Achinstein 2001).

Навпаки, у реляційному поданні об'єкт можна ідентифікувати як дані лише тоді, коли він розглядається як такий, що має цінність як доказ. Докази стають категорією ідентифікації даних, а не категорією використання даних, як у репрезентативному погляді (Canali 2019). Таким чином, доказ є конститутивним для самого поняття даних і не може бути відокремлений від нього. Це передбачає прийняття того, що умови, за яких даний об'єкт може служити доказом – і, отже, розглядатися як дані – можуть змінюватися; і якщо ця доказова роль повністю припиниться, об'єкт знову перетвориться на звичайний елемент, що не є датою. Наприклад, фотографія рослини, зроблена туристом у віддаленому регіоні, може стати доказом для дослідження морфології рослин з цієї конкретної місцевості; проте більшість фотографій рослин ніколи не розглядаються як докази для дослідження особливостей і функціонування світу, а з тих, які є, багато з них можуть згодом бути відкинуті як нецікаві або більше не стосуються поставлених питань.

Ця точка зору враховує мобільність і перепрофілювання, які характеризують використання великих даних, а також можливість того, що об'єкти, які спочатку не були згенеровані для того, щоб служити доказами, можуть бути згодом прийняті як такі. Розглянемо «мінімальний науковий принцип доказів» Мейо та Спаноса, який вони визначають таким чином:

Дані $\mathbf{x}_0$ надають погані докази для *H,* якщо вони є результатом методу чи процедури,

які мають незначну або зовсім не здатну знаходити недоліки в *H*, навіть якщо *H* є хибним. (Mayo & Spanos 2009b)

Цей принцип сумісний із реляційним поглядом на дані, оскільки він включає випадки, коли методи, що використовуються для генерування та обробки даних, можливо, не були спрямовані на перевірку гіпотези H: все, що він вимагає, це те, щоб такі методи могли бути релевантними для тестування H, у той момент, коли дані використовуються як докази для H (я повернуся до ролі гіпотез у обробці доказів у наступному розділі).

Розробка структурної схеми

Для побудови структурної схеми ми розглянемо автоматизовані системи наукових досліджень, яка зображена на рисунку 1.

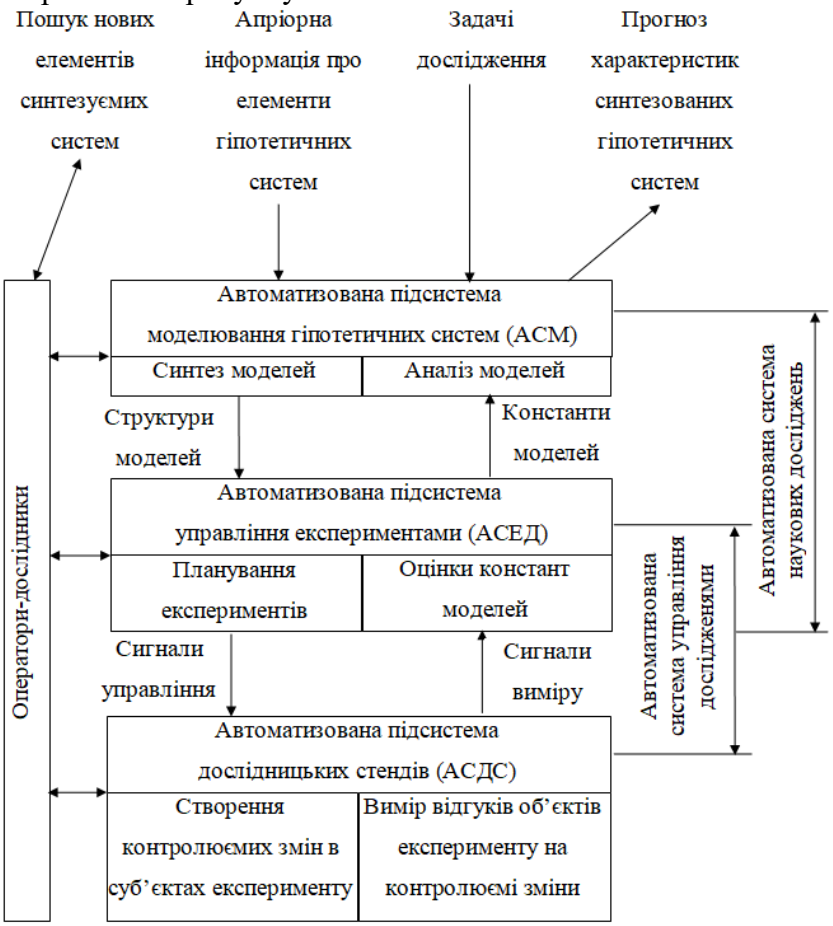


Рисунок 1 – Структурна схема системи

Висновки. У роботі наведені теоретичне узагальнення й рішення наукового завдання дослідження методів big data наукових досліджень.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем big data наукових досліджень.
– Досліджена система big data наукових досліджень.
– На основі отриманих результатів досліджень створена програмна реалізація системи big data наукових досліджень.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання big data наукових досліджень.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

Список літератури

1. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
2. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).
3. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
4. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
5. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
6. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
7. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
8. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
9. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
10. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
11. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
12. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
13. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
14. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
15. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
16. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
17. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 873-884. (Scopus).
18. Smirnov, O., Kuznetsov, A., Prokopovych-Tkachenko, D. «Hiding Data in Images Using a Pseudo-Random Sequence». ISCI'2020: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko, Victor A. Krasnobayev and Alexandr A. Kuznetsov. ASC Academic Publishing, USA, 2020. pp. 46-59. – ISBN: 978-1-7362833-0-1 (Hardback), ISBN: 978-1-7362833-1-8 (Ebook).
19. Smirnov, O., Kuznetsov, A., Shekhanin, K., Chepurko, I. Detecting Hidden Information in FAT. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 412-429. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
20. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

УДК 004

Б.Поляруш, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ БЮДЖЕТУВАННЯ ХМАРНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ЕФЕКТИВНОСТІ ЇЇ ВПРОВАДЖЕННЯ

У статті розроблено програмне забезпечення, яке призначено для системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. Метою розробки є дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. Об'єктом дослідження є процес бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. Предметом дослідження є методи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. Методи дослідження базуються на методах реалізації хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, бюджетування, хмарна інформаційна система, ефективність

Постановка проблеми. Однією з найважливіших задач створення й розвитку складних інформаційних комплексів підприємства (банку) є підвищення загальної конкурентоспроможності. Як приклад можна привести банківські інформаційні системи (БІС), що включають велику кількість підсистем, таких як центральні обчислювальні комплекси, автоматизовані системи обслуговування клієнтів, телекомунікаційні системи й т.п. Для підвищення конкурентоспроможності всієї БІС необхідно збільшувати ефективність роботи її компонентів. Наприклад, у частині локальних мереж ставиться задача вибору структури й параметрів налаштування стека протоколів. Звичайно проблеми виникають у системах, що вже функціонують досить тривалий час. Це пов'язане з тим, що обирані технології повинні бути сумісні з усім парком наявного устаткування, але при цьому бути перспективними. З обліком вищесказаного актуальність теми визначається необхідністю аналізу ефективності компонентів інформаційних систем, зокрема мережного середовища, з урахуванням показників різного характеру. Вибір набору показників і методик їхніх вимірів є складною науково-технічною задачею. У даному зв'язку основною метою дослідження є підвищення ефективності використання технологій локальних мереж у банківській сфері для збільшення загальної конкурентоспроможності всього підприємства (банку). Вибір об'єкта дослідження обумовлений тим, що такі мережі є в більшості випадків власністю банку, тому рішення поставленої в роботі задачі може дати тут найбільший ефект. Рішення поставленої задачі з використанням розроблених до теперішнього часу принципів відкритих систем на основі ТВС (технології відкритих систем) дозволяє реалізувати наступні найважливіші вимоги до кінцевого рішення: забезпечення масштабованості, необхідного рівня перспективності використовуваних технологій, а також узгодження протоколів на різних рівнях системи. Це дозволяє ефективно використовувати фінансові інвестиції на розвиток і супровід інформаційної системи підприємства (банку).

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її

впровадження. Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань: Огляд існуючих систем бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. Дослідження системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. Програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. *Об'єктом дослідження* є процес бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. *Предметом дослідження* є методи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. *Методи дослідження* базуються на методах реалізації хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Рішення задачі оцінки й підвищення ефективності функціонування БІС

Математична постановка задачі підвищення ефективності функціонування БІС

Приведемо особливості БІС:

– архітектуру мережі БІС (рисунок 1);

– основні задачі й функції, розв'язувані на різних рівнях ієрархії (таблиця 1);

– типи даних, переданих у мережах (таблиця 2).

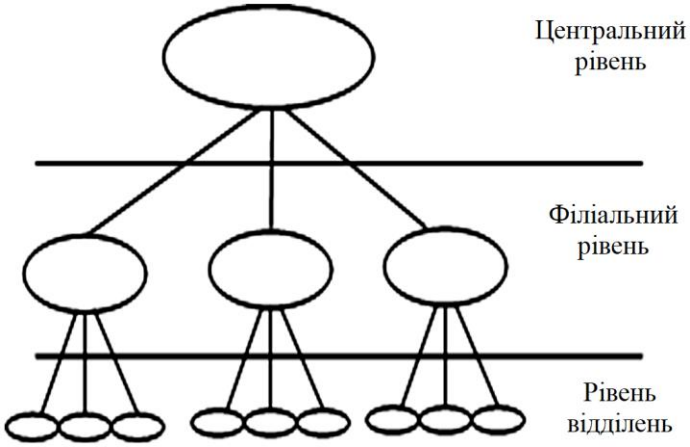


Рисунок 1 – Модель БІС

Таблиця 1 – Основні функції, виконувані на різних рівнях БІС

Функція	Рівні розміщення
1. Зберігання й обробка всієї банківської інформації. 2. Маркетинговий аналіз, розробка й просування послуг. 3. Внутрішній контроль і аудит. 4. Планування й проектування підрозділів.	Центр
1. Керування ризиками, витратами, капіталом. 2. Маркетинг ринків і клієнтів. 3. Аналіз конкурентів. 4. Електронні банківські послуги (банк-клієнт, І-банкинг).	Філії
1. Обслуговування клієнтів. 2. Валютні й інвестиційні операції. 3. Кредитування.	Відділення

Таблиця 2 – Структура мережних даних філії банку

Тип даних	Розмір переданих даних (Кб)	Частота появи (%)	Зразковий щоденний об'єм (Гб)
I	1 – 100	70 – 90	2 – 10
II	100 – 2 000	10 – 20	1 – 5
III	2 000 – 10 000	5 – 10	0,5 – 2

Математична постановка задачі підвищення ефективності функціонування БІС може бути визначена наступним чином. У якості вихідних даних приймається мережне середовище підприємства (банку), що може бути представлена в наступному виді:

$$\Psi = \{U, D, O, Q\}, \quad (1)$$

де:

– $U = \{u_i\}$, $i > 2$ – хости, підключених до локальної мережі підприємства (банку).

– $D = \{d_j\}$, $j \geq 1$ – типи даних переданих по мережі.

– $O = \{o_k\}$, $k > 2$ – типи мережних операцій.

– $^{Q} = \{q\}$, $l > 2$ – можливі стеки протоколів, що утворять мережне середовище.

– $q = \{x_1, x_2, .. x_m\}$, $1 \leq m \leq 7$, – набір протоколів різних рівнів, що утворять стек.

З огляду на, що замовник інформаційної системи висуває вимоги з використанням різних альтернативних показників, то необхідно використовувати вектор показників ефективності $\boldsymbol{\omega}$:

$$\omega = \{\omega_1, \omega_2, ...\omega_5\}, \quad (2)$$

де:

– $\omega = \mathrm{F}(\Psi, \mathrm{q})$ – вектор-функція й де кожний приватний показник може бути представлений у вигляді залежності $\omega_h = f_h(\Psi, \mathrm{q})$;

– h = 1...5 для різних стеків протоколів Q.

Таким чином, ми маємо багатокритеріальну задачу, для рішення якої можуть бути використані лексикографічні методи, методи головного й узагальненого показника, методи послідовних поступок і т.д. Аналіз перерахованих методів показав, що для рішення даної задачі доцільно використовувати метод узагальненого показника, що має наступний вид:

$$S = \sum_{h=1}^{5} \alpha_h \, \omega_h^*, \quad (3)$$

$$\sum_{h=1}^{5} \alpha_h = 1, \quad (4)$$

де:

– α_h – ваговий коефіцієнт h-*го* частки показника;

– ω_h^* – значення наведеного до однорідної величини показника ефективності.

У результаті багатокритеріальна задача зводиться до однокрітериальної і буде мати наступну постановку: визначити $\mathrm{q}_{opt} \subset Q$, при якому узагальнений показник ефективності приймає максимальне значення:

$$S_{\max} = \sum_{h=1}^{5} \alpha_h \, \omega_h^*(\Psi, \mathrm{q}_{opt}) \quad (5)$$

Таким чином, процедура рішення задачі дослідження зводиться до алгоритму, представленому на рисунку 1.

У результаті структурно-функціонального моделювання БІС філії виділені основні банківські функції й формовані ними мережні потоки.

Для оцінки формованого даними функціями мережного трафіку, вибору можливих технічних рішень їхньої реалізації, а також оцінці масштабованості обраних рішень расстрим процедуру побудови профілю середовища відкритої БІС філії.

Профіль середовища відкритої БІС філії

Опишемо побудову профілю середовища відкритої БІС філії відповідно до ТВС. При цьому до БІС застосовні основні вимоги відкритих систем: уніфікація обміну даними між компонентами БІС, забезпечення переносимості рішень між різними системами, а також забезпечення єдиного інтерфейсу для користувачів у різних системах. Даний профіль визначає набір стандартів і протоколів для локальних мереж, за допомогою яких варто реалізувати необхідний набір банківських функцій.

Відповідно до керівництва по проектуванню профілів середовища відкритої системи, проектування проводилося наступними етапами:

– аналіз вимог – декомпозиція функціональних служб на служби інформаційних систем;

– логічний проект – визначення зв'язку служб інформаційних технологій і інформаційних систем;

– фізичний проект – визначення базових стандартів, які можуть бути використані при побудові мережної структури локальної мережі.

На основі результатів фізичного проекту були проведені декомпозиції основних функцій банківської філії:

–по типах трафіку – визначення взаємозв'язку розглянутих функцій і типів даних, переданих у мережі (таблиця 3);

–по вимогах до бізнес-системи – знаходження можливих стеків протоколів локальних мереж для забезпечення даних функцій з урахуванням навантаження (таблиця 4).

Таблиця 3 – Таблиця розрахунку об'єму трафіку різного типу з урахуванням функціональних служб

Тип трафіку	Функції						Сумарний трафик по типах
	Офісна автоматизація	Системи керування й контролю	Системи фінансового прогнозування, аналізу й керування	Касові системи й системи обслуговування клієнтів	Системи на основі WEB і Банк-Клієнт	Автоматизовані банківські системи	
I	-	50 Мб	50 Мб	200 Мб	3000 Мб	5000 Мб	8300 Мб
II	700 Мб	50 Мб	100 Мб	-	2000 Мб	-	2850 Мб
III	300 Мб	-	50 Мб	-	-	-	350 Мб

Отримані таблиці можуть із успіхом застосовуватися при оцінці масштабування філіальної мережі. Для вибору оптимального стека протоколів з базових стандартів, отриманих на етапі фізичного проекту, надалі пропонується методика аналізу ефективності мережного середовища підприємства (банку).

Таблиця 4 – Матриця взаємозв'язку функціональних служб і технологічних вимог

Вимоги до бізнес-системі	Функції					
	Офісна автоматизація	Системи керування й контролю	Системи фінансового прогнозування, аналізу й управління	Касові системи й системи обслуж. клієнтів	Системи на основі WEB і Банк-Клієнт	Автоматизовані банківські системи
Число користувачів	20	2	5	50	1000	200
Число одночасних підключень	10	1	3	50	100	50
Об'єм даних у день	1Гб	100Мб	200Мб	200Мб	5Гб	5Гб
Транспортні й мережні протоколи	TCP/IP, IPX/SPX, NETBIOS	TCP/IP, IPX/SPX, NETBIOS	TCP/IP, IPX/SPX, NETBIOS	TCP/IP, IPX/SPX, NETBIOS	TCP/IP, IPX/SPX, NETBIOS	TCP/IP, IPX/SPX, NETBIOS
Канальні протоколи	Ethernet, Token Ring	Ethernet, Token Ring	Ethernet, Token Ring	Ethernet, Token Ring	Ethernet, Token Ring	Ethernet, Token Ring
Протоколи фізичного рівня	UTP cat 5	UTP cat 5	UTP cat 5	UTP cat 5	UTP cat 5	UTP cat 5

Методика аналізу ефективності мережного середовища підприємства (банку)

Розробимо методику аналізу ефективності мережного середовища підприємства (банку). На початковому етапі був визначений вектор показників ефективності $\boldsymbol{\omega}$, що відбивають основні аспекти поставленої задачі. Вибір показників був проведений методом експертних оцінок (метод анкетного опитування керівників підрозділів різних рівнів і напрямків). Для одержання комплексної оцінки були виділені показники різної природи (економічні, технічні й т.п):

– надмірність – показник ефективності використання пропускної здатності мережі;

– продуктивність – показник ефективності роботи мережі при різних навантаженнях;

– навантаження на устаткування – показник складності реалізації протоколів, що входять у стек;

– вартість рішення – показник ефективності фінансових вкладень;

– перспективність – показник можливості масштабування й розвитку даного рішення.

Розроблена методика розрахунку узагальненого показника ефективності S включає:

1. Вибір стеків протоколів, для яких проводяться дослідження Q.
2. Послідовне знаходження значень кожного приватного показника для кожного стека протоколів $\{\boldsymbol{\omega}\}$ (таблиця 5).
3. Приведення отриманих результатів по кожному показнику до безрозмірної величини від 1 до 4 (шляхом порівняння результатів на лінійній шкалі) {□□□□{□*□.
4. Знаходження значень узагальненого показника ефективності $\{S\}$.

5. Побудова таблиці для знаходження найбільш оптимальна стека q_{opt}.

Перші три показники є технічними, четвертий відбиває економічну ефективність. Методика розрахунку показника перспективності стосовно до даної предметної області вводиться вперше й відбиває можливості подальшого розвитку й застосування розглянутих технологій. При розрахунку значення узагальненого ефективності кожний показник уводиться з обліком його вагового коефіцієнта. Знаходження вагових коефіцієнтів є складною задачею. Для її рішення в роботі застосовувалися методи експертних оцінок (метод анкетного опитування фахівців інформаційних технологій в області локальних мереж).

Для визначення значень перших трьох показників при теоретичному дослідженні застосовувалися методи імовірнісного моделювання, а при експериментальному дослідженні – прямих і непрямих вимірів. Вартість рішення обчислювалася на основі маркетингового аналізу. Для визначення показника перспективності поряд з маркетинговим аналізом також використовувався аналіз рівня стандартизації розглянутих рішень.

Таблиця 5 – Методики оцінки показників ефективності стеків протоколів

Показники	Методики оцінки
Надмірність	$I = \frac{V_{служ.}}{V_{заг.}} = \frac{V_{заг.} - V_{кор.}}{V_{заг.}}$, де: – $V_{служ.}$ – об'єм службової інформації; – $V_{заг.}$ – загальний об'єм переданих даних; – $V_{кор.}$ – об'єм корисного навантаження.
Продуктивність	$C = T / V$, де: V – об'єм переданої інформації. $T = \frac{\sum_{1}^{N} T_i}{N}$, – де: T – час передачі інформації, обмірюваний на рівні додатку; N – число станцій.
Навантаження на устаткування	$L = (P^* + M^*)/2$ – значення навантаження на устаткування з розрахунку наведених безрозмірних значень P і M. $P = P_{експ.} - P_{поч.}$, $L = (P^* + M^*)/2$ де: – P – середнє значення лічильника навантаження на процесор; – $P_{експ.}$ – значення лічильника навантаження на процесор, обмірюване в експерименті при роботі з мережі; – $P_{поч.}$ – значення лічильника навантаження на процесор, обмірюване в експерименті при роботі локально. $M = M_{експ.} - M_{поч.}$ – середнє значення лічильника використання пам'яті, де: – $M_{експ.}$ – значення лічильника використання пам'яті, обмірюване в експерименті при роботі з мережі; – $M_{поч.}$ – значення лічильника використання пам'яті, обмірюване в експерименті при роботі локально.
Вартість рішення	$\Phi = \Phi_{CA} + \Phi_{H/S} + \Phi_{наст.}$, де: – Φ – вартість одного мережного підключення;

	– Φ_{NA} – вартість мережного адаптера; – $\Phi_{H/S}$ – вартість одного порту концентратора або комутатора; – $\Phi_{наст.}$ – вартість налаштування одного мережного підключення.
Перспективність	$A_{комп.i} = \frac{1}{4}\sum_{1}^{4} A_j$ – значення перспективності для кожного протоколу в стеці. $A = \frac{1}{n}\sum_{1}^{n} A_{комп.i}$ – загальна перспективність стека, де: – n – число протоколів у стеці 1. A_1 – кількість компаній виробників кінцевих рішень: одна (1), менш трьох (2), менш п'яти (3), більше п'яти (4); 2. A_2 – убудована підтримка в сучасні операційні системи: одна (1), дві (2), три (3), більше трьох (4); 3. A_3 – відношення до стандартизації: немає стандартів (1), готується до стандартизації (2), стандарт де-факто (3),стандарт де-юре (4); 4. A_4 – сфера застосування протоколу: скорочується (1), постійна (2), росте (3), швидко росте (4);
Узагальнений	$S = \sum_{h=1}^{5} \alpha_h \omega_h^*, \ \sum_{h=1}^{5} \alpha_h = 1,$ де – α_h – ваговий коефіцієнт h-го частки показника, – ω_h^* – значення наведеного до однорідної величини показника ефективності.

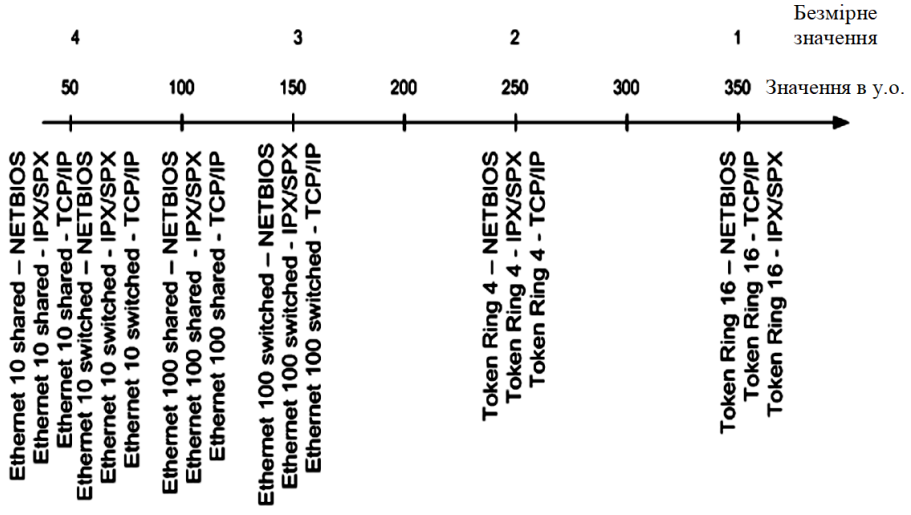


Рисунок 2 – Приведення показника вартості рішення до безрозмірної величини

Для розрахунку узагальненого показника значення кожного часного показника ефективності для всіх стеків протоколів необхідно привести до безрозмірної величини. Для зручності всі показники перетворяться в числа (використання єдиної порядкової шкали) $\{\omega\} \to \{\omega^*\}$, що відповідають місцю на шкалі розподілу від самого гіршого результату до найкращому: поганого (1), нижче середнього (2), середній (3), гарний (4). Приклад даного розрахунку наведений для показника вартості рішення на рисунку 6. Таким чином,

унікальність запропонованої методики полягає у виборі різнобічних показників ефективності, що забезпечують комплексний аналіз, алгоритмі розрахунку значень показників, а також методиці розрахунку узагальненого показника. Для перевірки розробленої методики оцінки ефективності стеків протоколів локальних мереж проведемо теоретичне й практичне дослідження сучасних протоколів локальних мереж.

Дослідження стеків протоколів локальних мереж

Дослідимо стеки протоколів локальних мереж типових банківських філій з використанням методики, запропонованої вище. Набір протоколів, обраний для дослідження, характерний саме для локальних мереж банківських філій, і при аналізі інших систем може відрізнятися.

На першому етапі порівняння проводилося для кожного окремого протоколу з урахуванням особливостей філіального мережного трафіку. Це дозволило оцінити й знайти оптимальні параметри кожного протоколу, що входить у стек.

Другим етапом було проведення комплексного аналізу стеків протоколів і знаходження найбільш ефективного рішення для типової банківської філії.

Результати досліджень представлені в таблиці 6.

Як показав аналіз існуючих банківських мереж, найбільш ефективними стеками протоколів для локальної мережі банківської філії є: NETBIOS + Ethernet 10Base switched і TCP/IP + Ethernet 10Base switched. Перший набір відрізняє більше високим рівнем продуктивності, другий – низкою вартістю й високою перспективністю.

Таким чином, результати дослідження показали, що перехід до технологій локальних мереж на основі Ethernet 100 Мбіт/с і вище на справжньому етапі розвитку вітчизняного банківського бізнесу є невиправданим, тому що приріст продуктивності буде нівельований високою вартістю модернізації технічних компонентів у філіях. Для перевірки отриманих результатів у п'ятому розділі проведені практичні дослідження сучасних протоколів локальних мереж з обліком основних банківських функцій.

Експериментальне дослідження ефективності протоколів локальних мереж

Представимо результати експериментальних досліджень ефективності стеків протоколів локальних мереж.

При проведенні тестування визначалися такі технічні показники, як: продуктивність і навантаження на устаткування. Як об'єкти тестування використовувалися практично всі відомі набори протоколів локальних мереж, використовуваних у банківських філіях. Однак, запропонована методика застосовна до будь-яких інших протоколів, використовуваним у сучасних локальних мережах. Для проведення експериментальних досліджень була розроблена методика функціонального тестування, заснована на імітації роботи типових мережних додатків (імітаційне моделювання).

Суть даної методики складається у вимірі показників ефективності для різних стеків протоколів залежно від числа одночасно працюючих робочих стацій. При цьому вимір проводиться на рівні додатків. Алгоритм роботи даної методики представлений на рисунку 2.

Для проведення тестування були обрані наступні мережні операції *O* типової банківської філії:

1. Копіювання по мережі файлу великого розміру (10Мб) по протоколі SMB (Server Message Block).

2. Копіювання по мережі 100 файлів невеликого розміру (по 2Кб) по протоколі SMB (Server Message Block).

3. Виконання 1000 транзакцій (загальний об'єм 20Кб) до бази даних по протоколі ODBC.

При виконанні даних операцій були обмірювані наступні показники:

1) час виконання мережної операції на клієнтській системі;

2) навантаження на процесор центральної системи (% Processor Time);

3) використання оперативної пам'яті центральної системи (% Committed Bytes In Use).

Відповідно до методики був розгорнутий тестовий стенд, що включає сервер центральної системи, 6 клієнтських систем, вузли реалізації мережного середовища (концентратори Ethernet, комутатори Ethernet, модулі множинного доступу Token Ring), мережні адаптери Ethernet і Token Ring, а також необхідна кількість кабелю UTP п'ятої категорії.

Таблиця 6 – Оцінка ефективності протоколів локальних мереж

№ п./п	Стек протоколів	Показники					Підсумкова оцінка
		надмірність*	продуктивність*	навантаження на устаткування*	вартість рішення*	перспективність*	
1.	Ethernet 10 shared – NETBIOS**	2	2	4	4	2	2,8
2.	Ethernet 10 shared – TCP/IP	1	2	2	4	3	2,4
3.	Ethernet 10 shared – IPX/SPX	2	2	3	4	2	2,6
4.	Ethernet 10 switched – NETBIOS	4	4	4	3	3	3,6
5.	Ethernet 10 switched – TCP/IP	3	3	3	3	4	3,2
6.	Ethernet 10 switched – IPX/SPX	3	3	3	3	2	2,8
7.	Ethernet 100 shared – NETBIOS	3	3	3	3	2	2,8
8.	Ethernet 100 shared – TCP/IP	3	2	2	3	3	2,6
9.	Ethernet 100 shared – IPX/SPX	3	2	3	3	2	2,6
10.	Ethernet 100 switched – NETBIOS	4	4	3	2	2	3
11.	Ethernet 100 switched – TCP/IP	3	4	2	2	4	3
12.	Ethernet 100 switched – IPX/SPX	3	3	2	2	4	2,8
13.	Token Ring 4 – NETBIOS	3	3	4	1	2	2,6
14.	Token Ring 4 – TCP/IP	2	3	3	1	3	2,4
15.	Token Ring 4 – IPX/SPX	2	3	3	1	2	2,2
16.	Token Ring 16 – NETBIOS	4	4	4	1	2	3
17.	Token Ring 16 – TCP/IP	3	3	2	1	3	2,4
18.	Token Ring 16 – IPX/SPX	3	3	3	1	2	2,4

* – у даній таблиці зазначені показники, наведені до безрозмірної величини.

** – для позначення протоколів і технологій, що входять у стек застосовуються загальноприйняті позначення.

Таблиця 7 – Результати експериментальної оцінки ефективності протоколів локальних мереж

Стек протоколів	Продуктивність*				Навантаження на устаткування*				
	Б.Ф.	М.Ф	Б.Д.	∑	Б.Ф.	М.Ф.	Б.Д.	∑	∑∑
Ethernet 10 shared-NetBIOS	3	3	3	3,0	4	4	4	4,0	3,5
Ethernet 10 shared-TCP/IP	3	3	3	3,0	4	4	4	4,0	3,5
Ethernet 10 shared-IPX/SPX	3	3	3	3,0	4	3	4	3,7	3,3
Ethernet 10 switched-NetBIOS	3	3	3	3,0	4	4	4	4,0	3,5
Ethernet 10 switched-TCP/IP	3	3	3	3,0	4	4	3	3,7	3,3
Ethernet 10 switched-IPX/SPX	3	3	3	3,0	4	4	4	4,0	3,5
Ethernet 100 switched-NetBIOS	4	4	3	3,7	2	3	3	2,7	3,2
Ethernet 100 switched-TCP/IP	4	4	3	3,7	2	4	4	3,3	3,5
Ethernet 100 switched-IPX/SPX	4	4	3	3,7	2	4	3	3,0	3,3
Token Ring 4-NetBIOS	2	2	3	2,3	3	4	3	3,3	2,8

Token Ring 4-TCP/IP	2	3	3	2,7	4	3	3	3,3	3,0
Token Ring 4-IPX/SPX	2	2	2	2,0	3	3	3	3,0	2,5
Token Ring 16-NetBIOS	3	3	4	3,3	2	3	3	2,7	3,0
Token Ring 16-TCP/IP	3	3	4	3,3	2	2	3	2,3	2,8
Token Ring 16-IPX/SPX	3	2	3	2,7	2	3	3	2,7	2,7

Для автоматизації тестування був розроблений пакет програм мовою VBScript, що імітують роботу кінцевих користувачів на стороні клієнтських систем і здійснюючий збір даних і попередню обробку результатів вимірів на стороні центральної системи. Отримані результати вимірів були оброблені відповідно до методи комплексної оцінки ефективності, розглянутої вище. Результати оцінки по двох технічних показниках представлені в таблиці 7. Практичне впровадження запропонованого стека дозволило підвищити ефективність функціонування локальних мереж даних фінансових установ приблизно на 15% при мінімальних фінансових витратах. Найкращі результати в цих організаціях були досягнуті при наступних параметрах налаштування протоколів: розмір кадру Ethernet – 512 байт, розмір датаграммы NETBIOS – 496 байт. На сучасному етапі розвитку технології й ринку мережних рішень перехід до більше швидкісних технологій (Ethernet 100 Мбіт/с і вище) став би невигідним вкладенням коштів, тому що інвестиції в модернізацію мережних компонентів локальної мережі не відбилися б на значному підвищенні ефективності роботи мережі в цілому.

Розробка структурної схеми

Однією з наших ключових цілей як постачальника інфраструктури як послуги є допомогти вам знайти правильне IT-рішення, яке відповідає вашим потребам. Крім того, надається комплексні послуги підтримки та забезпечуємо безперебійну роботу хмарного середовища, щоб ви могли безперешкодно займатися своїм основним бізнесом. Хмара – це ваше оптимальне рішення, коли ви хочете максимально контролювати свої операції. Водночас ви можете розраховувати на повну підтримку за допомогою експертів, які завжди доступні. Незалежно від того, чи потрібно вам просто розгорнути інфраструктуру, додаткову відповідну платформу (керовані служби) або важливий кластер Kubernetes для вашого проекту: усі ці компоненти можна організувати відповідно до ваших потреб таким чином, щоб ви могли досягти найкращих результатів. для ваших цілей. Крім того, хмара розроблена для будь-якої компанії, яка прагне керувати масштабованою хмарною інфраструктурою, незалежною від основних хмарних провайдерів, а також від їхніх моделей ліцензування та розташування даних. З хмарними службами firstcolo ви можете бути впевнені в безпеці даних. Фактично, сертифікати ISO 27001 і PCI DSS підтверджують це на папері.

Управління ресурсами

Усіма ресурсами можна керувати централізовано за допомогою інформаційної панелі. Віртуальні машини можна легко підключити та отримати доступ через єдиний інтерфейс.

Безпека розрахунку витрат – витрати завжди на виду

Ніяких прихованих витрат і прозорість у будь-який час. Жодних прихованих обмежень послуги, чітка та повна структура виставлення рахунків. Наприклад, трафік як фіксована місячна фіксована плата – дозволяє повністю використовувати запропоновану пропускну здатність.

CAPEX проти OPEX – ваш вибір із максимальною гнучкістю

Ви завжди можете гнучко керувати хмарними службами, які використовуєте, і додавати додаткові служби в будь-який час.

Гнучкі SLA

Ми пропонуємо вам оптимальну основу для будь-яких потреб і вимог, використовуючи багаторівневу модель SLA.

Високоякісні серверні рішення для ваших програм

Скористайтеся перевагами високого рівня досвіду в області виділених серверів – ми знаємо, як налаштувати правильну конфігурацію відповідно до ваших потреб: ми

порадимося з вами, яке індивідуальне рішення вам підходить, і налаштуємо його для вас разом із програмним забезпеченням і операційною системою для ваші цифрові проекти. Орендуйте або купуйте виділений сервер і насолоджуйтесь нашим широким спектром послуг. Виділений сервер призначений для кількох постійно діючих сервісів. Для повсякденних завдань не використовується. У порівнянні з хмарним сервером або спільним (віртуальним приватним) сервером, який зазвичай надається зі спільними ресурсами, виділений сервер доступний виключно одному користувачеві для прикладних цілей, таких як віртуалізація. Для дуже потребуючих продуктивності програм кілька виділених серверів зазвичай об'єднують у так звані кластери та працюють як одна логічна одиниця.

Використовуючи наші виділені сервери, менеджери IT-проектів комерційно-професійних розрахункових компаній хочуть уникнути незручностей, таких як зовнішні витрати на відрядження та обслуговування для спеціальних відповідей, які є звичайними для багатьох постачальників. Виділені сервери також забезпечують надійний захист і високу продуктивність. У нас ви можете отримати бажані сервери різного розміру за постійною місячною ціною, включаючи витрати на серверне житло. Завдяки нашому добре укомплектованому складу апаратного забезпечення ви отримуєте перевагу від найкоротшого часу реагування у разі необхідності заміни.

Ми разом з вами узгоджуємо необхідне апаратне обладнання відповідно до ваших індивідуальних вимог і побажань. У центрах обробки даних ми використовуємо відомі бренди виробників, такі як Dell, HP, Supermicro, Intel і Samsung. На відміну від багатьох конкурентів, ми не прив'язані до конкретних виробників чи технологій. Наш досвідчений персонал гарантує, що весь процес замовлення та введення в експлуатацію проходить гладко. Ваші виділені сервери виготовляються нами безпосередньо на місці та піддаються ретельній перевірці під час тестового запуску. Ми беремо на себе всі етапи – закупівлю фурнітури, збірку, встановлення та підтримку. У співпраці з вами ми налаштовуємо повністю індивідуальну систему.

З виділеним сервером ви отримуєте найбільшу гнучкість під час налаштування сервера (ми також раді зробити це за вас). У вашому розпорядженні весь об'єм, такий як простір на жорсткому диску та оперативна пам'ять, що значно скорочує час завантаження. Крім того, виділений сервер пропонує дуже високий рівень безпеки, а отже, надійний захист ваших даних.

Незалежно від того, чи це веб-додаток, інтернет-магазин чи інтернет-портал, переважна більшість компаній переносять важливу частину своєї інфраструктури в хмару як частину своєї цифрової стратегії. Частково це пов'язано з тим, що обчислювальні ресурси мають бути доступними швидко та за запитом у будь-який час, чого не завжди легко досягти за допомогою рішень на основі традиційних виділених серверів.

На багатьох веб-порталах під час певних щоденних піків кількість користувачів зростає дуже швидко лише на короткий період часу. Щоб мати можливість реагувати на ці піки, ресурси повинні бути завжди доступними під час використання виділених серверів, які залишаються невикористаними під час нижчих доходів користувачів. Це призводить до високої загальної вартості володіння. Таких витрат можна уникнути завдяки гнучкій масштабованості хмарних інфраструктур. Щоб задовольнити цей попит з боку наших ділових партнерів, ми покладаємося на наше власно розроблене хмарне середовище на основі OpenStack, а також на альтернативні хмарні рішення на основі VMware і Proxmox.

Надається безпечну віртуальну інфраструктуру для цифрових проектів. У цьому процесі ви завжди отримуєте переваги від орієнтованих на попит моделей використання та виставлення рахунків. Обчислювальні ресурси можна додавати та звільняти миттєво, коли вони більше не потрібні. Це призводить до набагато більш прибуткової структури витрат, і вам ніколи не доведеться ризикувати тим, що ви не зможете обслуговувати своїх клієнтів через брак ресурсів.

Хмарний консалтинг

Коли перший крок у хмару зроблено, часто виникає ряд додаткових вимог і перешкод. Особливо реалізація нових проектів і надання подальших послуг ставить перед компаніями проблеми, які не були повністю розглянуті на початку. Щоб гарантувати оптимальну взаємодію всіх підрозділів і швидку реакцію на нові вимоги, ви завжди можете покластися на наших експертів. Як провідний цифровий партнер, ми маємо великий досвід у сфері хмарних технологій у всьому та можемо допомогти вам із концептуальним проектуванням, впровадженням і безперебійною поточною роботою. Це може бути міграція існуючої хмарної інфраструктури або розширення до гібридної хмари - ми будемо раді вам порадити.

Управління публічною хмарою

Велика кількість компаній уже дотримується цілісної стратегії в області публічної хмари з провідними постачальниками, такими як Amazon Web Services (AWS) або Microsoft Azure. Природно, що гнучкість цих провайдерів майже незамінна, особливо для великих проектів. Тим не менш, через високу складність і великий набір функцій, ці проекти можуть швидко стати непрозорими і вимагати великих зусиль з налаштування та обслуговування. Щоб повністю використати потенціал вашої хмарної стратегії, ми пропонуємо можливість прямого підключення до звичайних провайдерів за допомогою гіперскейлерів. Це ми можемо поєднати з Direct-Cloud-Connect до провідних постачальників, таких як Azure, AWS і GCP, із пропускною спроможністю до 100 Гбіт/с із наших центрів обробки даних у Франкфурті. Зокрема, наша велика експертиза в середовищі Azure поширюється на всю команду firstcolo: технологія Microsoft неодноразово впроваджувалася в різні клієнтські проекти. У поєднанні з нашим комплексним керуванням хмарою це створює для вас «універсальний безтурботний пакет», у якому наші експерти беруть на себе адміністрування, щоб ви могли й надалі зосереджуватися на подальшому розвитку свого основного бізнесу.

Для деяких користувачів хмари ефективність хмари передбачає досягнення віддаленого підключення до їх бізнес-інфраструктури. Інші використовують його для опису використання ресурсів у своїх центрах обробки даних, тоді як багато хто зрозуміло пов’язує його з продуктивністю та надійністю.

Хоча всі ці визначення допустимі, ефективність хмари найкраще можна описати, поєднавши їх усі. Подумайте про це як про цілісну концепцію, яка спирається на численні динамічні фактори.

Простіше кажучи, можна сказати, що ефективність хмари означає вашу здатність використовувати ресурси хмари найкращим чином і з найменшими витратами, в той же час мінімізуючи витрати хмари.

Останнім часом досягнення хмарної ефективності стало особливо важливим, оскільки організації продовжують використовувати більше ресурсів, ніж їм насправді потрібно. За останні кілька років у компаній виникла тенденція сліпо розширювати свої хмарні технології, коли вони ростуть. Ось як вони зрештою опиняються в умовах надзвичайно складної хмарної архітектури, яка затьмарена неактивними ресурсами, що потім призводить до марних витрат на хмару.

У 2023 році, наприклад, перші дослідження Gartner показали, що в той час як ринок інфраструктури як послуги (IaaS) зросте до 50 мільярдів доларів до кінця року, компанії мають втратити понад 17 мільярдів доларів у вигляді марнотрачених витрат на хмару через невикористані ресурси..

Нижче наведено деякі основні проблеми, які особливо ускладнюють досягнення організаційною ефективністю витрат:

– **Відсутність інформації про витрати в хмарі**. Хоча загальні витрати на використання можуть бути очевидними, багато компаній не можуть розбити цифри, щоб отримати точні показники вартості для кожного ресурсу. Така відсутність видимості призводить до того, що широко відомо як розповсюдження хмари, коли організації продовжують випадково розширювати свої хмарні ресурси без належного керування.

– **Складність виставлення рахунків**. Для користувачів, які сподіваються на основі своїх рахунків отримати дані про витрати в хмарі, виявляється, що хмарні платформи ніколи не будуть такими перспективними. Ви отримуватимете свій щомісячний рахунок за хмару, але він міститиме незрозумілу інформацію про ваші витрати на хмару.

Amazon Web Services (AWS) є одним із прикладів сумно відомої платформи. Хоча його звіти про вартість і використання за умовчанням здаються вичерпними, користувачам часто важко зрозуміти складні технічні характеристики. Проблема настільки далекосяжна, що згідно з опитуванням 7500 користувачів AWS 95% визнали, що рахунки є найбільш заплутаною частиною платформи Amazon Cloud.

– **Неузгоджені стратегії оптимізації витрат у різних командах**. Деякі організації помилково вважають, що хмарне управління витратами призначене виключно для IT-команд. Таким чином, вони запроваджують стратегії скорочення витрат, які є односторонніми – інші відділи відходять на другий план.

– **Управління витратами та оптимізація вручну**. До недавнього часу для DevOps було звичайною справою вручну аналізувати тенденції використання хмари та обчислювати цифри витрат, перш ніж переходити до налаштування положень для кожної програми.

Увесь цей підхід не тільки трудомісткий і обтяжливий, але й схильний до помилок – отже, неможливо гарантувати ефективність хмари.

Тепер, коли ми виключили традиційні підходи, ось шість перевірених стратегій, які використовують найбільш просунуті підприємства для підвищення ефективності своїх хмар. Ось кроки, які вам слід виконати, щоб підвищити продуктивність, зменшити витрати на хмару та збільшити загальний результат:

1. Зведіть до мінімуму переміщення даних. Оскільки гібридна хмара є однією з найпопулярніших установок хмари в організаціях, величезні обсяги даних регулярно передаються між загальнодоступною хмарою та локальним середовищем. Весь цей процес «до-і-з» потребує як часу, так і ресурсів. Отже, що більше даних ви вирішите передати між середовищами, то більше ресурсів ви споживаєте, і тим довше знадобиться, щоб передавати все. Для оптимальної продуктивності системи вам слід мінімізувати переміщення даних між хмарними серверами та локальним середовищем компанії. Ви можете почати з ретельної класифікації даних, а потім вибрати ідеальне середовище для кожної категорії. Наприклад, критично важливі дані повинні зберігатися на локальних серверах, а віддалені центри обробки даних зарезервовані для некритичних даних і програм.

2. Виберіть найбільш підходящі екземпляри. Провідні постачальники IaaS пропонують різні типи обчислювальних екземплярів для різних видів робочого навантаження. На AWS ви знайдете екземпляри EC2, які надають різні комбінації мереж, пам’яті, ЦП і пам’яті. Деякі екземпляри оптимізовано для загальних обчислень, тоді як інші призначені для зберігання, прискорених обчислень тощо. Щоб зберегти найкращі ресурси за найнижчої вартості, вам слід приділити час, щоб вибрати найбільш підходящі екземпляри на основі ваших цілей хмарних обчислень. Якщо екземпляр замалий, ви можете заощадити гроші, але в кінцевому підсумку це призведе до зниження продуктивності. І якщо він виявиться завеликим, ваше робоче навантаження виграє від збільшення продуктивності, але ви будете потонути у втрачених хмарних витратах.

3. Скористайтеся автомасштабуванням. Вам не потрібно обмежувати свої обчислення потужністю за замовчуванням, наданою примірниками. Щоб задовольнити потреби, що динамічно змінюються, хмарні платформи здатні автоматично масштабувати ресурси користувача на вимогу. Відомо, що Google Cloud Platform (GCP), Microsoft Azure та AWS додають або видаляють екземпляри та пов’язані ресурси зі зміною робочого навантаження. Наприклад, ви можете скористатися їхніми балансувальниками навантаження, щоб уникнути перевантаження своїх інсталяцій під час стрибків навантаження. Після встановлення відповідних правил автомасштабування на основі очікуваних тенденцій використання балансувальник навантаження відстежуватиме та розподілятиме вхідний трафік між кількома примірниками.

4. Відстежуйте продуктивність. Балансувальник навантаження відстежуватиме ваш трафік і робочі навантаження, але він не надасть вам усіх необхідних показників. Щоб всебічно оптимізувати продуктивність хмари, вам слід зібрати й проаналізувати всі відповідні показники ваших робочих навантажень і тенденцій використання. Тут ви використовуєте не лише вбудовані аналітичні інструменти, а й сторонні сервіси, які здатні відстежувати продуктивність у режимі реального часу. Ви повинні мати можливість стежити за всім, що відбувається у вашому хмарному середовищі.

5. Доповнити хмарну мережу кешами. Хоча зберігання даних у хмарних серверах є хорошим способом зручного полегшення віддаленого доступу, передача всіх даних у вашу локальну мережу та з неї є зовсім іншою проблемою. Переміщення даних потребує часу, що може перешкоджати реагування ваших програм. Одним із способів ефективного прискорення процесу передачі є використання служб кешу, сумісних із вашою хмарною платформою. Це віддзеркалить ваші хмарні дані в мережі доставки вмісту з кількома кеш-серверами – мінімізуючи відносну відстань передачі даних. Замість того, щоб завантажувати файли з оригінального хмарного сервера, ваші програми швидко отримають дані з найближчого сервера кешу.

6. Впровадження хмарного аналізу витрат

Оскільки ефективності хмари неможливо досягти без мінімізації витрат, оптимізація продуктивності хмари завжди повинна йти рука об руку з керуванням витратами на хмару.

Щоб отримати найкращий результат, подумайте про використання хмарної платформи аналізу витрат. Збагачуючи ваші витрати метаданими послуг, телеметрією тощо, платформа аналізу витрат дає змогу бачити ваші витрати під будь-яким кутом. Ви можете розблокувати раніше неможливі показники вартості одиниці продукції (як-от вартість за функцію, вартість за клієнта тощо), а також деталізувати та зменшити вартість – з меншими зусиллями, ніж застарілі інструменти звітування про витрати. Система дає вам інформацію про те, чому, чому та куди ви інвестуєте AWS. Інженери можуть самостійно обслуговуватись і досліджувати вартість своєї архітектури та програм, що дозволяє їм приймати економічні інженерні рішення, які забезпечують прибутковість вашої компанії. Фінанси можуть виміряти рентабельність інвестицій у ваші технічні інвестиції – і відрізнити неконтрольований хмарний рахунок від ефекту масштабу.

На початковому етапі дослідження було виконано структурно-функціональне моделювання розподілу елементів БІС і інформаційні потоки між ними на різних рівнях. На першому етапі побудована модель розміщення елементів банківського середовища, що також ураховує взаємозв'язок рівнів ієрархії (центр, філія, відділення) і груп банківських систем (Back office, Middle office, Front office). Результати аналізу моделі відображені на структурній схемі (рисунок 3)

На структурній схемі наведені основні блоки системи, яка досліджується. Тобто системи з розподіленими ресурсами, ефективність якої оцінюють. Така система наведена на прикладі філії банку.

Front-office – це комплекс програмно-апаратних засобів, що підвищують ефективність спілкування. Це спеціалізовані системи, що автоматизують роботу співробітників, які спілкуються з "зовнішнім миром", які допомагають їм у повсякденній діяльності. Іншою стороною взаємодії учасників «телефонних» відносин є системи, що автоматизують внутрішні взаємодії в компанії. Back-office, – це "каркас" підприємства. Для ефективної роботи всієї фірми обидві сторони (Front-office і Back-office) повинні взаємодіяти один з одним по оптимальних алгоритмах.

Автоматизація Middle-офісних операцій

На жаль, у більшості українських банків діяльність відділів Middle-офісу дотепер майже не автоматизована. Основна аналітична звітність складається за допомогою не призначених для цього додатків. Відсутність єдиної інтегрованої системи для обліку всіх Front-офісних операцій значно обмежує можливості співробітників Middle-офісу по керуванню й аналізу діяльності банку.

Сучасні системи дозволяють: Оптимізувати внутрибанківські фінансові потоки між центрами виникнення прибутку й витрат; Управляти параметризацією всіх банківських операцій, у тому числі існує можливість додавання необмеженого числа аналітичних параметрів по кожній угоді; Управляти й здійснювати моніторинг кредитно-інвестиційного портфеля банку в режимі реального часу; Становити різні аналітичні звіти для оцінки й прогнозування діяльності банку; Становити різні аналітичні звіти для оцінки й прогнозування діяльності банків-конкурентів; Проектувати й управляти необмеженим числом лімітів, нормативів, у тому числі існує можливість аналізувати використання лімітів і нормативів у режимі реального часу; Переоцінювати відкриті позиції; Управляти ризиками: VAR-Аналіз, стратегії хеджування, інші звіти.

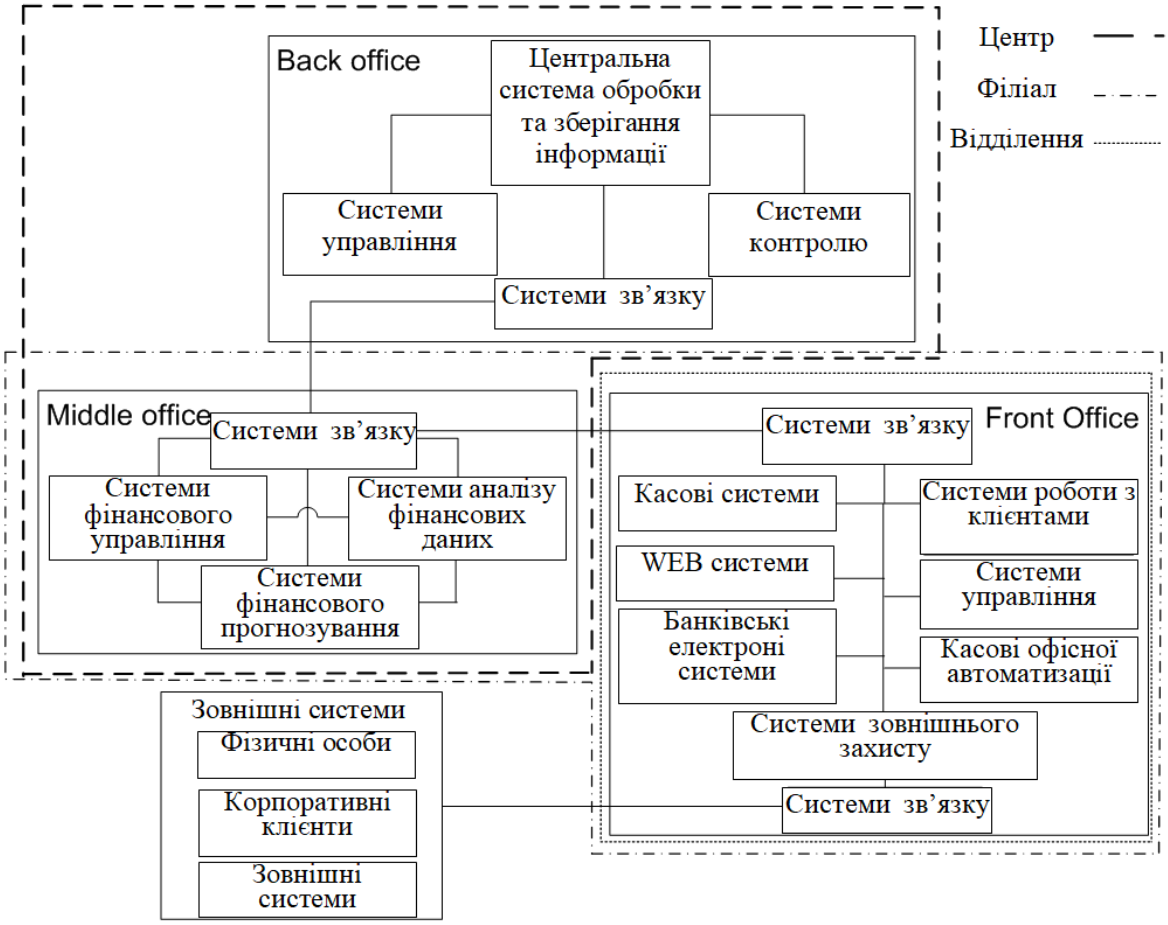


Рисунок 3 – Структурна схема розміщення елементів БІС

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. Досліджена система бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. На основі отриманих результатів досліджень створена програмна реалізація системи бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. Розроблені алгоритми дозволяють успішно вирішувати завдання бюджетування хмарної інформаційної системи для визначення ефективності її впровадження. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34. (Scopus).
2. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477. (Scopus).
3. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w (Scopus).
4. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
5. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).
6. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
7. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
8. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
9. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
10. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
11. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
12. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
13. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
14. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
15. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
16. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
17. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
18. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
19. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
20. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).

УДК 004

М.Середа, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПРОЕКТУВАННЯ СЕРВІСІВ АВТЕНТИФІКАЦІЇ

У статті розроблено програмне забезпечення, яке призначено для системи проектування сервісів автентифікації. Метою розробки є дослідження та програмна реалізація системи проектування сервісів автентифікації. Об'єктом дослідження є процес проектування сервісів автентифікації. Предметом дослідження є методи проектування сервісів автентифікації. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи проектування сервісів автентифікації. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, автентифікація

Постановка проблеми. Проблема з паролями полягає в тому, що над ними дуже легко втратити контроль. Люди передають свої паролі іншим людям. Люди записують їх, а інші читають. Люди надсилають їх електронними листами, і ці листи перехоплюються. Люди використовують їх для входу на віддалені сервери, і їхні комунікації прослуховуються. Паролі також легко вгадати. І коли щось із цього трапляється, пароль більше не працює як маркер автентифікації, оскільки ви ніколи не можете бути впевнені, хто вводить цей пароль.

У даній магістерській роботі буде розглянута автентифікація з використанням електронних ключів, більш конкретно з використанням як ключ USB-накопичувача.

Сервіс автентифікації, який описаний у даній роботі, Secure WEB-Logon – це апаратна «двофакторна автентифікація», яка вирішує проблему пароля. USB-ключ Сервіс автентифікації – це апаратний ключ, який замінює незахищений метод входу «Ідентифікатор користувача + пароль» для рішень Інтернету або внутрішньої мережі. Аутентифікація виконується за допомогою шифрування даних всередині безпечного апаратного забезпечення Сервіс автентифікації. Зашифрований маркер автентифікації змінюється щоразу, коли надходить запит на вхід, щоб переконатися, що перехоплений маркер не можна використовувати двічі.

Єдиний варіант – скористатися спеціально розробленою в результаті виконання магістерської роботи системою автентифікації з використанням електронних ключів.

Основна перевага розробленої системи системи полягає в мінімальній перебудові IT-інфраструктури організації, мінімальні витрати на адміністрування, одночасної автентифікації в службах каталогу Windows і NetWare, підвищеної безпека мережі за рахунок використання "сильних" паролів і зберігання їх у захищеній пам'яті електронних ключів.

Крім цього, істотно знижується "людський фактор", оскільки користувач просто не знає пароля, тому не може його нікому передати або записати на папірці.

Безпека підсистем автентифікації не може бути перевірена експериментально в ході випробувань на функціонування. Крім того, через достаток криптографічних алгоритмів і різноманіття завдань автентифікації ці системи найчастіше проектуються «з нуля», що збільшує трудомісткість розробки. Алгоритмізація процесу проектування й обґрунтування прийнятих рішень дозволять уникнути типових помилок, а також порівнювати різні варіанти побудови алгоритмів автентифікації й вибирати найкращий з них.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи проектування сервісів автентифікації.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи проектування сервісів автентифікації.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем проектування сервісів автентифікації.
– Дослідження системи проектування сервісів автентифікації.
– Програмна реалізація системи проектування сервісів автентифікації.

Об'єктом дослідження є процес проектування сервісів автентифікації.

Предметом дослідження є методи проектування сервісів автентифікації.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод проектування сервісів автентифікації.

Розроблено вітчизняний продукт проектування сервісів автентифікації, який має більш широкі можливості, на відміну від існуючих аналогів.

Виклад основного матеріалу.

Сервіс автентифікації – надійна система безпеки для захисту вашого програмного забезпечення від несанкціонованого відтворення.

Сервіс автентифікації для USB-портів

Функціонально модулі Сервіс автентифікації серій MLU і MKU для USB-інтерфейсу ідентичні модулям ML і MK для підключення LPT, а також забезпечують захист програмного забезпечення для ноутбуків і ПК, які не мають паралелі. інтерфейс принтера.

USB-ключі Сервіс автентифікації підтримують HID-режим, що означає, що системний драйвер USB не потрібен.

Сервіс автентифікації для паралельних портів

Модулі Сервіс автентифікації серій ML і MK для інтерфейсу принтера просто підключаються до інтерфейсу LPT, де вони функціонують ідеально, не створюючи проблем для периферійних пристроїв, таких як принтер, сканер тощо, які підключені нижче.

Під час розробки особлива увага приділялася прозорій поведінці, високому ступеню безпеки завдяки використанню процесора RISC, простому підключенню до програмного забезпечення та високій надійності при практичному використанні.

Сервіс автентифікації Secure WEB-Logon базується на двох компонентах:

1. Сервіс автентифікації-ключ і WEB-Client – програма на стороні клієнта.

2. WEB-сервер на стороні постачальника

Коли користувачеві потрібно ввійти на сервер, сервер надсилає довільний запит. Користувач повинен підключити USB-ключ Сервіс автентифікації і ввести особистий PIN-код.

Запит сервера, PIN-код користувача та деякі інші блоки даних будуть зашифровані всередині захищеного матричного ключа. Потім зашифрований маркер результату надсилається назад на сервер.

Конфіденційна інформація, наприклад ключ шифрування, безпечно зберігається в ключі Сервіс автентифікації і на сервері, але ніколи не передається між сервером і клієнтом.

Сервіс автентифікації Secure WEB-Logon можна використовувати з усіма WEB-браузерами і не базується на файлах cookie, плагінах або аплетах Java.

На стороні клієнта програма WEB-клієнт (EXE) використовується для передачі запитів сервера на ключ Сервіс автентифікації.

Щоб мати можливість протестувати тут Сервіс автентифікації Secure WEB-Logon, вам потрібен ключ Сервіс автентифікації!

Виконайте наступні кроки:

1. Створіть WEB-клієнтську програму за допомогою інструменту «Сервіс автентифікації WEB-Logon Wizard».

2. Запустіть програму WEB-Cleint і підключіть Сервіс автентифікації до ПК.

Коли Сервіс автентифікації підключено до USB-порту, автоматично запускається Інтернет-браузер із WEB-сайтом.

3. Зареєструйте на сайті входу параметри вашого ключ і PIN-код на ваш вибір для цього тесту.

4. Після реєстрації продовжте, натиснувши [Вхід] за допомогою підключеного ключа та зареєстрований PIN-код.

Сервіс автентифікації пропонує все, що для вас найважливіше:

– Кросплатформенність (Windows, Linux, Mac) без власних USB-драйверів.

– Істотно зменшує ваші зусилля з підтримки, оскільки він працює одразу після підключення без встановлення будь-яких USB-драйверів.

– Абсолютна надійність (10 років гарантії).

– Доступні кілька розмірів пам'яті.

– Програмний API, інструменти та підтримка безкоштовні.

– Найвища якість за розумною ціною.

– Сертифікований ISO.

– Два різних розміри корпусу (довгий або короткий) з однаковими технічними характеристиками за однаковою ціною.

Дуже відповідні інструменти для керування вашими ліцензіями:

– Просте впровадження API у ваших 16-, 32- та 64-розрядних програмах.

– Автоматичний захист файлів EXE.

– 64-розрядний API для Windows і Linux уже доступний.

– Віддалене оновлення: легко оновлюйте ключі безпосередньо у своїх клієнтів. Також доступний як інтеграція API у вашу систему керування клієнтами.

– Безпечний WEB-вхід для автентифікації користувача у ваших Інтернет- та Інтранет-додатках.

– Управління мережевими ліцензіями без додаткової плати.

– Той самий ключ можна використовувати локально або в мережі.

– Ліцензування мережі на основі TCP/IP доступне як приклад із відкритим кодом.

– Той самий ключ можна використовувати для захисту програмного забезпечення та/або WEB-автентифікації.

Сервіс автентифікації пропонує вищий ступінь безпеки:

– Шифрування відбувається повністю всередині апаратного забезпечення.

– Сучасна техніка, яка використовується для «Анти-Клонування-Безпека».

– Захист файлів EXE від налагодження та зворотного проектування. Також виконувані файли .NET v1.0, 1.1 і 2.0 підтримуються та захищені технологією «Anti-Process-Dump».

– Anti-Hacker-Lock ключ перестане працювати під час атаки.

– 128-бітне шифрування та дешифрування даних.

– Зберігання 128-бітних ключів, визначених вами, які неможливо прочитати з ключа.

– Використання «одноразових ключів», які дійсні лише для однієї послідовності шифрування/дешифрування.

– До 16 різних 128-бітних алгоритмів шифрування.

– Пам'ять серії MK/MKU захищена від запису та може бути змінена лише за наявності захисного ключу MasterKey-Dongle.

Технологія OTP

У сучасному цифровому середовищі, що швидко розвивається, безпека залишається головним пріоритетом для будь-якої платформи. Для забезпечення безпечної роботи користувача та захисту конфіденційних даних клієнтів необхідна надійна система

автентифікації користувачів. У цій вичерпній технічній статті ми розробимо вдосконалену систему автентифікації користувачів, яка використовує перевірку на основі OTP (одноразового пароля) для входу користувача та використовує JWT (веб-токени JSON) для авторизації користувачів. Крім того, ми запровадимо керування доступом на основі ролей, щоб надати користувачам доступ до певних ресурсів на основі їхніх призначених ролей і областей ролей. Ми також вивчимо обмеження швидкості та кешування OTP, щоб захистити систему від атак грубої сили та оптимізувати керування OTP.

Розуміння JWT, OTP і доступу на основі ролей користувача

JWT (веб-токени JSON):

JWT – це компактний, URL-безпечний формат маркера, який надійно представляє претензії між двома сторонами. Підпис забезпечує цілісність токена. JWT не мають статусу, і їх можна перевірити, не покладаючись на серверне сховище. Вони широко використовуються для автентифікації та авторизації в сучасних веб-додатках завдяки своїй ефективності та безпеці.

OTP (одноразовий пароль):

OTP – це одноразовий пароль з обмеженим часом, який зазвичай надсилається користувачеві через SMS, електронну пошту чи інші безпечні канали. Він забезпечує додатковий рівень безпеки, гарантуючи, що пароль користувача дійсний лише протягом короткого періоду часу та не може бути використаний повторно. OTP зазвичай використовуються для двофакторної автентифікації (2FA) і процесів скидання пароля. Використовуючи вхід на основі OTP, система автентифікації додає додатковий захід безпеки для перевірки ідентичності користувачів.

Доступ на основі ролей користувача:

Контроль доступу на основі ролей користувача визначає дозволи та обмеження, надані користувачам на основі їхніх призначених ролей. Кожна роль пов’язана з певними привілеями, які визначають, до яких дій і ресурсів може отримати доступ користувач. Цей підхід допомагає застосувати принцип найменших привілеїв, гарантуючи, що користувачі можуть отримати доступ лише до ресурсів, необхідних для виконання їхніх завдань. Впроваджуючи доступ на основі ролей користувача, система може ефективно та безпечно керувати контролем доступу, запобігаючи несанкціонованому доступу та потенційному витоку даних.

Мета реалізації

Поєднання JWT, OTP і доступу на основі ролей користувача слугує для створення надійної та безпечної системи автентифікації користувачів.

1. Токени JWT забезпечують безперебійний і безпечний метод авторизації користувачів, забезпечуючи доступ до ресурсів лише дійсним користувачам із підтвердженими претензіями. Природа без збереження стану усуває потребу в сховищі на стороні сервера, що робить його масштабованим і ефективним.

2. Вхід на основі OTP додає додатковий рівень безпеки, зменшуючи ризик несанкціонованого доступу через зламані паролі. Одноразові паролі обмежені за часом і можуть використовуватися лише один раз, покращуючи автентифікацію користувачів.

3. Доступ на основі ролей користувача гарантує, що користувачам надаються відповідні дозволи на основі їхніх ролей. Цей детальний контроль доступу обмежує доступ до критично важливих ресурсів, захищаючи конфіденційні дані та запобігаючи можливому зловживанню.

Інтегруючи ці технології, онлайн-платформи можуть підвищити рівень безпеки, завоювати довіру користувачів і захистити їхні цінні активи. Система автентифікації захищає від поширених загроз, таких як атаки грубої сили, неавторизований доступ і вразливість паролів, створюючи безпечне та надійне середовище для впевненої взаємодії користувачів.

Уразливості технологій ОТР

Технологія одноразових паролів вважається досить надійною. Однак об'єктивності заради відзначимо, що й у неї є свої недоліки, яким піддані всі системи, що реалізують принцип ОТР у чистому виді.

Деякі атаки застосовні тільки до окремих способів реалізації технології одноразових паролів. Для приклада можна знову взяти метод синхронізації по таймері. Як ми вже говорили, час у ньому враховується не з точністю до секунди, а в межах якогось установленого заздалегідь інтервалу. Це необхідно для обліку можливості розсинхронізації таймерів, а також появи затримок у передачі даних. І саме цим моментом теоретично може скористатися зловмисник для одержання несанкціонованого доступу до віддаленої системи. Для початку хакер "прослуховує" мережний трафік від користувача до сервера автентифікації й перехоплює відправлені "жертвою" логін і одноразовий пароль. Потім він відразу блокує його комп'ютер (перевантажує його, обриває зв'язок і т.п.), а сам відправляє авторизаційні дані вже від себе. І якщо він встигне зробити це так швидко, щоб інтервал автентифікації не встиг змінитися, то сервер визнає його за зареєстрованого користувача.

Розробка структурної схеми

Спершу розглянемо питання розробки методології формування процесу автентифікації. Найважливішою частиною підсистеми автентифікації є сукупність алгоритмів автентифікації, які задають набір захисних функцій, що визначають, що й при яких умовах може бути захищено.

Таблиця 1 – Упорядкованість захисних функцій автентифікації

Координати захисних функцій	**Упорядкованість**
Тип автентифікації	автентифікація повідомлення ⊇ упізнавання
Число сеансів на одному ключі	багаторазова автентифікація ⊇ однократна автентифікація
Тип використовуваного каналу зв'язку (можливість діалогу)	бездіалогова автентифікація ⊇ діалогова автентифікація
Довіра до верифікатору	недоверенный верифікатор ⊇ довірений верифікатор
Якість зв'язку (при однократної автентифікації потрібне надійне доведення інформації)	некритичність надійного доведення інформації ⊇ необхідність надійного доведення інформації
Наявність служби єдиного часу (необхідно для захисту від повторів або затримок інформації при бездіалогової автентифікації)	необов'язковість єдиного часу ⊇ наявність єдиного часу
Відносний обсяг переданої службової інформації (відношення обсягу переданих даних до ентропії	менший відносний обсяг службової інформації ⊇ більший відносний обсяг службової інформації

У роботі до складу алгоритмів автентифікації включені:

– власно протоколи автентифікації;

– швидкі алгоритми, що реалізують обчислення у відповідних математичних структурах;

– допоміжні алгоритми, що впливають на безпеку;

– алгоритми вибору параметрів підсистеми автентифікації;

– алгоритми керування ключами, включаючи зміну параметрів підсистеми автентифікації.

Стандартні алгоритми автентифікації не завжди задовольняють вимогам, пропонованим до критичних інформаційно-телекомунікаційних систем, тому з'являється необхідність проектування оригінальних алгоритмів автентифікації.

Безпека цих алгоритмів традиційно ґрунтується на складності рішення математичного завдання, для вибору якого в ході проектування пропонується трьохрівневасистематизація завдань.

До першого рівня віднесені класи уніфікованих математичних завдань (КУМЗ), у якості яких запропоноване розглядати наступні типи завдань, орієнтованих на побудову алгоритмів автентифікації:

– завдання про виконуваність (до якої зводяться завдання розкриття ключа, обіги й обчислення колізій хеш-функції);
– завдання визначення структури й порядку кінцевої групи;
– завдання обчислення індексу елемента кінцевої абелевої групи;
– завдання про укладання ранця;
– завдання обчислення морфізма між об'єктами категорії.

До другого рівня віднесені масові основні математичні завдання вибору (ОМЗ), отримані в результаті параметризації КУМЗ за допомогою математичних структур, що визначають область математики, до якої відноситься ОМЗ, а також класи зв'язаних завдань, до яких зводиться ОМЗ. Крім завдань вибору при дослідженні безпеки використовуються також додаткові завдання розпізнавання й пошуку.

До третього рівня віднесені приватні математичні завдання, що відповідають масовій ОМЗ.

Проектування алгоритмів автентифікації (рисунок 1) пропонується формально визначати як процес побудови ланцюжків відображень:

– {Класи уніфікованих математичних завдань} × {уніфіковані криптографічні примітиви} →{мінімізований набір захисних функцій} → {узагальнені протоколи автентифікації};

– {Класи уніфікованих математичних завдань} × {математичні структури} → {Основні математичні завдання} ∪ {додаткові завдання};

– {Основні математичні завдання} × {узагальнені протоколи автентифікації} → {алгоритми автентифікації} → {швидкі обчислювальні алгоритми};

– {Основні математичні завдання} ∪ {додаткові завдання} → {алгоритми генерації параметрів підсистеми автентифікації} → {частки математичні завдання} → {алгоритми керування ключами}.

Уніфіковані криптографічні примітиви містять у собі: симетричне шифрування, шифрування з відкритим ключем, безключову й ключову хеш-функцію, цифровий підпис, діалогові й бездіалогові докази з нульовим розголошенням знань, секретні гомоморфізми. Безлічі математичних завдань і криптографічних примітивів варто розглядати в їхньому розвитку.

При проектуванні алгоритмів автентифікації потрібно прогнозувати як зниження складності математичного завдання, так і ріст продуктивності обчислювальної техніки, що дозволяє вирішити це завдання, а також розвиток інших (не зв'язаних безпосередньо з обчисленнями) можливостей порушника, спрямованих на зниження безпеки.

Швидкість $s(t,T)$ падіння стійкості $S(t)$ на інтервалі часу $(T,\ T+t)$ запропоновано визначати по формулі:

$$s(t,\ T) = (\log S(T) - \log S(T+t))/(t \log S(T)).$$

Показано, що складність завдань падає приблизно з постійною швидкістю. Отримані оцінки дозволяють прогнозувати зниження складності й визначати час життя ключа.

Завдання, покладені в основу безпеки алгоритмів автентифікації, пропонується класифікувати по трьох типах: вибір, розпізнавання, пошук. У ході проектування звичайно

потрібно обґрунтувати складність завдання вибору й знайти або оцінити рішення завдань розпізнавання й пошуку.

Математичні структури, використовувані при проектуванні алгоритмів автентифікації в умовах довіреного верифікатора, як правило, не мають ефективно розв'язні алгебраїчні властивості й розрахункові статистичні характеристики. Це викликає необхідність введення додаткових завдань і обумовлює їхню складність.

Для автентифікації в умовах недовіренного верифікатора використовуються математичні структури з розпізнаваними алгебраїчними властивостями (групи, кільця, категорії) і відповідні ОМЗ.

На підставі аналізу особливостей застосування ОМЗ сформульовані вимоги до ОМЗ і відповідних класів зв'язаних завдань (КЗЗ), до яких поліномиально зводиться завдання порушення безпеки.

Для параметризації й рішення завдання проектування алгоритмів автентифікації пропонується чотирьохрівнева схема (рисунок 2), що характеризується, крім традиційного технічного рівня, наявністю математичного, криптографічного й сертифікаційного рівнів.

Математичний рівень є найбільш специфічним, у значній мірі визначає трудомісткість проектування, вимагає високої наукової кваліфікації виконавців і із цієї причини не може бути ефективно розпаралелений.

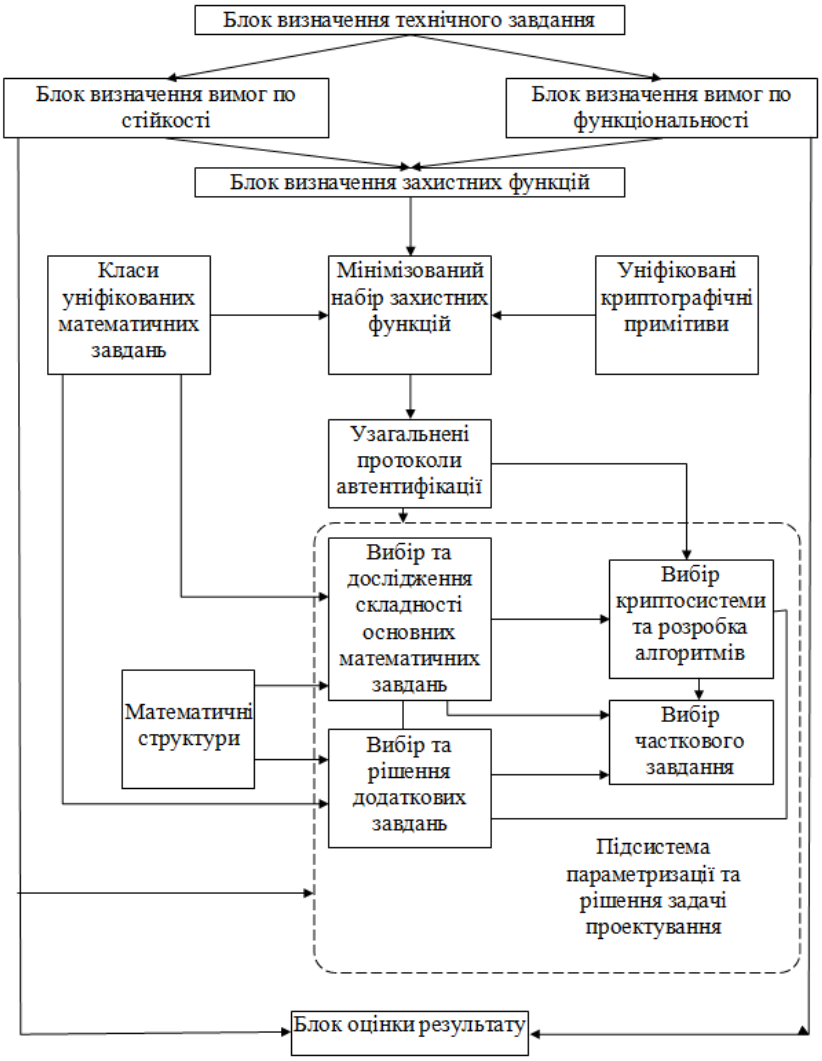


Рисунок 1 – Структурна схема системи проектування алгоритмів автентифікації

Розглянувши запропоновану методологію побудови систем автентифікації перейдемо до практичної побудови системи автентифікації з використанням наведених вище теоретичних відомостей.

За основу системи автентифікації візьмемо автентифікацію з одноразовим паролем з застосуванням USB-ключа.

Структурна схема такої системи наведена на рисунку 1.

З неї ми бачимо, що існують дві сторони процесу автентифікації. З однієї сторони це ЕОМ з операційною системою, у якій встановлений драйвер USB-ключа. З іншої сторони це користувач з USB-ключем, який потребує процесу автентифікації для доступу до системи.

На стороні ПЕОМ, крім драйвера USB-ключа, існує:

- Блок генерування частини одноразового паролю.
- Блок автоматичної генерації паролю.
- Блок перевірки ПІН-кода доступу до USB-ключа користувача.
- Блок синхронізації по часу.
- Блок підрахунку кількості повторів.
- Блок журналювання.
- Блок вибору методу автентифікації.
- БД користувачів з визначенням їх прав доступу.

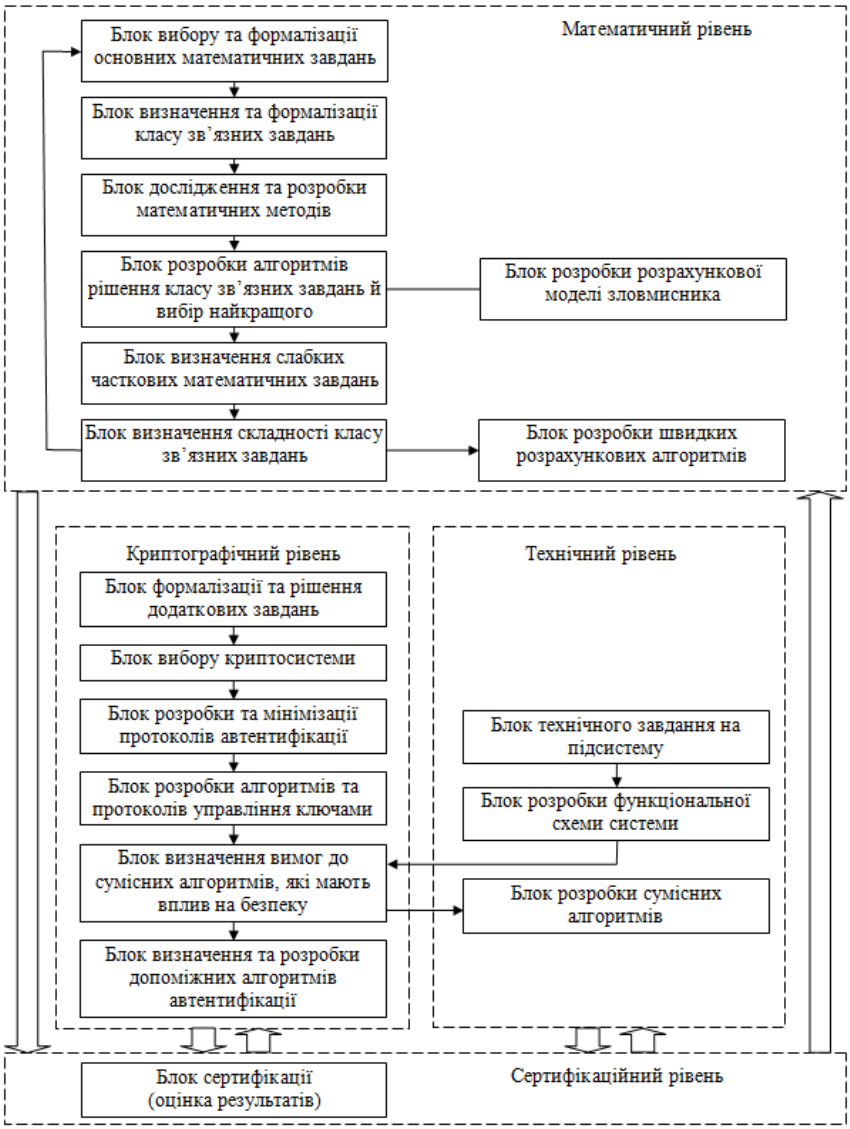


Рисунок 2 – Структурна схема системи параметризації та рішення завдання проектування

На стороні флеш-накопичувача існує:

- Блок криптографічних перетворень.
- БД параметрів користувача.

Процедура автентифікації відбувається наступним чином:

1. На ПЕОМ встановлюється драйвер USB-ключа.

2. Користувач під’єднує при вході у систему USB-ключ, для початку процедури автентифікації.

3. Система видає запит ПІН-коду доступу до USB-ключа.

4. Після введення ПІН-коду, видається вікно у якому потрібно ввести логін та пароль, необхідний для виконання процедури автентифікації й допуску користувача до системи.

5. Після введення паролю, на його основі формується у блоці генератора ключів пароль, частина якого відсилається до флеш-накопичувача.

6. Програмне забезпечення, встановлене на флеш-накопичувачі, згідно заданих таємних криптографічних алгоритмів перетворює цю частину ключа й надсилає відповідь.

7. На ПЕОМ відбуваються аналогічні перетворення, які заносяться у зашифрованому вигляді, до БД користувачів, та паролів.

8. Отримані з флеш-накопичувача дані порівнюються, з тими, які записані у БД.

9. Якщо дані співпадають, то користувач отримує доступ до системи з наданими йому правами. У іншому випадку, надається відмова у доступі.

Кількість повторів обмежується трьома спробами. Усі дії заносяться до журналу подій (у лог-файл). У випадку підозрілих дій видається сигнал адміністраторові ПЕОМ.

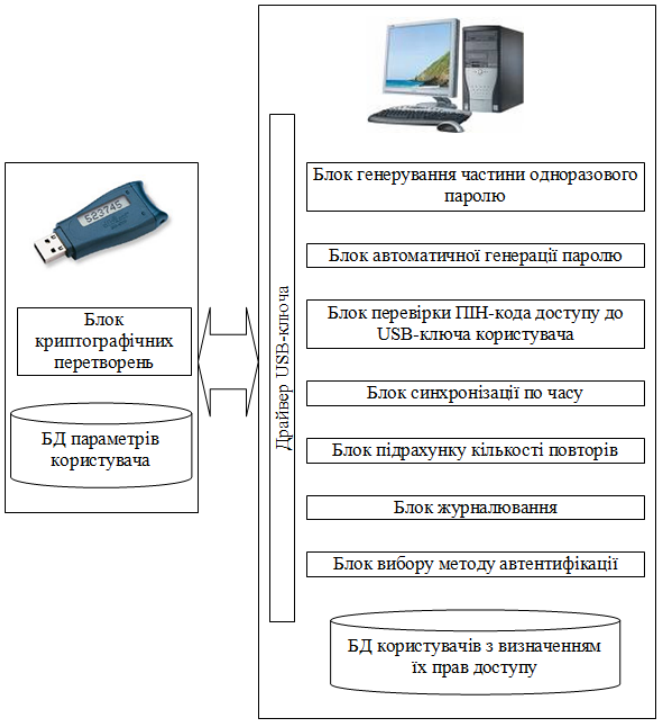


Рисунок 3 – Структурна схема системи автентифікації з використанням USB-ключа

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів проектування сервісів автентифікації. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем проектування сервісів автентифікації. Досліджена система проектування сервісів автентифікації. На основі отриманих результатів досліджень створена програмна реалізація системи проектування сервісів автентифікації. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання проектування сервісів автентифікації. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
2. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).
3. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
4. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
5. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
6. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
7. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
8. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
9. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
10. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
11. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
12. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
13. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
14. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
15. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
16. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
17. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).
18. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019. (Scopus).
19. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629. (Scopus).
20. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 873-884. (Scopus).

УДК 004

Є.Ситнік, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАННЯ ОБРАЗІВ У СТРУКТУРІ ТЕХНІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ БАНКІВСЬКОЇ УСТАНОВИ

У статті програмне забезпечення, яке призначено для системи розпізнання образів у структурі технічного захисту інформації банківської установи. Метою розробки є дослідження та програмна реалізація системи розпізнання образів у структурі технічного захисту інформації банківської установи. Об'єктом дослідження є процес розпізнання образів у структурі технічного захисту інформації банківської установи. Предметом дослідження є методи розпізнання образів у структурі технічного захисту інформації банківської установи. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи розпізнання образів у структурі технічного захисту інформації банківської установи. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, розпізнання образів, технічний захист інформації

Постановка проблеми. Завдяки експоненціальному зростанню ІКТ-технологій індустрія цифрового банкінгу досягла величезних успіхів у зручних, ефективних і швидких фінансових транзакціях. У результаті з'явилися численні нові банківські послуги, продукти та можливості для бізнесу. Розумна автентифікація за обличчям – це передова технологія, яка використовується в мобільному банкінгу. Користувачі можуть використовувати цю технологію для перевірки своєї ідентифікації за допомогою функції розпізнавання обличчя камери на своєму мобільному пристрої. Цей метод використовує складні алгоритми, які можуть аналізувати обличчя людини та виділяти відмінні характеристики, які можна побачити на ньому.

Атрибути зображень різних осіб потім класифікуються за допомогою алгоритмів навчання та методу кластеризації K-середніх. Для автентифікації осіб використовуються штучна нейронна мережа (ANN), адаптивна нейронна система нечіткого висновку (ANFIS) і комп'ютерна система дерева рішень (DT). У цьому запиті використовується обличчя. Крім того, метод Wild Horse Optimizer (WHO) використовувався для підвищення точності та оптимізації систем машинного навчання шляхом зважування функцій кластера. Нечітка логіка використовується для прийняття рішень щодо автентифікації на основі результатів алгоритмів машинного навчання. Найкраща функція з широкого набору даних вибирається за допомогою техніки, заснованої на еволюційних алгоритмах.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи розпізнання образів у структурі технічного захисту інформації банківської установи.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи розпізнання образів у структурі технічного захисту інформації банківської установи.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем розпізнання образів у структурі технічного захисту інформації банківської установи.

– Дослідження системи розпізнання образів у структурі технічного захисту інформації банківської установи.

– Програмна реалізація системи розпізнання образів у структурі технічного захисту інформації банківської установи.

Об'єктом дослідження є процес розпізнання образів у структурі технічного захисту інформації банківської установи.

Предметом дослідження є методи розпізнання образів у структурі технічного захисту інформації банківської установи.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Як можна використовувати розпізнавання зображень у фінансових установах:

1. Виявлення шахрайства

Однією з головних проблем у фінансовій сфері є запобігання та виявлення шахрайства, яке може призвести до значних збитків і репутаційної шкоди. Розпізнавання зображень може допомогти фінансовим установам перевірити особу та автентичність клієнтів, документів і транзакцій. Наприклад, розпізнавання зображень можна використовувати для сканування та перевірки паспортів, водійських прав та інших форм ідентифікації, а також для їх порівняння із зображеннями обличчя, зробленими камерами чи мобільними пристроями. Розпізнавання зображень також можна використовувати для виявлення аномалій і невідповідностей у чеках, рахунках-фактурах, квитанціях і контрактах, а також для позначення підозрілих дій і поведінки.

2. Кредитний скоринг

Іншим важливим аспектом фінансів є оцінка кредитоспроможності та профілю ризику позичальників і кредиторів. Розпізнавання зображень може допомогти фінансовим установам покращити свої моделі та алгоритми оцінки кредитоспроможності шляхом включення альтернативних джерел даних і функцій. Наприклад, розпізнавання зображень можна використовувати для аналізу профілів у соціальних мережах, поведінки в Інтернеті та особистих уподобань потенційних клієнтів, а також для отримання інформації із зображень їхніх активів, власності та способу життя. Розпізнавання зображень також можна використовувати для моніторингу ефективності та стану кредитів і застави, а також для прогнозування ймовірності дефолту та повернення.

3. Обслуговування клієнтів

Третій спосіб використання розпізнавання зображень у фінансах – покращення обслуговування клієнтів і покращення досвіду. Розпізнавання зображень може допомогти фінансовим установам пропонувати більш персоналізовані та зручні послуги та продукти своїм клієнтам, а також підвищити їхню лояльність і задоволеність. Наприклад, розпізнавання зображень можна використовувати для створення чат-ботів і віртуальних помічників, які можуть розпізнавати емоції, наміри та потреби клієнтів і реагувати на них, а також надавати відповідні рекомендації та рішення. Розпізнавання зображень також можна використовувати для біометричної автентифікації та способів оплати, таких як розпізнавання обличчя, сканування відбитків пальців і розпізнавання райдужної оболонки ока.

4. Аналіз ринку

Четвертий спосіб використання розпізнавання зображень у фінансах – це підтримка аналізу ринку та прийняття рішень. Розпізнавання зображень може допомогти фінансовим установам отримати розуміння та інформацію з різних джерел візуальних даних, таких як супутникові зображення, аерофотознімки та відеопотоки. Наприклад, розпізнавання зображень можна використовувати для вимірювання та прогнозування економічної діяльності, попиту та пропозиції в різних секторах, регіонах і країнах, а також для визначення тенденцій і закономірностей. Розпізнавання зображень також можна використовувати для оцінки та порівняння ефективності та вартості різних компаній, продуктів і брендів.

5. Відповідність нормативним вимогам

П’ятий спосіб використання розпізнавання зображень у фінансах – це забезпечення відповідності нормативним вимогам і звітності. Розпізнавання зображень може допомогти фінансовим установам дотримуватися правил і стандартів, встановлених владою та регуляторами, а також уникнути покарань і штрафів. Наприклад, розпізнавання зображень можна використовувати для автоматизації та оптимізації процесів збору, перевірки та подання даних, а також для зменшення помилок і ризиків. Розпізнавання зображень також можна використовувати для моніторингу та аудиту діяльності та операцій фінансових установ, а також для виявлення будь-яких порушень і повідомлень про них.

6. Майбутні перспективи

Розпізнавання зображень – це потужний і універсальний інструмент, який можна використовувати у фінансах для підвищення ефективності, точності, безпеки та інновацій. Однак розпізнавання зображень також стикається з деякими проблемами та обмеженнями, такими як якість даних, конфіденційність, етика та упередженість. Тому фінансовим установам необхідно обережно й обережно приймати та впроваджувати розпізнавання зображень, а також слідувати найкращим практикам і вказівкам. Розпізнавання зображень не замінює людське судження та досвід, а радше доповнює та сприяє. Розпізнавання зображень може допомогти фінансовим установам досягти своїх цілей і завдань, а також створити цінність і вплив на своїх клієнтів і зацікавлених сторін.

Ідентифікація облич людей у розумних банківських системах за допомогою штучних нейронних мереж

Використання мобільного телефону для здійснення банківських транзакцій стало звичайною та популярною практикою в епоху цифрових технологій та Інтернету [1]. Безпека та автентифікація користувачів стають все більш важливими з розвитком мобільного банкінгу та зростанням кількості користувачів [1]. Щоб покращити та полегшити це, інтелектуальна автентифікація обличчя була представлена як нова та потужна технологія [3]. Рівень безпеки біометричних пристроїв має бути підвищений, щоб забезпечити ефективну систему, особливо для онлайн-банкінгу [4]. Мобільна автентифікація може бути відповідним рішенням, яке дозволяє здійснювати онлайн-банкінг, мобільний банкінг і мобільні платежі таким чином, щоб легко забезпечити безпеку [5 , 6]. Лише автентифікація чутлива до атак; у випадках крадіжки або довірених третіх осіб безпеку можна легко порушити [7]. Хакери можуть легко зламати безпеку, оскільки більшість паролів виглядають слабкими. Безпечне банківське обслуговування дає клієнтам впевненість у тому, що їхня інформація в безпеці та що вони можуть з упевненістю здійснювати безпечні операції [8]. Для створення безпеки в системі онлайн-банкінгу, однією з яких є Мобільний банк, наразі були представлені різні методи [9]. Кожен із цих методів намагався виявити атаку за допомогою певної логіки та стратегії та запобігти проникненню в систему [10]. Незважаючи на багато зусиль, які були зроблені, ці методи все ще стикаються з проблемами безпеки та не можуть підтримувати належне покриття безпеки в цих системах [11]. Тому в цій статті ми представимо модель, яка використовує оптимізовані гібридні можливості для виявлення ідентичності зразків і автентифікації людей за допомогою онлайн-зображення мобільного телефону [12]. Запропонована в статті модель реалізована на основі підходів штучних нейронних мереж (ШНМ), адаптивних нейронних нечітких мереж (АНФІС) та дерев рішень (ДТ) для автентифікації особи. Однак ці методи самі по собі не мають високої точності. Тому ми використали алгоритм оптимізації дикого коня, щоб покращити продуктивність цих систем машинного навчання, і ми використали нечітку комбінацію результатів, щоб прийняти остаточне рішення. У цій роботі алгоритм автентифікації обличчя для мобільного банкінгу моделюється за допомогою програмного забезпечення MATLAB, а потім реалізуються та перевіряються плани, методи та інші запропоновані елементи для потрібної системи. Нарешті, результати моделювання порівнюються з іншими методами автентифікації. У цьому контексті ми спробуємо створити інтелектуальну технологію для безпечного та зручного банківського обслуговування, яка базуватиметься на ідентифікації людей на основі

зображень їхніх облич під час дзвінків з мобільного телефону. Поки що етап автентифікації використовувався в мобільному банкінгу, а існуючі алгоритми в цій галузі є проникними та мають слабкі місця в безпеці. У цьому контексті, після огляду існуючих методів автентифікації та їх порівняння, представлено новий інтегрований метод, заснований на гібридній моделі, оптимізованій за допомогою ВООЗ у мобільному банкінгу, для встановлення більшої безпеки та точності. Технологія розпізнавання обличчя (FRT) відома як обладнання для підтримки перевірки особи та автентифікації. Було досягнуто великих успіхів у розробці точних і стійких до втручання рішень FRT за допомогою технологій машинного навчання (ML) і штучного інтелекту (AI), як на чіпі, так і в хмарі. Ці розробки привели до більшої впевненості банків у використанні цієї технології для широкого спектру програм і варіантів використання. Використання еволюційних алгоритмів як нового підходу в цій статті може допомогти створити гібридну модель ідентифікації. Виходячи з цього, ми змогли допомогти підвищити безпеку FRT за допомогою проблеми зіставлення ознак, отриманої з набору зображень людей за допомогою генетичного алгоритму. Банки також безпосередньо використовують можливості машинного навчання, технології штучного інтелекту та еволюційних алгоритмів для покращення біометричних характеристик і розпізнавання особи. Це важливо і дає банкам впевненість, що біометрична технологія є безпечною та надійною. Звичайним методам виявлення та аналізу рис обличчя здебільшого не вистачає надійності та тривалий час обчислень. Ця стаття має на меті виявити способи, за допомогою яких машини можуть навчитися автоматично інтерпретувати інформацію в обличчях без необхідності ручної кластеризації функцій, використовуючи підхід глибокого навчання. Важливі внески поточної роботи підсумовуються таким чином:

– Вона представляє систему автентифікації за обличчям із реалізацією на основі машинного навчання з використанням гібридної моделі для динамічної автентифікації.

– Запропонована гібридна техніка була розроблена та перевірена за допомогою метаевристичного алгоритму ВООЗ у м’якому моделюванні на основі автентифікації людей для підвищення точності розпізнавання.

– Сегментація ознак, отриманих із різних типів зображень, базується на моделі кластеризації на основі K-середніх для трьох наборів методів машинного навчання: ANFIS, ANN і Дерево рішень (DT).

– Використання системи нечіткої логіки для прийняття рішень для ідентифікації людей з найвищою можливою точністю.

– Генетичний алгоритм використовувався для зіставлення ознак, вибору цих ознак і зменшення функцій шляхом видалення ознак, несумісних з реальними людьми в кожній системі. У цьому випадку мобільна реалізація виконується процесорами всіх типів телефонів.

В даний час системи автентифікації користувачів мобільних телефонів за допомогою PIN-коду, відбитків пальців і методів розпізнавання обличчя мають ряд обмежень. У статті [13] проведено порівняння одномодальних і мультимодальних поведінкових біометричних особливостей, тоді як досліджувані техніки розглядають різні дії, такі як набір тексту, прокручування, малювання чисел і натискання на екрані. Для кожної модальності реалізована окрема рекурентна нейронна мережа (RNN) з потрійними втратами. Потім виконується зважена комбінація різних модальностей на рівні балів.

Посилання [14] реалізує комплексний підхід до безпеки розумного дому, який покращує конфіденційність і безпеку за допомогою двох різних технологій, що розвиваються, а саме автентифікації обличчя та розпізнавання мовлення за допомогою його мобільного телефону/планшета/ПК. Для здійснення всього процесу використовуються нейронні мережі. Конфіденційність даних і обмеження ресурсів мобільних пристроїв були двома основними проблемами автентифікації, для вирішення яких у статті [15] запропоновано гібридне рішення. У першому часткове семантичне шифрування використовується для здійснення шифрування на основі алгоритму Пайє. На відміну від цього, останній розгортає глибоку згорточну нейронну мережу та локальну потрійну

комбінацію шаблонів для досягнення розпізнавання обличчя.

Оскільки глибокі нейронні мережі (DNN) не стійкі до вхідних збурень, моделі розпізнавання обличчя (FRM) у DNN страждають від цієї вразливості. Згідно з представленим методом [16], ворожі атаки розроблені після змін збереження ідентичності в обличчях, і в цій ситуації спостерігаються дефекти FRM для розпізнавання зображень, що належать до тієї самої ідентичності. Моделювання цих семантичних змін, що зберігають ідентичність, здійснюється через збурення, обмежені напрямком і величиною в прихованому просторі Style GAN. Важливим моментом є те, що семантична надійність FRM визначається статистичним описом збурень, які призводять до збоїв у FRM.

Щоб розвинути продуктивність розпізнавання обличчя на основі відео, пропонується нова семантична модель підпростору [17 , 18]. Важлива мета полягає в тому, щоб створити відповідний низьковимірний підпростір для кожної людини, на основі якого будується семантична модель для категоризації ключових кадрів людини в певні класи. Згодом, після семантичної класифікації, ключові кадри, що належать до тих же класів, використовуються для навчання лінійних класифікаторів для розпізнавання. Цікаво, що масштабні експерименти з базою даних відео з великими обличчями (XM2VTS) показують, що вищезгадана методологія досягає значного підвищення продуктивності порівняно з традиційними методами.

Як правило, для підтвердження особи користувача автентифікація користувача смартфона здійснюється за допомогою механізмів (пароль або шаблон безпеки). До переваг цих механізмів можна віднести простоту, дешевизну, швидкість для частого входу. З цим досвідом вони пошкоджуються так само, як напад плечем або розмазування. Цю проблему можна вирішити шляхом автентифікації користувачів за допомогою їх поведінки (тобто поведінки дотиків) під час використання смартфонів. Така поведінка включає тиск пальця, розмір і час натискання під час натискання клавіш. Вибір функцій (із цих поведінок) може відігравати важливу роль у продуктивності процесу автентифікації. Таким чином, мета статті [19 , 20] полягає в тому, щоб запропонувати добре організовану техніку автентифікації, яка забезпечує неявну автентифікацію для користувачів смартфонів, не накладаючи додаткових витрат на спеціальне обладнання та враховуючи обмежені можливості смартфона. Спочатку, відповідно до ставлень фільтра та оболонки, розміщуються методи вибору ознак оцінки, а потім використовується найкращий метод, щоб запропонувати метод неявної автентифікації. Слід зазначити, що оцінка цих методів проводиться відповідно до випадкового класифікатора лісу.

Розпізнавання обличчя вказує на те, що це єдині дані, доступні в реальному світі в багатьох функціональних програмах, що призводить до значного покращення продуктивності для більшості існуючих підходів FAR на основі глибокого навчання. Пропонується просторово-семантичне навчання (SSPL), метод, який вимагає двох кроків для навчання [11]. Щоб дізнатися про просторово-семантичні відносини з великомасштабних немаркованих даних обличчя, спочатку будуються три допоміжні завдання: завдання ротації фрагментів (PRT), завдання сегментації фрагментів (PST) і завдання класифікації фрагментів (PCT). Зокрема, PRT використовує самоконтрольоване навчання, щоб використовувати переваги просторової інформації, що міститься на фотографіях обличчя. На основі моделі аналізу обличчя PST і PCT відповідно охоплюють семантичну інформацію зображень обличчя на рівні пікселів і на рівні зображення. Другий крок – перенесення просторово-семантичних знань, отриманих із допоміжної діяльності, до завдання FAR. Це дає змогу точно налаштувати попередньо підготовлену модель за допомогою відносно невеликої кількості позначених даних. Описано технології побудови смарт-камер для семантичної обробки зображень на основі ядер ELcore [12]. Розглянуто етапи семантичного аналізу зображення для розпізнавання облич. На ELcore DSP-ядер виявлені та впроваджені на практиці ресурсомісткі алгоритми. Запропоновано метод автоматичного порівняльного маркування м’якої біометрії обличчя [13]. Проводяться подальші дослідження щодо необмеженого розпізнавання обличчя людини з використанням цієї порівняльної м’якої

біометрії в галереї з мітками людей (і навпаки).

Стаття [13] представила просту та ефективну нечітку нейронну мережу типу 2 на основі глибокого навчання на основі Фур'є для проблем великої розмірності. Правила будуються безпосередньо шляхом швидкого перетворення Фур'є. Вхідна матриця/вектор сегментована, і кожен сегмент представляє нечітке правило. Верхні/нижні межі спрацьовування правила отримують шляхом перетворення Фур'є. Вихід обчислюється простим методом редукції типу. Усі попередні та наступні параметри оптимізовано простим градієнтним спуском і розширеним фільтром Калмана на основі нечіткої коретропії. Розмір ядра звичайних фільтрів на основі коретропії визначається нечіткою системою. Збіжність методу навчання доводиться методом Ляпунова. Ефективність запропонованого підходу підтверджується задачею розпізнавання обличчя (1024 вхідних змінних), розпізнаванням цифр англійського рукописного тексту (1024 вхідних змінних) і задачею моделювання з набором даних реального світу (32 вхідні змінні). Моделювання та порівняння демонструють перевагу представленої схеми.

Згідно з дослідженнями, проведеними в цьому розділі, кожне дослідження представляло новий метод вирішення проблеми підтвердження особи. Досліджувані методики мають переваги та недоліки, які зазначені в табл. 1 . Важливим питанням, яке не було досліджено за допомогою всіх методів, є відсутність довіри до методів автентифікації людей у цих дослідженнях, які будуть використовуватися для мобільного банкінгу. Щоб забезпечити різноманітність методів, заснованих на гібридній моделі, у цій роботі було зроблено спробу посилити надійність запропонованої системи автентифікації. Критерій надійності для автентифікації в цьому дослідженні підвищується за допомогою техніки нечіткої логіки та нечітких правил, що її керують.

Дослідження щодо використання біометричних технологій у банках нещодавно набули важливого значення для кращого розпізнавання нових клієнтів, безпечної автентифікації існуючих клієнтів, захисту транзакцій з великою вартістю та боротьби з шахрайством. Цікаво, що більшість традиційних фізичних відділень банків використовують біометрію. У цьому контексті останні цифрові платформи також використовують біометрію. Ця технологія вважається єдиним надійним інструментом для гарантування ідентифікації та гарантування банківської безпеки в усіх каналах.

Тенденції, що призводять до впровадження біометрії серед банків, численні та включають наступне:

– Поява мобільних телефонів і багатогранної біометричної автентифікації на основі мобільних телефонів.

– Поява біометричних банківських карт означає «прощай з пін-кодами».

– Міжканальний прийом. Біометрія запроваджується в усіх банківських каналах – за підтримки відкритих банківських API, нормативних актів, таких як PSD2, які надихають на використання біометрії в сценаріях багатофакторної автентифікації, і пристроїв Інтернету речей, які підтримують голос і відео, і все частіше стикаються з біометрією.

У цій статті обговорювалися онлайн і мобільний банкінг і методи автентифікації. Також були досліджені проблеми безпеки в мобільному банкінгу. У цьому контексті було представлено новий метод вирішення основної проблеми безпеки та автентифікації в мобільному банкінгу. Запропонований метод є комбінацією методів інтелектуального аналізу даних, включаючи методи глибокого навчання, включаючи штучну нейронну мережу (ANN), адаптивну нейронну нечітку мережу (ANFIS) і алгоритм дерева рішень C4.5, усі покращені за допомогою оптимізації дикого коня BOO3 алгоритм. Далі описані етапи реалізації схеми автентифікації особи за допомогою облич людей на основі запропонованої гібридної моделі. Основою гібридного моделювання цієї роботи є сумісність рис, витягнутих із зображень обличчя людей для автентифікації.

Цей процес включає:

Етап 1: набір даних, який використовується в цій статті, може містити будь-які типи даних, які використовуються у сфері мобільного банкінгу. Однак, оскільки ця робота

зосереджена на автентифікації реальних осіб, ми спробуємо використати набір зображень. Різних людей слід використовувати з різних точок зору. Посилання [48 , 49] можуть бути серед наборів даних, використаних у цьому дослідженні.

Етап 2: розробка вдосконаленої методології є важливим питанням у літературі з аналізу даних, і його часто ігнорують як крок у процесі аналізу даних. Важливим моментом є те, що в реальних додатках машинного навчання спостерігається протилежна ситуація і уникається бажана точність. У цій ситуації, порівняно з існуючими методами машинного навчання, намагаються використовувати модифіковані або посилені підходи цих методів. Також для підготовки даних розглядаються дві основні задачі:

– Виконуючи проекти інтелектуального аналізу даних, організовуйте дані в стандартизованій формі, щоб вони були готові до обробки за допомогою інтелектуального аналізу даних та інших комп'ютерних інструментів.

– Набір даних має бути підготовлений таким чином, щоб забезпечити найкращу продуктивність методів аналізу даних.

Етап 3: пов'язаний з категоріями, вибір категорій для інтеграції має бути обраний таким чином, щоб ці категорії доповнювали одна одну, і кожну з них слід коротко пояснити. Розділення навчальної та тестової вибірок для категорій, які доповнюють одна одну, здійснюється на основі ознак, отриманих на етапі обробки даних. У цьому випадку ми застосували контрольовану техніку кластеризації нечітких K-середніх до набору зразків зображень, щоб навчити систему машинного навчання. Цей набір даних містить 77 ознак, отриманих із зображень обличчя суб'єктів.

Етап 4: на цьому етапі для кожної категорії ознак використовується техніка вибору ознак на основі генетичного алгоритму, щоб вибрати набір ознак, які мають найбільшу сумісність у правильній оцінці людей. На основі запропонованих алгоритмів класифікація та розділення вибірок виконується за допомогою комбінованої класифікації, і на цьому етапі виконується зіставлення даних на основі редукції ознак за допомогою запропонованого алгоритму прийняття рішень та застосування їх до даних навчання людей. Для цього кроку визначається цільова функція узгодженості, яка представлена в наступному розділі.

Етап 5: на цьому етапі нашої роботи ми будемо використовувати алгоритм ВООЗ для вдосконалення кожної системи машинного навчання на основі призначених їм функцій. Метою цього етапу є підвищення точності систем машинного навчання на основі зважування специфічних особливостей кожної системи.

Етап 6: Результати, отримані за категоріями, об'єднуються у формі більшості голосів. На цьому етапі вибрані значення для кінцевого результату отримують на основі категорій і на основі нечіткого колективного рішення щодо даних із відповідей. Нечіткі правила, що керують цим рішенням, представлені в наступному розділі.

Розробка структурної схеми

Структурна схема системи наведена на рисунку 1.

Структурно система складається з наступних частин:

1. Відеокамера спостереження, з якої поступає інформація систему розпізнання образів.

2. Система розпізнання образів, яка складається з наступних блоків:

– Блок читання картинки з відеокамери. Він призначений для читання картинок з відеокамери й подання даних на блок аналізу та виділення ознак за допомогою багатопрохідної схеми розпізнавання образів на основі кластерного аналізу.

– Блок аналізу та виділення ознак за допомогою багатопрохідної схеми розпізнавання образів на основі кластерного аналізу. Він є основою системи, й за допомогою нижчеописаного алгоритму проводить розпізнання осіб, та машин, які перетнули межу території банківської установи.

– Блок класифікації та опису об'єкта. Він дозволяє, виходячи з даних, отриманих від блоку аналізу та виділення ознак за допомогою багатопрохідної схеми розпізнавання образів на основі кластерного аналізу, розподілити куди заносити отримані дані, у поля бази даних,

які відповідають за осіб, або у поля бази даних, які відповідають за машини.

3. База даних журналювання розпізнаних образів. У цю баз даних заносяться усі дані, які відносяться до розпізнаних об’єктів, будь то людина, або автомобіль.

4. База даних образів облич. У цій базі даних зберігаються фотографії усіх працівників установи та відвідувачів, з виділенням характерних точок, для кожного обличчя, за якими можливо ідентифікувати або працівника банку, або відвідувача.

5. База даних образів цифр та букв на номерах автомобілів. У цій базі даних зберігаються образи усіх цифр та букв, з яких можуть складатися номера автомобілів, а також номера усіх автомобілів, які перетинали кордон приміщення банківської установи, який встаткований відеокамерами спостереження.

Так як розпізнавання образів відбувається за допомогою алгоритму багатопрохідної схеми розпізнавання образів на основі кластерного аналізу, то наведемо цей алгоритм.

Багатопрохідна схема розпізнавання образів на основі кластерного аналізу

Багатопрохідна схема складається в послідовній класифікації тих самих образів спочатку за допомогою образонезалежних алгоритмів розпізнавання, а потім – алгоритмів, що використовують особливості образів номерів автомобілів, і особливості розпізнавання осіб людини.

Метою багатопрохідної схеми розпізнавання з навчанням є адаптація до особливостей образів. При цьому на кількість і характеристики використовуваних образів не накладається істотних обмежень.

Багатопрохідна схема розпізнавання містить у собі попереднє розпізнавання образів, формування бази даних результатів попереднього розпізнавання, додаткове самонавчання на підставі отриманих результатів розпізнавання, наступні перерозпізнавання образів з урахуванням самонавчання, формування остаточних результатів.

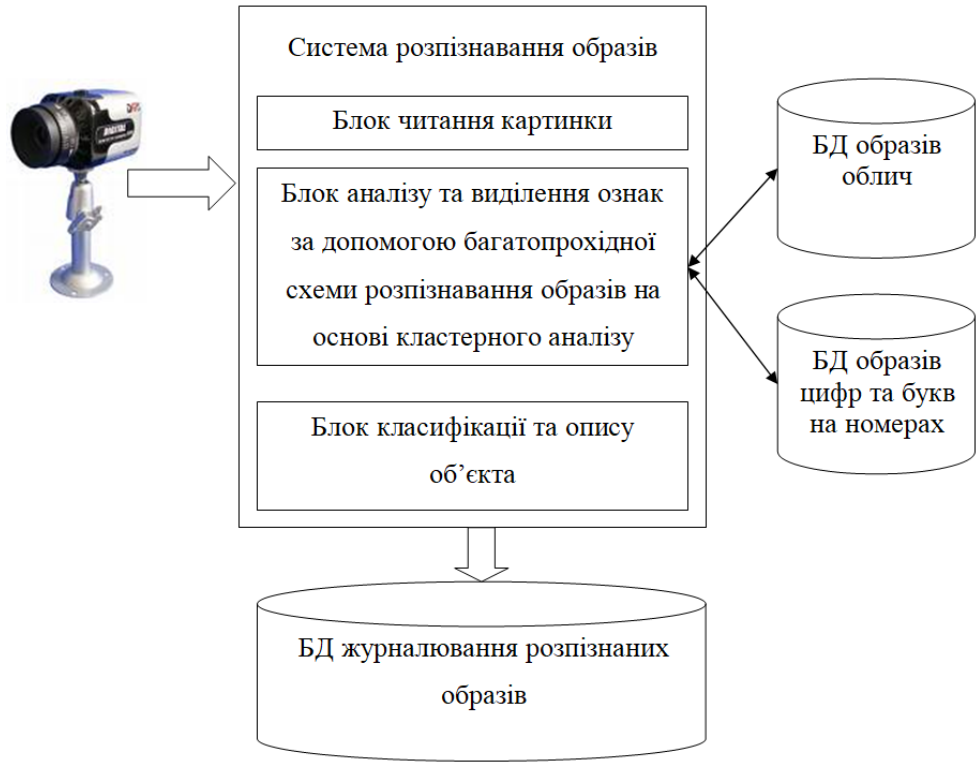


Рисунок 1 – Структурна схема системи

Навчальна вибірка готується першим проходом розпізнавання, що за допомогою образонезалежного алгоритму розпізнавання образів $\Re 1$. Алгоритм $\Re 1$ надає колекції альтернатив (варіанти розпізнавання з оцінками) образів, що відповідають розпізнаним образам, і атрибути образів.

Крім цього перший прохід забезпечує валидацію образів, тобто позначку надійно розпізнаних образів, за допомогою наступних двох механізмів:

– облік оцінок альтернатив, сформованих алгоритмом з монотонними оцінками
– словникове або контекстне підтвердження досить довгого слова.

Комбінація цих механізмів валідирує частину результатів розпізнавання. Таким чином, множина розпізнаних образів розбито на підмножини, що задаються розмірами й атрибутами образа. Як контейнер навчання, призначеного для зберігання розмічених образів, може виступати база даних, що формується на диску, або динамічна структура в оперативній пам'яті.

Метою навчання є побудова набору кластерів, частина яких надійно відділений друг від друга, а кластери, що залишилися, містять вказівки на їхню близькість до інших з погляду обраної Хаусдорфової метрики.

У нашій програмі для кластеризації ми використовували алгоритм ланцюгового розгорнення, алгоритм відноситься до групи методів одиночного зв'язку. Алгоритм кластеризації, що базується на відстані Хаусдорфа й наведений в [4] якісно відрізняється від широко відомих методів, насамперед, через структуру інформації, що підлягає класифікації під час навчання й цілей кластеризації. Мінімізація числа кластерів, швидкодія й інші питання класичної кластеризації мають для нас важливе, але не першорядне значення.

Результатом кластеризації служить множина об'єктів, кожний з яких містить суму стандартизованих ненормалізованих растрів, із близькими ознаками й загальними атрибутами, ланцюгова відстань між якими не перевищує певного заздалегідь межі. Кожний із кластерів крім успадкованих ознак має потужність (числом складових його растрів).

Ознаки кластера використовуються на етапі побудови еталонів накладення, що повинні сформувати базу надійних кластерів, а для кластерів, що залишилися, відповістити на запитання про можливість їхнього використання. Аналіз кластерів образів певного алфавіту містить кілька операцій:

– Перейменування кластера, що складається в розпізнаванні центра сумарного растра (або всієї суми растрів, розглянутого як напівтоновий образ) досить точним алгоритмом розпізнавання образів. Від перерозпізнавання очікується зняття систематичних помилок алгоритму першого проходу, що не зобов'язаний бути адаптивним до особливостей накреслень використовуваних образів.

– Об'єднання двох різноїменних прилеглих кластерів з метою перейменування одного з них. Вирішує завдання, аналогічні завданням перейменування кластера.

– Знищення кластера, тобто вивід ненадійно розпізнаного кластера з розгляду.

– Угруповання кластера з одним або декількома різноїменними з ним кластерами. Фіксує неможливість або ненадійність розрізнення результатів, отриманих накладенням кластерів з однієї групи.

Після первинного аналізу кластерів залишаються тільки надійні кластери, що володіють достатньої валідністю й потужністю. Серед надійних кластерів проводиться ітераційний процес пошуку образів, оскільки на картинці перебувають не просто якісь образи, а образи, що відбуваються з одного або декількох образів. Кластери розбиваються на кілька груп їх складових, і, можливо, по своїх атрибутах (якщо атрибути присутні). Якщо в процесі аналізу виявиться, наприклад, що на картинці присутня тільки один образ, але є кілька кластерів якої-небудь букви, то в остаточну вибірку кластерів треба взяти тільки один, кращий у деякому змісті, а інші можуть бути помилками розпізнавання, помилками сегментації, можуть виникнути через погану якість зображення.

Ітераційний процес аналізу первинних кластерів закінчується формуванням еталонів алгоритму накладення, що здатний відповісти на ряд питань, пов'язаних з можливістю розрізнення близьких образів. Алфавіт розпізнавання, містить ряд образів невідмінних друг від друга у всій множині накреслень цих образів, для яких, проте, необхідно на якімсь із етапів розпізнавання ухвалити рішення щодо виборі одного зі значень. Подібні по

накресленню друковані образи, близькість яких визначається властивостями алгоритмів розпізнавання образів, також є джерелом помилок образонезалежних алгоритмів.

У той же самий час у межах одного образа, як правило, деякі з образів алфавіту й родинних образів помітні геометрично, наявність же декількох образів на картинці може як утруднити, так і спростити розпізнавання близьких образів. Після завершення етапу аналізу кластерів, що досліджує подібні можливі конфліктні ситуації, стають відомим, наскільки добре побудовані еталони можуть дозволяти колізії родинних образів і образів з однаковим накресленням у різних образах. Інформація про це втримується в переліку груп різноїменних родинних образів і в списку відстаней між нерозрізненими образами.

З оброблених кластерів може бути витягнута множина еталонних кістякових і розширених образів, міра близькості до яких забезпечує достатні характеристики якості розпізнавання. Можливе подання еталонів, що складає із трьох об'єктів:

– кістякова підмножина SKEL, що містить значення растра кластерів у границях щирого кістякового образа;

– розширена підмножина COVER, що містить нулі в границях щирого розширеного образа, мабуть, COVER $\supseteq$ SKEL;

– опис штрафу, що залежить від відстані до найближчої точки кістяковий або розширений образи, що обчислюється за допомогою функцій Pen(i, j, M), аргументами якої є координати точки й множина M.

Кожне з підмножин є матрицею того ж розміру, що й стандартизовані растри, що підлягають кластеризації, і залежать від обсягу й інших характеристик кластера. Накладення, тобто обчислення міри близькості довільного образа й еталона E=||eij|| відбувається в кілька етапів, першим з яких є центрування образа. Для відцентрованого, тобто стандартизованого, образа R =||rij|| підраховується сума покомпонентних добутків значень растрів R і E:

$$\Sigma(R,E) = \Sigma\, \delta\,(rij=1 \wedge eij>0) \bullet eij -$$
$$\Sigma\, Pen(i,j,SKEL(S,\alpha))(\,(rij=0 \wedge eij>0) -$$
$$\Sigma\, Pen(i,j,COVER(S,\beta))(\,(rij=1 \wedge eij<0),$$

яка містить у собі як позитивні добутки точок растра, що потрапили в кістякову область, так і негативні компоненти точок, що не потрапили в розширену область, і штраф за недолік точок у кістяковій зоні. У цьому вираженні присутні як позитивні значення суми растрів, що склали кластер, так і негативні штрафні значення. У такий спосіб обчислена близькість відповідає на запитання про те, наскільки добре розпізнаваний образ відповідає розподілу даного кластера, тобто поліпшує імовірнісні властивості оцінок накладення. Зрозуміло, не слід забувати не тільки про евристичні штрафи, що не володіють імовірнісною природою, але й про обсяг кластера, що породжує кістякову область, тому що утворення малих кластерів не є чимсь винятковим для більшості картинок. Внесок штрафів у загальну суму також поліпшує оцінки за умови оптимізації штрафів за видалення від границі розширеного образа. Результатом накладення є альтернатива:

$$(S(E),\ W\bullet\Sigma(R,E)),$$

де S(E) – код образа кластера з растром E.

W – масштабний коефіцієнт для одержання оцінок, при цьому успадковуються властивості кластера (кегль, атрибути образа).

Спосіб центрування стандартизованого розпізнаваного растра, що полягає в сполученні геометричних центрів вихідного растра, розширюваного симетрично до стандартних розмірів, не може дати задовільних результатів у загальному випадку накладення. Пошук центра доповнюється зрушеннями розпізнаваного растра в невеликій околиці геометричного центра еталона з вибором найкращого результату накладення. Кластерному накладенню властивий ряд проблем, пов'язаних із проблемою пошуку геометричного центра образа. Центрування стандартизованих растрів не дозволяє розпізнавати образи із сильно деформованою рамкою. Для складних випадків центрування

необхідне залучення інших алгоритмів, наприклад, обчислення моментів або використання поліграфічних базових ліній.

Відзначимо, що обчислення досить трудомістких оцінок накладення може бути прискорено перериванням підрахунку суми в ситуації набору значного числа штрафів. Значна різниця в розмірах растрів R і E дозволяє прийняти рішення про відмову накладення даного еталона, це міркування в сукупності з фільтрацією по атрибутах еталонів також прискорюють алгоритм накладень.

Еталони містять ряд додаткових ознак, наприклад, для заборони перерозпізнавання образа по кластеру, породженого цим же образом без участі інших образів.

Побудована система еталонів дозволяє використовувати алгоритм накладення як алгоритм, що формує після перегляду всіх еталонів, що задовольняють розміру розпізнаваного образа, колекцію альтернатив, породжених найближчими еталонами. Також можливе використання еталонних накладень і як алгоритм-експерта, що перевіряє гіпотези про те, наскільки добре досліджуваний образ може бути розпізнаний з деяким заданим кодом образа. Можуть бути отримані наступними результатами перевірки близькості розпізнаваного образа одному з еталонів із заданим кодом:

– найменша відстань досягнута на еталоні, далекому від інших еталонів;

– найменша відстань досягнута на еталоні, що потрапив у групу близьких різноїменних еталонів;

– перевірка близькості не може бути зроблена через відсутність еталонів з даним кодом образа.

Отриманий результат може бути проінтерпретований на відміну від образонезалежного алгоритму-експерта, що відповідає тільки на питання про близькість досліджуваного образа до одному з еталонів, у такий спосіб. Перший результат залежно від того, чи є кластер досить представницьким (великий обсяг, валідність його растрів, що склали), може бути визнаний надійним або рекомендаційним як для обчислення автономних оцінок, так і для рішення конфліктів. Другий результат може бути визнаний надійним тільки для перевірки приналежності образа до всієї групи еталонів, у цьому випадку до колекції альтернатив, збагачуваної кластерною інформацією, можуть бути додані відсутні альтернативи родинних образів. Тобто другий результат фіксує конфлікт родинних або нерозрізнених альтернатив як нерозв'язний у рамках кластерної моделі, що може надалі вирішуватися за допомогою словникового пошуку, словникової корекції. Третій результат є відмовою кластерного накладення. У цьому випадку неможливе порівняння двох образів, код одного одного з яких є присутнім в еталонах, а іншого відсутній.

Очікувана відсутність ряду еталонів припускає комбінування результатів образонезалежного алгоритму й алгоритму кластерного накладення, адаптивного до образів у розпізнаваній картинці. Схема комбінування будується в припущенні, що значна частина (80-90%) картинок уже розпізнана без помилок, внаслідок чого для частки, що залишилася, дорозпізнаваємих образів можливе застосування достатнє трудомістких алгоритмів розпізнавання образів. Це означає, що комбінувати кластерне накладення доцільно з алгоритмом, точність якого перевищує точність алгоритму $\Re 1$ розпізнавання образів на першому проході. Комбінування з більше точним алгоритмом (наприклад, з нейронною мережею) забезпечує як збереження кластерних оцінок у випадку успішно розпізнаних обома алгоритмами образів і дозволених конфліктів, так і збереження точності алгоритму $\Re 1$ з одночасним зниженням його оцінок у випадку не підтвердження його результатів надійними кластерами. Це поліпшує й точність, і монотонність оцінок. Використання образонезалежного алгоритму дозволяє розпізнавати й виставляти оцінки образам, для яких не зібралися підтверджувальні еталони. Недоліком описаного комбінування є одержання оцінок різної природи: як образонезалежних, так і кластерних, зіставлення яких у загальному випадку важко. Початкові параметри схеми комбінування, у першу чергу відносяться порогів оцінок, по яких приймається рішення про надійність розпізнавання, можуть змінюватися

після завершення кластеризації, за рахунок чого відбувається додаткова адаптація до результатів першого проходу розпізнавання картинок.

Властиво дорозпізнавання, тобто другий прохід розпізнавання, містить у собі розпізнавання комбінованим алгоритмом як окремо стоячих, так і склеєних і розсипаних образів, які деяким чином були сегментовані на першому проході. Дорозпізнавання рядка образів починається з експертної оцінки як незмінених, так і сегментованих розпізнаних образів. Кожний розпізнаний образ піддається експертизі на предмет підтвердження оцінки його провідної альтернативи алгоритмом, використовуваним як експерт. Образи, що не одержали підтвердження, перерозпізнаються; деякі групи образів піддаються повторної сегментації. Сегментація, що навіть опирається на той самий перелік відрізків розрізування, при використанні іншого алгоритму розпізнавання образів може дати інші результати визначення границь образів. У той же час відзначимо, що часто успішно працює алгоритм розрізування (як, втім, і склейки), заснований винятково на кластерному розпізнаванні, без складання переліку відрізків можливого розрізування, хоча чисто кластерне розпізнавання працює не завжди у зв'язку із уже згадуваною проблемою можливої неповноти кластерів. Перерозпізнані ланцюжки образів конкурують зі своїми прототипами, утвореними на першому проході. Порівнянню підлягають слова, для яких можливо не тільки обчислення функцій над оцінками образів, що склали слово, але й підтвердження словниково-лінгвістичними методами.

Таким чином, побудований комбінований алгоритм (позначимо його $\mathfrak{R}2$) дозволяє поліпшувати точність і монотонність оцінок розпізнавання як за допомогою образонезалежного алгоритму, так і за рахунок надійних еталонів кластерного накладення. Крім цих поліпшень не можна не відзначити ще одного результату другого проходу, що складається в тому, що в перерозпізнаних рядках частина образів одержала додаткову валідацію від надійних еталонів. По суті справи частина образів, що не одержала такий валідації, може бути змінена наступними етапами розпізнавання, а валидированні образи з високою ймовірністю не можуть піддаватися змінам. Окремо слід зазначити валідацію системою еталонів, що містить один єдиний образ, така гіпотеза перевіряється під час кластеризації.

Висновки. У статті теоретичне узагальнення й рішення наукового завдання дослідження методів розпізнання образів у структурі технічного захисту інформації банківської установи. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем розпізнання образів у структурі технічного захисту інформації банківської установи. Досліджена система розпізнання образів у структурі технічного захисту інформації банківської установи. На основі отриманих результатів досліджень створена програмна реалізація системи розпізнання образів у структурі технічного захисту інформації банківської установи. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання розпізнання образів у структурі технічного захисту інформації банківської установи. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». Advanced Information Systems, 2023, 7(2), pp. 49-56.
2. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». CEUR Workshop Proceedings, Volume 3530, 2023, pp. 256-265.

3. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.
4. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022,
5. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.
6. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w
7. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.
8. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.
9. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.
10. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.
11. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.
12. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.
13. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.
14. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.
15. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.
16. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.
17. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.
18. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.
19. Zhurakovskyi, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.
20. Smirnov O. Kuznetsov A., Zaicenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.
21. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

УДК 004

Б.Сільман, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ FLASH DRIVE

У статті розроблено програмне забезпечення, яке призначено для системи Flash Drive. Метою розробки є дослідження та програмна реалізація системи Flash Drive. Об'єктом дослідження є процес Flash Drive. Предметом дослідження є методи Flash Drive. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи Flash Drive. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, захисту доступу, Flash Drive

Постановка проблеми. Захист конфіденційної інформації сьогодні необхідний практично будь-якому власникові Flash Drive. На сьогоднішній день під засобами захисту інформації на Flash Drive розуміють сукупність різних технічних і програмних систем і пристроїв, використовуваних для рішення різних завдань по захисту інформації, у тому числі попередження витоку, захисту даних на флешці (Flash Drive) від запису й забезпечення повного комплексу мер для безпеки информации, що захищається.

Сучасні засоби захисту інформації покликані забезпечити безпеку даних на Flash Drive, як то:

- захист файлів на флешці від запису;
- захист файлів на флешці від копіювання;
- захист файлів на флешці від видалення;
- інші несанкціоновані дії.

Всі частіше покупці прагнуть придбати не просто флеш-накопичувач, а саме захищені флеш-накопичувачі, і їх прекрасно можна зрозуміти.

Захищені Flash Drive дозволили нам вступити в зовсім нову еру не просто швидкої, але й безпечної передачі й використання інформації.

На флеш носіях, на даний час переноситься дуже багато конфіденційної інформації. І якщо при утраті флеш носія, ця інформація попаде у руки конкурентів то це приведе до значних економічних збитків, у кращому випадку.

У гіршому випадку, наслідки можуть бути катастрофічними для підприємства або якоїсь установи.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи Flash Drive.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи Flash Drive.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем Flash Drive.
- Дослідження системи Flash Drive.
- Програмна реалізація системи Flash Drive.

Об'єктом дослідження є процес Flash Drive.

Предметом дослідження є методи Flash Drive.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Понад 20 років тому флеш-накопичувачі USB, також відомі як флеш-накопичувачі, вважалися проривом у технології портативного зберігання даних. Сьогодні вони розглядаються як серйозна загроза безпеці.

Експерти з кібербезпеки сходяться на думці, що шифрування USB-флешки є найкращим рішенням для захисту конфіденційних даних. Зашифровані USB-накопичувачі є дуже ефективними інструментами для усунення прогалин у ризиках і допомагають забезпечити безпеку даних, пропонуючи комплексний захист паролем (тобто найсучасніші зашифровані диски забезпечують сертифікацію FIPS, відому як FIPS 140-2 рівня 3 або шифрування військового рівня). додаткові функції включають; видалення даних у випадках, коли спроби введення пароля перевищують встановлений ліміт, технологія захисту від несанкціонованого доступу, щоб запобігти доступу хакерів до внутрішніх компонентів накопичувача (з використанням розширеного апаратного 256-бітного шифрування AES у режимі XTS і надійного захисту від несанкціонованого доступу або «On-Device Crypto chip», який забезпечує додатковий рівень укріплення), антивірусний захист і фізичний рівень захисту від несанкціонованого доступу для дисків із платами з епоксидним покриттям або металевими корпусами, наповненими епоксидною смолою, щоб захистити фізичне сховище від несанкціонованого доступу або з часом пошкодження.

Шифрування портативного флеш-накопичувача USB запобігає потраплянню конфіденційних даних у чужі руки. У цьому посібнику крок за кроком показано, як зашифрувати флеш-накопичувач USB у Windows за допомогою вбудованого засобу шифрування, відомого як BitLocker.

Хоча ОС Windows 10 може розблокувати флеш-накопичувач, зашифрований у Windows 7, і навпаки, USB-накопичувачі, зашифровані за допомогою ОС Windows, не можна відкрити в macOS.

Щодня з’являються нові загрози безпеці ваших даних із заражених апаратних носіїв; наприклад, жорсткі диски, компакт-диски чи картки пам’яті, до пошкодженого програмного забезпечення та шкідливих програм. Переважно з онлайн-джерел; веб-сайти, блоги, електронні листи. Тому дуже важливо підтвердити, що ваш захист від вірусів/шкідливих програм активний у кожній точці входу.

Також важливо переконатися, що ваш антивірус оновлений. Важливо також оснастити хости кінцевих точок для будь-якого пристрою за межами брандмауера найновішим антивірусним програмним забезпеченням. Також зверніть особливу увагу на програми з розширеним захистом від шкідливих програм на USB-пристроях, коли вони використовуються на інших пристроях.

Ключовим фактором, який може поставити під загрозу рівень безпеки флеш-накопичувача, навіть незважаючи на шифрування, є якість накопичувача. Дослідження показують, що безкоштовні накопичувачі, якими користуються на конференціях, ділових зустрічах тощо, часто не можна порівняти з фактичними флеш-накопичувачами найвищої якості, які можуть коштувати вам трохи.

Переваги/відмінності використання цих якісних флеш-накопичувачів USB, на відміну від дешевих роздаткових матеріалів, можна побачити в їх продуктивності та довговічності з часом. Отже, важливо знати різні типи та якості флеш-накопичувачів USB і гарантувати, що ви сплачуєте витрати на безпеку своїх даних.

Окрім фізичного захисту файлів, настійно рекомендується також захистити важливі дані на онлайн-серверах, які називаються «хмарою». До таких серверів належать Dropbox, Google Drive або iCloud. Ці інструменти дозволяють легко зберігати та отримувати файли з будь-якої точки світу.

Зашифрований хмарний сервіс усуває ризик втрати даних через неправильне розташування флешки. Усе, що вам знадобиться, це підключення до Інтернету та ваш

пароль. Важливо також зазначити, що не всі хмарні служби зберігання повністю захищають вашу конфіденційність. Більшість хмарних служб можуть допомогти зашифрувати ваші файли, але збережуть ваші ключі шифрування; тобто вони можуть мати доступ до ваших файлів у будь-який час. Це робить ці служби більш вразливими до витоку даних у разі злому системи.

Ви можете обійти це, використовуючи лише хмарні служби з наскрізним шифруванням; це означає, що ваші дані спочатку шифруються перед надсиланням на їхні сервери, і, таким чином, навіть якщо їхні сервери якимось чином незаконно отримали доступ, лише ви зможете розшифрувати їх, щоб отримати доступ до даних у цих файлах.

Якщо вам потрібно використовувати флеш-накопичувач USB для зберігання даних, є способи покращити захист ваших даних. Читайте далі, щоб дізнатися, як захистити флешку.

Перш ніж зберігати їх на флеш-накопичувачі, важливо спочатку визначити, що вважати конфіденційними даними, а також ширші вимоги щодо безпеки. Конфіденційні дані – це будь-яка конфіденційна інформація, для доступу до якої потрібен дозвіл.

Конфіденційні дані включають таку конфіденційну інформацію – оригінальну або скопійовану:

– Захищена медична інформація (PHI).
– Інформація, що дозволяє ідентифікувати особу (PII).
– Записи про освіту.
– Інформація споживача.
– Дані власника картки.
– Конфіденційна інформація про персонал.
– Конфіденційні дані.

Організації повинні мати суворі методи захисту даних та інформаційної безпеки, щоб гарантувати, що ці дані не будуть скомпрометовані через несанкціонований доступ. Вони також повинні дотримуватися відповідного законодавства про персональні дані, наприклад Загального регламенту захисту даних (GDPR).

Деякі організації можуть запровадити заходи безпеки конфіденційних даних, зокрема:

1. Управління даними.
2. Безпечне керування привілейованим доступом.
3. Шифрування.
4. Програми навчання персоналу.
5. Тестування безпеки даних.
6. Класифікація даних.
7. Плани реагування на інциденти.
8. Регулярне резервне копіювання даних систем зберігання.
9. Безпечні процеси видалення.
10. Моніторинг сторонніх і четвертих постачальників.

Як безпечно зберігати дані на флеш-накопичувачі USB.

Настійно рекомендуємо не зберігати конфіденційні дані на флеш-накопичувачі USB, а натомість вибрати безпечніші пристрої для збереження даних. Їх невеликий розмір дозволяє легко транспортувати, але також легко втратити або вкрасти. Це падіння збільшує ризик втрати даних, витоків і порушень даних, що коштує значних витрат для організацій.

Якщо ви все ж використовуєте флеш-накопичувач, дотримуйтеся цих 7 порад, щоб захистити свої дані.

1. Придбайте зашифрований USB

Шифрування захищає конфіденційну інформацію, роблячи її доступною лише для тих, хто має ключ дешифрування. Купуючи флеш-накопичувач, вам слід вибрати флеш-накопичувач військового класу з 256-бітним апаратним шифруванням AES – найнадійнішим алгоритмом шифрування.

Серед інших функцій зашифрованого USB-накопичувача:

– Захист від злому.

– Антивірусне сканування.
– Захист від грубої сили.
– Захист паролем.
– Відповідність TAA.
– Можливості віддаленого керування.
– Сертифікація FIPS 140-2 (рівень 3).
– Відповідність галузевим стандартам безпеки, таким як HIPAA, SOX і GLBA.

2. Використовуйте програмне забезпечення для шифрування USB

В якості альтернативи придбанню зашифрованого флеш-накопичувача користувачі операційної системи Microsoft Windows можуть використовувати BitLocker для шифрування своїх флеш-накопичувачів. Зауважте, що апаратне забезпечення шифрування забезпечує кращий захист, ніж програмне забезпечення.

Інструкції Microsoft щодо ввімкнення BitLocker доступні нижче:

– Перегляньте інструкції щодо ввімкнення BitLocker у Windows 10.
– Перегляньте інструкції щодо ввімкнення BitLocker у Windows 11.

3. Майте резервну копію

У разі втрати, викрадення чи пошкодження флеш-пам’яті ви можете ніколи не відновити дані, що зберігаються на ній. Навіть якщо втрачену або вкрадену флеш-накопичувач повернули, не слід використовувати його знову, оскільки на ньому потенційно може бути встановлено програмне забезпечення-вимагач або інший тип шкідливого програмного забезпечення. Найкраща гарантія відновлення даних на вашому флеш-пам’яті – це резервна копія всіх файлів, збережена в іншому окремому місці зберігання, наприклад у хмарному сховищі.

4. Видалити дані після використання

Після того, як ви зберегли, відредагували та передали свої дані з USB-накопичувача, найбезпечніше негайно повністю видалити їх. Потім вам слід вийняти флеш-накопичувач із USB-порту та зберігати його в надійному місці, щоб уникнути втрати або крадіжки.

5. Встановіть антивірусний захист

З огляду на те, що різні типи зловмисного програмного забезпечення з’являються щодня, підтримувати ваше програмне забезпечення в актуальному стані є надзвичайно важливим. Використовуйте антивірусне програмне забезпечення, яке забезпечує захист від зловмисного програмного забезпечення на всіх кінцевих точках, включаючи жорсткі диски, USB-пристрої та SD-карти – можна заразити всіх.

6. Оновлюйте програмне забезпечення

Експлойти нульового дня використовують невиправлені вразливості програмного забезпечення – поширений вектор атак, який може мати руйнівні наслідки. Кіберзлочинці можуть легко отримувати доступ, редагувати та викрадати дані з уразливих систем і пристроїв, включаючи USB-накопичувачі.

Якнайшвидше встановлення оновлень програмного забезпечення не дозволить кіберзлочинцям скористатися цими вразливими місцями. Більшість операційних систем, включаючи Microsoft Windows, Mac OS/Apple iOS і Linux, пропонують автоматичні оновлення, щоб забезпечити ваш захист.

7. Використовуйте альтернативні методи зберігання

Зрештою, флеш-накопичувачі не є відповіддю, якщо ви хочете серйозно поставитися до безпеки своїх даних. Навіть найбезпечніші USB-накопичувачі не підходять для сучасних методів зберігання даних, як-от хмарне сховище. Хмарні служби пропонують багато інноваційних функцій безпеки, наприклад Secure Access Service Edge (SASE).

SASE – це хмарна модель безпеки, яка використовує брандмауери, брокери послуг хмарного доступу (CASB), захищений веб-шлюз (SWG) і доступ до мережі без довіри (ZTNA). Інші механізми безпеки хмари включають Cloud Security Posture Management і Cloud Infrastructure Entitlement Management (CIEM).

Незважаючи на потужні можливості безпеки, хмарні служби, як і всі сторонні

постачальники, несуть ризики для третіх сторін та інші ризики, пов'язані з їх функціональністю. Організації та окремі особи повинні проводити належну перевірку, щоб переконатися, що їхні хмарні постачальники дотримуються відповідних вимог безпеки даних.

Найкращі методи безпечного зберігання конфіденційних даних на флеш-накопичувачах USB діляться на дві категорії: профілактичні та реактивні. Профілактичні заходи вимагають глибокого розуміння того, де і коли USB-накопичувачі використовуються у вашій організації, а також їх потенційної можливості бути каналом для кібератак. Реактивні стратегії охоплюють інший кінець спектру та включають методи відновлення даних. Застосуйте обидві стратегії, щоб забезпечити безпечне використання USB-накопичувачів у вашій організації.

6 найкращих методів безпеки для USB-накопичувачів

Застосуйте політику використання USB-накопичувача для всієї організації

Розробіть і запровадьте детальну політику використання USB-накопичувачів, яка описує належне використання, обмеження та вказівки щодо реагування на інциденти.

Проведення аудитів управління активами

Керуйте інвентаризацією флеш-накопичувачів USB, які використовуються у вашій організації. Періодично перевіряйте інвентаризацію, щоб переконатися, що користувачі дотримуються політики використання USB-накопичувача вашої організації.

Використовуйте шифрування для захисту конфіденційних даних

Шифруйте конфіденційну інформацію, що зберігається на USB-накопичувачах. У разі несанкціонованого використання або втрати цих дисків зашифровані файли будуть марними для зловмисників.

Слідкуйте за діями копіювання та передачі файлів

Виявляйте та блокуйте неавторизовану передачу даних і вимагайте автентифікації, коли користувачі копіюють або передають важливі файли на USB.

Резервне копіювання даних із виведених з експлуатації USB-накопичувачів

Якщо USB-накопичувач виведено з експлуатації, створіть резервні копії необхідних файлів і папок, щоб захистити їх розташування, і зітріть усі дані з диска, щоб запобігти витоку даних.

Розгорніть повний антивірусний захист

Періодично перевіряйте кінцеві точки та флеш-накопичувачі USB кожного разу, коли вони використовуються, щоб уникнути зараження зловмисним програмним забезпеченням, яке виникло за межами вашої організації.

USB-накопичувач і безпека - Як захистити вміст USB-накопичувача

USB-накопичувач (USB-накопичувач) дуже корисний, коли дані потрібно транспортувати з одного місця в інше. USB-накопичувач легкий і невеликий за розміром, його можна, наприклад, зберігати в кишені або гаманці. Ви також можете зберігати великі обсяги даних на флеш-накопичувачах USB, на деяких флешках до 64 ГБ, тому, якщо вам потрібно перенести багато даних, використання USB-накопичувачів може бути дуже зручним.

Ризики безпеці

USB-флеш-накопичувачі корисні, але під час носіння USB-флеш-пам'яті слід враховувати певні ризики для безпеки – його можна загубити або вкрасти. Це справді викликає серйозне занепокоєння, якщо накопичувач містить конфіденційну інформацію, наприклад фінансову інформацію, бізнес-плани, вихідний код програмного забезпечення, дані про співробітників, технічні креслення тощо. Щоб запобігти потраплянню інформації в чужі руки, існують флеш-накопичувачі USB які можуть захистити дані, що зберігаються на диску. Дані зберігатимуться в зашифрованому вигляді, і ніхто не зможе отримати до них доступ без правильного пароля, пін-коду, відбитка пальця чи іншої інформації для автентифікації.

Приклади флеш-накопичувачів USB (USB-накопичувачів), які можуть захищати дані

Нижче ми наведемо кілька прикладів флеш-накопичувачів USB, безпека яких була в центрі уваги під час їх створення. Вони використовують всю апаратну систему для захисту вмісту накопичувача.

Sandisk Cruzer Professional – USB-накопичувач Sandisk Cruzer Professional використовує апаратну систему шифрування для шифрування даних, а конфіденційна інформація зберігається в спеціальному захищеному паролем розділі на USB-накопичувачі. Менш конфіденційну інформацію можна зберігати у загальнодоступному місці для легкого доступу та спільного використання. Для захисту даних використовується надійне 256-бітне шифрування AES. На USB-накопичувачі можна зберігати до 8 ГБ.

Corsair Flash Padlock 2 – флеш-накопичувач USB Corsair Flash Padlock 2 використовує вбудоване 256-бітне апаратне шифрування AES для захисту даних, а для доступу до даних необхідно ввести PIN-код із 4–10 цифр (безпосередньо на USB-накопичувачі).. Міцна гумова кришка захищає USB-накопичувач від випадкового фізичного пошкодження. На USB-накопичувачі можна зберігати до 16 ГБ.

Kingston DataTraveler 5000 – флеш-накопичувач USB Kingston DataTraveler 5000 – це флеш-накопичувач USB із сертифікатом FIPS 140-2 рівня 2, який використовує апаратне 256-бітне шифрування AES (у режимі XTS) для захисту даних, що забезпечує дуже високий рівень безпеки. Функції шифрування в DataTraveler були розроблені компанією Spyrus, яка також виготовляє безпечні флеш-накопичувачі USB. Одним із замовників Spyrus є армія США, яка має дуже високі вимоги до безпеки. Ви можете прочитати більше про співпрацю Kingston і Spyrus тут. На цьому USB-накопичувачі можна зберігати до 16 ГБ.

IronKey Enterprise S200 – флеш-накопичувач USB IronKey Enterprise S200 – це флеш-накопичувач USB із сертифікатом FIPS 140-2 рівня 3, який забезпечить дуже високий рівень безпеки. IronKey Enterprise захищає дані за допомогою надійного 256-бітного апаратного шифрування AES, а хмарна система під назвою IronKey Enterprise Management Service дає адміністраторам повний контроль над розгорнутими флеш-накопичувачами USB через Інтернет. Адміністратор може віддалено вимкнути пристрої та стерти дані, якщо це необхідно. IronKey Enterprise також має вбудований активний захист від шкідливих програм. На цьому USB-накопичувачі можна зберігати до 16 ГБ. На ринку доступні інші флеш-накопичувачі USB, які можуть захистити дані. Наприклад, Kingston також має USB-накопичувач (Kingston DataTraveler 6000), який має сертифікат FIPS 140-2 рівня 3. Існують також USB-накопичувачі з меншим рівнем безпеки, які замість цього використовують програмне забезпечення для захисту даних, наприклад SanDisk Cruzer Switch.

Стандарт FIPS 140-2

FIPS 140-2, згаданий у тексті вище, є стандартом комп’ютерної безпеки, який використовується для акредитації криптографічних модулів. Стандарт FIPS 140-2 був створений NIST (Національний інститут стандартизації технологій) і визначає 4 різні рівні безпеки:

– Рівень безпеки 1. Це найнижчий рівень безпеки. Необхідно використовувати принаймні один схвалений алгоритм або схвалену функцію безпеки, але не вимагається жодного фізичного механізму безпеки, окрім основної вимоги для компонентів виробничого класу.

– Рівень безпеки 2. Це другий найнижчий рівень, і він вимагає, щоб було неможливо відкрити або втручатися в фізичний пристрій, не залишивши слідів.

– Рівень безпеки 3. Рівень безпеки 3 вимагає, щоб пристрій виявляв, коли хтось намагається його відкрити, і намагався захистити інформацію різними способами.

– Рівень безпеки 4. Це найвищий рівень безпеки, який вимагає, щоб уся конфіденційна інформація (наприклад, криптографічні ключі та дані автентифікації) була негайно знищена, якщо зловмисник намагається відкрити пристрій або намагається отримати до нього доступ іншим способом.

Захист програмного забезпечення та шифрування

Існують системи безпеки програмного забезпечення, які можуть захищати інформацію незалежно від апаратного забезпечення чи від конкретного виробника USB-накопичувача. Програмні рішення в деяких випадках менш безпечні, ніж апаратні рішення, але в основному рівень безпеки, який вони забезпечують, є достатнім для загального використання. Однією з найбільших переваг використання програмного рішення є те, що воно набагато дешевше. Захищені апаратні рішення часто коштують досить дорого, тому, якщо вам потрібно придбати велику кількість USB-накопичувачів, ви виберете USB-накопичувачі з апаратним захистом, що коштуватиме багато.

Одним із прикладів програмного рішення є інструмент SamLogic CD-Menu Creator, який, незважаючи на свою назву, також можна використовувати з флеш-накопичувачами USB і для захисту даних на диску. Інструмент має вбудовані функції для шифрування та обробки паролів, і ці функції можна використовувати для захисту документів, зображень, малюнків, відео тощо. Функції безпеки в CD-Menu Creator можуть запобігти несанкціонованому доступу до файлів, якщо, наприклад, USB палицю втрачено або вкрадено. Усі конфіденційні файли зберігаються на флеш-накопичувачі USB у зашифрованому вигляді. BitLocker To Go у Windows 10/11 також може захистити флеш-накопичувач USB BitLocker To Go – це нова функція в Windows 10/11, яку можна використовувати для шифрування даних на флеш-накопичувачі USB. Коли ви підключаєте USB-накопичувач до комп'ютера з інстальованою Windows 10/11, вам буде запропоновано ввести пароль, і ви повинні ввести правильний пароль, щоб розблокувати диск і отримати доступ до вмісту. Також можна отримати доступ до вмісту з Windows Vista та Windows XP, якщо запустити спеціальну програму під назвою BitLocker To Go Reader, яка поширюється разом із флеш-накопичувачем (вона автоматично встановлюється на диск Windows 10/11). Але одна відмінність порівняно з Windows 10/11 полягає в тому, що ви можете лише переглядати файли та копіювати їх, але ви не можете записати назад будь-який вміст. USB-накопичувач буде лише для читання.

Розробка структурної схеми

Структурна схема наведена на рисунку 1. З неї ми бачимо, що розроблена система складається з наступних структурних блоків.

1. Дані, які записуються на флешку.
2. Блок шифрування за допомогою алгоритму AES.
3. Блок розшифрування за допомогою алгоритму AES.
4. Флешка на яку записані зашифровані дані.

Основним блок системи є блок шифрування AES. Розглянемо його більш детальніше.

Алгоритм шифрування AES працює наступним чином:

1. Дані для шифрування input, розбивається на блоки та копіюються до установочного масиву State, згідно визначеного правилу.

2. Формується сеансовий ключ Round Key з ключа шифрування Cipher Key, за допомогою функції KeyExpansion().

3. Визначається число раундів в залежності від довжини ключа 10, 12, або 14 разів.

4. Виконання операції шифрування, тобто виконання раундів шифрування визначену в пункті 3 кількість раз:

– застосування SubBytes();
– застосування ShiftRows();
– застосування MixColumns();
– застосування AddRoundKey().

5. Формування блоку зашифрованих даних, для цього після завершення останнього раунду трансформації, State копіюється в output за визначеним правилом.

Алгоритм розшифрування AES працює наступним чином:

1. Дані для розшифрування input, розбивається на блоки та копіюються до установочного масиву State, згідно визначеного правилу.

2. Формується сеансовий ключ Round Key з ключа шифрування Cipher Key, за допомогою функції KeyExpansion().

3. Визначається число раундів в залежності від довжини ключа 10, 12, або 14 разів.

4. Виконання операції розшифрування, тобто виконання раундів шифрування визначену в пункті 3 кількість раз:

– застосування InvShiftRows(), яке призначене для трансформації при розшифруванні яка є зворотною стосовно ShiftRows();

– застосування InvSubBytes();

– застосування InvAddRoundKey().

– застосування InvMixColumns().

5. Формування блоку роз зашифрованих даних, для цього після завершення останнього раунду трансформації, State копіюється в output за визначеним правилом.

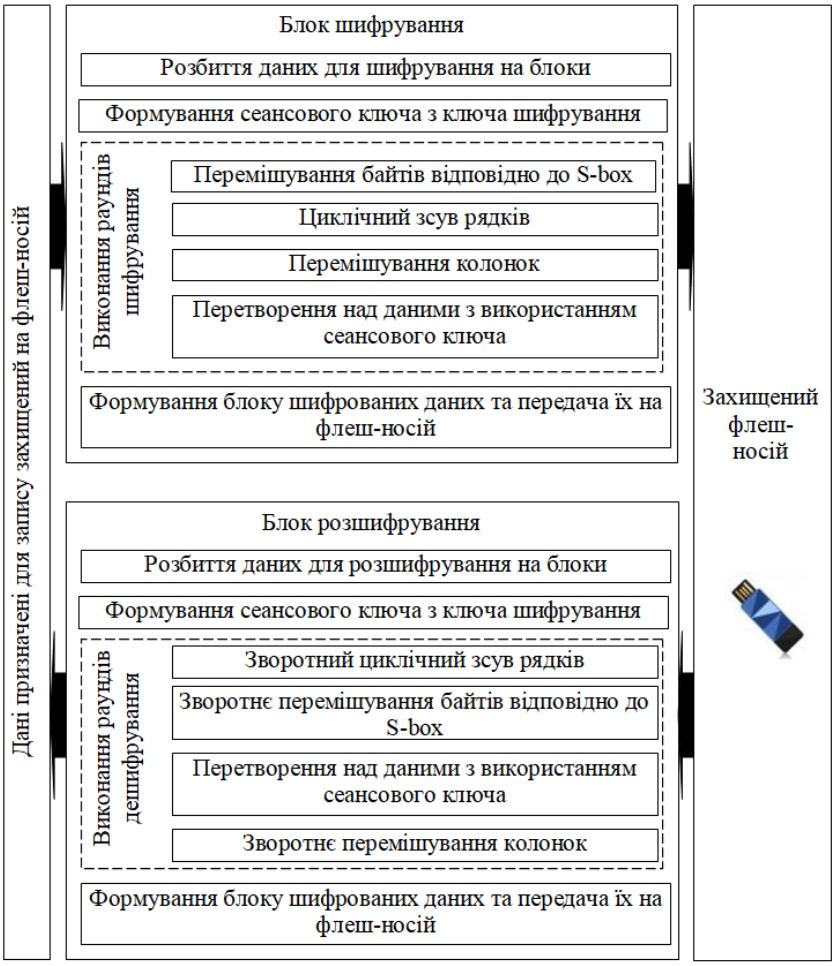


Рисунок 1 – Структурна схема системи

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів Flash Drive. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем Flash Drive. Досліджена система Flash Drive. На основі отриманих результатів досліджень створена програмна реалізація системи Flash Drive. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання Flash Drive. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34. (Scopus).
2. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477. (Scopus).
3. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w (Scopus).
4. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
5. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).
6. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
7. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
8. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
9. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
10. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
11. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
12. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
13. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
14. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
15. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
16. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
17. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
18. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
19. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
20. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).

УДК 004

Р.Соловйов, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПОДІЛУ КЛЮЧІВ В МЕРЕЖІ CISCO SD-WAN, ЩО БАЗУЄТЬСЯ НА ХМАРНІЙ АРХІТЕКТУРІ

У статті розроблено програмне забезпечення, яке призначено для системи розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі. Метою розробки є дослідження та програмна реалізація системи розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі. Об’єктом дослідження є процес розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі. Предметом дослідження є методи розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі. Методи дослідження базуються на методах захисту інформації та хмарних обчислень, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп’ютерні науки, розподіл ключів, Cisco SD-WAN, хмарна архітектура

Постановка проблеми. Однією із ключових задач удосконалювання інформаційних комунікацій є задача побудови безпечної комп'ютерної мережі. Інтерес до неї обумовлюється зростаючими об'ємами переданими між учасниками інформаційного обміну конфіденційної інформації, і швидким ростом таких показників інформації, як вартість втрати конфіденційності, вартість схованого порушення цілісності, вартість втрати інформації. Цій проблемі присвячена велика кількість наукових робіт і монографій.

Захищеність комунікацій у безпечній мережі включає забезпечення конфіденційності й цілісності переданої інформації. Ці властивості забезпечуються використовуваними криптографічними системами, успішне функціонування яких припускає використання на приймальні й передавальній сторонах захищеного каналу криптографічних ключів, бінарних наборів достатньої довжини.

До 1976 року використовувалися лише симетричні криптосистеми, у яких ключі передавальної й приймаючої сторони повинні бути секретними й практично однаковими, що пов'язане із проблемою доставки ключа від одного учасника іншому або від довіреного центра обом учасникам. З винаходом У. Діффі, М. Хеллманом публічної криптографії один із ключів може бути відкритим і доставлятися його стороні, що використовує, не по закритому, а по автентичному каналі, що вимагає реєстрації цього ключа в центрах сертифікації інфраструктури розподілу відкритих ключів.

Помітимо, що використання криптосистем з відкритим ключем саме по собі не досить для забезпечення захищеності комунікацій у комп'ютерних мережах, оскільки алгоритми криптографії з відкритим ключем через необхідність виконання алгебраїчних операцій в алгебраїчних структурах високих порядків на три порядки повільніше алгоритмів симетричних криптосистем.

Тому криптосистеми з відкритим ключем можуть використовуватися для захисту лише невеликих обсягів інформації й в основному відіграють допоміжну роль, забезпечуючи захист передачі секретних ключів для симетричних криптосистем.

Безпосереднє використання цього способу доставки секретного ключа кожним

учасником комп'ютерної мережі приводить до багаторазового (по числу учасників) виконанню дорогих актів сертифікації й використанню ключів, що генерується учасниками без належного контролю їхньої якості.

Для спрощення процедури формування й доставки секретних ключів у криптографії запропоновані й досліджені різноманітні схеми попереднього розподілу ключів. У них процедура доставки секретного ключа учасникам комп'ютерної мережі виконується у два етапи: кожному учасникові довіреним центром доставляється пакет ключової інформації (у вигляді наборів двійкових слів достатньої довжини), склад якого (можливо, з деякою додатковою відкритою інформацією про ці слова) публікується.

При цьому кожний учасник, знаючи склади пакетів і опубліковані дані, може, використовуючи тільки набори зі свого пакета, обчислити для захищеної комунікації з будь-яким іншим учасником мережі ключ, що не може обчислити ніякий третій учасник.

Розвиваючи цей підхід, криптографи запропонували й схеми більш загального характеру, що дозволяють попередньо розподіляти ключову інформацію для обчислення ключів привілейованих груп учасників, недоступних забороненим групам учасників. Але й цей підхід зустрічає труднощі відносно до великих обчислювальних мереж, оскільки припускає використання єдиного центра генерації й доставки пакетів ключової інформації кожному учасникові мережі. Використання для доставки пакетів криптосистем з відкритим ключем припускає сертифікацію відритих ключів учасниками.

Розглянутий стан проблеми розподілу ключів дозволяє вважати **актуальними** наступні задачі:

– розробка й обґрунтування архітектурних мережевих рішень нецентралізованого попереднього розподілу ключової інформації в комп'ютерній мережі на основі незалежного попереднього її розподілу в сегментах або доменах мережі;

– розробка й обґрунтування способу реалізації попереднього розподілу ключової інформації в сегментах або доменах обчислювальної мережі на основі використання пропонованої модифікації протоколу Kerberos;

– розробка й обґрунтування нових схем попереднього розподілу ключів;

– розробка програмного забезпечення для обчислення пакетів ключової інформації й принципів побудови системного програмного забезпечення для обчислювальних систем з нецентралізованим попереднім розподілом ключів.

Варто підкреслити відповідність цих напрямків дослідження особливостям мобільних обчислювальних мереж, у яких практично важко реалізовані вимоги сертифікації відкритих ключів і актуальна задача обліку умови розширюваності мережі.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі.

– Дослідження системи розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі.

– Програмна реалізація системи розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі.

Об'єктом дослідження є процес розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі.

Предметом дослідження є методи розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі.

Методи дослідження базуються на методах захисту інформації та хмарних обчислень, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Опис методу побудови схем попереднього розподілу ключів

У цьому розділі наведені приклади поліпшення схем попереднього розподілу системних ключів за рахунок аналізу привілейованого й забороненого сімейств.

Визначення: Нехай:

$\sum^1$ – множина всіх можливих KDP($\boldsymbol{P}_1$,$\boldsymbol{F}_1$)-схем.

$\sum^2$ – множина всіх можливих KDP($\boldsymbol{P}_2$,$\boldsymbol{F}_2$)-схем.

$\boldsymbol{S}^1$ – KDP($\boldsymbol{P}_1$,$\boldsymbol{F}_1$)-схема ($\boldsymbol{S}^1 \in \sum^{1)}$.

$\boldsymbol{S}^2$ – KDP($\boldsymbol{P}_2$,$\boldsymbol{F}_2$)-схема ($\boldsymbol{S}^2 \in \sum^{2)}$.

Схеми $\boldsymbol{S}^1$ і $\boldsymbol{S}^2$ є взаємозамінними, якщо $\boldsymbol{S}^1 \in \sum^2$ і $\boldsymbol{S}^2 \in \sum^{1\cdot}$

Для взаємозамінних схем по визначенню KDP-схеми виконано:

$$\forall P \in \boldsymbol{P}_1, F \in \boldsymbol{F}_1, P \cap F = \varnothing: \ \bigcap_{i \in P} S_i^2 \not\subset \bigcup_{j \in F} S_j^2, \tag{1}$$

$$\forall P \in \boldsymbol{P}_2, F \in \boldsymbol{F}_2, P \cap F = \varnothing: \ \bigcap_{i \in P} S_i^1 \not\subset \bigcup_{j \in F} S_j^1 \cdot \tag{2}$$

Помітимо, що відношення взаємозамінності є симетричні і рефлексивні, але не є транзитивні, тобто з того, що схеми $\boldsymbol{S}^1$ і $\boldsymbol{S}^2$ взаємозамінні, $\boldsymbol{S}^2$ і $\boldsymbol{S}^3$ взаємозамінні, не слідує, що $\boldsymbol{S}^1$ і $\boldsymbol{S}^3$ взаємозамінні. Відношення взаємозамінності є відношенням толерантності.

Твердження 1. Якщо $\boldsymbol{F}_1 \subseteq \boldsymbol{F}$, де $\boldsymbol{F}_1$ це сімейство всіх підмножин множини U потужності w, причому $\forall P \in \boldsymbol{P}: |P| + w < n$ $\left(\max_{P \in \boldsymbol{P}} |P| + w < n\right)$, то KDP($\boldsymbol{P}$,$\boldsymbol{F}$) і KDP($\boldsymbol{P}$,$\boldsymbol{F}'$)-схеми є взаємозамінними, де $\boldsymbol{F}'$ – це об'єднання сімейства $\boldsymbol{F}$ і всіх підмножин множини U потужність яких не перевершує w.

Твердження 2. Якщо існує сімейство $\boldsymbol{P}_1 \subseteq \boldsymbol{P}$, таке що $\boldsymbol{P}_1$ це сімейство всіх підмножин множини U потужності g, причому $\forall F \in \boldsymbol{F}: |F| + g < n$, то KDP($\boldsymbol{P}$,$\boldsymbol{F}$) і KDP($\boldsymbol{P}'$,$\boldsymbol{F}$)-схеми є взаємозамінними, де $\boldsymbol{P}'$ – це об'єднання сімейства $\boldsymbol{P}$ і всіх підмножин множини U, потужність яких не перевершує g.

Твердження 3. Якщо $g+w<n$, то схеми KDP(g,w) і KDP($\leq g$, $\leq w$) є взаємозамінними.

Для імовірнісного методу запропонована більше низька оцінка кількості системних ключів, необхідних для побудови схеми.

На основі вивчених методів запропонований метод побудови загального випадку схем розподілу системних ключів.

Суть імовірнісних методів полягає в тому, що формується деяким випадковим образом таблиця. Далі перевіряється, чи є сформована таблиця KDP-схемою. Якщо немає – то таблиця формується заново.

Розглянемо імовірнісний метод побудови KDP($\boldsymbol{P}$,$\boldsymbol{F}$)-схем.

Нехай елементи множини Ψ пронумеровані: $\{\psi_1, \psi_2, \ldots, \psi_k\}$.

Позначимо $X_{is} = \begin{cases} 1, & якщо\ \psi_s \in S_i \\ 0, & якщо\ \psi_s \notin S_i \end{cases}$, s=1, …, k; i=1, …, n...

Таблиця X заповнюється в такий спосіб:

$$X_{is} = \begin{cases} 1, & з\ ймовірністю\ \ p \\ 0, & з\ ймовірністю\ 1-p \end{cases}. \tag{3}$$

Верхня оцінка ймовірності того, що KDP-схема не буде побудована імовірнісним методом:

$$E = E(k, p) = \sum_{P \in \boldsymbol{P}} \sum_{\substack{F \in \boldsymbol{F} \\ P \cap F = \varnothing}} \prod_{s=1}^{k} \left(1 - \prod_{i \in P} p \prod_{j \in F} (1-p)\right) = \sum_{P \in \boldsymbol{P}} \sum_{\substack{F \in \boldsymbol{F} \\ P \cap F = \varnothing}} \left(1 - p^{|P|}(1-p)^{|F|}\right)^k \tag{4}$$

Розглянуто імовірнісний метод, що дозволяє побудувати KDP$(\boldsymbol{P}^g, \boldsymbol{F}^w)$ -схеми, де:

$$\boldsymbol{P}^g = \{P : P \in \boldsymbol{P}, |P| = g\},\ \boldsymbol{F}^w = \{F : F \in \boldsymbol{F}, |F| = w\}. \tag{5}$$

Мінімізуючи $E(k, p)$ по p, маємо:

$$p_0 = \frac{g}{w+g}. \tag{6}$$

Кількість ключів $k_0^{g,w}$, необхідних для побудови такої схеми обчислюється по формулі:

$$k_0^{g,w} = \left\lfloor \frac{(g+w)^{g+w}}{g^g w^w} \ln\left(\frac{\sum\limits_{P \in \boldsymbol{P}^g} \sum\limits_{\substack{F \in \boldsymbol{F}^w \\ P \cap F = \varnothing}} 1}{1 - \mathrm{E}} \right) \right\rfloor + 1. \tag{7}$$

Уведено поняття об'єднання KDP-схем і запропоновані способи зменшення об'єму ключового матеріалу за рахунок комбінування різних методів побудови схем KDP (або схем, одержуваних одним методом, але з різними параметрами).

Визначення. Нехай:

$\{S_1^1,\ldots,S_n^1\}$ – KDP($\boldsymbol{P}_1$,$\boldsymbol{F}_1$)-схема.

$\{S_1^2,\ldots,S_n^2\}$ – KDP($\boldsymbol{P}_2$,$\boldsymbol{F}_2$)-схема.

При цьому $\Psi_1 \bigcap \Psi_2 = \varnothing$.

Тоді об'єднанням KDP-схем KDP($\boldsymbol{P}_1$,$\boldsymbol{F}_1$)$\bigcup$ KDP($\boldsymbol{P}_2$,$\boldsymbol{F}_2$) є сімейство $\{S_1,\ldots,S_n\}$: $S_i = S_i^1 \bigcup S_i^2$.

Позначимо $G = \{g : \boldsymbol{P}^g \neq \varnothing\}$, $W = \{w : \boldsymbol{F}^w \neq \varnothing\}$.

Тоді:

$$\boldsymbol{P} = \bigcup_{g \in G} \boldsymbol{P}^g,\ \boldsymbol{F} = \bigcup_{w \in W} \boldsymbol{F}^w \text{ і } \mathrm{KDP}(\boldsymbol{P}, \boldsymbol{F}) = \bigcup_{w \in W, g \in G} \mathrm{KDP}(\boldsymbol{P}^g, \boldsymbol{F}^w). \tag{8}$$

При цьому можна використовувати різні методи побудови KDP$(\boldsymbol{P}^g, \boldsymbol{F}^w)$ -схем.

Розглянемо комбінування імовірнісних і тривіальних методів побудови схем попереднього розподілу системних ключів.

Якщо виконуються умови:

$$H_1(g,w) = \{ \sum_{g \in G} \min\{k_0^{g,w}, |\boldsymbol{P}^g|\} \geq |\boldsymbol{F}^w| \}, \tag{9}$$

або

$$H_2(g,w) = \{ \sum_{w \in W} \min\{k_0^{g,w}, |\boldsymbol{F}^w|\} \geq |\boldsymbol{P}^g| \}, \tag{10}$$

то для побудови KDP$(\boldsymbol{P}^g, \boldsymbol{F}^w)$ -схеми раціонально використовувати тривіальні методи, інакше – імовірнісні.

Таким чином, KDP($\boldsymbol{P}$,$\boldsymbol{F}$)-схема:

$$\bigcup_{\substack{w \in W, g \in G, \\ \neg H_1(g,w), \\ \neg H_2(g,w)}} \mathrm{KDP}(\boldsymbol{P}^g, \boldsymbol{F}^w) \bigcup \bigcup_{\substack{w \in W, \\ H_1(w,g)}} \mathrm{KDP}(\cdot, \boldsymbol{F}^w) \bigcup \bigcup_{\substack{g \in G, \\ H_2(g,w)}} \mathrm{KDP}(\boldsymbol{P}^g, \cdot) \tag{11}$$

Кількість системних ключів |Ψ|, необхідних для побудови схеми:

$$k_0 = \sum_{\substack{w \in W, g \in G, \\ \neg H_1(g,w), \\ \neg H_2(g,w)}} k_0^{g,w} + \sum_{\substack{w \in W, \\ H_1(w,g)}} |\boldsymbol{F}^w| + \sum_{\substack{g \in G, \\ H_2(g,w)}} |\boldsymbol{P}^g| \tag{12}$$

Імовірнісний метод побудови схеми попереднього розподілу ключів з геш-функцією.

Визначення. HAKDP(***P***,***F***,*L*)-схемою, де ***P*** і ***F*** – це сімейства підмножин множини *U* = {1, …, *n*}, називається всяка пара сімейств (***S***,***D***), ***S***={S_1,…,S_n} підмножин кінцевої множини $\Psi \subseteq \Omega$ (|Ψ|=*k*) і ***D***={D_1,…,D_n} підмножин множини {1,…,*L*}, причому |D_t| = |S_t| для *t*=1,…,*n*, задовольняючій умові: $\forall P \in \boldsymbol{P}, F \in \boldsymbol{F}, P \cap F = \varnothing : \bigcap_{i \in P} S_i \not\subset \bigcup_{j \in F} S_j$ або не виконана умова $D_{F,P} \le D_P$, де D_P – набір значень $\max_t(D_i(t))$ відповідним співпадаюніж елементам множин S_i з *P* і $D_{F,P}$ є набір значень $\min_t(D_i(t))$ всіх тих елементів з *F*, які відповідають співпадаюніж елементам множин S_i з *F*.

Зауваження. HAKDP(***P***,***F***,0)-схема є KDP(***P***,***F***)-схемою.

Приклад HAKDP(2,1,1)-схеми для *n*=4 абонентів і *k*=3 ключів:

S_1={1,2,3}, D_1=(1,1,1),

S_2={1,2}, D_2=(0,0),

S_3={1,3}, D_3=(0,0),

S_4={2,3}, D_4=(0,0)

Нехай, як і у випадку KDP-схем, таблиця *X* отримана деяким випадковим способом Pr{ X_{is} =1}=p_0. Таблиця ***D*** також отримана випадковим способом Pr{D_t(*s*)=*a*}=$\frac{1}{L}$, $a \in \{1,...,L\}$. При цьому p_0 обчислюється по формулі:

$$p_0 = \frac{2}{3(1+P_L)}, \tag{13}$$

кількість ключів, необхідних для побудови HAKDP(2,1,*L*)-схеми:

$$k = -\frac{\log(n(n-1)(n-2))}{\log(1-p^2(1-p)-p^3P_L)}, \tag{14}$$

де P_L – імовірність того, що $D_{rs} < D_i(s)$ і $D_r(s) < D_j(s)$.

Дану ймовірність можна порахувати по формулі:

$$P_L = \sum_{m=0}^{L} \frac{1}{L+1}\left(\frac{L-i}{L+1}\right)^2 = \frac{2L^2+L}{6L^2+12L+6}. \tag{15}$$

Переваги використання KDP і HAKDP-схем випливають із даних, представлених у таблиці 1 на прикладі KDP(10^4, 173) і HAKDP(10^4, 114, 10)-схем.

Таблиця 1 – Порівняння централізованої KDP і HAKDP-схем

	Без попереднього розподілу	KDP(10^4,173)-схема	HAKDP(10^4,114,10)-схема
Середня довжина пакета	9 999	115	71
Число переданих ключів	99 990 000	1 150 000	710 000

Пропонується три способи організації безпечної мережі на основі протоколу Kerberos і схем попереднього розподілу ключів.

Перший спосіб припускає використання стандартного протоколу Kerberos, у якому як сервер додатків виступає абонент мережі, якому адресується повідомлення.

Другий спосіб припускає використання стандартного протоколу Kerberos для попереднього розподілу ключового матеріалу в мережі, відповідно до KDP-схеми. При цьому KDP-схема й ключовий матеріал формується на стороні сервера додатків.

Третій спосіб припускає використання модернізованого протоколу Kerberos для

розподілу ключового матеріалу. При цьому KDP-схема й ключовий матеріал формуються на стороні сервера TGS.

Розглянемо нецентралізований розподіл ключової інформації. У складних мережах, що складаються з декількох доменів, використання централізованого розподілу ключової інформації може бути важко.

Опишемо технологію захищених комунікацій у комп'ютерній мережі, з використанням ключової інформації, попередньо розподіленої в її сегментах. Хоча б один з елементів у кожному сегменті відіграє роль сервера додатків. Інші елементи сегмента представляють мережевих клієнтів. Мережеві сервери додатка є довіреними вузлами для всіх елементів комп'ютерної мережі. Вони зашифровують і розшифровують конфіденційну інформацію користувачів, коли відповідний шифротекст одного сегмента передається користувачеві з іншого сегмента. При цьому використовуються ключі, що обчислюються по пачках, розподіленим елементам одного сегмента, для розшифрування й іншого для зашифрування.

Захищені комунікації в мережі здійснюються з використанням ключів, що обчислюються на підставі часток ключової інформації, виділюваних окремим елементам або різним парам елементів того самого домена (рисунок 1).

Дана технологія дозволяє зменшити час обчислення схем попереднього розподілу ключів, а також об'єм ключової інформації, переданої по секретних каналах від генератора до елементів комп'ютерної мережі.

Передбачаються наступні особливості мережі:

1) комп'ютерна мережа складається з N сегментів (або доменів) $D_1,\ldots, D_N$;

2) хоча б один учасник кожного домена виконує роль мережевого сервера додатків NAS (Network Application Server), інші елементи є клієнтами мережі.

3) кожний NAS є елементом двох сусідніх сегментів, він одержує свої частки ключової інформації, розподіленої в обох сегментах;

4) граф, вершини якого відповідають сегментам, а ребра – мережевим серверам додатків NAS зв'язний;

5) кожний учасник може передавати зашифровану інформацію будь-якому іншому учасникові по відкритих каналах;

6) мережеві сервери додатків NAS є довіреними вузлами для всіх елементів комп'ютерної мережі;

7) кожний мережевий сегмент (або домен) D_j, j=1,…,N, прикріплений до певного довіреного центра TA_j;

8) кожний довірений центр TA_j, j=1,…,N, виробляє схему попереднього розподілу ключів для прикріпленого сегмента й розподіляє між його елементами пакети обчисленої ключової інформації;

9) генерація схем розподілу ключів і доставка ключової інформації для елементів різних сегментів здійснюється незалежно й, можливо, одночасно.

10) конфіденційна комунікація між учасниками комп'ютерної мережі організується з використанням зашифрування-розшифрування й забезпечення цілісності тих самих або різних пакетів ключової інформації, що належать одному або декільком учасникам того самого сегмента.

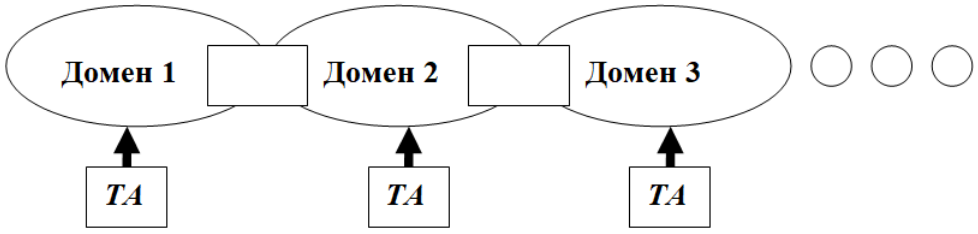


Рисунок 1 – Нецентралізований попередній розподіл ключів

Розглянемо окремий випадок мережі, у якій кожний абонент мережі може передати інформацію іншому абонентові мережі.

Розподіл ключової інформації здійснюється незалежно у всіх доменах мережі. Для цього в кожному домені будується KDP(2,1)-схема. Такі схеми можуть мати однакову або різні структури, але в кожній з них елементи ключової інформації, що розподіляється, виробляються випадково й незалежно.

Нехай по завершенню етапу попереднього розподілу ключів (централізованому для кожного домена образом, але незалежним у кожному домені) всі учасники комп'ютерної мережі мають пакети ключової інформації. Це дозволяє їм обчислити ключі для конфіденційного обміну в межах одного домена.

Для конфіденційної передачі інформації від абонента А абонентові В того самого домена, абонент А шифрує інформацію ключем k_{AB}, що будується на основі ключової інформації абонента А і KDP-схеми даного домена.

Для конфіденційної передачі інформації від абонента А абонентові В сусіднього (що має з доменом абонента А загальний сервер додатка С) домена, А передає С інформацію, зашифровану ключем k_{AC}. Сервер додатка С зашифровує інформацію за допомогою ключа k_{CB}, потім передає інформацію абонентові В, розшифровану ключем k_{AC}.

Для конфіденційної передачі інформації від абонента А абонентові В домена, що не має з доменом елемента А загального сервера додатків, будується шлях від абонента А до абонента В через сервера додатків C_1, C_2, …, C_l... Далі абонент А передає серверу додатків C_1 інформацію, зашифровану ключем k_{AC_1}. Сервер додатків C_1 зашифровує інформацію за допомогою ключа $k_{C_1C_2}$, потім передає інформацію серверу додатків C_2, розшифровану ключем k_{AC_1}, і так далі. Наприкінці сервер додатків C_l передає абонентові В інформацію, зашифровану ключем k_{C_lB}.

Помітимо, що відкритий текст відправника доступний для атак зловмисника через сервери додатків NAS, вони є довіреними вузлами для всіх учасників інформаційного конфіденційного обміну. Якщо шифратор відповідає властивості перестановочності (як, наприклад, блоковий шифратор у режимі гаммування), то розшифруванню в довірених вузлах зв'язку може передувати вторинне зашифрування, що було вище.

Для скорочення операцій розшифрування й зашифрування у вузлах зв'язку можливий наступний варіант: абонент А передає абонентові В ключ K_{AB}, обчислений на основі його ключового пакета, а потім здійснює передачу інформації, зашифрованої ключем K_{AB}.

Ключ зашифрування є значенням геш-функції, застосованої до ідентифікатора абонента В. Передача ключа K_{AB} і інформації здійснюється описаним вище способом.

У результаті виконання магістерської роботи досліджений також спосіб організації конференц-зв'язку коаліції абонентів з різних доменів в умовах нецентралізованого розподілу ключової інформації.

Модифікація протоколу Kerberos для генерації й розподілу ключової інформації

Генерація й розподіл ключової інформації для безпечної мережі може бути організована з використанням модифікації протоколу Kerberos. Запропонована модифікація протоколу Kerberos для генерації й розподілу ключової інформації в комп'ютерній мережі відрізняється тим, що в серверах TGS обчислюються пакети ключової інформації, що доставляються клієнтам із сервера додатків у складі посилок TGT.

Нехай є один або більше серверів автентифікації (AS) і, принаймні, така ж кількість серверів видачі квитків (TGS).

Кожний сервер TGS відповідає одному серверу AS. Кожний домен мережі зв'язується із сервером TGS і абоненти домена прив’язуються до цього TGS.

Якщо сервер додатків NAS утримується в декількох доменів, то він прив’язується до всіх TGS приналежних даним доменам. Всі абоненти мережі підтримують службу одноразових паролів OTPS (One-Time Password Service) з декількома AS, до яких прив'язані

відповідні TGS.

Сервер додатків NAS підтримує службу одноразових паролів однієї або більше OTPS. Абонент A_i і сервер автентифікації AS_j з OTPS мають одноразовий пароль k_{A_i,AS_j}.

На етапі ініціалізації (рисунок 2) кожний сервер AS_j обчислює ключі k_{A_i,TGS_t} для зв'язку з TGS_t і прив'язаних до нього абонентів A_i, k_{NAS_p,TGS_t} для зв'язку TGS_t і прив'язаних до нього NAS_p.

Припустимо, що в кожного AS_j є ключ k_{AS_j,TGS_t} для зв'язку із прив'язаними до нього TGS_t і кожний TGS будує KDP(P,F) - схему для відповідного домена мережі: для кожного абонента A_i або NAS_p цього домена обчислюються пакети S_i, S_s.

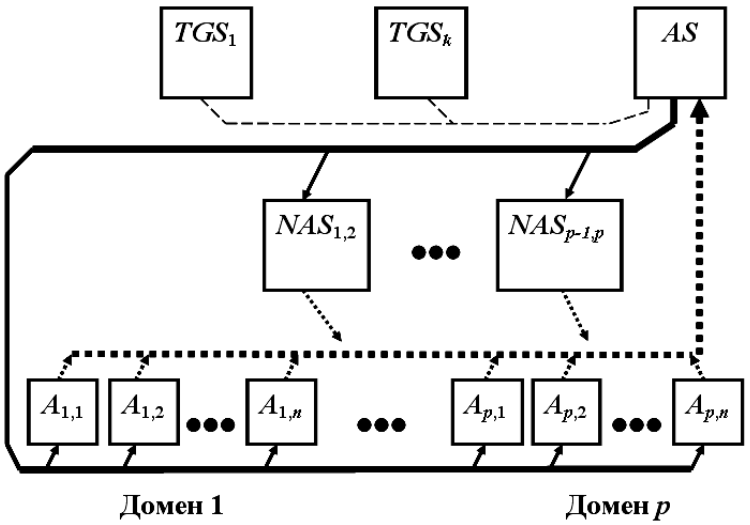


Рисунок 2 – Протокол обміну із сервером автентифікації з метою одержання дозволу на видачу ключової інформації

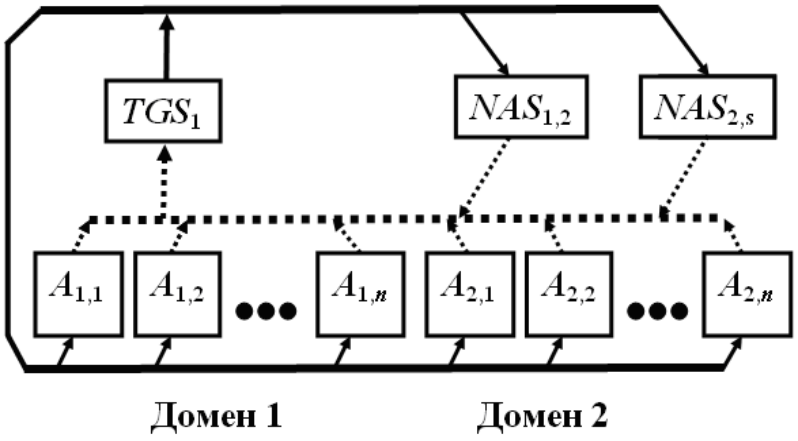


Рисунок 3 – Протокол обміну із сервером TGS з метою одержання ключового матеріалу абонентами мережі А і серверами додатків NAS

Тепер кожний учасник A_i підтримуючу службу одноразового пароля з відповідним сервером автентифікації AS_j, може одержати свій пакет K_i ключової інформації, ініціалізуя й виконуючи протоколи:

a) протокол обміну із сервером автентифікації, з метою одержання дозволу TGT на одержання ключової інформації (рисунок 2);

b) протокол обміну із сервером TGS, з метою одержання ключової інформації (рисунок 3);

Для безпечного обміну інформації між абонентами мережі використовується:

с) протокол комунікації абонентів мережі, у тому числі через сервер додатків NAS (рисунок 4).

Дано оцінки ключової інформації, необхідної для організації захищених комунікацій, для мереж різних типів.

Дані, представлені в таблиці 2 показують переваги використання нецентралізованих 10(KDP(1002, 56) і 10(HAKDP(1002, 56, 10)-схем.

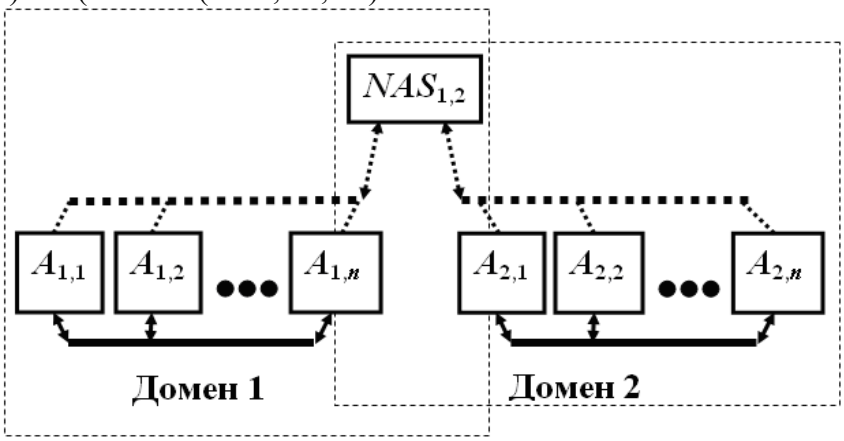


Рисунок 4 – Протокол комунікацій абонентів мережі, у тому числі через сервер додатків NAS

У нецентралізованій мережі в порівнянні із централізованої (таблиця 1), число ключів, що пересилаються від ТА учасникам, скорочується. Застосування схем з гешуванням приводить до ще більшого скорочення середньої довжини пакетів ключової інформації й об'єму ключової інформації, переданої від довіреного центра ТА.

Таблиця 2 – Порівняння нецентралізованої KDP і HAKDP

	Без попереднього розподілу	KDP(10^4,173)-схема	HAKDP(10^4,114,10)-схема
Середня довжина пакета	1 001	37	28
Число переданих ключів	10 030 020	370 740	280 560

Також досліджуються особливості побудови програмних засобів попереднього розподілу ключів. Показано, що при цьому перевага віддається імовірнісним методам двухетапного синтезу схем.

На першому етапі випадково генерується таблиця розподілу ключів, а на другому – перевіряється її відповідність умовам необхідної схеми. Аналітично обґрунтовані параметри керування імовірнісного етапу. Працездатність запропонованих методів підтверджена експериментами з використанням розроблених програмних засобів при різних параметрах вимог до мережі.

Розробка структурної схеми

Структурна схема розробленої системи показана на рисунку 5.

Для реалізації мережі WAN було вибрано технологію приватної мережі на орендованих каналах.

Мережа складається з головного офісу та декількох філіалів. У приміщенні головного офісу знаходяться наступні сервери:

1. Сервер автентифікації.
2. Сервер квитанцій (білетів).
3. Ресурсний сервер.

У приміщеннях філіалів знаходяться:

1. Ресурсні сервери.
2. Робочі станції (хости).

Орендовані територіальні канали прокладаються провайдером транспортних територіальних послуг у його первинній мережі FDM, PDH, SDH або мережі з інтегральними послугами ISDN. При оренді каналу в таких мережах підприємство ділить пропускну здатність магістральних каналів і комутаторів цієї мережі з іншими абонентами даного провайдеру.

На рисунку 5 показаний приклад використання орендованих каналів для побудови корпоративної мережі підприємства із трьома філіями.

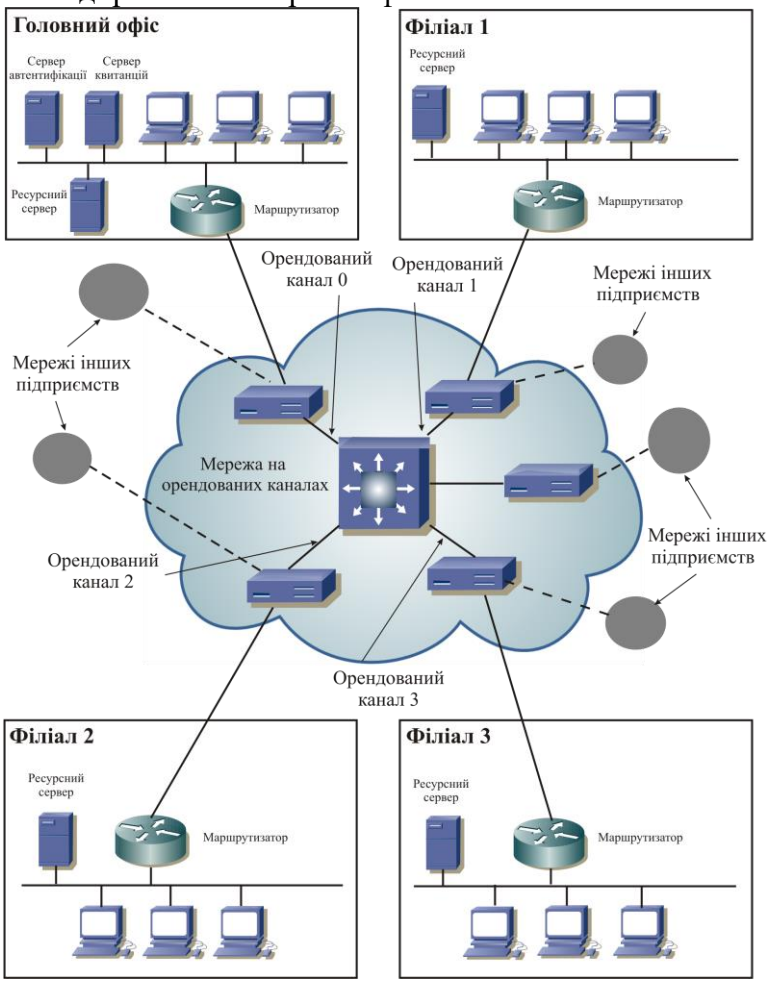


Рисунок 5 – Структурна схема розробленої системи

Канали, що зв'язують центральну мережу підприємства з мережами філій, проходять через мультиплексор, що поєднує канали всіх абонентів у магістральний канал. Незважаючи на те, що територіальні канали в цьому випадку не відносяться до власності підприємства, корпоративні мережі, побудовані на орендованих каналах, також називають часними, принаймні по двох причинах.

По-перше, смуга пропускання орендованого каналу повністю виділяється підприємству, і тому є в деякому змісті його "приватною власністю". Це повною мірою відноситься до орендованих цифрових каналів, які підтримуються провайдером на базі первинної цифрової мережі з технікою мультиплексування TDM. Орендар такого каналу одержує у своє повне розпорядження всю його пропускну здатність – 64 Кбіт/с, 128 Кбіт/с, 2 Мбіт/с, або вище. У застарілих мережах із частотним мультиплексуванням FDM орендар самостійно розпоряджається не пропускною здатністю, а заздалегідь відомою смугою пропущення каналу. У кожному разі, пропускну здатність каналу підприємство-орендар не

ділить ні з ким, і це дуже важливо для створення корпоративної мережі зі стабільними характеристиками.

Наявність гарантованої пропускної здатності дає можливість адміністраторові мережі планувати роботу додатків через глобальні канали зв'язку: розподіляти пропускну здатність каналу між додатками, оцінювати можливі затримки повідомлень, обмежувати обсяг генеруємого територіального трафіку, визначати максимальну кількість активних додатків і т.п.

По-друге, приватний характер мереж, побудованих на орендованих каналах, підтверджується достатньою конфіденційністю даних. Корпоративні дані практично не доступні для абонентів, що не є користувачами корпоративної мережі або співробітниками організації-провайдеру каналів. Дійсно комутацію каналів у первинних мережах може виконати тільки оператор мережі, а рядовому користувачеві така операція недоступна. Це спричиняє більший ступінь захищеності даних, переданих по каналах первинних мереж. Наприклад, тут неможлива типова для Інтернету атака – відгалуження й аналіз "чужого" трафіка іншим користувачем. Таким чином забезпечується прийнятна безпека переданих даних від зовнішніх атак.

Для мережі центрального офісу найкраще підійдуть маршрутизатори Cisco 3620 або Cisco 3640. Конкретна модель маршрутизатора й кількість установлених модулів буде залежати від покладеного на маршрутизатор завдання.

Моделі серії 3600 надають функціонально повне рішення для організації віддаленого доступу. Інакше кажучи, дані пристрої можуть використовуватися як потужний сервер доступу. Будь-яка модель Cisco 3600 може забезпечувати надійний доступ до Вашої WAN численних віддалених і мобільних користувачів. При цьому вони зможуть не тільки працювати з файловим господарством WAN, як зі своїм власним, але також використовувати загальні програмні додатки.

Дана серія маршрутизаторів є відмінними засобом вкладення коштів, при якому Ви зможете легко в майбутньому змінювати конфігурацію Вашої глобальної мережі в міру росту вимог до кількості з'єднань або в міру появи нових технологій для глобальних обчислювальних мереж. Все це забезпечує захист інвестицій у встаткування й, отже, економію грошей. Моделі Cisco 3600 мають достатню масштабованість. Наприклад, модель Cisco 3640 підтримує інтерфейси ISDN PRI (Primary Rate Interface) або ISDN BRI (Basic Rate Interface) у тому самому шасі: максимальне число підтримуваних PRI-з'єднань – шість, BRI-зєднань – 24, а модель Cisco 3620 може бути сконфігурована з одним портом Ethernet і одним портом ISDN PRI, або одним портом Ethernet і чотирма портами ISDN BRI. В усі моделі сімейства Cisco 3600 інтегрована міжмережева операційна система Cisco IOS, що підтримує встановлення з'єднань на вимогу, що забезпечує об'єднання локальних мереж, безпека доступу й даних і оптимізацію з'єднань із глобальними обчислювальними мережами. Завдяки підтримці повного набору функціональних можливостей Cisco IOS, маршрутизатори серії Cisco 3600 дають надійні й гнучкі засоби роботи з Інтернетом/Інтранетом мультимедійними додатками. Cisco IOS робить легкою не тільки роботу через Інтернет, але спрощує й підвищує ефективність роботи у корпоративній мережі Інтернет. Маршрутизатори сімейства Cisco 3600 надають відмінну можливість вибору конфігурації. Так, наприклад, модель Cisco 3640 має чотири слота під мережеві модулі, а модель Cisco 3620 – два слоти. У кожний слот по вибору можуть бути встановлені мережеві модулі для Інтернет і Token Ring, а також можна вибирати із цілого набору інтерфейсних карт для з'єднання із глобальними обчислювальними мережами. Кожне рознімання для мережевих модулів може прийняти інтерфесні карти мережевих модулів різного типу, включаючи ISDN PRI, ISDN BRI, і асинхронні/синхронні послідовні інтерфейси.

Інакше кажучи, по наявній лінії зв'язку можна легко об'єднати за допомогою Cisco 3640 WAN із чотирма іншими локальними мережами, розташованими в інших кінцях міста, і працювати в них, як у власній мережі й навіть використовувати їхнє устаткування (наприклад, високоякісний принтер).

Сімейство Cisco 3600 надає користувачеві можливість ефективного й недорогого використання додатків Інтернету/Інтранету і можливість масштабування подібних рішень при збільшенні потреб або зміні структури віддаленого доступу.

Пропонуючи повну інтеграцію мережевого маршрутизатора й сервера для ISDN, асинхронних і синхронних з'єднань WAN в одному продукті, Cisco 3600 дає користувачеві нову платформу для майбутніх додатків, що є негайним рішенням сьогоднішніх проблемам.

Для віддалених офісів (філій) найкраще підійдуть маршрутизатори серій Cisco 2500 або Cisco 2600. Конкретна модель маршрутизатора й кількість встановлених модулів буде залежати від покладеного на маршрутизатор завдання.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі. Досліджена система розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі. На основі отриманих результатів досліджень створена програмна реалізація системи розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання розподілу ключів в мережі Cisco SD-WAN, що базується на хмарній архітектурі. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
2. Смірнов О.А., Дрєєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.
3. Смирнов А.А., Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений. Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.
4. Смирнов А.А., Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения. Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.
5. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 3(69). С. 93-98. 2022. (Фахове видання. Категорія «Б»)
6. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.
7. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.
8. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
9. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
10. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
11. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).

12. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
13. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
14. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
15. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
16. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
17. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
18. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
19. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).
20. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019. (Scopus).

УДК 811.161.1'276:33

Н. Глевацька, магістр гр. ПР-71м
Центральноукраїнський національний технічний університет

ДЕЯКІ ОСОБЛИВОСТІ ПЕРЕКЛАДУ УКРАЇНСЬКОЮ МОВОЮ МЕТАФОРИ НА МАТЕРІАЛАХ НАУКОВО-ЕКОНОМІЧНОГО СТИЛЮ

Стаття досліджує актуальну проблему перекладу метафор у науково-економічних текстах. Для досягнення цих цілей використовуються методи зіставного аналізу, метод словникових дефініцій, описовий метод та метод міжмовної перекладної особливості. Автор статті проводить аналіз специфіки економічного стилю, вивчає термінологію та виявляє особливості перекладу метафоричних висловів та концептів. Об'єкт дослідження є способи перекладу економічних термінів, а предметом дослідження є безпосередньо економічні тексти, які підлягають перекладу. Узагальнюючи, стаття пропонує важливий внесок у вивчення проблем перекладу метафор у науково-економічних текстах.

переклад метафор; українська мова; науково-економічний стиль; особливості перекладу; метафоричні вислови.

Постановка проблеми. Дослідження проблеми перекладу метафор на матеріалах науково-економічного стилю має велику актуальність в сучасному світі. Зростаюча глобалізація та міжнародна співпраця у галузі економіки створюють потребу в ефективному спілкуванні між різними мовами та культурами. Відповідно, переклад метафор у науково-економічних текстах має важливе значення для забезпечення чіткості, точності та зрозумілості інформації.

Проблема полягає у тому, що метафори в науково-економічному стилі мають специфічні концептуальні значення та контекстуальні нюанси, які можуть бути складні для передачі в інші мови. Додатково, культурні різниці та відмінності у менталітеті можуть

впливати на сприйняття та розуміння метафор, що потребує уважного вивчення та аналізу.

Аналіз останніх досліджень і публікацій. Деякі особливості перекладу метафор на українську мову у науково-економічному стилі викликають широкий і різноманітний інтерес серед науковців. Велика кількість дослідників, зокрема І. Брага, В. Володіна, Х. Дацишина, Ю. Караулова, Л. Павлюка, Г. Солганика, О. Сербенська, та інші, займалися вивченням метафори як мовного засобу комунікації. Лінгвісти, такі як Д. Девідсон та Г. Скляревська, аналізували функціональний аспект метафоризації. Перекладачі, зокрема М. Джонсон, Дж. Лакофф та В. Москвін, займалися класифікацією метафор. В. Комісаров та П. Ньюмарк у своїх дослідженнях розглядали особливості передачі метафор у перекладі[1].

Ці дослідження свідчать про значення метафори як засобу комунікації та її роль у передачі економічних концепцій.

Мета й завдання дослідження. Мета статті полягатиме у виявленні особливостей перекладу метафор на матеріалах науково-економічного стилю в контексті української мови. Дослідження буде спрямоване на визначення проблемних аспектів перекладу метафор, виявлення можливих труднощів та розробку стратегій та рекомендацій щодо їх ефективного перекладу.

Стаття буде корисною для перекладачів, лінгвістів, студентів та науковців, що цікавляться лінгвістикою та перекладознавством, а також для фахівців у галузі економіки та фінансів, які займаються міжнародними комунікаціями та перекладом науково-економічних текстів.

З метою досягнення поставленої мети, стаття має такі завдання:

- з'ясувати та проаналізувати особливості мовного перекладу економічних текстів у процесі перекладу з англійської на українську. Це передбачає вивчення специфіки економічного стилю, аналіз термінології та виявлення особливостей, які виникають під час перекладу метафоричних висловів та концептів;
- -дослідити найбільш часті перекладацькі трансформації та підходи, що використовуються для подолання труднощів з перекладом економічних текстів. Це включає аналіз різних стратегій перекладу метафор, виявлення ефективних методів передачі концептуальних значень та контекстуальних нюансів метафоричних висловів.

Такі завдання дозволять виявити особливості перекладу метафор на матеріалах науково-економічного стилю, проаналізувати проблеми, які виникають під час перекладу та запропонувати практичні рекомендації та стратегії для ефективного перекладу економічних текстів з урахуванням метафоричної природи висловів.

Предметом дослідження в статті є способи перекладу економічних термінів.

Об'єктом дослідження є безпосередньо економічні тексти, що вивчаються.

Для дослідження цих особливостей перекладу українською мовою метафори на матеріалах науково-економічного стилю в даній статті були використані такі методи: метод зіставного аналізу, метод словникових дефініцій, описовий метод та метод міжмовної перекладної особливості. Ці методи дозволили з'ясувати та проаналізувати особливості мовного перекладу економічних текстів, а також дослідити найбільш часті перекладацькі трансформації та підходи, що використовуються для подолання труднощів з перекладом економічних текстів.

Початковою особливістю перекладу економічних текстів є необхідність точності передачі інформації. Це вимагає від перекладача не лише знання економічної теорії та специфіки термінології, але й розуміння загальних принципів економічної науки.

У зв'язку з використанням метафор в економічних текстах, виникає потреба в дослідженні їх перекладу на українську мову. Метафори можуть містити складні образні вирази, які потребують особливої уваги при перекладі, оскільки неправильне тлумачення або невдале перекладення можуть призвести до спотворення семантики та експресивності метафоричних висловлювань.

Метафора є одним з найпоширеніших засобів виразності в мові, вона допомагає передати складні ідеї та концепції шляхом порівняння з відомими образами. У перекладі

метафоричних висловів з однієї мови на іншу важливо зберегти їхню смислову та емоційну забарвленість [1; 2]. Особливості перекладу українською мовою метафори на матеріалах науково-економічного стилю можуть вимагати додаткової уваги та навичок в мові.

По-перше, науково-економічний стиль відрізняється своєрідністю термінології та специфічними поняттями [3]. Перекладачу потрібно ретельно вивчити фахову лексику та особливості економічних теорій, щоб зрозуміти контекст і передати метафоричний вираз належним чином. Наприклад, метафора "ринок знань" може використовуватися в науково-економічному тексті для опису обміну інформацією та навичками. Перекладачеві важливо вибрати еквівалентну метафору в українській мові, яка відображатиме аналогічність ідеї та збереже фаховий зміст.

По-друге, українська мова має свої унікальні метафоричні вирази, які можуть відрізнятися від англійської чи інших мов. Перекладачеві потрібно бути уважним до таких особливостей та знати національну культуру та традиції, щоб правильно використовувати ці унікальні вирази. Наприклад, українська мова використовує вираз "витратити останню копійку", щоб виразити ідею повного витрачання ресурсів або фінансового вкладу. У такому випадку перекладач може шукати відповідність українському еквіваленту, наприклад, "витратити останні гроші" або "витратити останні ресурси". Важливо знати та розуміти українські метафоричні вирази, щоб зберегти їхню ефективність та виразність при перекладі.

По-третє, науково-економічний стиль часто використовує абстрактні поняття та процеси, які можуть бути виражені за допомогою метафор. Перекладачеві важливо знайти еквівалентну метафору, яка передасть суть поняття та його відношення до економічних процесів. Наприклад, метафора "економіка на шляху до підйому" використовується для опису поліпшення економічної ситуації. Перекладач може використати українську метафору, яка виражає аналогічну ідею, наприклад, "економіка на шляху до покращення" або "економіка витягнулася з ями".

Крім того, важливо враховувати стиль та аудиторію, до якої звертається науково-економічний текст. Деякі метафори можуть бути занадто розмовними або несприйнятними для формального стилю. Тому перекладачеві потрібно уміти адаптувати метафори до наукового реєстру, зберігаючи їхню смислову цілісність та ефективність.

Українська мова також має свої особливості в уживанні метафор, які варто враховувати при перекладі науково-економічного стилю. Наприклад, українська мова може використовувати рослинні метафори для опису економічних процесів. Наприклад, "економічні засоби вкладаються у розвиток" може бути перекладено як "економічні ресурси приживаються до розвитку". Використання таких метафор допомагає створити зв'язок з природними процесами та зрозуміти ідею вкладання ресурсів у певний напрямок.

До інших особливостей українського перекладу метафор науково-економічного стилю відносяться використання ідіоматичних виразів та культурних референцій Це можуть бути національні вирази, які мають визначене значення в українській культурі. Перекладачеві слід знати ці вирази та здати їх відповідним чином, щоб зберегти метафоричну силу та ефективність тексту. Наприклад, метафора "відкривати нові горизонти" може бути перекладена як "відкривати нові горизонти знань" або "відкривати нові горизонти можливостей"[4].

Нарешті, у перекладі науково-економічного стилю важливо враховувати точність та ясність передачі інформації. Метафори повинні бути зрозумілі та сприйнятні для аудиторії, а також відповідати контексту та специфіці тематики тексту. Перекладачеві слід бути уважним до деталей і знати основні принципи перекладу метафор, щоб досягти максимальної точності і зберегти інтегральність метафоричного виразу. Наприклад, у науково-економічному тексті можна зустріти метафору "економіка втрачає пульс" для опису спаду економічного зростання. У перекладі ця метафора може бути передана, наприклад, як "економіка втрачає темпи" або "економіка втрачає свою динаміку". Важливо знайти правильне поєднання слів та виразів, щоб зберегти ідею та ефект метафори.

Переклад метафор науково-економічного стилю на українську мову вимагає від

перекладача не лише знання лексики та граматики, але й розуміння контексту, культурних особливостей та фахових термінів. Важливо зберегти ефективність та точність метафор, враховуючи специфічні особливості української мови та національної культури. Перекладач повинен бути креативним і вміти знайти адекватні вирази, які передадуть метафоричний зміст і збережуть стиль тексту.

Враховуючи ці особливості, переклад метафор науково-економічного стилю на українську мову може стати викликом, але він також відкриває можливість творчого та ефективного виразу ідей та концепцій [4]. Правильний переклад метафор допомагає зберегти точність і виразність тексту, сприяючи зрозумінню та ефективному сприйняттю інформації українською аудиторією науково-економічних текстів. Переклад метафор науково-економічного стилю на українську мову вимагає від перекладача не лише знання лексики та граматики, але й розуміння контексту, культурних особливостей та фахових термінів. Важливо зберегти ефективність та точність метафор, враховуючи специфічні особливості української мови та національної культури. Перекладач повинен бути креативним і вміти знайти адекватні вирази, які передадуть метафоричний зміст і збережуть стиль тексту.

Враховуючи ці особливості, переклад метафор науково-економічного стилю на українську мову може стати викликом, але він також відкриває можливість творчого та ефективного виразу ідей та концепцій. Правильний переклад метафор допомагає зберегти точність і виразність тексту, сприяючи зрозумінню та ефективному сприйняттю інформації українською аудиторією науково-економічних текстів. Однією з важливих особливостей перекладу метафор науково-економічного стилю є необхідність збереження наукової точності та чіткості. Враховуючи те, що економічні поняття та процеси можуть бути складними і абстрактними, перекладач повинен добре розуміти їх сутність та використовувати метафори, які передають ці поняття без спотворень.

При перекладі метафор науково-економічного стилю на українську мову, перекладачеві також слід звернути увагу на відповідність стилю та реєстру мовлення. Науково-економічний стиль вимагає формальності та точності, тому метафори повинні бути перекладені з врахуванням цих особливостей.

Крім того, важливо пам'ятати, що метафори можуть мати культурні особливості та контекстуальні відтінки, які можуть відрізнятися між різними мовами. Перекладачеві потрібно знати українську культуру та мовленнєві традиції, щоб знайти відповідні метафори та вирази, які були б зрозумілі та ефективні для українських читачів.

Важливо також враховувати, що переклад метафор - це творчий процес, і іноді не існує єдиного правильного варіанту перекладу. Перекладачеві можуть знадобитися гнучкість і творчість, щоб знайти найкращий спосіб передачі метафоричного виразу на українську мову. Це може включати вибір аналогічних метафор, переформулювання або використання інших стилістичних засобів для досягнення бажаного результату.

Крім того, у перекладі науково-економічного стилю слід враховувати контекст, у якому використовується метафора. Важливо розуміти метафору в контексті конкретного економічного терміну або процесу, щоб забезпечити точність і зрозумілість перекладу. Наприклад, метафора "економічний шторм" може використовуватися для опису фінансової кризи. При перекладі цієї метафори, перекладачеві слід знати специфіку фінансової кризи та використовувати відповідні українські метафоричні вирази, щоб передати аналогічну ідею.

Також варто звернути увагу на стиль і мету комунікації. У наукових текстах, де пріоритетом є точність та об'єктивність, метафори можуть бути менш поширеними. Але якщо метафори використовуються, їх переклад повинен відповідати формальному стилю та мовним нормам наукових текстів.

Враховуючи всі ці особливості, перекладач науково-економічного стилю повинен мати широкі знання економічних термінів, розуміння культурних контекстів і навички використання метафор української мови. Він повинен бути уважним до деталей та здатним зберігати ефективність метафор, забезпечуючи чіткість, точність та стильову відповідність перекладу. Науково-економічний стиль має свої особливості, які впливають на переклад

метафор. Однією з них є використання специфічних термінів та концепцій, які потребують точного та акуратного перекладу. При перекладі метафор науково-економічного стилю, перекладач повинен бути ознайомлений з фаховою лексикою та термінологією, щоб забезпечити вірність передачі значення метафори.

Другою особливістю є використання абстрактних концепцій та ідей у науково-економічному стилі. Це можуть бути такі метафори, як "економіка у спаді", "зростання як тягар" тощо. При перекладі таких метафор важливо знайти еквівалентні вирази, які будуть передавати абстрактне значення та ідею метафори. Наприклад, "економіка у спаді" може бути перекладено як "економіка знижується", а "зростання як тягар" - як "зростання стає обтяжливим".

Третя особливість полягає у використанні стандартів та показників у науково-економічному стилі, які можуть мати свої відповідники у метафорах. Наприклад, "досягнення показників" може бути перекладено як "досягнення мети", а "відхилення від стандартів" - як "відхилення від норми". При перекладі таких метафор важливо врахувати специфіку контексту та передати значення стандартів та показників у вибраній метафорі [4; 5; 6; 7].

Остання особливість полягає у використанні економічних теорій та моделей, які можуть відображатися у метафорах. Наприклад, "пропускна здатність резервуару" може бути перекладено як "максимальний обсяг резервуару" або "межа потужності резервуару". При перекладі таких метафор, перекладач повинен мати розуміння економічних концепцій та моделей, щоб забезпечити точний та зрозумілий переклад.

Важливо також враховувати стилістичні відтінки метафор та їх вплив на сприйняття тексту. Науково-економічний стиль характеризується формальністю та об'єктивністю, тому переклад метафор повинен враховувати цей стиль та підтримувати наукову атмосферу тексту. Перекладач повинен забезпечити відповідну форму та мовний рівень перекладу, щоб зберегти стильову відповідність.

Окрім того, переклад метафор науково-економічного стилю також залежить від аудиторії, до якої звертається текст. Якщо це наукове видання або спеціалізована аудиторія, то переклад може бути більш технічним та використовувати специфічні терміни. У разі загальної аудиторії або популярних наукових видань, перекладач повинен забезпечити більш зрозумілі та доступні метафори, які передадуть ідею без втрати точності.

Висновок. Переклад метафор науково-економічного стилю на українську мову вимагає від перекладача розуміння фахових термінів, контексту та стилістичних особливостей тексту. Перекладач повинен знайти відповідні українські метафори, які збережуть точність, чіткість та стиль тексту. Такий переклад допоможе зрозуміти ідеї та концепції науково-економічного стилю українським читачам, сприяючи їхньому кращому розумінню інформації та збереженню наукової цінності тексту.

Важливо також пам'ятати про контекстуальну відповідність при перекладі метафор. Конкретні економічні терміни та процеси можуть мати відмінності у використанні метафор в різних мовах. Тому перекладач повинен враховувати культурні особливості та вибрати метафору, яка була б належним чином сприйнята українським читачем і передавала б суть оригінальної метафори.

Нарешті, важливо зазначити, що переклад метафор науково-економічного стилю - це творчий процес, де перекладач має використовувати свої знання, досвід і вміння. Іноді може бути необхідно експериментувати з різними варіантами перекладу, вибирати найкращий спосіб передачі ідеї метафори на українську мову.

Отже, переклад метафор науково-економічного стилю на українську мову є складним завданням, яке вимагає від перекладача розуміння економічних концепцій, вміння використовувати відповідні метафори та збереження наукової точності. Тільки шляхом уважного вибору і перекладу метафор можна забезпечити ефективну комунікацію і передачу інформації українським читачам в контексті науково-економічного стилю.

Список літератури

1. Herrmann, J. B. Metaphor in specialist discourse. J. B. Herrmann, T. Berber Sardinha. – Amsterdam, Netherlands: John Benjamins Publishing Company, 2015. 185 p.
2. Kuhn T.S. Metaphor in science. Ortony A. Metaphor and thought. - Cambridge: Cambridge University Press, 2014. 409–419 p.
3. Newmark Peter. A Textbook of Translation. Harlow: Pearson Education Limited, 2008. 292 p.
4. Яковенко Р. В. Тлумачний англо-український словник економічних термінів з елементами теорії та проблематики. Дидактичний довідник. Роман Яковенко. – [Вид. 2–ге, випр.]. – Кіровоград: видавець Лисенко В.Ф., 2015. – 130 с. URL: http://dspace.kntu.kr.ua/jspui/bitstream/123456789/5566/1/Tlumachniy%20slovnyk.pdf (дата звернення: 12.10.2022
5. Abbreviations Dictionary. URL: http://www.acronymfinder.com (дата звернення: 12.09.2022).
6. Collins English Dictionary. Harper Collins Publishers, 2006. 774 p.
7. Levy J. Translation as a Decision Process. Translation Studies Reader. London and New York, 2003. p. 148–159.

УДК 004

В.Сушков, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ХМАРНОГО СЕРВІСУ З ВИКОРИСТАННЯМ АЛГОРИТМУ TDEA

У статті розроблено програмне забезпечення, яке призначено для системи хмарного сервісу з використанням алгоритму TDEA. Метою розробки є дослідження та програмна реалізація системи хмарного сервісу з використанням алгоритму TDEA. Об'єктом дослідження є процес хмарного сервісу з використанням алгоритму TDEA. Предметом дослідження є методи хмарного сервісу з використанням алгоритму TDEA. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи хмарного сервісу з використанням алгоритму TDEA. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, хмарний сервіс, TDEA

Постановка проблеми. Хмарні обчислення – це архітектура для надання обчислювальних послуг через Інтернет на вимогу та платного доступу до пулу спільних ресурсів без їх фізичного отримання. Таким чином, це заощаджує кошти та час для організації. Хмарні обчислення є новою парадигмою. Яка стала найактуальнішою сферою досліджень сьогодні завдяки своїй здатності зменшувати витрати, пов'язані з обчислювальною технікою. Сьогодні це найцікавіша та приваблива технологія, яка пропонує своїм користувачам послуги на вимогу через Інтернет. Оскільки хмарні обчислення зберігають дані та їх розповсюджені ресурси в середовищі, безпека стала основною перешкодою, яка заважає розгортанню хмарних середовищ [1]. Багато користувачів використовують хмару для зберігання своїх особистих даних. Таким чином, необхідна безпека зберігання даних на носії інформації [3]. Ця концепція використовується алгоритмом DNS. Це симетричний блоковий шифр, який можна використовувати як додаткову заміну DES або IDEA. Він використовує ключ змінної довжини, від 32 біт до 448 біт, що робить його ідеальним як для внутрішнього використання, так і для експорту. Це алгоритм шифрування, який можна використовувати як заміну алгоритмам DES або IDEA. Ця стаття використовується, щоб дізнатися про безпеку хмарних обчислень за допомогою алгоритму DES [4]. Практична атака Sweet32 на набори шифрів на основі 3DES у TLS вимагала блоків

(785 ГБ) для повної атаки, але дослідникам пощастило отримати зіткнення одразу після блоків, що зайняло лише 25 хвилин. На безпеку TDEA впливає кількість блоків, оброблених одним набором ключів [8]. Один пакет ключів не повинен використовуватися для застосування криптографічного захисту (наприклад, шифрування) більш ніж 64-бітних блоків даних.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи хмарного сервісу з використанням алгоритму TDEA.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи хмарного сервісу з використанням алгоритму TDEA.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем хмарного сервісу з використанням алгоритму TDEA.
– Дослідження системи хмарного сервісу з використанням алгоритму TDEA.
– Програмна реалізація системи хмарного сервісу з використанням алгоритму TDEA.

Об'єктом дослідження є процес хмарного сервісу з використанням алгоритму TDEA.

Предметом дослідження є методи хмарного сервісу з використанням алгоритму TDEA.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу

Більшість організацій вже досить давно використовують переваги хмарних обчислень. Важко нехтувати перевагами гнучкості, гнучкості та масштабованості, коли було б важко підтримувати лише фізичне обладнання.

Де ускладнюється пошук способів захисту робочих ресурсів, які розміщені за межами вашого приміщення. Це відкриває шлях до хмарної безпеки як дисципліни захисту хмарних систем даних. Давайте заглибимося в тему, щоб знайти все, що потрібно знати про хмарні рішення безпеки та як вони працюють.

Що таке хмарна безпека?

Хмарна безпека – це набір процедур і технологій, призначених для захисту даних і захисту від зовнішніх і внутрішніх загроз. Із зростанням інтеграції з хмарою зростають і потенційні ризики, і компаніям потрібні рішення для захисту їх мережевої інфраструктури. Встановлення правильного балансу між продуктивністю та безпекою має першочергове значення.

Хмарні рішення безпеки розгортаються подібно до інструментів, що використовуються для захисту фізичного обладнання. Ключова відмінність полягає в тому, що ними також можна керувати та розгортати віддалено. Відповідальність за захист даних розподіляється між постачальником хмарних технологій і клієнтом. Перший постачальник повинен забезпечити безпеку налаштування свого апаратного забезпечення та правил доступу, тоді як другий має подбати про шифрування сховища та різні конфігурації політик безпеки.

Це одна з ключових причин, чому вважається, що хмарну безпеку підтримувати набагато важче, ніж локальні моделі. Оскільки є більше залучених сторін, це також означає, що щось важливе може бути пропущено. Не кажучи вже про те, що використання зовнішніх постачальників позбавляє клієнта значної видимості та контролю.

Чому хмарна безпека важлива?

Організації значною мірою покладаються на хмарні обчислення для багатьох своїх повсякденних операцій. Динамічний характер хмарної інфраструктури надає багато чудових можливостей для компаній, які прагнуть отримати переваги під час досягнення своїх бізнес-цілей. Оскільки потенціал великий, компанії, які знаходять способи приборкати хмарні

обчислення, можуть подолати багато викликів ІТ.

Однак, оскільки хмарні обчислення все ще є новою територією для більшості підприємств, ризики, пов'язані зі збереженням ваших даних поза межами, є більш помітними. Відповідно до домовленості між хмарним провайдером, кожен клієнт несе відповідальність за безпеку своїх даних. Таким чином, кожна організація має розглянути, як підійти до хмарної безпеки для свого унікального бізнес-випадку.

Кібербезпека завжди вимагає активного внеску з боку організації. В іншому випадку вони ризикують привернути небажану увагу з боку хакерів, які спеціально націлені на хмарні мережі. Тому хмарні обчислення актуальні незалежно від розміру вашої організації чи галузі.

Основні переваги хмарної безпеки

Хмарна безпека приносить користь організаціям кількома способами:

– Допомагає запобігти кібератакам. Хмарна безпека може бути основою для стримування або припинення вхідних спроб злому.

– Покращує безпеку даних. Різні технології допомагають захистити конфіденційні дані, допомагаючи захистити дані, щоб вони не потрапили в чужі руки.

– Полегшує обслуговування хмари. Більшість хмарних служб пропонують моніторинг і підтримку в реальному часі, що допомагає підвищити надійність обслуговування.

– Швидше одужання. У разі порушення даних хмарні інструменти безпеки допомагають легше організувати процес відновлення.

– Відповідність нормативним вимогам. Часто хмарна безпека є обов'язковою вимогою для безпечної акредитації на відповідність нормативним вимогам.

Як працює хмарна безпека?

Хмарна безпека допомагає організаціям, надаючи різні елементи керування для захисту від загроз для програм даних і хмарних систем. Оскільки платформи хмарних обчислень є основним рішенням для більшості підприємств, загрози, націлені на бізнес, часто спрямовані на хмару.

Тому хмарні рішення безпеки допомагають бізнесу кількома способами:

– Збільшити прозорість. Набагато легше захистити організацію, коли мережеві адміністратори знають, до чого мають доступ користувачі.

– Моніторинг стану мережі. Знання про те, яка діяльність відбувається в хмарі, може допомогти зупинити різні ризики на їх шляху.

– Підвищує рівень безпеки. Найважливіші ресурси можна краще захистити від доступу неавторизованих користувачів до конфіденційної інформації.

– Забезпечує ефективніше керування ідентифікацією. Збільшення вимог до доступу допомагає захистити облікові записи користувачів від захоплення.

– Порівнює безпеку з вимогами відповідності. Оскільки більшість компаній зберігають багато конфіденційної інформації, хмарна безпека допомагає їм відповідати визначеним стандартам безпеки.

Типи хмарних середовищ

Незважаючи на такий загальний термін, хмарні обчислення можна налаштувати різними способами. Важливо також відзначити, що навіть один і той же тип хмари може бути організований по-різному. Проте кожен тип хмарних обчислень має слабкі та сильні сторони, які можуть суттєво вплинути на ваш бізнес.

Загальнодоступні хмари

Загальнодоступна хмара – це середовище, яке на вимогу розповсюджується через загальнодоступний Інтернет постачальником послуг. Деякі загальнодоступні хмари є безкоштовними для всіх, тоді як інші вимагають підписки або тарифікуються згідно з моделями оплати за користування. Найбільші публічні хмарні постачальники включають Google Cloud, Amazon Web Services, Microsoft Entra ID і IBM Cloud.

Такі послуги допомагають перспективним компаніям переносити робочі навантаження назовні та легко збільшувати або зменшувати масштаб відповідно до своїх потреб. Це звільняє локальних мережевих адміністраторів і допомагає знизити ІТ-витрати.

Набагато дешевше використовувати спільну інфраструктуру, якою керує третя сторона, ніж мати такий самий масштаб налаштування всередині компанії.

Приватні хмари

Приватна хмара – це хмарне середовище, у якому всі апаратні та програмні ресурси зарезервовані та доступні для одного клієнта. Часто ці середовища захищені брандмауером групи. Це створює повністю ізольований доступ без збігів з іншими користувачами хмари.

Більшість компаній віддають перевагу налаштуванням приватної хмари, оскільки це набагато простіший спосіб забезпечити безпеку та відповідати вимогам відповідності. Однак один великий недолік цієї установки полягає в тому, що вона не така масштабована, як публічна хмара. Приватні хмари зазвичай мають фіксований розмір, і їх неможливо миттєво збільшити або зменшити. Для підвищення масштабу приватної хмари знадобляться додаткові ліцензії на обладнання та програмне забезпечення.

Гібридні хмари

Гібридна хмара – це середовище, у якому програми працюють із різних джерел: хмарних і локальних. Цей метод є найпопулярнішим у хмарних обчисленнях, оскільки більшість компаній отримують найкраще з обох світів. Більшість компаній використовують інфраструктуру, яку вони створили протягом тривалого часу, і розширюють її за допомогою хмарних доповнень.

Підключення хмарних і локальних середовищ зазвичай здійснюється за допомогою локальних мереж (LAN), глобальних мереж (WAN), віртуальних приватних мереж (VPN) та інших методів. Вся установка керується з інтегрованої платформи керування та оркестровки.

Багатохмарність

Мультихмари – це комбінації різних типів хмар, публічних або приватних. Це налаштування створюється, коли різні хмари (часто від різних постачальників послуг) об'єднуються певним методом інтеграції чи оркестровки. Це допомагає уникнути прив'язки до постачальника та створювати більш гнучкі рішення, адаптовані до конкретних потреб бізнесу.

Часто такі налаштування створюються для однієї хмари, яка функціонує як резервна копія на випадок запобігання втраті даних. У разі аварії дані організації можна безпечно відновити з резервної копії.

Типи моделей Cloud Service

Хмарні обчислення можна надати як три різні моделі послуг, кожна з яких надає унікальний набір переваг, які можуть задовольнити різні потреби бізнесу.

IaaS

Інфраструктура як послуга фактично пропонує типові компоненти інфраструктури центру обробки даних, такі як обладнання, обчислювальна потужність, простір для зберігання або мережеві ресурси. Доступ до ресурсів здійснюється через віртуальні або приватні мережі, і клієнт може швидко використовувати їх. Цей метод вирішує проблему підтримки фізичного обладнання для малих, середніх і великих компаній.

SaaS

Програмне забезпечення як послуга – це ліцензія та модель продажу, яка використовується для доставки програмного забезпечення через загальнодоступний Інтернет. Зазвичай користування здійснюється за підпискою. Після сплати комісії ви можете користуватися послугою протягом встановленого періоду часу. Постачальник – це той, хто контролює весь стек обчислень. Тим часом користувач може безпосередньо взаємодіяти з програмним забезпеченням з його кінцевої точки.

PaaS

Платформа як послуга пропонує повний набір інструментів середовища розробки. Це значно спрощує процес розробки програмного забезпечення та корисно під час створення нових програм. Ця структура миттєво надає інструменти проектування, тестування та доставки, що дозволяє клієнтам швидко розпочати роботу над новими проектами.

Типи хмарних рішень безпеки

Доступно кілька типів хмарних рішень безпеки, кожне з яких підходить для конкретного завдання.

Керування ідентифікацією та доступом (IAM)

Управління ідентифікацією та доступом (IAM) – це структура бізнес-процесів, яка полегшує політику та технології для керування цифровою ідентифікацією. IT-менеджери можуть використовувати IAM для контролю доступу до ресурсів організації. IAM створює цифрові ідентифікатори для кожного користувача, що полегшує їх моніторинг і обмеження.

Запобігання втраті даних (DLP)

Запобігання втраті даних (DLP) – це набір інструментів і процесів, які використовуються для забезпечення безпеки бізнес-даних. Він використовує різні інструменти, як-от шифрування, профілактичні заходи та сповіщення про виправлення, щоб захистити дані під час передавання чи зберігання.

Інформація про безпеку та керування подіями (SIEM)

Управління інформацією про безпеку та подіями (SIEM) – це підхід до управління безпекою для оркестрування IT-безпеки організації. Він використовує різні інструменти керування інформацією та подіями для створення єдиної інформаційної панелі за допомогою ШІ для кореляції даних на кількох платформах. Це дозволяє легко мати повний панорамний огляд безпеки організації.

Безперервність бізнесу та аварійне відновлення

Інструменти безперервності бізнесу (BC) і аварійного відновлення (DR) надають організаціям інструменти, послуги та протоколи для відновлення організації після аварії. Ці послуги допомагають організаціям зменшити ризик втрати даних і шкоди репутації та покращити поточні бізнес-операції.

Хмарні загрози безпеці

Хмарні системи піддаються тим самим ризикам, що й ваша локальна інфраструктура. Однак залучення додаткових сторін збільшує загальну суму ризиків.

– Відсутність повного контролю. Оскільки хмарні сервіси існують за межами корпоративних мереж, організації не повністю контролюють усі сфери кібербезпеки.

– Багатоквартирність. Коли кілька клієнтів орендують послуги в одного постачальника, ви можете потрапити під лавину, коли один із ваших сусідів буде зламаний.

– Shadow IT. Хмарні середовища сумно відомі тіньовими IT-налаштуваннями, особливо коли активна політика «принеси свій власний пристрій» (BYOD).

– Неправильні конфігурації. Однією з найпоширеніших причин витоку даних є неправильні налаштування. Інсайдерські аварії часто призводять до витоку клієнтської інформації, що засмучує, навіть якщо налаштування безпеки належні. Дізнайтеся більше про хмарні загрози безпеці, ризики та вразливості

Хмарні інструменти безпеки

Ось деякі з конкретних інструментів, які використовуються для захисту хмари:

– Cloud Workload Protection Platform (CWPPs) – система безпеки, призначена для захисту робочих навантажень

– Cloud Access Security Brokers (CASB) – посередник між хмарними клієнтами та хмарним сервісом, який забезпечує дотримання політик безпеки

– Cloud Security Posture Management (CSPM) – набір інструментів безпеки, які полегшують моніторинг і виявлення неправильної конфігурації

– Secure Access Service Edge (SASE) – конвергенція різноманітних засобів безпеки та мережевих інструментів, щополегшує керування безпекою мережі

Нарешті, численні доповнення, як-от веб-служби IAM, інструменти DLP та інші інструменти безпеки, допомагають користувачам хмари.

Як захистити хмару

Ось кілька порад про те, як краще захистити інформацію в хмарі.

– Шифрування. Для каналів зв'язку та постійного зберігання слід використовувати

шифрування. Таким чином дані будуть недоступні під час передачі та коли ваш сервер зламано.

– Безпечні конфігурації. Дотримання належної гігієни управління послугами кібербезпеки. Це передбачає зміну паролів за замовчуванням і отримання додаткової інформації про елементи керування безпекою хмарного постачальника.

– Використовуйте надійні паролі. Жодне налаштування безпеки не допоможе, якщо ваші користувачі повторно використовують ті самі паролі. Надійні паролі піднімають смугу входу в організацію, ускладнюючи проникнення.

– Обмежити дозволи. Їх не слід надавати, якщо для виконання певної посадової ролі не потрібні дозволи. Хоча це здається обмежувальним, це також допомагає запобігти багатьом ризикам кібербезпеки.

Нарешті, для користувачів, які покладаються на сторонніх постачальників, неможливо недооцінити, наскільки важливим є аналіз умов обслуговування. Чіткий розподіл обов'язків допоможе уникнути сірих зон, які можуть бути використані. Це важливий документ, який допомагає зрозуміти слабкі сторони вашого поточного налаштування та кроки, які можна вжити, щоб виправити його налаштування.

TDEA

Сервіси хмарного зберігання даних швидко стають все більш популярними. Користувачі можуть зберігати свої дані в хмарі та отримувати доступ до них будь-де в будь-який час. Через конфіденційність користувачів дані, що зберігаються в хмарі, зазвичай зашифровані та захищені від доступу інших користувачів.

Беручи до уваги властивість хмарних даних працювати над співробітництвом, шифрування на основі атрибутів (ABE) вважається однією з найбільш підходящих схем шифрування для хмарних сховищ [7]. Існує багато запропонованих схем ABE.

Більшість із запропонованих схем припускають, що постачальники послуг хмарного сховища або довірені треті сторони, які займаються керуванням ключами, є надійними та не можуть бути зламані; однак на практиці деякі суб’єкти можуть перехоплювати зв’язок між користувачами та постачальниками хмарних сховищ, а потім змушувати постачальників сховищ розкривати секрети користувачів, використовуючи владу уряду чи інші засоби [3]. У цьому випадку вважається, що зашифровані дані відомі, а постачальники сховищ мають надати секрети користувача. Google надав ФБР документи користувача після отримання ордера на обшук. У 2013 році Едвард Сноуден розкрив існування глобальних програм стеження, які збирають такі хмарні дані, як електронні листи, текстові та голосові повідомлення від деяких технологічних компаній. Після зламу постачальників хмарних сховищ усі схеми шифрування втрачають свою ефективність [4].

Хоча ми сподіваємося, що постачальники хмарних сховищ зможуть боротися з такими організаціями, щоб зберегти конфіденційність користувачів через законні шляхи, це, здається, стає все складнішим.

Наприклад, Lavabit була компанією, що надає послуги електронної пошти, яка захищала всі електронні листи користувачів від зовнішнього зловмисного впливу; на жаль, це не вдалося, і він вирішив закрити свою службу електронної пошти.

Оскільки боротися із зовнішнім зловмисним впливом важко, ми мали на меті створити схему шифрування, яка могла б допомогти постачальникам хмарних сховищ уникнути цієї скрутної ситуації. У цій роботі пропонується постачальникам хмарних сховищ засоби для створення підроблених секретів користувачів. Враховуючи такі фальшиві секрети користувача, сторонні особи можуть отримати лише підроблені дані зі збереженого зашифрованого тексту користувача. Щойно зловмисники подумають, що отримані секрети справжні, вони будуть задоволені, і, що більш важливо, постачальники хмарних сховищ не розкриють жодних справжніх секретів [8]. Тому конфіденційність користувачів все ще захищена.

Ця концепція походить від особливого виду схеми шифрування, яка називається шифруванням із запереченням, запропонована вперше [1]. Шифрування, яке можна

заперечувати, включає відправників і одержувачів, які створюють переконливі підроблені докази підроблених даних у зашифрованих текстах, щоб сторонні зловмисники особи були задоволені. Зверніть увагу, що заперечення випливає з того факту, що зловмисники не можуть довести, що запропоновані докази неправильні, і тому не мають підстав відхиляти надані докази. Цей підхід намагається повністю заблокувати зусилля зловмисного впливу, оскільки зловмисники знають, що їхні зусилля будуть марними.

Ми використовуємо цю ідею, щоб постачальники хмарних сховищ могли надавати послуги зберігання без аудиту.

У сценарії хмарного сховища власники даних, які зберігають свої дані в хмарі, є схожими на відправників у схемі забороненого шифрування [3]. Ті, хто має доступ до зашифрованих даних, відіграють роль одержувачів у схемі шифрування, яка забороняється, включаючи самих постачальників хмарних сховищ, які мають загальносистемні секрети та повинні мати можливість розшифрувати всі зашифровані дані.

У цій роботі ми описуємо заперечувальну схему ABE для хмарних служб зберігання. Ми використовуємо характеристики ABE для захисту збережених даних за допомогою точного механізму контролю доступу та забороненого шифрування для запобігання зовнішньому аудиту. Наша схема заснована на схемі шифрування на основі атрибутів політики Waters (CP-ABE) [4]. Ми вдосконалюємо схему Уотерса від білінійних груп простого порядку до білінійних груп складеного порядку. Згідно з припущенням про проблему вирішення підгрупи, наша схема дозволяє користувачам мати можливість надавати підроблені секрети, які здаються законними зовнішнім зловмисникам.

Загалом, потрійний DES із трьома незалежними ключами має довжину ключа 168 біт (три 56-бітні ключі DES), але завдяки зустрічі в середині атаки ефективний захист, який він забезпечує, становить лише 112 біт. Варіант ключа 2 зменшує ефективний розмір ключа до 112 біт (оскільки третій ключ такий самий, як і перший). Однак цей параметр сприйнятливий до певних атак із відкритим текстом або відомим текстом, і, таким чином, NIST визначає лише 80 біт безпеки. Це можна вважати несправним, оскільки доступний споживач може ретельно шукати весь простір ключів 3des.

Сахай і Уотерс першими представили концепцію ABE, у яку власники даних можуть вбудовувати спосіб обміну даними з точки зору шифрування [4]. Тобто лише ті, хто відповідає умовам власника, можуть успішно розшифрувати збережені дані. Тут ми зауважимо, що ABE – це шифрування для привілеїв, а не для користувачів. Це робить ABE дуже корисним інструментом для хмарних служб зберігання, оскільки обмін даними є важливою функцією для таких служб [2]. Користувачів хмарних сховищ так багато, що власникам даних непрактично шифрувати свої дані за допомогою парних ключів. Крім того, для багатьох людей також непрактично шифрувати дані багато разів. За допомогою ABE власники даних вирішують лише те, які користувачі можуть отримати доступ до їхніх зашифрованих даних [5]. Користувачі, які задовольняють умови, можуть розшифрувати зашифровані дані.

Існує два типи ABE: CP-ABE і Key-Policy ABE (KP-ABE). Різниця між цими двома полягає в перевірці політики. KP-ABE – це ABE, у якому політику вбудовано в секретний ключ користувача, а набір атрибутів – у зашифрований текст.

І навпаки, CP-ABE вбудовує політику в зашифрований текст, а секрет користувача має набір атрибутів [3].

Goyal та ін. запропонувала першу KP-ABE. Вони створили виразний спосіб зв'язати будь-яку монотонну формулу як політику для секретних ключів користувача.

Бетенкур запропонував перший CP-ABE в [4]. Ця схема використовувала деревоподібну структуру доступу для вираження будь-якої монотонної формули через атрибути як політику в зашифрованому тексті.

Перший повністю експресивний CP-ABE був запропонований Вотерсом, у якому використовувалися схеми лінійного секретного обміну (LSSS) для побудови політики шифрованого тексту [5].

Левко та ін. покращив схему Waters до повністю безпечного CP-ABE, хоча й з деякою втратою ефективності.

На сучасному етапі аналізується техніко-економічне обґрунтування проекту та висувається бізнес-пропозиція з дуже загальним планом проекту та деякими оцінками витрат. Під час аналізу системи має бути проведено техніко-економічне обґрунтування запропонованої системи. Це зроблено для того, щоб запропонована система не була тягарем для компанії. Для аналізу здійсненності важливе певне розуміння основних вимог до системи. Три ключові міркування, пов'язані з аналізом здійсненності:

– економічний аналіз.

– технічний аналіз.

– соціальний аналіз.

Економічний аналіз

Це дослідження проводиться для перевірки економічного впливу від система матиме на організації. Сума коштів, яку компанія може влити в дослідження та розробку системи, обмежена. Витрати мають бути обґрунтованими. Таким чином, розроблена система також в межах бюджету, і це було досягнуто завдяки тому, що більшість використовуваних технологій знаходяться у вільному доступі. Треба було купувати лише індивідуальні продукти.

Технічний аналіз

Це дослідження проводиться для перевірки технічної здійсненності, тобто технічних вимог системи. Будь-яка розроблена система не повинна мати високих вимог до наявних технічних ресурсів. Це призведе до високих вимог до наявних технічних ресурсів. Це призведе до високих вимог до клієнта. Розроблена система повинна мати скромні вимоги, оскільки для впровадження цієї системи потрібні лише мінімальні або нульові зміни.

Соціальний аналіз

Аспектом дослідження є перевірка рівня прийняття системи користувачем. Це включає процес навчання користувача ефективному використанню системи. Користувач не повинен відчувати загрозу з боку системи, натомість повинен прийняти її як необхідність. Рівень прийняття користувачами залежить виключно від методів, які використовуються для навчання користувача системі та ознайомлення його з нею. Його рівень довіри потрібно підвищити, щоб він також міг зробити певну конструктивну критику, що вітається, оскільки він є кінцевим користувачем системи.

Вхід та вихід алгоритму DES

Дизайн вводу є ланкою між інформаційною системою та користувачем. Він включає в себе розробку специфікації та процедур для підготовки даних, і ці кроки, необхідні для переведення даних транзакцій у придатну для використання форму для обробки, можуть бути досягнуті шляхом перевірки комп'ютера для зчитування даних із письмового чи друкованого документа або це може відбутися за допомогою людей, які вводять ключі дані безпосередньо в систему.

Дизайн введення зосереджується на контролі необхідного обсягу введення, контролі помилок, уникненні затримок, уникненні додаткових кроків і збереженні простоти процесу.

Вхід розроблений таким чином, щоб забезпечити безпеку та легкість використання зі збереженням конфіденційності.

– Дизайн вхідних даних – це процес перетворення орієнтованого на користувача опису вхідних даних в комп'ютерну систему. Цей дизайн важливий, щоб уникнути помилок у процесі введення даних і показати правильний напрямок керівництву для отримання правильної інформації з комп'ютеризованої системи.

– Це досягається шляхом створення зручних екранів для введення даних для обробки великого обсягу даних. Метою проектування вхідних даних є полегшення введення даних і відсутність помилок. Екран введення даних розроблено таким чином, що можна виконувати всі операції з даними. Він також надає можливість перегляду записів.

– Після введення даних буде перевірено їх дійсність. Дані можна вводити за

допомогою екранів. Відповідні повідомлення надаються, коли це необхідно, щоб користувач не перебував у кукурудзяному стані.

Таким чином, мета дизайну вхідних даних полягає в тому, щоб створити макет вхідних даних, яким легко слідувати.

Вихідні дані

Якісні вихідні дані відповідають вимогам кінцевого користувача та чітко представляють інформацію. У будь-якій системі результати обробки повідомляються користувачам та іншій системі через виходи. У проекті виводу визначається, як інформація має бути переміщена для негайної потреби, а також виведення друкованої копії. Це найважливіше і пряме джерело інформації для користувача. Ефективна та інтелектуальна конструкція виводу покращує взаємозв'язок системи, щоб допомогти користувачеві приймати рішення:

– Розробка комп'ютерного виходу повинна відбуватися організовано, добре продумано; необхідно розробити правильний результат, забезпечуючи, щоб кожен вихідний елемент був розроблений таким чином, щоб люди могли легко й ефективно використовувати систему. Під час аналізу проектування вихідних даних комп'ютера вони повинні визначити конкретний результат, який необхідний для задоволення вимог.

– Виберіть методи для представлення інформації.

– Створення документів, звітів або інших форматів, які містять інформацію, створену системою.

Вихідна форма інформаційної системи повинна досягати однієї або більше з наступних цілей.

– Передавати інформацію про минулу діяльність, поточний стан або прогнози – майбутнього.

– Сигналізуйте про важливі події, можливості, проблеми або попередження.

– Викликати дію.

– Підтвердити дію.

Безпека.

Загалом, потрійний DES із трьома незалежними ключами має довжину ключа 168 біт (три 56-бітні ключі DES), але завдяки атаці зустрічі посередині ефективний захист, який він забезпечує, становить лише 112 біт. Варіант 2 ключа зменшує ефективний розмір ключа до 112 біт (оскільки третій ключ такий самий, як і перший). Однак ця опція сприйнятлива до певних атак обраного відкритого тексту або відкритого тексту, і, таким чином, NIST визначає лише 80-бітний захист [3]. Це можна вважати несправним, оскільки доступний споживач може ретельно шукати весь простір ключів 3des.

Апаратна атака сьогодні на пакети шифрів на основі 3DES у TLS вимагала блоків (785 ГБ) для повної атаки, але дослідникам пощастило отримати колізію відразу після блоків, що зайняло лише 25 хвилин [4]. Безпека TDEA постраждала за кількістю блоків, оброблених одним пакетом ключів.

Крім того, ми забезпечуємо, що цей автентифікатор може бути ефективно згенерований власником даних одночасно з процедурою кодування.

Широкий аналіз показує, що наша схема є доведено безпечною, а оцінка продуктивності показує, що наша схема є високоефективною та може бути реально інтегрована в систему хмарного зберігання на основі відновленого коду.

Розробка структурної схеми

На рисунку 1 зображена структурна схема системи хмарного сервісу з використанням алгоритму TDEA.

Забезпечення безпеки інформації при зберіганні й обробці більших інформаційних масивів – одна із самих актуальних проблем сучасних інформаційних технологій. Інтенсивний розвиток методів розподіленої обробки даних і різке збільшення обсягів інформації, що накопичується в комп'ютерних системах, привели останнім часом до кардинальної зміни методів довгострокового зберігання даних. Традиційні підходи до

організації зберігання більших інформаційних масивів перестали задовольняти зрослим вимогам до ємності носіїв і швидкості доступу до даних. Всі частіше споживач довіряє зберігання своєї власної інформації зовнішнім центрам або мережам зберігання даних (так званий аутсорсинг). Одна з основних сфер застосування мережного зберігання даних – формування банків даних електронних документів, а також електронних архівів і бібліотек. Такі сховища даних можуть бути як публічними, так і обмеженого доступу, залежно від характеру документів, що накопичуються в них. Нерідко перед приміщенням документів у мережні сховища вони піддаються стиску або іншим спеціальним видам кодування. У зв'язку із цим загострюється необхідність забезпечення керованості, надійності й безпеці зберігання й доступу до електронних документів, а також процедур їхньої передачі між прикладними програмами й пристроями зберігання.

Якщо навіть дані зберігаються локально, виникає інша проблема: адміністратори, що управляють СУБД і персонал так чи інакше звичайно мають права доступу до всієї збереженої інформації. Для захисту від їхніх несанкціонованих дій у деяких випадках доцільне застосування апаратно-програмних засобів шифрування даних перед записом їх на засоби зберігання. Часто шифрувальні модулі вбудовуються, наприклад, у засоби резервного копіювання даних. Однак при зберіганні шифрованих масивів утруднений пошук окремих файлів і оперативний доступ до елементів масиву, необхідним для роботи прикладних програм. Тому що масив зберігається в зашифрованому виді, і серверу, на якому він зберігається (або СУБД), не можуть бути довірені ключі шифрування, користувач (або прикладна програма від його ім'я) змушений завантажувати копії всіх файлів масиву, розшифровувати їх і потім виконувати пошук на локальній машині. Очевидно, що такий спосіб пошуку дуже неекономічний. У зв'язку із цим вимальовується проблема забезпечення можливості пошуку даних по шифрованим і (або) стислим даним, що може бути конкретизована залежно від застосовуваної в системі моделі шифрування даних.

Для шифрування великих масивів даних, що поміщаються в зовнішні стосовно власника інформації сховища, ефективні лише симетричні схеми шифрування. Можливості їхнього практичного застосування, мабуть, визначаються можливостями організації схеми керування секретними ключами, для яких необхідно забезпечити виконання двох почасти суперечливих вимог: забезпечення високої схоронності ключів (зокрема, за рахунок резервування) і обмеження середовища їхнього поширення тільки тими пристроями, яким довіряє власник інформації.

У зв'язку із цим у деяких випадках більше раціональним виглядає застосування схем відкритого шифрування, що дає можливість невизначеному колу осіб поміщати свої дані в сховище, але доступ до них залишати лише для власників секретного ключа. Така схема може бути корисна, наприклад, для систем електронної пошти або систем планування потоків завдань (workflows), де циркулюють переважно повідомлення невеликої довжини. Для таких схем тим більше необхідні механізми пошуку за шифрованим даними, що операції розшифрування в асиметричних криптосхемах, як відомо, виконуються на кілька порядків повільніше в порівнянні із симетричними.

Інша проблема, пов'язана із забезпеченням конфіденційності пошуку в масивах даних, пов'язана з бажанням унеможливити одержання адміністратором СУБД і сторонніми особами відомостей про те, до яких саме записів (або фрагментів) бази даних здійснювався доступ при кожному конкретному запиті. У закордонній літературі це завдання зветься “Private Information Retrieval” (PIR). Вона особливо актуальна, наприклад, при обробці й зберіганні електронних документів, що містять відомості приватного характеру: фінансові, юридичні, майнові, медичні й інші.

Якщо навіть самі поля бази даних зашифровані, характер і частота запитів до них уже можуть дати зловмисникові деяку непряму інформацію, розголошення якої небажано для власника. Ці завдання до визначеної міри аналогічні виникаючої в телекомунікаційних системах завданню маскування інтенсивності трафіка між вузлами, що, як відомо, вирішується шляхом суцільного заповнення каналу псевдовипадковими послідовностями.

Виходячи зі структурної схеми системи зображеної на рисунку 3, система хмарного сервісу з використанням алгоритму TDEA, працює наступним чином.

Спершу при вході в систему, користувач звертається до блоку розмежування доступу.

Блок розмежування доступу отримує пароль користувача, та звертається до менеджеру паролів, де отримує сеансовий пароль перевірки правильності паролю користувача, та правильності прав доступу користувача, які зберігаються у відповідних зашифрованих базах даних.

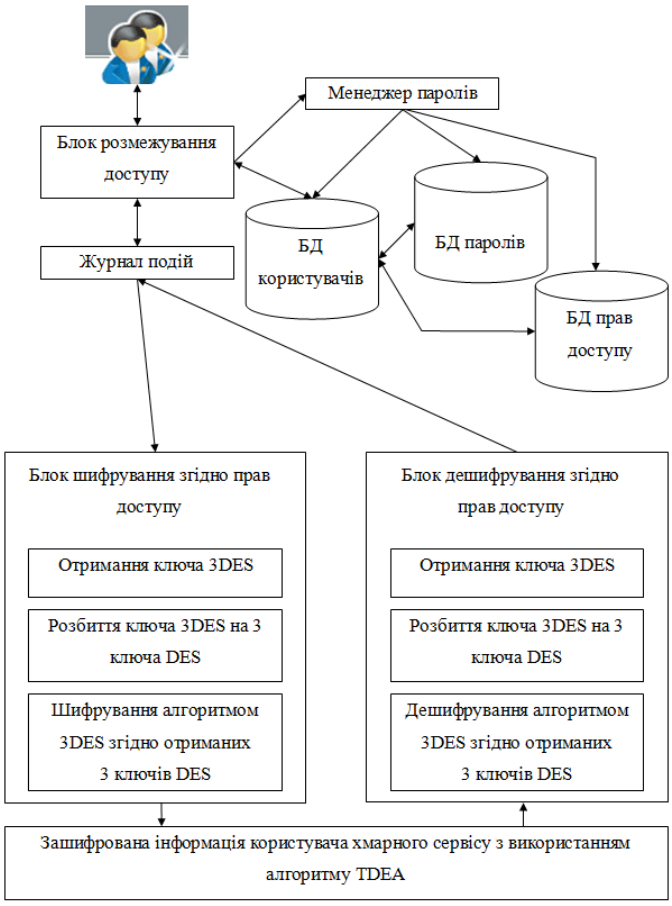


Рисунок 1 – Структурна схема системи

Розмежування цих баз зроблено з метою підвищення стійкості системи зберігання інформації.

Після підтвердження прав доступу, та правильності введеного паролю, користувачеві видається сеансовий ключ 3DES для роботи з інформацією.

У блоці шифрування згідно прав доступу, з отриманого ключа 3DES добуваються 3 ключа алгоритму DES, за допомогою яких й відбувається шифрування інформації алгоритмом 3DES. Процедура дешифрування відбувається аналогічним чином.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів хмарного сервісу з використанням алгоритму TDEA. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем хмарного сервісу з використанням алгоритму TDEA. Досліджена система хмарного сервісу з використанням алгоритму TDEA. На основі отриманих результатів досліджень створена програмна реалізація системи хмарного сервісу з використанням алгоритму TDEA. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання хмарного сервісу з використанням алгоритму TDEA. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності

предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
2. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 3(69). С. 93-98. 2022.
3. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.
4. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.
5. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95
6. Смирнов А., Кузнецов А., Кузнецова Т. «Шумоподобные дискретные сигналы для асинхронных систем кодового разделения радиоканалов». Радиотехника, № 2(205), 175–183. 2021.
7. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.
8. Смірнов, О.А., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю.Усік П.С., «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.
9. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.
10. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.
11. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12..
12. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022..
13. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34..
14. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477..
15. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w.
16. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143.
17. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207..
18. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58..
19. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256..
20. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114..

УДК 004

М.Фадєєв, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ З ВИКОРИСТАННЯМ МУЛЬТИВАРІАНТНОГО ЦЕНТРУ РЕАЛІЗАЦІЇ КРИПТОАЛГОРИТМІВ

У статті розроблено програмне забезпечення, яке призначено для системи з використанням мультиваріантного центру реалізації криптоалгоритмів. Метою розробки є дослідження та програмна реалізація системи з використанням мультиваріантного центру реалізації криптоалгоритмів. Об’єктом дослідження є процес з використанням мультиваріантного центру реалізації криптоалгоритмів. Предметом дослідження є методи з використанням мультиваріантного центру реалізації криптоалгоритмів. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи з використанням мультиваріантного центру реалізації криптоалгоритмів. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп’ютерна інженерія, захист інформації

Постановка проблеми. Останнім часом досить широке поширення одержали різні програмно-апаратні системи захисту інформації, призначені для шифрування даних, що зберігаються на жорстких дисках. Як приклади можна привести відомі PGPdisk і BestCrypt, StrongDisk, Zdisk і Zserver від SecurIT, і т.д.

Крім цього, існують вартим особняком системи шифрування окремих файлів і каталогів, найбільш відома й розповсюджена з яких – EFS (Encrypted File System), що входить до складу MS Windows, починаючи з Windows 2000.

Всі ці системи відрізняються друг від друга способом шифрування, алгоритмами, можливостями й т.д. настільки, що потенційний користувач таких систем часом губиться, і не завжди може зрозуміти, які саме можливості надає та або інша система, і навіщо йому все це потрібно.

Незважаючи на масу розходжень, всі сучасні системи шифрування даних працюють за принципом «прозорого» шифрування. Суть цього принципу полягає в тому, що шифрування даних не є окремою операцією, що повинен виконувати користувач у процесі роботи, а здійснюється одночасно з роботою користувача, автоматично, при читанні й записі даних. Користувач тільки повинен включити шифрування, ввівши при цьому якийсь пароль або ключ шифрування.

У наших українських умовах легко уявити собі ситуацію, коли з комп'ютера, що зберігає конфіденційну інформацію, витягається жорсткий диск і підключається до іншого комп'ютера. А там бажаючий ознайомитися з інформацією знає свій пароль і має права адміністратора. З урахуванням такої можливості покладатися на один тільки пароль досить легковажно.

Разом з тим шифрування безсило проти різних програмних і апаратних закладок, «троянів», мережного злому й інших атак, яким може піддатися працюючий комп'ютер із завантаженими ключами шифрування, коли користувач або адміністратор може просто не знати, що на комп'ютер проникнув сторонній.

У цьому випадку зловмисник тим або іншим способом прикидається легальним користувачем, і одержує доступ до інформації також, як і легальний користувач. На жаль, шифрування не вміє перевіряти права доступу користувачів на доступ до інформації.

Тому, шифрування даних – це лише один з важливих елементів системи інформаційної безпеки, але зовсім не достатній.

Необхідна наявність грамотна настроєної системи розмежування доступу, контролю цілісності операційного середовища, засобів виявлення проникнень, антивірусного й антитроянського захисту й т.д.

У різних системах можуть використовуватися різні способи шифрування даних. Це може бути шифрування на рівні файлів, або шифрування на рівні секторів диска.

Аналізуючи дані з відомих джерел, можна рекомендувати використовувати файл-контейнер для захисту даних окремих користувачів на їхніх комп'ютерах; у цьому випадку навантаження не занадто високе, і падіння продуктивності не так помітно. Більше проста процедура установки й керування такою системою навіть якоюсь мірою компенсує ці недоліки.

Для захисту ж корпоративних серверів, до яких пред'являються більше високі вимоги по продуктивності й надійності, рекомендується використовувати блокове шифрування розділів диска.

У цьому випадку додаткові, до того ж, як правило, разові роботи з інсталяції й настроювання системи захисти цілком виправдуються більше високим ступенем надійності й меншим падінням продуктивності.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи з використанням мультиваріантного центру реалізації криптоалгоритмів.

Мета й завдання дослідження.

Метою роботи є дослідження та програмна реалізація системи з використанням мультиваріантного центру реалізації криптоалгоритмів.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем з використанням мультиваріантного центру реалізації криптоалгоритмів.

– Дослідження системи з використанням мультиваріантного центру реалізації криптоалгоритмів.

– Програмна реалізація системи з використанням мультиваріантного центру реалізації криптоалгоритмів.

Об'єктом дослідження є процес з використанням мультиваріантного центру реалізації криптоалгоритмів.

Предметом дослідження є методи з використанням мультиваріантного центру реалізації криптоалгоритмів.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Опис технологій центру шифрування

Розроблене програмне забезпечення дозволяє виконувати наступні функції захисту інформації:

– проводити шифрування;

– будувати геш-функції масивів інформації;

– реалізовувати алгоритми генерації псевдовипадкових чисел;

– підраховувати контрольну суму файлів.

Розглянемо ці технології захисту інформації більш докладно.

Алгоритми шифрування реалізовані у програмному забезпеченні базуються на симетричних алгоритмах.

Криптографічні алгоритми відіграли вирішальну роль в історії та розвитку криптовалюти. Ці алгоритми постійно вдосконалювалися, щоб забезпечити безпечні та приватні транзакції в цифровій сфері.

Протягом багатьох років було досягнуто значного прогресу, перейшовши від простих методів шифрування до більш складних і витончених алгоритмів. Не можна недооцінювати вплив цих досягнень на зростання криптовалют.

У цьому розділі ми досліджуємо історичний розвиток криптографічних алгоритмів у світі крипто, висвітлюючи ключові моменти, які сформували ландшафт цифрових транзакцій. Розуміючи роль криптографічних алгоритмів, читачі зможуть краще зрозуміти їх значення в поточному розвитку та впровадженні криптовалют.

Приєднуйтесь до нас, коли ми розгадаємо подорож криптографічних алгоритмів та їхній вплив на світ криптовалюти, що постійно розвивається.

Ранні криптографічні алгоритми

На ранніх етапах розвитку криптовалюти криптографічні алгоритми відігравали вирішальну роль у забезпеченні безпеки та цілісності транзакцій. Ці алгоритми лягли в основу криптографічної еволюції, яка сформувала історію криптовалют.

Найпоширенішим криптографічним алгоритмом у цей час був SHA-256 (Secure Hash Algorithm 256-bit), який використовувався при створенні Bitcoin. Алгоритм SHA-256 забезпечив безпечний метод для перевірки цілісності транзакції та запобігання втручанню.

Оскільки криптовалюти набули популярності, виникла потреба в більш досконалих криптографічних алгоритмах. Це призвело до розробки алгоритмів блокчейну, включаючи алгоритми Proof of Work (PoW) і Proof of Stake (PoS). Ці алгоритми запровадили нові механізми для захисту транзакцій і підтримки цілісності блокчейну.

Еволюція криптографічних алгоритмів у сфері криптовалют мала значний вплив на розвиток галузі. Забезпечуючи безпечну та децентралізовану систему для перевірки транзакцій, ці алгоритми вселили довіру до криптовалют. Крім того, вони проклали шлях для розробки нових програм і варіантів використання, таких як смарт-контракти та децентралізовані фінанси.

Введення геш-функцій

Впровадження геш-функцій стало важливою віхою в еволюції криптографічних алгоритмів у сфері криптовалют. Геш-функції, які є математичними алгоритмами, приймають вхідні дані (або повідомлення) і створюють рядок символів фіксованого розміру, відомий як геш-значення або геш-код. Ці функції створені для того, щоб бути швидкими та ефективними, генеруючи унікальний вихід для кожного унікального входу.

У контексті криптовалют використання геш-функцій має кілька важливих наслідків. По-перше, вони забезпечують цілісність транзакцій і даних шляхом створення унікального ідентифікатора для кожного блоку транзакцій. Це дозволяє учасникам мережі перевіряти автентичність даних, не покладаючись на центральний орган. По-друге, геш-функції забезпечують певний рівень конфіденційності та безпеки, унеможливлюючи зворотне проектування початкового введення з геш-значення.

Щоб краще зрозуміти вплив геш-функцій на криптовалюти, давайте розглянемо деякі ключові віхи в криптоіндустрії:

– 2008: Білий документ про біткойн – Сатоші Накамото представляє біткойн, децентралізовану криптовалюту, яка використовує геш-функції для захисту транзакцій. Ця революційна концепція усуває потребу в посередниках і забезпечує однорангові транзакції.

– 2013: Представлення Scrypt – Scrypt, функція отримання ключів, представлена, щоб зробити майнінг більш доступним і стійким до спеціалізованого обладнання. Це заохочує ширшу участь у майнінгу та допомагає підтримувати децентралізований характер криптовалют.

– 2015: Ethereum Blockchain – Ethereum представляє концепцію розумних контрактів, розширюючи використання геш-функцій за межі безпеки транзакцій. Це дозволяє розробляти децентралізовані програми (DApps) і полегшує створення нових токенів і проектів на основі блокчейну.

– 2021: Proof of Stake (PoS) – багато криптовалют переходять від Proof of Work (PoW) до консенсусних алгоритмів PoS, які покладаються на геш-функції для перевірки транзакцій.

Цей перехід зменшує споживання енергії та покращує масштабованість, роблячи криптовалюти більш стійкими та ефективними.

Впровадження геш-функцій мало глибокий вплив на розвиток криптографічних алгоритмів у криптовалютах. Це проклало шлях для створення децентралізованих систем, покращило безпеку та конфіденційність, а також уможливило зростання інноваційних програм. Оскільки криптоіндустрія продовжує розвиватися, геш-функції, безсумнівно, відіграватимуть вирішальну роль у формуванні її майбутнього.

Геш-функції у світі криптовалют служать багатьом важливим цілям, крім безпеки. Ці математичні алгоритми перетворюють дані в рядки фіксованого розміру, які називаються геш-значеннями, забезпечуючи цілісність і автентичність цифрових транзакцій.

Однак їхнє значення виходить за межі безпеки. Геш-функції відіграють важливу роль у майнінгу, який є основою багатьох криптовалют. Вони надають унікальні ідентифікатори для транзакцій і блокувань, уможливлюючи перевірку та перевірку, таким чином гарантуючи легітимність і захист від несанкціонованого доступу.

Крім того, геш-функції знаходять застосування для зберігання даних, цифрових підписів і алгоритмів підтвердження роботи.

Багатогранна природа геш-функцій у криптовалютному ландшафті досліджується в цій статті, проливаючи світло на їхню важливість за межі їх ролі в забезпеченні безпеки.

Підвищення цілісності даних

Використання геш-функцій у криптовалюті відіграє вирішальну роль у підвищенні цілісності даних. Ці математичні алгоритми генерують рядок символів фіксованого розміру, відомий як геш-значення, унікальний для вхідних даних. Застосовуючи геш-функції до крипто-транзакцій, зберігається цілісність даних. Будь-яка зміна даних транзакції призведе до іншого геш-значення, попереджаючи систему про можливе втручання.

Геш-функції також відіграють значну роль у видобутку блокчейнів. Кожен блок у блокчейні містить геш-значення, отримане з геш-значення попереднього блоку, створюючи ланцюжок блоків. Це гарантує, що будь-яка зміна блоку призведе до зміни геш-значення, розриваючи ланцюжок і роблячи весь блокчейн недійсним. Таким чином, геш-функції використовуються для перевірки та захисту даних, що зберігаються в блоках.

Крім ролі в забезпеченні цілісності даних, геш-функції мають різні інші застосування в сфері криптовалют. Вони використовуються, зокрема, у цифрових підписах, зберіганні паролів і створенні адрес. Надійність і ефективність геш-функцій робить їх важливим інструментом у світі криптовалют.

Перевірка автентичності транзакції

Автентичність транзакцій у криптовалюті забезпечується за допомогою геш-функцій. Ці функції відіграють вирішальну роль у перевірці цілісності та дійсності транзакцій шляхом генерації унікальних вихідних даних фіксованої довжини, відомих як геш, коли вони застосовуються до даних транзакцій. Потім цей геш використовується для перевірки автентичності транзакції.

Одним із способів використання геш-функцій для перевірки автентичності транзакцій є використання цифрових підписів. У програмі цифрової валюти цифровий підпис створюється шляхом шифрування гешу транзакції за допомогою закритого ключа відправника. Потім одержувач може розшифрувати підпис за допомогою відкритого ключа відправника, тим самим перевіряючи автентичність транзакції. Цей процес гарантує, що транзакція не була підроблена під час передачі та походить від заявленого відправника.

Щоб додатково проілюструвати важливість геш-функцій для перевірки автентичності транзакцій, у наведеній нижче таблиці висвітлено деякі ключові переваги, які вони надають у додатках цифрових валют:

- Забезпечує цілісність даних.
- Запобігає фальсифікації транзакцій.
- Вмикає безпечні цифрові підписи.
- Сприяє ефективній перевірці транзакцій.

– Підтримує невідмову від транзакцій.

Запобігання подвійним витратам

Використання геш-функцій є вирішальним аспектом у запобіганні подвійним витратам у криптовалюті. Подвійне витрачання означає шахрайську діяльність під час спроби витратити ту саму криптовалюту кілька разів. Геш-функції відіграють важливу роль у забезпеченні цілісності та безпеки криптовалютних транзакцій.

Коли транзакція ініціюється, вона проходить процес, відомий як гешування, де геш-функція застосовується до даних транзакції. Цей процес генерує унікальний результат, який називається геш, який служить цифровим відбитком для транзакції. Потім геш зберігається в блокчейні. Будь-яка зміна даних транзакції, незалежно від того, наскільки мала, призведе до зовсім іншого гешу.

Щоб запобігти подвійним витратам, мережа блокчейну покладається на консенсусні алгоритми, такі як proof-of-work або proof-of-stake, які широко використовують геш-функції. Ці алгоритми перевіряють і підтверджують транзакції шляхом вирішення складних математичних головоломок або за допомогою механізмів на основі ставок. Використовуючи геш-функції в ці алгоритми, мережа гарантує, що кожна транзакція є унікальною, тим самим запобігаючи будь-яким спробам витратити ту саму криптовалюту більше одного разу.

Крім того, геш-функції також використовуються в майнінгу, процесі додавання нових транзакцій до блокчейну. Майнери змагаються, щоб знайти геш, який відповідає певним критеріям, що вимагає значної обчислювальної потужності. Цей процес не тільки захищає мережу, але й запобігає подвійним витратам, забезпечуючи додавання в блокчейн лише дійсних транзакцій.

Забезпечення ефективних процесів майнінгу

Геш-функції відіграють вирішальну роль в оптимізації процесів майнінгу в екосистемі криптовалюти. У контексті криптовалюти майнінг включає перевірку та додавання нових транзакцій до блокчейну, що вимагає значної обчислювальної потужності та має важливе значення для підтримки цілісності та безпеки мережі.

Ефективні процеси майнінгу значною мірою залежать від властивостей геш-функцій. Криптографічні геш-функції, як-от SHA-256, спеціально розроблені, щоб розв’язувати їх обчислювально, але їх легко перевірити. Така конструкція робить їх ідеальними для майнінгу, оскільки це гарантує, що процес не можна легко маніпулювати або підробити.

Майнери використовують геш-функції, щоб змагатися у вирішенні складних математичних головоломок, відомих як алгоритми підтвердження роботи. Перший майнер, який розгадає головоломку, отримує винагороду щойно відкарбованими монетами та комісією за транзакції. Це стимулює майнерів інвестувати в потужне обладнання та конкурувати за швидше вирішення цих головоломок, тим самим підвищуючи ефективність процесу видобутку.

Крім того, геш-функції полегшують створення майнінг-пулів, де кілька майнерів об’єднують свої обчислювальні потужності, щоб підвищити ймовірність вирішення головоломки та отримання винагороди. Таке об’єднання ресурсів забезпечує більш ефективний і послідовний процес видобутку, оскільки майнери колективно вирішують головоломки швидше.

Забезпечення механізмів консенсусу

Включення механізмів консенсусу є важливим аспектом геш-функцій у індустрії криптовалют. Ці механізми відіграють вирішальну роль у перевірці транзакцій і підтримці цілісності мережі.

Роль у перевірці транзакцій

Механізми консенсусу в криптовалюті значною мірою покладаються на геш-функції для підтвердження транзакцій. Геш-функції відіграють вирішальну роль у забезпеченні цілісності та безпеки мережі криптовалют.

Нижче наведено три ключові способи, за допомогою яких геш-функції забезпечують перевірку транзакцій:

1. Цілісність даних: геш-функції генерують унікальні вихідні дані фіксованого розміру для заданого вхідного даних. Застосовуючи геш-функції до даних транзакцій, будь-яка зміна даних призведе до іншого геш-значення. Це дозволяє учасникам мережі перевірити, чи дані транзакції не були підроблені.

2. Перевірка транзакцій: геш-функції використовуються для створення цифрових підписів, які перевіряють автентичність і цілісність транзакцій. Порівнюючи геш-значення транзакції з цифровим підписом, учасники можуть переконатися, що транзакція не була підроблена.

3. Ефективний консенсус: геш-функції використовуються в механізмах консенсусу, таких як підтвердження роботи (PoW) і підтвердження частки (PoS), щоб забезпечити згоду щодо дійсності транзакції. Виконуючи складні обчислення за допомогою геш-функцій, учасники можуть досягти консенсусу щодо порядку та дійсності транзакцій, уможливлюючи децентралізований і недовірливий консенсус.

Вплив на масштабованість мережі

Вплив геш-функцій на масштабованість мережі та ефективну роботу механізмів консенсусу є ключовим аспектом, який слід враховувати в контексті криптовалюти. Геш-функції відіграють важливу роль у створенні консенсусних механізмів, таких як Proof of Work (PoW) і Proof of Stake (PoS), які необхідні для підтримки цілісності та безпеки криптовалютної мережі.

Ці механізми покладаються на геш-функції для підтвердження та перевірки транзакцій, гарантуючи додавання в блокчейн лише дійсних транзакцій. Однак обчислювальна складність геш-функцій може мати наслідки для масштабованості мережі. Зі збільшенням кількості транзакцій та учасників криптовалютної мережі зростають і обчислювальні ресурси, необхідні для механізмів консенсусу. Це потенційно може призвести до вузьких місць і сповільнення часу обробки транзакцій.

Тому оптимізація ефективності геш-функцій має вирішальне значення для забезпечення масштабованості та безперебійної роботи мереж криптовалюти.

Увімкнення функції смарт-контракту

Реалізація функції смарт-контракту в криптовалютах передбачає інтеграцію геш-функцій, які забезпечують цілісність і безпеку процесу виконання контракту. Геш-функції відіграють вирішальну роль у забезпеченні безперебійного виконання смарт-контрактів, надаючи механізми перевірки та підтвердження.

Нижче наведено три ключові способи, за допомогою яких геш-функції забезпечують роботу смарт-контракту:

– Цілісність даних: геш-функції використовуються для перевірки цілісності даних, що зберігаються в розумному контракті. Згенерувавши унікальне геш-значення для даних контракту, можна легко виявити будь-яке підроблення або модифікацію. Це гарантує, що виконання контракту базується на точній і незмінній інформації.

– Умовне виконання: розумні контракти часто вимагають виконання певних умов, перш ніж їх можна буде виконати. Геш-функції дають змогу створити умовне виконання шляхом включення конкретних умов у контракт. Ці умови можуть бути закодовані як геш-значення, і контракт буде виконано, лише якщо геш-значення відповідає попередньо визначеній умові.

– Незмінний код: геш-функції використовуються для створення унікальних ідентифікаторів для смарт-контрактів, відомих як адреси контрактів. Ці адреси походять від коду контракту, що забезпечує незмінність коду. Ця незмінність має вирішальне значення для гарантії того, що логіка та функціональність контракту не можуть бути змінені після розгортання.

Підтримка технології незмінного блокчейну

Геш-функції відіграють вирішальну роль у підтримці незмінності технології блокчейн у криптовалютах. Ці функції забезпечують цілісність даних, генеруючи унікальні геш-коди

для кожного блоку в блокчейні. Цей процес надзвичайно ускладнює зміну минулих транзакцій без виявлення.

Запобігаючи втручанню та шахрайству, геш-функції дозволяють здійснювати ненадійні транзакції, усуваючи потребу в посередниках.

Як наслідок, використання геш-функцій підвищує безпеку та прозорість мережі блокчейн.

Забезпечення цілісності даних

Цілісність даних є фундаментальним аспектом технології блокчейн, оскільки вона забезпечує точність і незмінність інформації, що зберігається децентралізованим способом. Геш-функції відіграють вирішальну роль у підтримці цілісності даних у мережі блокчейн.

Наступні три ключові способи демонструють, як геш-функції забезпечують цілісність даних:

1. Перевірка: геш-функції генерують унікальні геш-значення для кожного блоку даних, діючи як цифровий відбиток. Порівнюючи ці геш-значення, користувачі можуть перевірити цілісність даних і переконатися, що вони не були підроблені.

2. Консенсус: геш-функції використовуються в механізмах консенсусу блокчейн-мереж, таких як підтвердження роботи (PoW) або підтвердження частки (PoS). Завдяки перевірці транзакцій і блоків за допомогою гешування досягається консенсус щодо правильної версії блокчейну, додатково гарантуючи цілісність даних.

3. Ланцюжок блоків: геш-функція, яка використовується в технології блокчейн, створює ланцюжок блоків, причому кожен блок містить геш-значення попереднього блоку. Це гарантує, що будь-яка модифікація блоку вимагатиме перерахунку гешу всіх наступних блоків, що робить фактично неможливим зміну даних без виявлення.

Запобігання втручанню та шахрайству

Геш-функції відіграють вирішальну роль у запобіганні підробці та шахрайству в технології блокчейн. Ці функції приймають вхідні дані, такі як транзакція або блок даних, і генерують вихідні дані фіксованого розміру, відомі як геш-значення. Це геш-значення є унікальним для вхідних даних, тобто навіть незначна зміна вхідних даних призведе до зовсім іншого геш-значення.

Зберігання цих геш-значень у блокчейні гарантує, що будь-яка спроба підробити дані буде негайно виявлена.

Крім того, геш-функції використовуються в процесі майнінгу для забезпечення цілісності блокчейну. Майнери змагаються, щоб знайти конкретне геш-значення, яке відповідає певним умовам. Цей процес робить практично неможливим втручання в минулі транзакції без повторного виконання всіх наступних блоків.

У результаті геш-функції підтримують незмінність і надійність технології блокчейн, роблячи її високостійкою до шахрайства та маніпуляцій.

Увімкнення безнадійних транзакцій

Геш-функції відіграють вирішальну роль у підвищенні безпеки та надійності технології блокчейн. Вони забезпечують безнадійні транзакції, усуваючи потребу в посередниках або перевірених третіх сторонах. Це досягається шляхом надання унікального цифрового відбитка, або геш-значення, для кожної транзакції, забезпечуючи її цілісність і автентичність.

Існує три ключові способи, за допомогою яких геш-функції забезпечують безнадійні транзакції та підтримують незмінність блокчейну:

1. Перевірка: учасники можуть перевірити цілісність транзакцій, порівнюючи геш-значення, згенероване з даних транзакцій, із геш-значенням, що зберігається в блокчейні.

2. Консенсус: геш-функції використовуються в алгоритмах консенсусу, таких як Proof of Work, щоб гарантувати, що всі вузли в мережі погоджуються щодо дійсності транзакцій.

3. Цілісність блокчейну: геш-функції створюють ланцюжок блоків, пов’язуючи геш-значення кожного блоку з геш-значенням попереднього блоку. Це надзвичайно ускладнює зміну минулих транзакцій без виявлення.

Геш-функції відіграють вирішальну роль у підвищенні ефективності процесів майнінгу криптовалюти. Забезпечуючи цілісність даних і запобігаючи подвійним витратам, вони сприяють безперебійній роботі системи. Геш-функції забезпечують швидкий і безпечний спосіб перевірки транзакцій, дозволяючи майнерам перевіряти блоки та конкурувати за винагороду в децентралізованій мережі. Їх впровадження забезпечує надійність і надійність системи криптовалют, роблячи її більш ефективною та безпечною для всіх учасників.

Геш-функції в криптовалюті служать різним цілям, крім забезпечення цілісності даних і запобігання подвійним витратам. Вони мають різноманітні додатки, включаючи створення унікальних ідентифікаторів, створення цифрових підписів і сприяння безпечному спілкуванню між сторонами в децентралізованій мережі.

Геш-функції відіграють вирішальну роль у забезпеченні функції смарт-контрактів у технології блокчейн. Вони забезпечують цілісність і незмінність даних, надають унікальний ідентифікатор для кожного контракту та забезпечують безпечну та ефективну перевірку виконання контракту. Використовуючи геш-функції, мережа блокчейну може перевірити справжність транзакцій і запобігти будь-яким підробкам або несанкціонованим змінам контракту. Крім того, геш-функції допомагають оптимізувати зберігання та пошук даних контракту шляхом створення компактного представлення вмісту контракту. Це дозволяє швидше та ефективніше обробляти смарт-контракти в блокчейні. Загалом геш-функції є важливим компонентом технології блокчейн, що забезпечує надійність і безпеку функціональності смарт-контракту.

Геш-функції відіграють вирішальну роль у підтримці механізмів консенсусу, які використовуються в криптовалютах. Вони надають засоби перевірки цілісності даних транзакцій, гарантуючи, що кожен блок у ланцюжку блоків залишається унікальним, незмінним і стійким до втручання. Застосування геш-функцій посилює безпеку та надійність усієї системи.

Геш-функції мають відомі вразливості та обмеження, які потенційно можуть вплинути на безпеку криптовалют. Ці вразливості включають атаки на зіткнення, атаки на попередні зображення та ймовірність того, що майбутні квантові комп’ютери порушать поточні геш-функції.

Геш-функції в криптовалюті відіграють вирішальну роль, а не просто забезпечення безпеки. Вони пропонують кілька переваг, які підвищують загальну функціональність і надійність екосистеми цифрової валюти. Ці переваги включають:

1. Підвищення цілісності даних: геш-функції гарантують, що дані залишаються недоторканими та незмінними. Вони генерують унікальні ідентифікатори (геш-значення) для кожного фрагмента інформації, що дозволяє легко виявити будь-які зміни чи підробку.

2. Перевірка автентичності транзакцій: геш-функції використовуються для перевірки автентичності транзакцій. Порівнюючи геш-значення транзакції з її відповідним відкритим ключем, користувачі можуть переконатися, що транзакція не була змінена чи підроблена.

3. Запобігання подвійним витратам: геш-функції допомагають запобігти проблемі подвійних витрат, коли користувач намагається витратити ту саму криптовалюту двічі. Генеруючи унікальні геш-значення для кожної транзакції, блокчейн може ідентифікувати та відхиляти будь-які повторювані транзакції.

4. Забезпечення ефективних процесів майнінгу: геш-функції є невід’ємною частиною процесу майнінгу криптовалют, таких як біткойн. Майнери використовують обчислювальну потужність для вирішення складних математичних головоломок, а геш-функції допомагають забезпечити справедливість і безпеку цього процесу.

5. Підтримка механізмів консенсусу: механізми консенсусу, такі як Proof of Work або Proof of Stake, покладаються на геш-функції для досягнення згоди щодо дійсності транзакцій і створення нових блоків. Геш-функції відіграють вирішальну роль у підтримці цілісності та безпеки цих механізмів консенсусу.

6. Увімкнення функцій смарт-контракту: смарт-контракти — це самовиконувані контракти з попередньо визначеними правилами та умовами. Геш-функції використовуються для забезпечення цілісності та безпеки цих контрактів, що робить їх надійними та захищеними від підробки.

7. Полегшення використання незмінної технології блокчейну: геш-функції необхідні для створення незмінності технології блокчейну. Кожен блок у блокчейні містить унікальне геш-значення, яке залежить від даних у блоці. Будь-яка зміна в даних призведе до іншого геш-значення, роблячи очевидним, що блок було змінено.

Пропонуючи ці переваги, геш-функції сприяють довірі, надійності та безпеці всієї екосистеми криптовалюти. Вони є важливими компонентами для забезпечення цілісності транзакцій, блоків і загального функціонування цифрових валют у світі, що постійно розвивається.

Симетричні ключові алгоритми набувають популярності

Алгоритми із симетричним ключем набули значного значення в еволюції криптографічних алгоритмів у сфері криптовалют. Ці алгоритми використовують той самий ключ як для шифрування, так і для дешифрування, що робить їх високоефективними та придатними для різноманітних криптографічних програм.

Зростання популярності алгоритмів із симетричним ключем можна пояснити їх здатністю забезпечувати безпечні та швидкі процеси шифрування та дешифрування. У світі криптовалют, де транзакції потрібно проводити швидко та безпечно, це надзвичайно важливо. Алгоритми симетричного ключа, такі як Advanced Encryption Standard (AES) і Data Encryption Standard (DES), стали кращими виборами для шифрування конфіденційних даних, забезпечуючи конфіденційність і цілісність транзакцій.

Крім того, симетричні ключові алгоритми пропонують масштабованість, забезпечуючи ефективне шифрування та дешифрування великих обсягів даних. Ця масштабованість особливо важлива в контексті криптовалют, де обсяг транзакцій і даних може бути значним. Використовуючи симетричні ключові алгоритми, криптовалютні платформи можуть забезпечити безперебійну роботу своїх систем без шкоди для безпеки.

Криптографія з відкритим ключем революціонізує криптовалюту

Криптографія з відкритим ключем зробила революцію у світі криптовалют, змінивши спосіб проведення та захисту транзакцій у цифрових валютах. Цей криптографічний алгоритм запровадив революційний підхід, який усуває обмеження алгоритмів із симетричним ключем.

На відміну від алгоритмів із симетричним ключем, які покладаються на один ключ як для шифрування, так і для дешифрування, у криптографії з відкритим ключем використовується пара різних ключів: відкритий ключ і закритий ключ. Відкритий ключ широко поширений і використовується для шифрування, тоді як закритий ключ, відомий лише власнику, використовується для дешифрування та цифрових підписів.

Однією з ключових переваг криптографії з відкритим ключем є її здатність забезпечувати безпечний зв’язок і цифрові підписи без необхідності довіреної третьої сторони. Ця система дозволяє здійснювати безпечні транзакції, які можна перевірити, оскільки закритий ключ можна використовувати для підтвердження права власності без розкриття конфіденційної інформації.

Завдяки запровадженню криптографії з відкритим ключем криптовалюти стали більш безпечними, прозорими та стійкими до шахрайства. Він усунув необхідність безпечного обміну ключами та забезпечив безпечний метод для цифрових підписів. Цей прогрес значно підвищив безпеку та функціональність криптовалют, проклавши шлях для подальших інновацій у цій галузі.

Крім того, криптографія з відкритим ключем відкрила нові можливості для конфіденційності та анонімності в транзакціях з криптовалютою. Тепер користувачі можуть використовувати унікальні пари ключів для кожної транзакції, гарантуючи, що їх

ідентифікаційні дані залишаються прихованими, зберігаючи при цьому цілісність процесу транзакції.

Поточний стан і майбутнє криптографічних алгоритмів у криптовалюті

Поточний стан і майбутні перспективи криптографічних алгоритмів у криптовалюті зазнали значної еволюції. Важливість безпечних і ефективних криптографічних алгоритмів неможливо переоцінити, оскільки криптоіндустрія продовжує розвиватися. Щоб забезпечити безпечні транзакції та цифрові підписи, більшість криптовалют наразі покладаються на асиметричні криптографічні алгоритми, такі як криптографія з еліптичною кривою (ECC) і RSA (Rivest-Shamir-Adleman).

ECC набув популярності завдяки своїй здатності забезпечувати той самий рівень безпеки, що й RSA, але з меншими розмірами ключів, що робить його ефективнішим з точки зору обчислень. Ця ефективність особливо важлива в контексті технології блокчейн, де підтримка децентралізованої мережі залежить від ефективності.

Крім ECC і RSA, інші криптографічні алгоритми, такі як SHA-256 (Secure Hash Algorithm 256-bit) і HMAC (Keyed-Hash Message Authentication Code), використовуються для цілісності даних і автентифікації. Ці алгоритми гарантують, що дані залишаються незмінними, і дозволяють сторонам, залученим у транзакцію, перевірити автентичність даних.

Дивлячись у майбутнє, розробка квантових комп’ютерів становить потенційну загрозу для традиційних криптографічних алгоритмів. Квантові комп’ютери мають здатність зламати типові алгоритми шифрування, роблячи їх неефективними. Щоб вирішити цю проблему, дослідники вивчають використання квантово-стійких криптографічних алгоритмів, таких як криптографія на основі решітки та багатоваріантна криптографія, для захисту криптовалют у постквантову еру.

в криптовалютах, були менш безпечними та простішими порівняно з передовими методами шифрування, які використовуються сьогодні. Еволюція криптографічних алгоритмів призвела до розробки більш складних методів, які забезпечують безпеку та цілісність криптовалют. Ці досягнення посилили захист конфіденційних даних і транзакцій в екосистемі криптовалют. Сучасні криптографічні алгоритми, які використовуються в криптовалютах, включають сильніші алгоритми шифрування, геш-функції та схеми цифрового підпису, що робить їх більш стійкими до атак і підробки. Ці посилені заходи безпеки забезпечують вищий рівень довіри та надійності в системі криптовалюти, захищаючи цілісність і конфіденційність цифрових активів.

Впровадження геш-функцій у криптографічних алгоритмах у криптовалюті мало значний вплив як на безпеку, так і на ефективність цих алгоритмів. Геш-функції відіграють вирішальну роль у перевірці цілісності даних і захисту від підробки. Крім того, вони сприяють прискоренню обчислень, підвищуючи загальну ефективність криптографічних алгоритмів, які використовуються в криптовалюті.

Алгоритми з симетричним ключем отримали провідне місце в еволюції криптографічних алгоритмів у криптовалюті завдяки своїй ефективності та простоті. Ці алгоритми дозволяють виконувати швидкі процеси шифрування та дешифрування, що робить їх придатними для захисту великих обсягів даних у децентралізованій мережі. Нижче наведено переваги алгоритмів симетричних ключів у криптовалюті:

1. Ефективність: алгоритми симетричного ключа можуть швидко шифрувати та дешифрувати дані, забезпечуючи ефективні транзакції та безпечний зв’язок у мережі криптовалюти.

2. Простота: ці алгоритми відносно легко реалізувати та зрозуміти, що зменшує складність криптографічних операцій у системах криптовалюти.

3. Масштабованість: симетричні ключові алгоритми можуть обробляти великі обсяги даних, завдяки чому вони добре підходять для захисту транзакцій і підтримки цілісності блокчейну.

4. Надійність: ці алгоритми забезпечують надійний захист від несанкціонованого доступу та втручання, забезпечуючи безпеку та цілісність транзакцій криптовалюти.

5. Сумісність. Алгоритми симетричного ключа широко підтримуються та сумісні з різними платформами та пристроями, що дозволяє бездоганно інтегрувати в системи криптовалюти.

Криптографія з відкритим ключем зробила революцію в безпеці та транзакційних можливостях криптовалют, запровадивши систему асиметричних ключів. Цей інноваційний підхід забезпечив більш безпечний спосіб шифрування та дешифрування даних, забезпечуючи конфіденційність, цілісність і автентичність цифрових транзакцій. Замість того, щоб покладатися на один ключ як для шифрування, так і для дешифрування, криптографія з відкритим ключем використовує пару математично пов’язаних ключів: відкритий та закритий ключ. Відкритий ключ вільно розповсюджується та використовується для шифрування, тоді як закритий ключ зберігається в таємниці та використовується для дешифрування. Ця система забезпечує безпечний зв’язок і перевірку між сторонами без необхідності довіреної третьої сторони. Завдяки криптографії з відкритим ключем можна надійно зберігати криптовалюти, впевнено проводити транзакції, а загальна безпека цифрової економіки значно підвищується.

Масштабованість, квантова стійкість і пошук правильного балансу між безпекою та зручністю використання є одними з поточних проблем у розробці криптографічних алгоритмів для криптовалют. У майбутньому галузь може досліджувати нові алгоритми, вдосконалювати функції конфіденційності та вирішувати проблеми з регулюванням для подальшого прогресу в цій галузі.

Еволюція криптографічних алгоритмів мала значний вплив на розвиток і впровадження криптовалют. Спочатку використовувалися базові методи шифрування, але з часом були представлені більш складні та безпечні алгоритми.

Ці криптографічні алгоритми постійно покращують безпеку та конфіденційність цифрових транзакцій. Оскільки криптовалюти продовжують розвиватися, майбутнє криптографічних алгоритмів містить величезний потенціал для подальшого прогресу в забезпеченні цілісності та конфіденційності цих цифрових валют.

Розробка структурної схеми

На рисунку 1 зображено структурну схему розробленого програмного забезпечення.

Програмне забезпечення структурно складається з наступних блоків:

– Головний модуль програми.
– База даних алгоритмів шифрування.
– База даних геш-функцій.
– База даних алгоритмів генерації псевдовипадкових чисел (ГПВЧ).
– База даних алгоритмів підрахунку контрольних сум (CRC).
– Шифрування файлів та папок.
– Шифрування даних на змінних носіях інформації.
– Шифрування даних, які передаються по мережі.
– Шифрування листів e-mail.
– Створення зашифрованих архівів, що саморозпаковуються.
– Генератор псевдовипадкових чисел.
– Перевірка цілісності та автентичності файлів.
– Створення віртуального зашифрованого диску.
– Створення паролю.

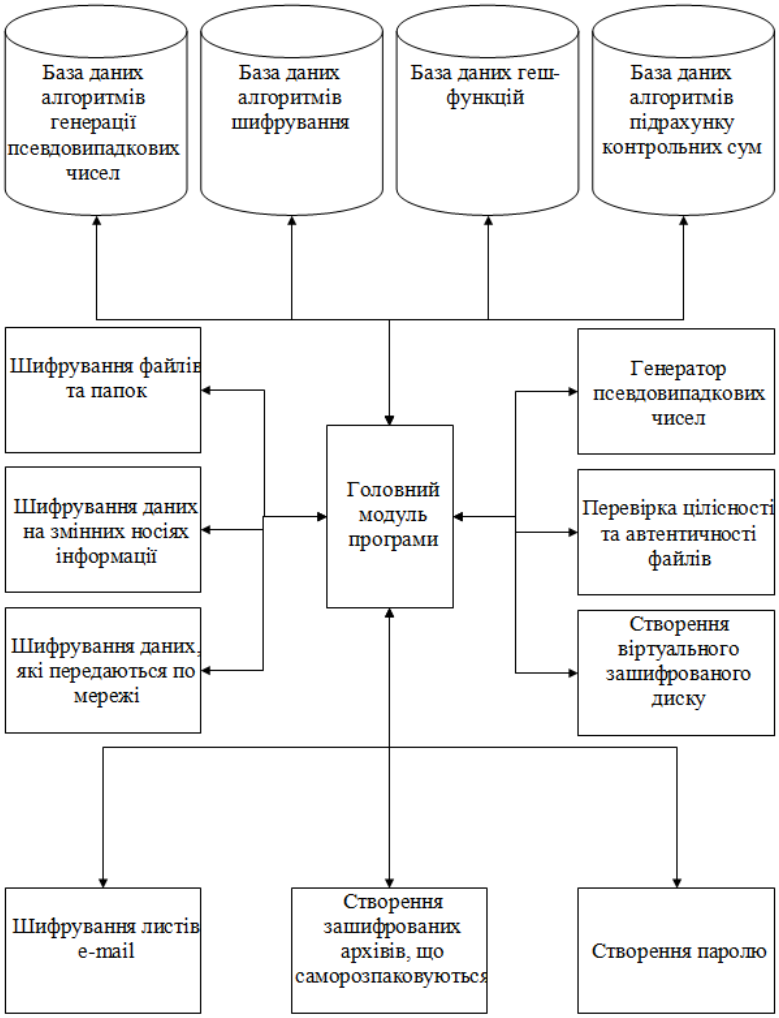


Рисунок 1 – Структурна схема розробленого програмного забезпечення

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів з використанням мультиваріантного центру реалізації криптоалгоритмів.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем з використанням мультиваріантного центру реалізації криптоалгоритмів.

– Досліджена система з використанням мультиваріантного центру реалізації криптоалгоритмів.

– На основі отриманих результатів досліджень створена програмна реалізація системи з використанням мультиваріантного центру реалізації криптоалгоритмів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання з використанням мультиваріантного центру реалізації криптоалгоритмів.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143.

2. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207..
3. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58..
4. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256..
5. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114..
6. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346..
7. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131..
8. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14..
9. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84..
10. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587..
11. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136..
12. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379..
13. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43..
14. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645..
15. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.,.
16. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407..
17. Смирнов А.А. Анализ и сравнительное исследование перспективных направлений развития цифровых телекоммуникационных систем и сетей / А.А.Смирнов, В.В.Босько, Е.В.Мелешко // Системи обробки інформації. – Х.: ХУ ПС, 2008. – Вип.7(74). – С.120-123.
18. Смирнов А.А. Усовершенствование метода управления очередями в многопротокольных узлах телекоммуникационной сети / А.А.Смирнов, Е.В.Мелешко // Збірник тез та доповідей другої всеукраїнської науково-практичної конференції «Системний аналіз. Інформатика. Управління». Запоріжжя. Тези доповідей. Запоріжжя: КПУ, 2011.
19. Смирнов С.А. Метод безопасной маршрутизации метаданных в облачные антивирусные системы / А.К. Дидык, С.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.
20. Смирнов С. А. Сравнительные исследования математических моделей технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Абу Таам Гани, А. А. Смирнов, А. В. Коваленко, С. А. Смирнов // Системи обробки інформації: зб. наук. праць. – Х.: ХУПС, 2014. – Вип. 9(125). – 105-110.

УДК 004

Б.Федоров, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЛЯ ПРОТИДІЇ ДЕКОМПІЛЯЦІЇ КОДУ

У статті розроблено програмне забезпечення, яке призначено для системи для протидії декомпіляції коду. Метою розробки є дослідження та програмна реалізація системи для протидії декомпіляції коду. Об'єктом дослідження є процес для протидії декомпіляції коду. Предметом дослідження є методи для протидії декомпіляції коду. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи для протидії декомпіляції коду. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерна інженерія, протидія декомпіляції коду

Постановка проблеми.

На сучасному етапі для протидії декомпіляції коду використовується найпоширеніша методика автоматичного захисту програм від аналізу на основі технології віртуалізації машинного коду. Дана методика ставить у відповідність кожній машинній інструкції одну або кілька інструкцій автоматично згенерованого віртуального процесора, називаних байт-кодом або псевдокодом. У тіло програми, що захищається, вбудовується захищений від аналізу інтерпретатор, завданням якого є виконання згенерованого на етапі захисту байта-коду. Основним недоліком такого підходу є низька швидкість роботи захищеного в такий спосіб коду. Більше того, у більшості випадків все-таки можливе створення автоматичного декомпілятора псевдокоду у вихідний машинний код.

У зв'язку із цим для підвищення стійкості до автоматичних засобів деактивації захисту й забезпечення більш високої швидкодії захищеної програми в порівнянні з існуючими методиками необхідне створення нових підходів. В основі їх лежить принцип програмного «чорного ящика», реалізація якого припускає використання технологій заплутування коду й даних програми або обфускації

Дана обставина визначає актуальність розробки методик обфускації коду й даних програми, що не вимагають для своєї роботи вихідного коду програми й забезпечуючих стійкість стосовно автоматичних утиліт деактивації захисту.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи для протидії декомпіляції коду.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи для протидії декомпіляції коду.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем для протидії декомпіляції коду.
- Дослідження системи для протидії декомпіляції коду.
- Програмна реалізація системи для протидії декомпіляції коду.

Об'єктом дослідження є процес для протидії декомпіляції коду.

Предметом дослідження є методи для протидії декомпіляції коду.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу

Обфускація коду – це техніка, яку використовують автори зловмисного програмного забезпечення та інші зловмисники, щоб приховати справжні наміри свого коду та уникнути виявлення програмним забезпеченням безпеки.

Обфускація коду – це процес ускладнення розуміння, аналізу та зворотного проектування програмного коду. Це техніка, яка використовується авторами зловмисних програм та іншими зловмисниками, щоб приховати справжні наміри свого коду та уникнути виявлення програмним забезпеченням безпеки.

У цій статті ми розглянемо різні прийоми та методи, які використовуються для обфускації коду, а також те, як це зробити.

Як працює обфускація коду?

Обфускація коду працює шляхом перетворення вихідного коду у форму, яку важко зрозуміти й проаналізувати. Цього можна досягти за допомогою різних методів, включаючи шифрування, упаковку коду та обфускацію потоку керування.

Мета обфускації коду полягає в тому, щоб аналітикам і програмному забезпеченню безпеки було важко зрозуміти функціональність коду, що ускладнює його виявлення та видалення, а також робить код більш стійким до зворотного проектування.

Хоча обфускація може стримувати випадкові спроби зрозуміти код, важливо зазначити, що рішучі та досвідчені зловмисники все одно можуть реконструювати обфускований код, витративши достатньо зусиль і часу.

Обфускація – це лише один із рівнів захисту, і на неї не слід покладатися виключно для захисту критичних аспектів системи.

Методи обфускації коду

Методи обфускації коду передбачають перетворення коду, щоб зробити його більш складним для розуміння, зворотне проектування або підробку. Ось кілька поширених прийомів:

1. Шифрування рядків

Ця техніка передбачає шифрування рядків у коді, щоб аналітикам було важко зрозуміти передбачувані дії коду.

Наприклад, у зразку зловмисного програмного забезпечення, який шифрує рядок «Видалити всі файли», аналітикам буде важко зрозуміти намічені дії зловмисного програмного забезпечення без попереднього розшифрування рядка.

2. Обфускація потоку керування

Ця техніка передбачає використання складних розгалужених і циклічних структур у коді, щоб аналітикам було важко зрозуміти потік керування кодом.

Наприклад, зразок зловмисного програмного забезпечення, який використовує кілька вкладених операторів if-else і кілька операторів goto, ускладнить аналітикам розуміння потоку керування шкідливим програмним забезпеченням.

3. Антиналагодження

Цей метод передбачає використання різних методів для виявлення та запобігання налагодженню коду, що ускладнює аналітикам розуміння поведінки коду.

Наприклад, аналітикам буде важко проаналізувати та зрозуміти зразок зловмисного програмного забезпечення, який перевіряє наявність налагоджувача та завершує роботу, якщо його виявлено.

4. Упаковка коду

Ця техніка передбачає стиснення та шифрування коду, щоб аналітикам було важко видобувати та аналізувати функціональні можливості коду.

Наприклад, зразок зловмисного програмного забезпечення, який використовує пакувальник UPX для стиснення та шифрування коду, аналітикам буде важко витягнути та проаналізувати функціональність коду.

5. Введення коду

Ця техніка передбачає введення коду в законні процеси або системні бібліотеки, щоб аналітикам було важко виявити та ізолювати код.

Наприклад, аналітикам буде важко виявити та виділити код зловмисного програмного забезпечення, яке впроваджується в процес explorer.exe.

6. Поліморфізм

Ця техніка передбачає постійну модифікацію коду, щоб уникнути систем виявлення на основі сигнатур.

Наприклад, системам виявлення на основі сигнатур буде важко виявити зразок шкідливого програмного забезпечення, який генерує новий код під час кожного запуску.

7. Метаморфізм

Ця техніка є більш просунутою формою поліморфізму, яка передбачає використання методів генерації коду для створення кількох версій зловмисного програмного забезпечення, які мають різні коди, але виконують однакові шкідливі дії.

Наприклад, системам виявлення на основі сигнатур буде важко виявити зразок шкідливого програмного забезпечення, який створює новий код для кожного нового зараження.

8. Безфайлове шкідливе програмне забезпечення

Цей метод передбачає використання методів на основі пам’яті для запуску зловмисного програмного забезпечення без запису у файлову систему, що ускладнює аналітикам виявлення та аналіз зловмисного програмного забезпечення.

Наприклад, аналітикам буде важко виявити та проаналізувати зразок шкідливого програмного забезпечення, який повністю працює в пам’яті та не залишає жодних слідів у файловій системі.

9. Перейменування функцій API

Перейменування функцій API – це стратегія, спрямована на те, щоб зробити код зловмисного ПЗ більш стійким до аналізу шляхом зміни імен функцій, які він використовує для взаємодії з операційною системою. Це частина ширшого набору методів, які використовуються для приховування справжньої природи та призначення шкідливого коду.

10. Стиснення коду

Стиснення коду – це техніка, яка використовується в обфускації коду для зменшення розміру виконуваного файлу шляхом стиснення коду. Це може ускладнити аналіз і зворотне проектування коду.

Які приклади обфускації?

Щоб ви зрозуміли, як працює обфускація коду та як автори зловмисного програмного забезпечення використовують такі методи або техніки. Нижче наведено кілька прикладів обфускації коду, які допоможуть вам краще його зрозуміти.

Одним із прикладів обфускації коду є використання шифрування рядків у шкідливих програмах. Цей метод передбачає шифрування рядків у коді зловмисного програмного забезпечення, що ускладнює аналітикам розуміння намічених дій зловмисного програмного забезпечення.

Наприклад, у зразку зловмисного програмного забезпечення, який шифрує рядок «Видалити всі файли», аналітикам буде важко зрозуміти намічені дії зловмисного програмного забезпечення без попереднього розшифрування рядка.

Іншим прикладом є використання методів захисту від помилок у зловмисному програмному забезпеченні. Цей метод передбачає використання різних методів для виявлення та запобігання налагодженню коду зловмисного програмного забезпечення, що ускладнює аналітикам розуміння поведінки зловмисного програмного забезпечення.

Наприклад, аналітикам буде важко проаналізувати та зрозуміти зразок зловмисного програмного забезпечення, який перевіряє наявність налагоджувача та завершує роботу, якщо його виявлено.

Які є деякі інструменти обфускації коду?

Існує декілька комерційних інструментів із відкритим вихідним кодом для обфускації коду, таких як ConfuserEx, Skater.NET Obfuscator і Crypto Obfuscator.

Ці інструменти можна використовувати для шифрування рядків, упаковки коду та

виконання обфускації потоку керування. Важливо зазначити, що хоча ці інструменти можна використовувати для обфускації коду, вони також можуть використовуватися зловмисниками, щоб приховати справжні наміри свого коду.

Обфускація коду – це техніка, яка використовується для ускладнення розуміння, аналізу та зворотного проектування програмного коду. Він використовується авторами зловмисного програмного забезпечення та іншими зловмисниками, щоб приховати справжні наміри свого коду та уникнути виявлення програмним забезпеченням безпеки.

Обфускацію коду можна досягти за допомогою різноманітних технік і методів, включаючи шифрування, упаковку коду та обфускацію потоку керування.

Інструменти обфускації доступні, щоб допомогти розробникам захистити свій код, але важливо розуміти, що ці інструменти також можуть використовуватися зловмисниками, щоб приховати справжні наміри свого коду.

Обфускація може ускладнити розуміння коду, вона не забезпечує справжнього шифрування чи надійної безпеки. Обфускація більше схожа на маскування – вона просто перетворює код, щоб він став менш читабельним, але основна логіка все ще присутня у формі, яку можна переробити.

Якщо ваша головна турбота про безпеку, покладатися лише на обфускацію недостатньо. Фактично, багато експертів з безпеки вважають обфускацію відносно слабкою формою захисту. Рішучі зловмисники, які володіють потрібними інструментами та навичками, все одно можуть перепроектувати обфускований код.

Для більшої безпеки краще використовувати шифрування. Шифрування передбачає перетворення даних таким чином, що лише авторизовані сторони можуть скасувати перетворення. У контексті коду це може включати шифрування конфіденційних частин коду та їх дешифрування під час виконання.

Обфускація коду в React Native

Є кілька причин, чому ви можете захотіти обфускати свій код React Native:

– **Захист інтелектуальної власності:** обфускація вашого коду ускладнює іншим людям викрасти ваші ідеї або скопіювати вашу роботу.

– **Запобігайте зворотному проектуванню:** обфускація може ускладнити зловмисникам зворотне проектування вашої програми та виявлення вразливостей.

– **Покращення безпеки:** обфускований код складніше використовувати, оскільки зловмисникам буде важче зрозуміти, як він працює.

Як обфускати рідний код React

Існує кілька різних способів обфускати рідний код React. Деякі з найпопулярніших методів включають:

– **Метод плагіна:** ви можете використовувати плагін для обфускації свого коду React Native. Доступно кілька різних плагінів, наприклад:

– obfuscator-io-metro-plugin

– @smartface/obfuscator-io-metro-plugin

– react-native-obfuscating-transformer

– **ProGuard:** ProGuard – це безкоштовний інструмент із відкритим кодом, який можна використовувати для обфускації коду Java. Він включений до Android SDK і може використовуватися для обфускації коду React Native, написаного на Java.

– **R8:** R8 – це новіший інструмент, який також включено в Android SDK. Він призначений для заміни ProGuard і пропонує ряд переваг, включаючи покращену продуктивність і підтримку новіших версій Java.

– **Hermes:** Hermes – це механізм JavaScript, розроблений Facebook. Він розроблений, щоб бути більш ефективним і безпечним, ніж механізм JavaScript за замовчуванням у React Native. Hermes містить вбудований обфускатор, який можна використовувати для обфускації коду React Native, написаного на JavaScript.

– **DexGuard:** DexGuard – це комерційний обфускатор, спеціально розроблений для програм Android, який пропонує низку розширених функцій.

– **iXGuard:** iXGuard – це комерційний обфускатор, спеціально розроблений для додатків iOS, який пропонує ряд розширених функцій.

Сьогодні я зосереджуся на методі плагінів для обфускації нашого коду в React Native. Я поясню кожен крок, як ми можемо обфускати наш код за допомогою obfuscator-io-metro-plugin.

Обфускація рідного коду React за допомогою плагіна Metro

Щоб використовувати плагін Metro Obfuscator.io, вам потрібно буде встановити його з npm. Ви можете зробити це, виконавши таку команду:

npm i -D плагін obfuscator-io-metro

Після встановлення плагіна вам потрібно буде додати його до файлу конфігурації Metro. Ви можете зробити це, додавши такий код до свого metro.config.jsфайлу:

```
const jsoMetroPlugin = require ("obfuscator-io-metro-plugin")(
  {
    // для цих опцій шукайте параметри бібліотеки javascript-obfuscator зверху url
    compact: false,
    sourceMap: false, // вихідна карта, згенерована після обфускації, не є корисною правильно тепер використовуйте значення за замовчуванням, тобто false
    controlFlowFlattening: true,
    controlFlowFlatteningThreshold: 1,
    numbersToExpressions: true,
    simplify: true,
    stringArrayShuffle: true,
    splitStrings: true,
    stringArrayThreshold: 1,
  },
  {
    runInDev: false  /* необов’язково */,
    logObfusc atedFiles: правда  /* необов’язкові згенеровані файли будуть розташовані в./.jso */,
  }
);
модуль. exports = {
  transformer: {
    getTransformOptions: async () => ({
      transform: {
        experimentalImportSupport: false,
        inlineRequires: false,
      },
    }),
  },
  ...jsoMetroPlugin,     /*додайте цей рядок у свій попередній модуль після визначеного вище */
 };
```

Як перевірити, чи працює обфускація

Після того, як ви успішно налаштували плагін Obfuscator.io Metro, ви можете перевірити, чи код обфускований чи ні, створивши збірку за допомогою assembleReleaseабо bundleReleaseабо IPA з Xcode і перевіривши код у такому місці:

/. jso

Якщо код у цьому місці обфускований, плагін працює правильно.

Обфускація коду є цінним інструментом, який може допомогти вам захистити вашу інтелектуальну власність, запобігти зворотній інженерії та покращити безпеку вашої програми React Native. Однак важливо ретельно зважити всі «за» і «проти», перш ніж

вирішити, чи варто обфускати код.

Розробка структурної схеми

На рисунку 1 зображена структурна схема програмного забезпечення, яке реалізує процес обфускації коду.

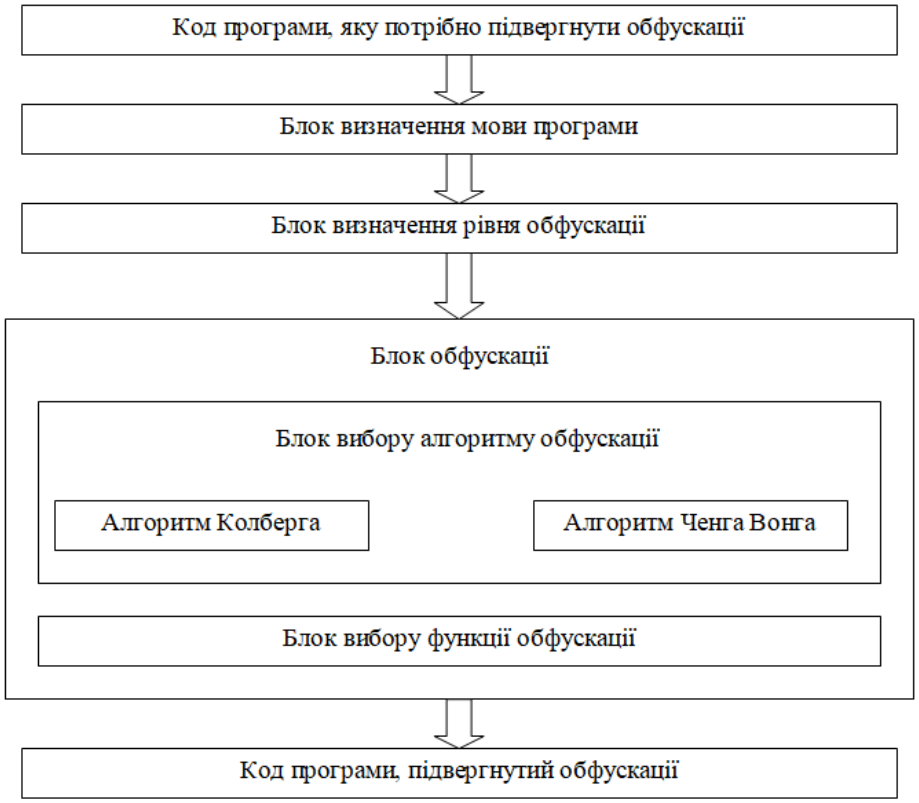


Рисунок 1 – Структурна схема системи

Обфускація коду Java/Kotlin

Обфускацію коду Java/Kotlin надзвичайно легко деобфускати та провести зворотне проектування. По-перше, конфіденційну частину коду ніколи не слід писати цими мовами програмування. Якщо з якоїсь причини це єдиний спосіб дій, тоді найкраще захистити цю чутливу частину коду за допомогою спеціального рішення, а не покладатися лише на обфускацію.

Як найкраща практика для розробки високовартісних програм, слід використовувати мови, які компілюються безпосередньо для складання, наприклад C/C++/Rust тощо.

Конфіденційний код: рівень бізнес-логіки програми, який обробляє/надсилає/отримує конфіденційну інформацію про користувачів або процеси, які мають велике значення для постачальника послуг.

Обфускація – це чудово, але вона приносить побічні ефекти, коли застосовується до всієї програми

Багато разів перетворення, які застосовуються для отримання заплутаного коду, означають, що:

a) Рекламний код буде введено в програму.

b) Буде створено додаткові шари тупикового циклу.

c) Буде введено додаткові символи та значення для перенаправлення коду та багато іншого.

У цих випадках дуже поширеним результатом є те, що програма має тенденцію до збільшення розміру. Це також може означати, що продуктивність програми також знижується, що є червоним прапорцем з точки зору програмування та досвіду користувача. Крім того, існує ймовірність «зламання» або збою програми.

Повноцінні програми, до яких застосовується обфускація, ускладнюють розробнику

виявлення помилок і їх виправлення. Для продовження розробки їм знадобиться певне технологічне рішення відображення, яке допоможе поєднати проблемну область у заплутаному коді з вихідним простим кодом.

Код додатка (повний додаток; написаний на Java/Kotlin), який обфускований за допомогою сучасних інструментів, можна легко деобфускатувати та декомпілювати за допомогою таких інструментів, як інструмент JEB. Для нативного коду такі інструменти, як Ghidra або IDA Pro, можна використовувати для досягнення тих самих результатів. Подивіться на декомпільований код (схожий на вихідний код) і зрештою виконайте маніпуляції з кодом. Є відео під назвою «JEB в дії», у яких новачок може дослідити класи та визначити логіку, а також статті, у яких обфусковані програми розглядаються за допомогою таких інструментів, як інструмент JEB. Важливо розуміти, що мета полягає в тому, щоб запобігти маніпулюванню програмою. Це вимагає додаткових кроків і більш прямого підходу до обфускації.

На додаток до попереднього пункту, обфускований двійковий файл не повністю захищений від зворотного проектування. Справжній захист від зворотного проектування реалізується або централізовано, наприклад, ретрансляція ресурсів програми з центральним сервером для перевірки шаблонів послідовності інструкцій, системних викликів тощо, або децентралізовано, наприклад, безпечне вбудовування важливої інформації в захищену частину програми та впровадження захисних кодів і контрольних сум над важливою частиною програми, що ускладнює проникнення зловмисника.

Де обфускація справді має значення?

Обфускація має значення в чутливій частині коду, де обробляється важлива інформація. Двійковий блок, який відповідає за обробку конфіденційних даних, має бути частиною, де потрібно реалізувати обфускацію та інші функції безпеки, щоб повністю захистити конфіденційні дані та всі процеси навколо них. Навіть у цьому випадку безпека через невідомість (обфускацію) відіграє лише невелику роль у збереженні інформації. Це може викликати запитання з точки зору зловмисника, якщо він бачить лише певну частину коду обфускованою – чи не приклав би він всю свою енергію та зосередився на розумінні та деобфускації цієї частини коду? Відповідь: так, він буде. І саме тому важливо впроваджувати набагато більше з точки зору безпеки (зверніться до розділу нижче, щоб отримати докладнішу інформацію), щоб досягти рівня, який буде займати зловмисника протягом тривалого часу в надії, що він зрештою дасть вгору.

Елементи, які мають набагато більший пріоритет, ніж обфускація, у безпеці програми

Щоб отримати механізми безпеки, які максимально відбивають зловмисників, мають бути інші механізми. Наприклад, захист від використання програми в середовищі підвищених привілеїв, як-от рутований або зламаний телефон, емулятори, фреймворки підключення, такі як FRIDA або Xposed, налагоджений телефон/додаток тощо.

Крім того, зв’язок між програмою та серверами має бути захищений від перехоплення та атак типу Man-in-the-Middle (MitM). Це означає, що конфіденційні програми не можуть покладатися лише на безпеку, яку пропонують TLS або інші публічні протоколи, і повинні мати більше рівнів безпеки, наприклад шифрування корисного навантаження та закріплення сертифіката. Захист ключів API – таким чином захист серверної частини від доступу невідомих сторін є не менш важливим.

Зберігання інформації в зашифрованому вигляді в пісочниці мобільного додатка та можливість «самознищення» цієї бази даних із пам’яті після виявлення атаки готує програму до найгіршого та захищає у разі атаки. Подібним чином захист і шифрування активів, які входять до складу програми, таких як сертифікати сервера для TLS, важливі для збереження цілісності процесів і уникнення масштабованих атак (коли один екземпляр програми на пристрої зламано, інші екземпляри не впливають на це).).

Сильна зовнішня прив’язка довіри також важлива для безпечної атестації примірника програми, який запитує ініціалізацію – це підвищує довіру постачальника послуг до програми, яка намагається підключитися до серверної частини служби для виконання бізнес-

логіки. Забезпечення додаткового рівня довіри на додаток до рівнів, вбудованих у мобільний додаток.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів для протидії декомпіляції коду. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем для протидії декомпіляції коду. Досліджена система для протидії декомпіляції коду. На основі отриманих результатів досліджень створена програмна реалізація системи для протидії декомпіляції коду. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання для протидії декомпіляції коду. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Kovalenko Oleksandr Qualitative risk analysis of software development / Oleksandr Kovalenko, Jamil Al-Azzeh, Oleksii Smirnov, Anna Kovalenko, Serhii Smirnov // Asian Journal of Information Technology. – Volume 17 Issue 3. – Medwell Journals. – 2018. – P. 218-230. ISSN: 1682-3915. URL: http://medwelljournals.com/abstract/?doi=ajit.2018.218.230 Doi: ajit.2018.218.230
2. Kovalenko Oleksandr, The mathematical model of the testing technology for DOM XSS vulnerabilities / O. Kovalenko, O. Smirnov, A.Kovalenko, S. Smirnov, V. Vialkova // Scientific & practical cyber security journal (SPCSJ) Volume 2 Issue 1, P. 22-28. Georgia. Tbilisi. Scientific Cyber Security Association (SCSA), 2018 ISSN: 2587-4667. URL: https://journal.scsa.ge/wp-content/uploads/2018/12/04-3-o.kovalenko-a.kovalenko-o.smirnov-s.smirnov-v.vialkova.pdf
3. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
4. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
5. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
6. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
7. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
8. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
9. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
10. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
11. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
12. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).
13. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019. (Scopus).
14. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629. (Scopus).
15. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 873-884. (Scopus).

16. тСмирнов А.А., Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений. Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.
17. Смирнов А.А., Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения. Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.
18. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 3(69). С. 93-98. 2022.
19. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.
20. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.

УДК 004

В.Шевченко, магістр гр. КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПРОТОКОЛІВ СТЕКУ TCP/IP У ХМАРНИХ СЕРВІСАХ

У статті розроблено програмне забезпечення, яке призначено для системи доступу до хмарних сервісів з використанням технології PKI. Метою розробки є дослідження та програмна реалізація системи доступу до хмарних сервісів з використанням технології PKI. Об’єктом дослідження є процес доступу до хмарних сервісів з використанням технології PKI. Предметом дослідження є методи доступу до хмарних сервісів з використанням технології PKI. Методи дослідження базуються на методах захисту інформації та хмарних обчислень, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи доступу до хмарних сервісів з використанням технології PKI. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп’ютерні науки, хмарні сервіси, PKI

Постановка проблеми. Сьогодні більшість компаній так чи інакше використовують Інтернет, і із цим зв'язані проблеми захисту віддалених і мобільних користувачів інформаційних систем компанії, захисту корпоративних хмарних сервісів (інтранет-сайту й будь-якого додатка компанії, що працює по http протоколу). Інтернет – це зона підвищеного ризику, відповідно, потрібні спеціальні засоби захисту при роботі віддалених користувачів з WEB-додатками по SSL-протоколу. Таким чином, підсистема захисту WEB-ресурсів вирішує наступні задачі:

– забезпечення єдиного інтерфейсу до додатків;
– інтегрований контроль доступу до корпоративних хмарних сервісів;
– захист клієнтських браузерів;
– захист хмарних сервісів.

Існує багато технологій захисту хмарних сервісів. У магістерському проекті пропонується система захисту основана на використанні протоколів SSL/TLS, які побудовані з використання інфраструктури відкритих ключів (PKI).

У протоколі SSL/TLS використовується ряд симметричних алгоритмів, асиметричних

алгоритмів та геш-функцій. Тому одним із завдань, які потрібно вирішити у даному магістерському проекті є вибір того, або іншого алгоритму, які використовуються на різних етапах побудови системи захисту доступу до хмарних сервісів.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи доступу до хмарних сервісів з використанням технології PKI.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи доступу до хмарних сервісів з використанням технології PKI.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем доступу до хмарних сервісів з використанням технології PKI.

– Дослідження системи доступу до хмарних сервісів з використанням технології PKI.

– Програмна реалізація системи доступу до хмарних сервісів з використанням технології PKI.

Об'єктом дослідження є процес доступу до хмарних сервісів з використанням технології PKI.

Предметом дослідження є методи доступу до хмарних сервісів з використанням технології PKI.

Методи дослідження базуються на методах захисту інформації та хмарних обчислень, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу

Інфраструктура відкритих ключів (PKI) керує ідентифікацією та безпекою в Інтернет-зв'язку для захисту людей, пристроїв і даних.

Організації покладаються на рішення PKI для автентифікації та шифрування інформації, що проходить через веб-сервери, цифрові ідентифікатори, підключені пристрої та програми. Встановлення захищених комунікацій має першочергове значення для забезпечення безперервності бізнесу та проактивного управління ризиками, оскільки організації все більше покладаються на Інтернет для критичних бізнес-систем.

PKI є важливим компонентом архітектури нульової довіри , як описано Національним інститутом стандартів і технологій (NIST), де довіра ніколи не надається неявно та має постійно оцінюватися.

Криптографія з відкритим ключем є основною технологією, яка забезпечує PKI за допомогою двох окремих, але пов'язаних ключів для шифрування та дешифрування. Отриману пару ключів, відкритий ключ, який використовується для шифрування повідомлення, і пов'язаний закритий ключ для його дешифрування, також називають асиметричною криптографією. Пара ключів використовує криптографічні алгоритми, щоб гарантувати, що зашифрований зв'язок може бути розшифрований лише призначеним одержувачем, власником секретного ключа.

Як сертифікати на основі PKI забезпечують безпечну цифрову ідентифікацію

Стандарт X.509 Міжнародного союзу електрозв'язку (ITU) визначає формат сертифікатів на основі PKI. Ненав'язливі та всюдисущі користувачі щодня стикаються з цифровими сертифікатами під час використання веб-сайтів, мобільних додатків, онлайн-документів і підключених пристроїв. Загалом, термін цифровий сертифікат описує всі сертифікати X.509. Це список, який включає сертифікати SSL/TLS, сертифікати підпису електронної пошти, сертифікати підпису коду та сертифікати підпису документів.

Цифровий сертифікат, який часто називають «онлайн-обліковими даними»:

- Перевіряє особу власника.
- Надає зашифрований відкритий ключ власника.
- Видається надійним центром сертифікації (CA), який перевіряє автентичність.

Як обговорювалося раніше, відкриті ключі створюються за допомогою складного

асиметричного алгоритму для поєднання їх із пов'язаним закритим ключем. У криптографії з відкритим ключем ключ шифрування (який може бути відкритим або закритим ключем) використовується для шифрування звичайного текстового повідомлення та перетворення його в закодований формат, відомий як зашифрований текст. Потім інший ключ використовується як ключ дешифрування для розшифровки зашифрованого тексту, щоб одержувач міг прочитати вихідне повідомлення. Все це відбувається автоматично і невидимо для користувача.

Оскільки відкритий ключ публікується для всього світу, інші створюються за допомогою складного криптографічного алгоритму, щоб поєднати їх із пов'язаним приватним ключем шляхом генерації випадкових числових комбінацій різної довжини, щоб їх неможливо було використати за допомогою атаки грубою силою. На відміну від загальнодоступного ключа, закритий ключ є секретним ключем, відомим лише його власнику. Приватні ключі генеруються за допомогою тих самих алгоритмів, які створюють відкриті ключі, щоб створити надійні пари ключів, пов'язані математично.

Найпоширеніші криптографічні алгоритми, які використовуються для генерації ключів:

- Рівест–Шамір–Адлеман (RSA).
- Криптографія еліптичної кривої (ECC).
- Алгоритм цифрового підпису (DSA).

Ці алгоритми використовують різні методи обчислення для створення випадкових числових комбінацій різної довжини. Розмір ключа або довжина біта допомагає визначити силу захисту. Такі організації зі стандартизації, як CA/Browser Forum, визначають базові вимоги до підтримуваних розмірів ключів. Наприклад, типове використання 2048-бітних ключів RSA включає сертифікати SSL, цифрові підписи, сертифікати підпису коду та інші цифрові сертифікати. Ця довжина ключа забезпечує достатню криптографічну міцність, щоб утримати хакерів від злому алгоритму.

Чому б не використати два секретні ключі замість одного відкритого та одного закритого? Хоча це може здатися суперечливим, використання пари ключів, що складається з двох закритих ключів для шифрування та дешифрування конфіденційної інформації, процес, який називається алгоритмом симетричного ключа, не є більш безпечним, ніж асиметричне шифрування. Крім того, використання алгоритмів із симетричним ключем вимагає, щоб обидві сторони у спілкуванні мали доступ до секретних ключів, що збільшує ризик, оскільки тепер обом потрібно зберігати секрет. Нарешті, ці типи алгоритмів не можуть легко масштабуватися, оскільки майже неможливо скоординувати величезну кількість з'єднань, необхідних для приватного обміну всіма комбінаціями закритих ключів.

Переваги PKI

Криптографічні ключі забезпечують механізм перевірки, який захищає ідентифікаційні дані та дані від несанкціонованого доступу або використання. Вони є важливою частиною корпоративної програми кібербезпеки для захисту веб-сайтів, транзакцій електронної комерції, документів, електронних листів, серверів та інших активів від атак кіберзлочинців.

PKI забезпечує масштабовану безпеку даних і цифрової ідентифікації, яка може захистити мільярди повідомлень, якими організації щодня обмінюються через власні мережі та через Інтернет. Ця масштабованість забезпечує широке та відкрите поширення відкритих ключів без зловмисників, які зможуть виявити закритий ключ, необхідний для розшифровки повідомлення. Крім того, PKI розвинулася, щоб стати ще більш універсальною, забезпечуючи взаємодію, тривалий час безвідмовної роботи та керування.

За своєю суттю PKI забезпечує конфіденційність, дозволяючи двом сторонам, що спілкуються, надсилати та отримувати конфіденційні дані приватно. Організаційні переваги великі та вимірні, забезпечуючи безпечний зв'язок, який:

- Захистіть клієнтів.
- Захист інтелектуальної власності підприємства.

- Посилити програми відповідності.
- Запобігайте витоку даних.
- Підтримуйте зростаючу віддалену, розподілену робочу силу.
- Захистіть все більше хмарних програм і пристроїв Інтернету речей (IoT).

По суті, це найкращий захист цифрової ідентифікації ваших людей, пристроїв і даних, що дозволяє окремим особам, організаціям і навіть пристроям встановити довіру в цифровому світі.

Загальні програми PKI

Існує багато застосувань технології PKI, включаючи безпеку веб-сервера, цифрові підписи та підпис документів, а також цифрові ідентифікатори.

Безпека веб-сервера

Криптографія з відкритим ключем є основою для протоколів рівня безпечних сокетів (SSL) і транспортного рівня безпеки (TLS), які є основою захищених підключень веб-браузера HTTPS. Сертифікати SSL/TLS шифрують Інтернет-зв'язок і забезпечують надійне з'єднання клієнт-сервер. Без них кіберзлочинці могли б використовувати Інтернет або інші IP-мережі, використовуючи різноманітні вектори атак, щоб перехоплювати повідомлення та отримувати доступ до їх вмісту.

Цифрові підписи та підпис документів

Окрім використання для шифрування повідомлень, пари ключів можна використовувати для цифрових підписів і підписів документів. PKI використовує приватний ключ відправника для перевірки його цифрової ідентичності. Ця криптографічна перевірка математично прив'язує підпис до вихідного повідомлення, щоб переконатися, що воно не було змінено.

Підписання коду

Підписання коду дозволяє розробникам додатків додавати рівень гарантії шляхом цифрового підпису додатків, драйверів і програмного забезпечення, щоб кінцеві користувачі могли переконатися, що третя сторона не змінила або не скомпрометувала отриманий код. Щоб переконатися, що код безпечний і надійний, ці цифрові сертифікати забезпечують цілісність контейнерів, коду, який вони запускають, і робочих програм, які їх використовують.

Сертифікати електронної пошти

Сертифікати S/MIME перевіряють відправників електронної пошти та шифрують вміст електронної пошти для захисту від дедалі складніших атак соціальної інженерії та фішингу. Шляхом шифрування/дешифрування повідомлень електронної пошти та вкладень, а також шляхом перевірки особи сертифікати електронної пошти S/MIME гарантують користувачам, що електронні листи є автентичними та незмінними.

Цифрові ідентифікатори

Цифрова автентифікація є важливим елементом стратегії нульової довіри для автентифікації людей, даних або програм. Захист ідентифікаційних даних за допомогою цифрових сертифікатів X.509 є більш важливим, ніж будь-коли, оскільки дані та програми виходять за межі традиційних мереж на мобільні пристрої, загальнодоступні хмари, приватні хмари та пристрої IoT. Сертифікати цифрової ідентифікації на основі цього стандарту дозволяють організаціям покращити безпеку шляхом заміни паролів, які зловмисники стають все більш вправними у крадіжці.

Встановлення довіри: роль центрів сертифікації в PKI

Критично важливим компонентом розгортання сертифікатів X.509 є довірений центр сертифікації (CA) або агент для видачі сертифікатів і публікації відкритих ключів, пов'язаних із закритими ключами окремих осіб. Без цього довіреного центру сертифікації відправники не могли б знати, що вони насправді використовують правильний відкритий ключ, пов'язаний із закритим ключем одержувача, а не ключ, пов'язаний зі зловмисником, який має намір перехопити конфіденційну інформацію та використати її для негідних цілей.

Довірені сторонні організації, такі як Sectigo, діють як органи сертифікації, але багато

підприємств і постачальників технологій також вирішують виступати в якості власного ЦС. Вони також можуть вирішити використовувати самопідписані сертифікати.

Незалежно від підходу до розгортання, ЦС слід довіряти:

– Перевіряйте та ручайтеся за особу всіх відправників, для відкритих ключів яких вони публікують.

– Переконайтеся, що ці відкриті ключі справді пов'язані з особистими ключами відправників.

– Захищати рівень інформаційної безпеки у власній організації для захисту від зловмисних атак.

Значення автоматизації PKI та керування сертифікатами

Керування сертифікатами може сприйматися як проста повсякденна задача для IT-або веб-адміністратора, але забезпечення індивідуальної дійсності сертифікатів займає багато часу та коштує. Наприклад, навіть мінімальна видача SSL-сертифікату вручну з одним веб-сервером і доменом включає кілька етапів. Завдання може легко зайняти кілька годин, а витрати на оплату праці сягають понад 50 доларів США на веб-сервер.

Тепер помножте ці зусилля на тисячі чи мільйони сертифікатів PKI на всіх мережевих пристроях і ідентифікаторах користувачів у глобальній організації. Нарешті, додайте роботу, необхідну для керування *життєвим циклом* сертифікатів, щоб включити процеси виявлення, встановлення, моніторингу та оновлення.

Результатом ручного керування сертифікатами є дорогий, трудомісткий і технічно складний виклик для зайнятих IT-команд. Важливо те, що ручний підхід може наражати організації на раптові збої або збої в критично важливих бізнес-системах, а також на зломи та атаки з боку кіберзлочинців.

Автоматизація наскрізного процесу видачі сертифікатів, конфігурації та розгортання забезпечує чітку віддачу від інвестицій (ROI) для IT-директорів та ОГС, які прагнуть зменшити ризики, відповідати нормативним вимогам, контролювати операційні витрати та швидше виводити послуги на ринок.

Сучасні рішення PKI забезпечують функціональність, яка покращує адміністрування та керування сертифікатами життєвого циклу за допомогою:

– Автоматизація: виконання окремих завдань із мінімізацією ручних процесів.

– Координація: використання автоматизації для керування широким набором завдань.

– Масштабованість: керування сертифікатами, що обчислюються сотнями, тисячами чи навіть мільйонами.

– Криптова гнучкість: Оновлення криптографічної надійності, анулювання та заміна сертифікатів із загрозою на квантово-безпечні сертифікати дуже швидко у відповідь на нові чи зміни загроз.

– Видимість: перегляд статусу сертифіката за допомогою єдиного скла в усіх випадках використання.

Економте час і зберігайте контроль за допомогою Sectigo PKI Automation

Враховуючи різні системи, програми та пристрої, які використовують цифрові сертифікати, IT-команди часто керують окремими службами автоматизації від багатьох різних постачальників. Запуск кількох платформ автоматизації означає, що вони часто не настільки ефективні, як могли б бути.

Sectigo надає рішення для автоматизації сертифікатів, які дозволяють підприємствам бути гнучкими та ефективними, зберігаючи контроль над усіма сертифікатами у своєму середовищі. Sectigo підтримує автоматичне встановлення, відкликання та оновлення сертифікатів SSL/TLS і не-SSL за допомогою провідних у галузі протоколів, API та сторонніх інтеграцій.

Єдина інформаційна панель керування сертифікатами, яка автоматизує виявлення, розгортання та керування життєвим циклом у всіх випадках використання та платформах постачальників, може забезпечити ефективність, яку обіцяє автоматизація. Крім того, із

Sectigo ви ніколи не зіткнетеся з обмеженням обсягу сертифікатів, як це можливо з альтернативами з відкритим кодом. Рішення автоматизації Sectigo дозволяють вашій групі безпеки легко застосовувати політику криптографічної безпеки; захистити комунікації; запобігти втраті даних через несанкціонований доступ; і перспективні системи, програми та пристрої в масштабах підприємства.

Запуск PKI у хмарі швидко стає новою нормою. Ми розглядаємо переваги хмарної PKI разом із деякими важливими кроками, пов'язаними з цим підходом.

Оскільки все більше компаній переходять на хмару, важливо розуміти переваги хмарної PKI (інфраструктури відкритих ключів) над традиційною локальною PKI. Cloud PKI стосується розміщення та підтримки центру сертифікації (CA) організації та середовища PKI для надання та керування сертифікатами в хмарі. Хмарні рішення PKI пропонують низку переваг порівняно з локальними рішеннями, зокрема нижчу загальну вартість володіння (TCO), підвищену безпеку та спрощене керування сертифікатами.

Нижче ми пояснюємо кожну з цих переваг більш детально.

Що таке PKI і як він працює?

PKI – це система цифрових сертифікатів , яка вважається золотим стандартом автентифікації та шифрування цифрових ідентифікаторів користувачів, пристроїв і програм. Він має низку варіантів використання для захисту зв'язку, захисту даних і ввімкнення контролю доступу, зокрема:

- Сертифікати SSL/TLS для веб-серверів.
- Сертифікати пристроїв і кінцевих точок для комп'ютерів, мобільних пристроїв і пристроїв Інтернету речей.
- Сертифікати підпису коду, зокрема для робочих процесів DevOps.
- Цифрові підписи.
- Безпечний доступ до API і веб-служб.
- VPN на основі сертифікатів.

PKI базується на криптографії з відкритим ключем, який є надійним механізмом шифрування, який покладається на пару відкритого та закритого ключів. Ці два відповідні ключі використовуються разом для шифрування та дешифрування повідомлення; вони засновані на криптографічних алгоритмах для захисту ідентифікаційних даних і даних від несанкціонованого доступу або використання, захисту від атак з боку кіберзлочинців та інших зловмисників.

Організації, які хочуть використовувати PKI, повинні налаштувати центр сертифікації та/або використовувати надійний сторонній ЦС, наприклад Sectigo. Центр сертифікації відповідає за видачу, перевірку, відкликання та поновлення сертифікатів, а також гарантує автентичність і надійність виданих сертифікатів у системі.

Центри сертифікації повинні створити та підтримувати дуже безпечну інфраструктуру для розміщення середовища PKI. Ця інфраструктура включає кореневий ЦС, безпечне зберігання кореневого ключа ЦС в апаратному модулі безпеки (HSM) і сервер керування сертифікатами. Інфраструктура має бути добре захищена за допомогою комплексної безпеки мережі та програм.

Бути CA – велика відповідальність. Якщо хакерам вдається зламати середовище PKI, вони, по суті, мають «ключі від замку» і можуть видати себе за довіру, щоб непомітно отримати доступ до критичних бізнес-систем. Зважаючи на це, організації часто покладаються на надійних третіх сторін, які спеціалізуються на PKI, або прагнуть контролювати власне приватне середовище PKI.

Є два основних способи розміщення інфраструктури PKI: у хмарі, яка відома як хмара PKI, або у власних локальних центрах обробки даних організації.

Cloud PKI

Це стосується PKI, яка розміщена та підтримується в хмарі. За допомогою хмарного керування PKI хмарний постачальник керує всіма ресурсами, необхідними для розгортання PKI, включаючи апаратне забезпечення, програмне забезпечення та безпеку хмарного

середовища. Постачальник не лише створює та підтримує спеціалізовану інфраструктуру PKI, але й залучає до роботи необхідних спеціалістів.

Хмарна PKI усуває приховані витрати на розгортання PKI. Насправді ліцензування програмного забезпечення, серверне обладнання та встановлення часто є лише невеликим компонентом загального середовища PKI. Розглянемо наступні додаткові важливі елементи:

– Розгортання масштабованого сервера для підтримки постійної доступності та резервування.

– Генерація кореневого ключа для встановлення кореневого сертифіката ЦС.

– Програмне забезпечення для резервного копіювання та технологія відновлення після збоїв для забезпечення безперервної роботи та аварійного відновлення.

– Регулярні перевірки безпеки, щоб підтримувати відповідність і уникнути штрафів.

– Спеціалізовані експерти PKI контролюватимуть усе (їхні зарплати набагато вищі, ніж у адміністраторів початкового рівня).

Усі хмарні PKI-рішення не однакові. Підтримка кількох випадків використання цифрової ідентифікації, застосування принципів нульової довіри до мережі , налаштування IT, сумісність сертифікатів і гнучкість шифрування – все це в середовищі, створеному для зручності використання – це лише деякі з можливостей, які характеризують передові рішення для розгортання, які спрямовані на поточні та майбутні загрози.

Крім того, організації традиційно дивляться на локальну PKI як на спосіб забезпечити більше контролю, але для цього потрібні додаткові ресурси. Запущені процеси, пов’язані з розгортанням, поточне технічне обслуговування та спеціалізований персонал – усе це призводить до високих капітальних витрат і постійних операційних витрат.

Переваги Cloud PKI

Давайте детальніше розглянемо кожну з переваг хмарної PKI порівняно з традиційною локальною PKI, включаючи нижчу загальну вартість користування, підвищену безпеку та простіше керування сертифікатами та ключами.

Низька TCO. Однією з найбільших переваг PKI у хмарі є те, що вона може допомогти зменшити TCO. Це пояснюється тим, що це допомагає усунути потребу у дорогих внутрішніх IT-фахівцях і зменшує капітальні витрати, пов’язані з локальним обладнанням, створенням центру обробки даних і програмним забезпеченням. Крім того, хмарні середовища зменшують поточні операційні витрати за рахунок зниження витрат на послуги, підтримку та обслуговування, а також усувають непрямі витрати, такі як незаплановані простої.

Безпека: сучасний ландшафт загроз підняв міжгалузеві дебати щодо безпеки локальної та хмарної безпеки на новий рівень. Хоча ця розмова, безумовно, не нова, кілька гучних локальних зломів, незважаючи на брандмауери та інші блокпости, ставлять під сумнів переконання, що близькість центру обробки даних означає непроникний захист. Ця проблема також стосується використання локальної архітектури для середовищ інфраструктури відкритих ключів. Організації, які вважають архітектуру PKI безпечнішою, оскільки кореневий ключ і сервер керування сертифікатами знаходяться в їх центрі обробки даних, а не в хмарі, можливо, піддають себе ризику. Хмарні PKI-рішення забезпечують найвищий рівень безпеки мережі та додатків, мають відповідати нормам відповідності, таким як SOC2 та SOC3, і підлягають ретельній зовнішній перевірці.

Простіше керування сертифікатами. Однією з найбільших переваг хмарної PKI є те, що вона може спростити керування сертифікатами. З хмарним постачальником послуг, таким як Sectigo, усіма вашими сертифікатами можна керувати в одному централізованому місці, незалежно від того, чи є вони приватними чи публічними. Це полегшує відстеження всіх ваших сертифікатів, автоматизує надання та встановлення сертифікатів, гарантує, що вони дійсні та поновлені до закінчення терміну дії, а також підтримує централізований список відкликаних сертифікатів (CRL).

Масштабованість: ще одна перевага хмарної PKI полягає в тому, що її можна легко масштабувати відповідно до потреб бізнесу. Завдяки хмарному рішенню ви можете легко

додавати або видаляти сотні, тисячі або навіть мільйони сертифікатів за потреби, не купуючи нове обладнання чи масштабуючи його. Ця гнучкість дозволяє легко підтримувати ваше рішення PKI в актуальному стані в міру зростання вашої організації.

Amazon Web Services (AWS) і Microsoft Azure широко використовують хмарні середовища для багатьох різних бізнес-додатків. Обидва також пропонують рішення для цифрових сертифікатів для своїх середовищ. Проте є деякі важливі відмінності, які організації повинні враховувати.

AWS – це приватна комерційна ЦС. Оскільки він не є членом CA/Browser Forum, наразі він не вважається довіреним загальнодоступним центром сертифікації, необхідним для загальнодоступних веб-сайтів і програм. Клієнти AWS можуть надавати та розгортати сертифікати AWS Certificate Manager (ACM) у різних службах AWS, включаючи AWS Elastic Load Balancing, Amazon CloudFront, Amazon API Gateway, екземпляр Amazon EC2 та інші інтегровані служби, якими керує консоль керування AWS. Деякі додаткові міркування щодо сертифікатів сервера AWS включають:

- ACM не надає розширеної перевірки чи сертифікатів перевірки організації, лише перевірку домену.
- ACM надає лише сертифікати для протоколів SSL/TLS.
- ACM не можна використовувати для шифрування електронної пошти.

Так само Microsoft Azure пропонує хмарну версію керованої PKI через Microsoft CA (MSCA), яка є вбудованим центром сертифікації для продуктів Microsoft. Хоча це зручно для керування сертифікатами для пристроїв, які використовують операційні системи Microsoft, це не охоплює всі основи. Наприклад, якщо у вас є пристрої, які не використовують операційні системи Microsoft, вам усе одно потрібно буде надавати сертифікати та керувати ними через інший ЦС. Ця додаткова складність може легко піддати організацію ризику збоїв і порушень у разі закінчення терміну дії сертифікатів.

Sectigo CLM пропонує хмарне рішення PKI

Очевидно, що хмарний центр сертифікації має ряд переваг перед локальним центром сертифікації. Платформа керування життєвим циклом сертифікатів (CLM) Sectigo спеціально створена в хмарі для захисту ідентифікаційної інформації та даних для захисту ваших користувачів, пристроїв і програм. Ця платформа керування сертифікатами забезпечує єдине середовище керування скляною панеллю, яке автоматизує наскрізний життєвий цикл публічних і приватних цифрових сертифікатів. Крім того, Sectigo CLM не залежить від ЦС , тобто це універсальна платформа, здатна керувати публічними та приватними сертифікатами від Sectigo та інших провідних ЦС.

Можливо, найбільшою перевагою є те, що хмарна PKI Sectigo є більш безпечною, ніж локальна PKI. Як надійний публічний центр сертифікації, який пропонує десятиліття непроникної довіри, Sectigo захищає хмарні приватні корені на тому ж рівні, що застосовується до сотень мільйонів публічних цифрових сертифікатів, які ми видали по всьому світу. Покращена безпека мережі, суворий подвійний контроль для дозволів фізичного доступу, складне керування HSM для зберігання ключів, сервери високої доступності та аварійного відновлення, а також віддані експерти з безпеки поєднуються, щоб забезпечити найкращий у своєму класі хмарний захист.

Платформа Sectigo CLM також сертифікована WebTrust і SOC3, а також підлягає ретельним зовнішнім перевіркам. Завдяки прямим зв’язкам із форумом веб-переглядача центру сертифікації та вибраними державними установами Sectigo отримує завчасні сповіщення про будь-які проблеми з безпекою PKI, включно з проблемами, пов’язаними з еволюцією квантових обчислень.

SASE та доступ до мережі з нульовою довірою (ZTNA)

Ми розглядаємо значення терміну служби безпечного доступу (SASE), значення доступу до мережі без довіри (ZTNA), як вони працюють разом тощо. Останніми роками парадигма безпеки істотно змінилася. У старій парадигмі брандмауери були центральними точками довіри. Усе, що знаходилося за брандмауером, вважалося надійним і безпечним (або

вважалося безпечним), тоді як усе поза ним вважалося ворожим і небезпечним. Сьогодні все не так просто. Важко захистити периметр мережі, коли майже неможливо визначити, де цей «периметр» навіть лежить. Ця нова хмарна реальність у поєднанні з широко розповсюдженими операціями віддаленої роботи у 2020 році зробила новий акцент на Secure Access Service Edge, більш відомому під абревіатурою SASE.

SASE

Secure Access Service Edge (SASE) – це комплексна система безпеки, яка забезпечує безпечний доступ до програм і даних на основі надійної цифрової ідентифікації незалежно від місця розташування користувача чи машини. SASE спочатку був визначений Gartner у 2019 році як «новий пакет технологій, включаючи програмно визначену глобальну мережу (SD-WAN), безпечний веб-шлюз (SWG), посередників безпеки доступу до хмари (CASB), доступ до мережі з нульовою довірою (ZTNA) і брандмауер як послуга (FWaaS) як основні можливості, з можливістю ідентифікації конфіденційних даних або зловмисного програмного забезпечення та здатністю розшифровувати вміст на швидкості лінії з безперервним моніторингом сеансів для рівня ризику та довіри».

За своєю суттю SASE є синонімом «надійної автентифікації для всіх ваших активів». Кількість пристроїв, які підключаються до корпоративних мереж, зросла експоненціально: від персональних пристроїв, таких як ноутбуки та смартфони, до підключених активів, таких як пристрої Інтернету речей і хмарні сервіси. У результаті групи IT-безпеки перемістили фокус з периметра на автентифікацію кожного окремого пристрою. Підприємства приймають менталітет нульової довіри, де ніщо не вважається безпечним, доки не буде перевірено.

Тут на допомогу приходить Secure Access Service Edge. Давайте розберемо:

– **«Безпечний доступ»** стосується того факту, що ноутбуки, телефони та інші пристрої повинні якось підключитися до мережі. Пристрої IoT також вимагають підключення до мережі. Щоб переконатися, що ці з'єднання безпечні, потрібна надійна автентифікація.

– **«Service Edge»** означає той факт, що ці пристрої існують на межі мережі – те, що колись вважалося «поза брандмауером». Сьогодні edge означає кожен окремий актив від ноутбуків до API і мобільних пристроїв, і єдиний спосіб надійно захистити це – надати йому цифрову ідентичність, надавши цьому пристрою сертифікат ідентифікації.

Рішення SASE, які називаються «нахабними», пропонують гнучкий, багатофункціональний підхід до технології безпеки, який добре підходить для сучасного IT-ландшафту, який є неймовірно складним, враховуючи гібридні та багатохмарні середовища, безліч підключених пристроїв і розподілену робочу силу.. Ці рішення використовують цифрову ідентифікацію для захисту від складних і масштабованих векторів атак, особливо націлених на вразливості, що виникають через цю складність. Цей підхід до технологій безпеки також буде застосовний до майбутніх корпоративних IT-ландшафтів.

Як працює SASE?

SASE працює, поєднуючи SD-WAN, SWG, CASB, ZTNA та FWaaS і керуючи цими рішеннями в рамках єдиного набору політик безпеки та ідентифікації. Давайте розглянемо кожен із цих компонентів:

– SD-WAN, або програмно визначена глобальна мережа, може підвищити продуктивність і безпеку WAN-з'єднання, будь то приватне, широкосмугове підключення до Інтернету, LTE та/або 5g, шляхом встановлення політик, визначення пріоритетів, маршрутизації та оптимізації трафіку в глобальній мережі підприємства..

– SWG, або захищений веб-шлюз, може захистити користувачів від веб-загроз, таких як зловмисне програмне забезпечення, і заборонити незахищеному Інтернет-трафіку доступ до внутрішніх систем шляхом дотримання корпоративної політики прийнятного використання.

– CASB, або брокер безпеки доступу до хмари, може ідентифікувати та захищати конфіденційні дані, перебуваючи між користувачами хмарних служб і хмарними

програмами, до яких вони мають доступ. Це допомагає організаціям застосовувати політику безпеки, навіть якщо хмарні служби знаходяться поза межами прямого контролю.

– ZTNA, або доступ до мережі з нульовою довірою, можна використовувати для забезпечення безпечного та детального контролю доступу. ZTNA – це модель, у якій довіра ніколи не надається неявно, і її необхідно постійно оцінювати.

– FWaaS або брандмауер як послуга може захищати програми та дані від несанкціонованого доступу за допомогою хмарного брандмауера, який включає можливості брандмауера наступного покоління (NGFW) і засоби контролю доступу, такі як системи запобігання вторгненням (IPS), фільтрація URL-адрес і захист DNS.

Які переваги SASE?

Цифрова трансформація призвела до нової ери корпоративних послуг безпеки. Важливим каталізатором для SASE є необхідність покращити взаємодію різних рішень безпеки. У сучасному підприємстві більше не домінує єдиний стек технологій, і підприємства більше не можуть зосереджуватися лише на захисті центрів обробки даних і забезпеченні захисту в рамках мережевої архітектури з міжмережевим екраном.

Сучасні складні середовища тепер включають мобільні пристрої, багатохмарні технології, DevOps, BYOD, Інтернет речей тощо. Звичайно, вам потрібна надійна автентифікація для кожної з цих систем. У цьому розширеному середовищі ідентифікація є новим периметром, і SASE розроблено для цього середовища.

Крім надійної безпеки ідентифікації, яку забезпечує SASE, підприємства також можуть отримати такі переваги:

– Більша гнучкість.
– Швидке впровадження нових технологій.
– Підвищення ефективності IT.
– Менші адміністративні витрати.

Ці рішення надають організаціям гнучкість, необхідну для безпечного доступу до своїх програм і даних, незалежно від того, де вони знаходяться, локально чи в хмарі. Цей надійний підхід до цифрової ідентичності допомагає надавати детальний доступ і дозволи кожному користувачеві, пристрою та процесу в мережі. Ці можливості дозволяють організаціям швидко впроваджувати інновації, включаючи програми SaaS, пристрої IoT та інструменти віддаленого доступу, і робити це, одночасно блокуючи свою інфраструктуру від атак і зберігаючи контроль над тим, хто і які системи мають доступ до конкретних програм і даних. Крім того, завдяки консолідації всіх мережевих функцій і функцій безпеки, які традиційно постачаються в точкових продуктах і рішеннях, архітектура SASE забезпечує єдиний підхід для IT-адміністраторів для керування своїми мережами та безпекою. Це максимізує ефективність і продуктивність IT-команд, дозволяючи їм визначати єдиний набір політик безпеки та централізовано керувати кількома технологіями відповідно до цих політик. Підприємства також можуть скоротити адміністративні витрати. SASE розгортається як єдиний стек програмного забезпечення, що усуває потребу в кількох пристроях. Це зменшує як капітальні витрати на проекти, так і поточні експлуатаційні витрати.

SASE проти ZTNA: відмінності, подібності та як вони працюють разом

І SASE, і ZTNA є важливими компонентами сучасної архітектури безпеки, однак це два різні рішення. SASE забезпечує комплексну багатогранну структуру безпеки, тоді як ZTNA є більш вузькою моделлю, орієнтованою на обмеження доступу до ресурсів, яка є частиною SASE. При спільному використанні вони можуть забезпечити більш комплексне рішення безпеки, здатне захистити програми та дані, незалежно від місцезнаходження кінцевого користувача. Доступ до мережі з нульовою довірою , який часто називають програмно-визначеним периметром (SDP), означає відмову в доступі до ресурсів, якщо користувачеві чи машині не надано явного дозволу, що забезпечує більш суворий підхід до безпеки, який особливо корисний у разі злому. Крім того, права доступу для кожної особи постійно оцінюються та затверджуються або відхиляються відповідно. «Ніколи не довіряй,

завжди перевіряй» – це фундаментальна філософія мережі без довіри та ключова відмінність між мережею без довіри та іншими мережевими моделями. З нульовою довірою немає неявних довірчих відносин. Натомість усі кінцеві користувачі та пристрої розглядаються як ненадійні, доки їх не можна перевірити. Цей процес перевірки є основою моделі нульової довіри. Це здійснюється за допомогою різних методів, включаючи автентифікацію, авторизацію та перевірку, і базується на таких критеріях, як ідентифікація користувача, місцезнаходження, версія операційної системи та мікропрограми, а також тип апаратного забезпечення кінцевої точки. Переваги моделі нульової довіри очевидні: покращена кібербезпека за рахунок усунення прогалин у безпеці та контролю бокового переміщення в мережі, а також підтримка мобільних і віддаленого доступу працівників. Крім того, модель нульової довіри захищає дані як у хмарі, так і в локальних центрах обробки даних, забезпечуючи надійний захист від програм-вимагачів, шкідливих програм, фішингових атак і розширених загроз.

Перевага комбінованих рішень

Простіше кажучи, поєднання SASE та Zero Trust допомагає компаніям застосовувати політику в усій їхній мережі. Такий підхід забезпечує кілька ключових переваг, зокрема посилену безпеку мережі, спрощене керування мережею, менші витрати та єдиний огляд усієї мережі. SASE та ZTNA також можуть допомогти компаніям зменшити ризик витоку даних і зменшити площу атаки. Поєднуючи ці два підходи, компанії можуть створити надійний периметр кібербезпеки, куди зловмисникам важко проникнути. Це допомагає гарантувати, що лише авторизовані користувачі та пристрої мають доступ до конфіденційних даних і систем, а користувачі та машини мають доступ лише до тих ресурсів, які їм потрібні для виконання своєї роботи.

Чи є SASE VPN?

Ні, SASE – це не VPN (віртуальна приватна мережа), а структура, яка забезпечує безпечний доступ до програм і даних, тоді як VPN використовуються для забезпечення безпечного з'єднання користувача з Інтернетом. Хоча VPN можуть забезпечити безпечне з'єднання, вони не завжди ефективні для захисту програм і даних. SASE та ZTNA можна використовувати разом, щоб забезпечити більш безпечне рішення, яке здатне захистити програми та дані від несанкціонованого доступу. Оскільки SASE включає ZTNA, його можна використовувати на додаток до VPN або замінити їх. Його здатність забезпечувати принципи найменших привілеїв для доступу в режимі реального часу особливо корисна для безпеки хмари, особливо в сучасні часи дедалі більш віддаленої робочої сили та хмарних робочих навантажень. Zero Trust Network Access має велику перевагу перед VPN, коли мова йде про деталізацію. За допомогою ZTNA підприємства можуть обмежувати доступ на більш точному рівні порівняно з віртуальними приватними мережами.

Як керувати цифровими ідентифікаторами в SASE

Застосування SASE покладається на надійну цифрову ідентичність для всіх користувачів, пристроїв і процесів у всьому підключеному IT-ландшафті. У рамках цієї безпеки, перш за все, ідентифікаційні дані, для компаній дуже важливо перевірити автентичність і зашифрувати всі цифрові ідентифікаційні дані людини чи машини. Цифрові сертифікати, видані центрами сертифікації (CA), такими як Sectigo, є базовою технологією, яка використовується для автентифікації ідентифікаційних даних людини чи машини та встановлення цифрової довіри. Захистити ідентифікаційні дані та керувати ними в рішеннях SASE непросто, враховуючи вибухове зростання обсягу, різноманітності та швидкості цифрових ідентифікаційних даних у нових сценаріях використання, включаючи гібридні та багатохмарні середовища, цифрові підписи, контейнери DevOps, код, роботизовану автоматизацію процесів. (RPA) та інші корпоративні програми. У сукупності ці виклики ідентифікації становлять майже неможливе завдання, незважаючи на зусилля, щоб запобігти збою в управлінні ідентифікацією та захистити вашу мережу та дані від злому та крадіжки. Дослідження EMA у 2021 році серед IT-керівників показало, що 81% підприємств вважають складним керування цифровими ідентифікаторами. Найкращий спосіб для CISO та їхніх

команд застосувати SASE та забезпечити цифрову довіру зараз і в майбутньому – це автоматизувати життєвий цикл кожного окремого ідентифікатора в усій IT-екосистемі. Управління життєвим циклом сертифіката (CLM) – це комплексне рішення, яке автоматизує життєвий цикл сертифіката, від надання та розгортання до відкликання.

CLM гарантує належне встановлення, моніторинг і оновлення всіх сертифікатів, надаючи організаціям масштабованість, видимість і контроль, необхідні для забезпечення безпеки та сумісності цифрових середовищ за допомогою SASE. Крім того, сучасним підходом до CLM є агностичне хмарне рішення Sectigo CA. Sectigo Certificate Manager надає єдиний портал адміністрування для захисту та керування зростаючою кількістю цифрових ідентифікацій як людини, так і машини, з інтеграцією в провідних постачальників технологій, які ефективно працюють у будь-якому IT-середовищі.

Зі збільшенням віддаленої роботи в осяжному майбутньому та безперервним розширенням таких областей, як IoT, SASE ставатиме ще більш важливим, оскільки організації шукатимуть більш надійні способи захисту своїх мереж.

Розробка структурної схеми

Розроблене програмне забезпечення представляє із себе набір компонентів призначених для забезпечення політики безпеки як у вже існуючих, так і в створюваних мережних інформаційних системах.

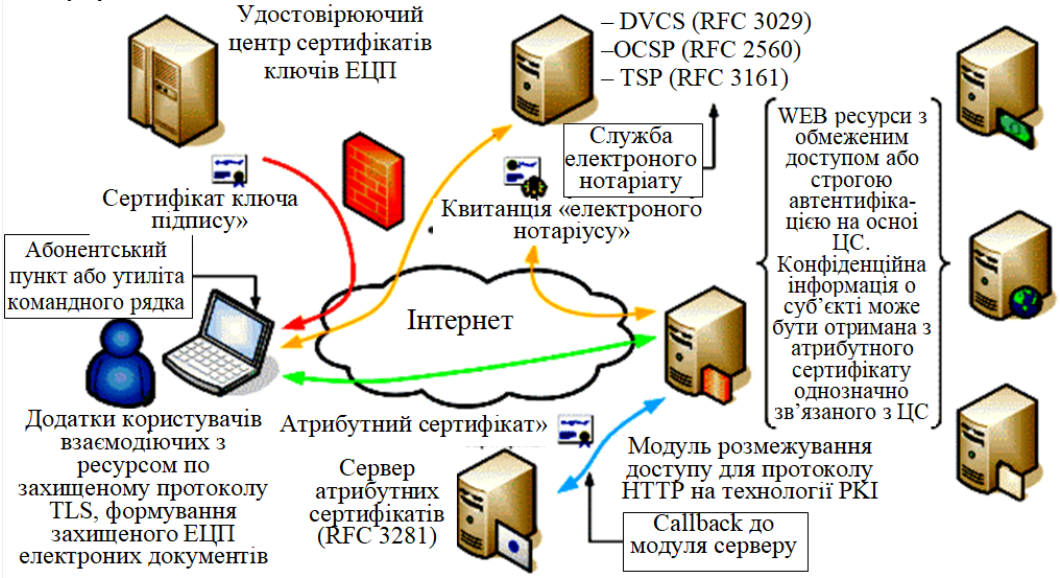


Рисунок 1 – Структурна схема

На рисунку 1 представлена структурна схема розробленої системи. На цій схемі введені наступні позначення:

– ЕЦП – електроний цифровий підпис;
– ЦС – цифровий сертифікат;
– PKI – інфраструктура відкритих ключів;
– DVCS – Data Validation and Certification Server Protocols – протокол підтвердження даних та сертифікації серверу;
– OCSP – Online Certificate Status Protocol – онлайн протокол статусу сертфікату;
– TSP – Time-Stamp Protocol – протокол часових міток;
– TLS – криптографічний протокол, що забезпечує захищену передачу даних між вузлами в мережі Інтернет;
– RFC – документ, у якому описується той або інший стандарт.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів доступу до хмарних сервісів з використанням технології PKI. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем доступу до хмарних сервісів з використанням технології PKI; Досліджена система

доступу до хмарних сервісів з використанням технології PKI; На основі отриманих результатів досліджень створена програмна реалізація системи доступу до хмарних сервісів з використанням технології PKI; Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання доступу до хмарних сервісів з використанням технології PKI; Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
2. Смірнов О.А., Дрєєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.
3. Смирнов А.А., Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений. Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.
4. Смирнов А.А., Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения. Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.
5. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 3(69). С. 93-98. 2022.
6. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.
7. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.
8. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95
9. Смирнов А., Кузнецов А., Кузнецова Т. «Шумоподобные дискретные сигналы для асинхронных систем кодового разделения радиоканалов». Радиотехника, № 2(205), 175–183. 2021.
10. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.
11. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12. (Scopus).
12. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022. (Scopus).
13. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34. (Scopus).
14. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477. (Scopus).
15. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w (Scopus).
16. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
17. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).

18. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
19. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
20. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).

УДК 004

І.Шевчук, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДАНИХ КАРТИ ТАХОГРАФА

У статті розроблено програмне забезпечення, яке призначено для системи даних карти тахографа. Метою розробки є дослідження та програмна реалізація системи даних карти тахографа. Об'єктом дослідження є процес даних карти тахографа. Предметом дослідження є методи даних карти тахографа. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи даних карти тахографа. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерна інженерія, захисту доступу, тахограф

Постановка проблеми. У цей час пластикові карти одержали широке поширення в додатках захисту інформації: це й таксофонні карти, і SIM-карти в стільникових телефонах, це, звичайно ж, платіжні карти різних типів, карти медичного страхування, проїзду в міському транспорті, карти постійного покупця, що стимулюють попит, називані дисконтними, контейнери криптографічних ключів, карти-ключі, що відкривають електронний замок у дверях, електронні посвідчення особи, засоби підтвердження оплати й дійсності абонента в стільниковій телефонії й супутниковому телебаченні, засоби автентифікації користувачів обчислювальної системи й т.д.

Основними завданнями розвитку технології вітчизняних чіп-карт на сьогодні є:

– пошук методів збільшення ефективності використання ресурсів кристала, в умовах неможливості переходу на іншу норму проектування;

– пошук шляхів інтеграції чіп-карт закордонного виробництва в системи, що використовують вітчизняні криптографічні стандарти;

– проектування захищених малоресурсоємних протоколів електронних платежів і ідентифікації на основі чіп-карт;

– проектування захищених безконтактних карт, що несуть як ідентифікаційну, так і платіжну функціональність, які задовольняють вітчизняним стандартам в області захисту інформації;

– пошук нових областей застосування для чіп-карт.

Рішенню комплексу вищевказаних теоретичних і практичних питань і присвячена дана магістерська робота. Крім того, розроблені методи зменшення ресурсоємності, застосовувані для рішення набору даних прикладних завдань, можуть бути використані й в інших областях, у яких є подібні завдання. Останнє робить магістерську роботу актуальною не тільки для розглянутої предметної області.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи даних карти тахографа.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи даних карти тахографа.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем даних карти тахографа.
– Дослідження системи даних карти тахографа.
– Програмна реалізація системи даних карти тахографа.

Об'єктом дослідження є процес даних карти тахографа.

Предметом дослідження є методи даних карти тахографа.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу.

Опис функціонування системи

Розглянемо основні уразливості, при роботі з чіп-картами. До основних класів типових атак на АС на основі чіп-карт, відносяться наступні:

– Соціальна інженерія.
– Соціальна інженерія із застосуванням апаратних засобів.
– Доступ до каналів зв'язку.
– Підміна / модифікація устаткування.
– Інженерне проникнення, DPA/ DFA-атаки, криптоаналіз.
– Закладки, залишені розроблювачами системи.

Розглянуті в магістерській роботі протоколи безпеки спроектовані таким чином, що захищають від атак зловмисників описаних рівнів до рівня 3 включно на доступ до каналів зв'язку й підміну / модифікацію устаткування, якщо в описі протоколу не зроблене уточнення. Захист від атак видів "соціальна інженерія", "соціальна інженерія із застосуванням апаратних засобів" повинна бути забезпечена організаційно-адміністративними мірами. Захист від атак класу "інженерне проникнення, DPA/DFA-атаки, криптоаналіз" здійснюється розроблювачами кристалів чіп-карт, і базових криптоалгоритмів. Захист від закладок, залишених розроблювачами системи, не здійснюється.

Вирішимо завдання побудови архітектури компактної файлової системи мікропроцесорної карти, збільшення ефективності використання ресурсів існуючого кристала вітчизняної мікропроцесорної карти (у першу чергу, EEPROM).

Для цього розробимо архітектуру компактної файлової системи, що дозволяє використовувати ресурси EEPROM істотно більш раціональним образом, рекомендації зі зміни архітектури мікроконтролера карти для забезпечення можливості виконання внутрішніх скриптів безпосередньо мікроконтролером карти й швидкодіюча реалізація ДСТ 28147:2009 на чіп-картах.

Самим ресурсномістким і дефіцитним видом пам'яті сьогоднішніх інтелектуальної карти є EEPROM. Вона займає більше половини кристала і його розмір обмежує можливості використання карти. Для рішення поставленого завдання були проведені:

– Розробка організації файлової системи таким чином, щоб перенести незмінні частини файлів додатків карти (а їх – до 90% від усього обсягу прикладних даних) у більше дешеве й менш дефіцитне масочне ПЗП. Таким чином, байти того самого файлу зберігаються, залежно від їхнього призначення, у різних пристроях зберігання. При цьому таке зберігання є прозорим для операційної системи й додатків карти (

– Рисунок 1). Дане завдання було вирішено за допомогою FAT із кластерами змінної довжини.

– Зменшення розмірів службових областей.

– Був зроблений перехід від блок-орієнтованої організації файлової системи (що приводить до втрат при вирівнюванні до границі блоку) до байт-байт-орієнтованого.

– Замість розрахунку CRC на файл (коли для читання хоча б одного байта було потрібно перечитати весь файл, щоб перевірити CRC) був реалізований підрахунок CRC на сектор (рис.2).

Розроблена архітектура файлової системи має наступні властивості:

– Мінімізовано втрати від вирівнювань.

– Кількість рівнів файлової системи становить не менш трьох.

– Для додатків, емісія яких становить достатній обсяг, з'являється можливість переміщення всіх константних даних з EEPROM у вільну область масочного ПЗП.

– Реалізовано можливість видалення з карти додатків, що мають константні дані в масочному ПЗП для забезпечення можливості використання карти в інших додатках. Оскільки видалення даних з масочного ПЗП неможливо, віддаляються лише посилання на них з EEPROM.

– Для забезпечення можливості використання кристалів з окремими збійними ділянками EEPROM, зсуву даних, збережених в EEPROM, не зберігаються в масочному ПЗП.

– Можливе видалення файлів (що дозволяється лише в деяких файлових системах чіп-карт), однак видалення файлів і додатків не повинні носити масового характеру.

– Залежно від типу, файли можуть мати заголовки різної довжини.

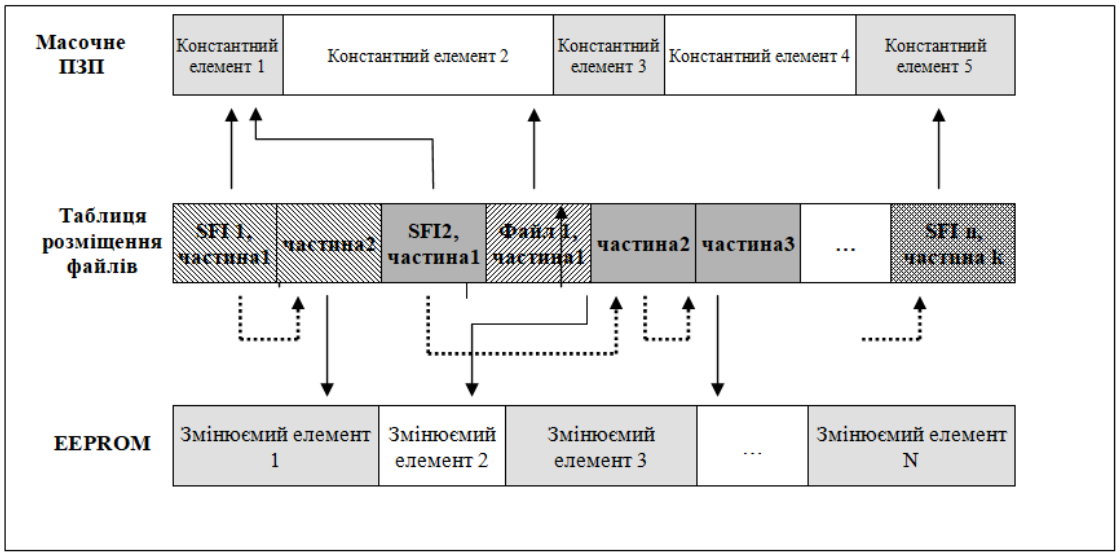


Рисунок 1 – Таблиця розміщення файлів

– Структура даних файлової системи поліпшена з погляду мінімізації часу звертання до файлів.

– Файлова система дозволяє забезпечувати збалансоване навантаження по перезапису на сектори EEPROM, тобто немає секторів, перезаписуваних істотно частіше, ніж інші.

Оцінка практичної ефективності (на прикладі EMV-сумісного додатка) запропонованої методики показала, що розмір що вимагається EEPROM може зменшуватися до 3-4 разів.

Далі розглянемо завдання вироблення рекомендацій зі зміни архітектури мікроконтролера карти для забезпечення можливості виконання внутрішніх скриптів безпосередньо мікроконтролером карти, що має метою зменшення простору масочного ПЗП, займаного кодом ОС карти, а також забезпечення можливості кастомізації конфігурації ОС на етапі персоналізації шляхом додавання різних додаткових модулів ОС в EEPROM.

Для забезпечення ізольованості друг від друга додатків, що перебувають на карті, виберемо шлях введення програмного супервізора й введемо аналог захищеного режиму, що

дозволяє додатку робити лише безпечні операції, а виконання операцій, критичних з погляду безпеки, здійснюється під контролем (або за допомогою) супервізора.

Рекомендуємо, до застосування, реалізацію мінімального достатнього набору модифікацій в архітектурі кристала інтелектуальної карти КБ5004ВЕ1 (An15M04):

– введення прапора наявності віртуального режиму в регістрі стану процесора;

– об'єднання адресного простору масочного ПЗП і EEPROM в один адресний простір, для того, щоб програмний код, збережений в EEPROM, був доступний для виконання кристалом;

– поділ набору команд на привілейовані й непривілейовані;

– введення в систему команд кристала додаткової команди перемикання на супервізор;

– використання двох переривань під порушення захисту й під супервізор;

– спроектовані протоколи дозволяють захиститися від атак зловмисників рівнів 1-3, описаних у розділі 1 на доступ до каналів зв'язку й підміну / модифікацію устаткування.

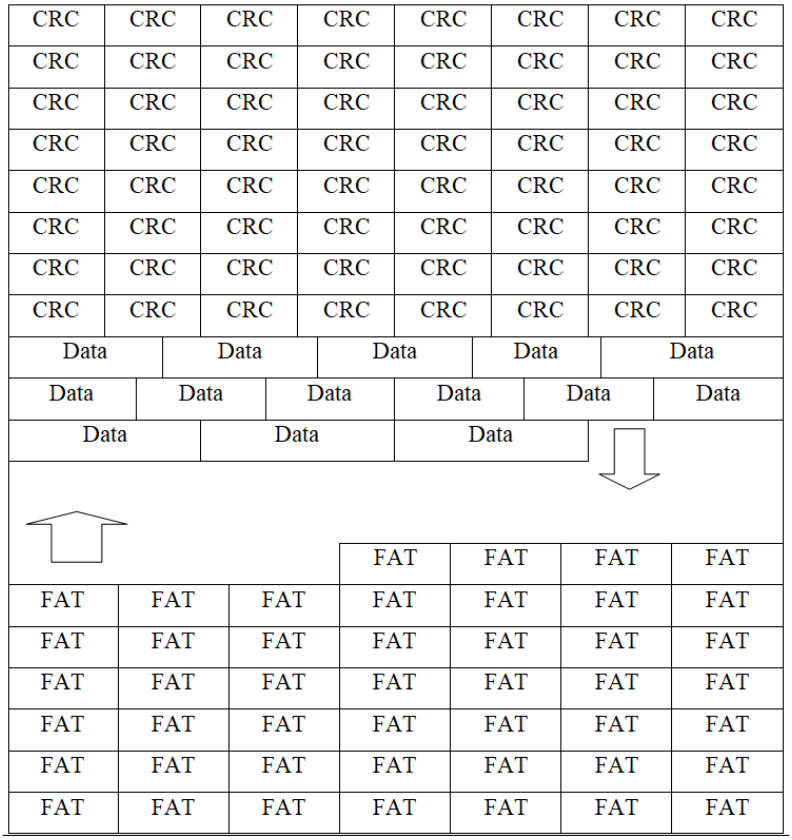


Рисунок 2 – Заповнення EEPROM карти файловою системою

Вироблені рекомендації дозволять розвантажити простір у масочному ПЗП, тому що розмір реалізації супервізора представляється істотно меншим у порівнянні із традиційним інтерпретатором скриптів віртуальної машини. По оцінці, заснованій на розмірах реалізацій інтерпретаторів віртуальної машини в існуючих операційних системах для кристала КБ5004ВЕ1, абсолютна величина виграшу може скласти від одного до півтора кілобайт із шістнадцяти на сьогоднішній день.

Розглянемо реалізацію вітчизняного криптоалгоритму ДСТ 28147:2009 на закордонних картах без необхідності модифікації їх масочного ПЗП, з поліпшеними швидкісними характеристиками. Дане рішення необхідно для забезпечення можливості використання закордонних чіп-карт (не підтримуючих ДСТ 28147:2009) у додатках, що вимагає використання вітчизняної криптографії й наявності обсягів EEPROM, істотно перевищуючих існуючі EEPROM чіп-карт російського виробництва.

ДСТ 28147:2009 орієнтований на мікропроцесори із тридцятидвобітною архітектурою, з порядком байтів BigEndian. Ніяких складностей у реалізації ДСТ 28147:2009 на інтелектуальній карті в принципі немає, якщо тільки карта підтримує опціональний для інтелектуальної карти тридцятидвобітний тип int. На жаль, на сьогоднішній день більшість реалізацій інтелектуальної карти тип int не підтримують, а карти, на яких інтелектуальної карти підтримують int, мають досить високу вартість. Виходить, необхідно робити тридцятидвобітне додавання як пари шістнадцятибітних, контролюючи переповнення, благо шістнадцятибітний тип short зобов'язаний бути присутнім у будь-якій реалізації інтелектуальної карти. Однак в інтелектуальній карті неможливо здійснити контроль переповнення за допомогою прапора, тому для контролю переповнення при додаванні двох чисел певного типу доводиться перетворювати їх до старшого типу й потім порівнювати з маскою, переконуючись, що результат приводиться до вихідного типу без втрат. Таким чином, при традиційному підході, одне додавання тридцятидвобітних чисел виливається в додавання чотирьох восьмибайтових чисел, і це навіть без обліку інкрементів.

Пропонована схема дозволить здійснювати додавання тридцятидвобітних чисел з будь-якими значеннями шляхом виконання двох шістнадцятибітних додавань, одного інкремента й трьох шістнадцятибітних операцій перевірки знака.

Для прискорення реалізації криптоалгоритма були отримані наступні теоретичні результати:

Визначення 1: Назвемо знаковою інтерпретацією беззнакового числа $x \in \overline{0,2^n-1}$ конструкцію виду:

$$s(x) := \begin{cases} x, & x < 2^{n-1} \\ x - 2^{n-1}, & x \geq 2^{n-1} \end{cases}, \qquad (1)$$

Уведемо знакові інтерпретації для що складаються A, B і їхньої суми:

$$a := s(A),\ b := s(B),\ c := s((A+B) \bmod 2^n). \qquad (2)$$

Під сумою доданків у цьому випадку розуміється операція додавання без обліку переповнення, тобто за модулем 2^n.

Лема 1: $c \geq 0 \Leftrightarrow A + B \in \left[0, 2^{n-1}\right) \cup \left[2 \cdot 2^{n-1}, 3 \cdot 2^{n-1}\right)$.

Лема 2: $\forall n \geq 2 \qquad 2 \cdot (2^n - 1) \geq 3 \cdot 2^{n-1}$.

Твердження 1 (Критерій наявності переносу): Перенос при додаванні чисел A і B (тобто $A + B \geq 2^n$) виникає тоді й тільки тоді, коли щира умова:

$$\begin{cases} b < 0 \,\&\, c \geq 0, & a \geq 0 \\ b < 0 \vee c \geq 0, & a < 0 \end{cases}. \qquad (2)$$

Отриманий результат дозволяє одержати реалізацію, істотно більш швидку в порівнянні з існуючими аналогами.

Розробимо протокол гнучких і малоресурсоємних типових додатків чіп-карт, що відповідають сучасним вимогам безпеки, придатних до використання в широкому колі виникаючих прикладних завдань, у тому числі з урахуванням специфіки додатків безконтактних карт.

Уведемо поняття універсального облікового додатка й розробимо його в також виробимо ряд рекомендацій з розробки архітектури вітчизняних безконтактних карт, що задовольняють вимогам вітчизняних стандартів в області захисту інформації, на базі кристала MIFARE компанії Philips Semiconductor.

Основною проблемою в даній області є складність розбору виникаючих нестандартних ситуацій, що виникають внаслідок атак або ненавмисних аварійних ситуацій, таких як, переривання живлення або розриви зв'язку в процесі виконання транзакції.

Звичайно в подібній ситуації тримач карти приречений очікувати два тижні – час, протягом якого банк повинен зібрати всі оффлайн-журнали для відновлення послідовності подій.

Для забезпечення можливості невідкладного й однозначного трактування всіх відомих атак і помилок, що виникають у процесі проведення платіжних транзакцій були уведені три платіжних лічильники:

– лічильник запитів сертифіката балансу;
– лічильник дебетований;
– лічильник онлайн-операцій.

Розроблено алгоритм виявлення атак шляхом трактування станів лічильників.

Розроблена архітектура універсального платіжного додатка задовольняє наступним вимогам:

– можливість захищеного онлайн-поповнення, дебетування, синхронізації й віддаленої зміни платіжних лімітів;

– можливість безпечних оффлайн-дебетування й, можливо, оффлайн-скасування з виконанням всіх вимог безпеки.

– можливість безпечного дебетування на дрібні суми без уведення PIN-коду;

– стійкість до збоїв зв'язку під час онлайн-транзакції: у цьому випадку клієнт не повинен втратити кошти з балансу карти, навіть тимчасово;

– оффлайн- і онлайн-операції рознесені, тобто клієнт навіть після невдало завершеної онлайн-транзакції повинен мати можливість, як і колись, проводити оффлайн-операції;

– можливість використання симетричних криптоалгоритмів, і можливість застосування (у випадку потреби й наявності криптографічного співпроцесора на кристалі карти) асиметричних криптоалгоритмів без серйозних архітектурних змін.

Спроектовані протоколи дозволяють захиститися від атак зловмисників рівнів 1-3, описаних у розділі 1 на доступ до каналів зв'язку й підміну / модифікацію устаткування.

Спроектовано універсальний механізм, називаний універсальним обліковим додатком, на зразок універсального платіжного додатка, що має однакові принципи функціонування в зовсім різних платіжних проектах. Забезпечується унікальність криптограм, передані дані захищаються від модифікації при читанні/запису.

Були зроблені наступні кроки:

– Вичленовано загальні прикладні особливості в різних по призначенню облікових додатків.

– Виходячи з вимог атомарності перезапису секторів з урахуванням контролю автентичності записуваних даних був обраний розмір сектора.

– Обрано й описана конфігурація маски доступу.

– Розроблено набір прикладних команд універсального облікового додатка.

Отриманий універсальний обліковий додаток має наступні властивості:

– Універсальність: наскільки це можливо, додаток задовольняє вимогам різних дисконтних схем.

– Низька ресурсоємність. Функціональність додатка легко реалізується на різних чіп-картах, як вітчизняних, так і закордонних, що дозволяють розширювати набір команд за допомогою скриптів або апплетів інтелектуальної карти. Також повинна бути можливість апаратної реалізації додатку в безконтактній карті без мікропроцесора.

– Для спрощення реалізації додатка на безконтактних картах додаток не вимагає наявності апаратного ДВЧ.

– Гнучка схема розмежування доступу.

– Можливість реалізації декількох облікових додатків на одній карті.

– Можливість розмежування доступу до прикладним даних шляхом двосторонньої криптографічної автентифікації між картою й пристроєм прийому карт (ППК).

– Можливість автентифікації тримача карти за паролем.

– Наявність убудованого в додаток криптографічного контролю цілісності прикладних даних, переданих як з карти, так і на карту.

– Мінімізовано кількість обмінів між картою й ППК.

– Наявність механізму забезпечення унікальності криптограм.

– Спроектовані протоколи дозволяють захиститися від атак зловмисників рівнів 1-3, описаних у розділі 1 на доступ до каналів зв'язку й підміну / модифікацію устаткування.

Розробимо архітектуру вітчизняної безконтактної карти. Вирішимо наступні підзадачи:

– Вибір продукту з лінійки MIFARE як прототип вітчизняної безконтактної карти. Після детального дослідження лінійки кристалів MIFARE компанії Phillips Semiconductors, як прототип кристала вітчизняної безконтактної карти був обраний кристал MF1 IC S50.

– Механізми криптографічної автентифікації були наведені у відповідність із вітчизняними стандартами.

– Організація пам'яті була наведена у відповідність із розмірами ключа застосовуваного криптоалгоритмом ДСТ 28147:2009;

– Механізм автентифікації став забезпечувати можливість контролю цілісності переданих в обох напрямках по ефіру даних.

– Забезпечено унікальність (неповторюваність) даних, записуваних на карту.

Знайдемо шляхи застосування чіп-карт вітчизняного виробництва в системах захисту ПЗ від несанкціонованого копіювання. У ній вирішуються проблеми:

– Можливості використання вітчизняних чіп-карт у системах захисту ПЗ від несанкціонованого копіювання шляхом розробки протоколу симетричної автентифікації суб'єктом, що не зберігає секретний ключ автентифікації.

– Криптографічний протокол голосової активації ПЗ, що захищається від несанкціонованого копіювання, що володіє архітектурою, схожою з попереднім протоколом архітектурою.

– Спроектовані протоколи дозволяють захиститися від атак зловмисників рівнів 1-3, описаних у розділі 1 на доступ до каналів зв'язку й підміну / модифікацію устаткування.

Пропонується протокол симетричної однобічної автентифікації. Загалом, на автентифікуючій стороні зберігається таблиця еталонних відповідей на запити автентифікації, що не дозволяє, проте, відтворювати свій вміст в емуляторі зловмисника.

Стійкість протоколу забезпечується тимчасовими і ємнісними міркуваннями, а сам протокол показує можливість застосування чіп-карт, що здійснюють лише симетричні криптографічні перетворення, у системах захисту ПЗ від несанкціонованого копіювання.

Діаграма обмінів протоколу автентифікації, наведена в таблиці 1.

Істотною вимогою є забезпечення цілісності програмно-апаратного середовища автентифікуючій сторони. При цьому допускається можливість дослідження зловмисником алгоритмів функціонування автентифікуючій сторони, у т.ч. можливість читання зловмисником таблиці еталонних відповідей.

Розробимо криптопротокол віддаленої активації що захищається ПЗ через голосовий телефонний канал, що задовольняє наступним вимогам:

– можлива передача даних тільки один раз від клієнта до сервера й потім один раз назад;

– розмір переданих даних для кожного напрямку не повинен перевищувати 64-128 бітів, більший обсяг передати через голосовий телефонний канал (диктування) представляється скрутним;

– ПЗ, що захищається від несанкціонованого копіювання, повинне бути захищене від модифікації зловмисником;

– допускається наявність у зловмисника повної інформації про протокол активації;

– ПЗ, що захищається від несанкціонованого копіювання, повинне бути захищене від надання апаратним середовищем і операційною системою нав'язувальних зловмисником даних, що представляють собою ідентифікатори й метрики устаткування, а також джерела інформації для програмних ДВЧ;

– на стороні клієнта не повинні вимагатися які-небудь додаткові апаратні засоби;

– на клієнті не повинні зберігатися секретні криптографічні ключі;

– повинно бути розрахунково складно для зловмисника земулювати відповідь сервера на запит клієнта;

– можливе використання різних базових криптоалгоритмів одного класу в криптографічному протоколі активації.

Обмеження, що накладаються у вищеописаних вимогах, на розмір і кількість переданих повідомлень унеможливлюють використання асиметричних криптоалгоритмів у силу того, що розміри ЕЦП відомих авторові криптоалгоритмів істотно перевищують задані межі. Використання ж традиційної схеми автентифікації на базі симетричного криптоалгоритма, що припускає зберігання секретного ключа на обох сторонах, також неможливо.

Таблиця 1 – Діаграма обмінів протоколу автентифікації

Мікропроцесорна карта		Система захисту ПЗ від несанкціонованого копіювання
Ініціалізація даних		
Ключ K генерується й міститься на карту		Таблиця $T'=\{h(i) \mid 0 \le i < N\}$, що відповідає ключу K генерується й міститься до пам'яті ПЗ, що захищається, де $h(i) = H(v(i))$, $N = 2^{25} = 33\,554\,432$, розмір таблиці T' складе 128Мб.
Автентифікація карти		
		Генерується випадкове число $r \in \overline{0, N-1}$, і передається карті.
	←	
Формується й вертається відповідь $t = v(r)$ довжиною більше 1 кб		
	→	
		Обчислюється хеш-значення $w = H(t)$, потім по таблиці T', перевіряється рівність $w = h(r)$. Автентифікація вважається успішною у випадку збігу, і неуспішною в противному випадку.

Для підготовки до виконання криптопротоколу на сервері пропонується згенерувати таблицю пар виду:

$$\{(r,c) \mid r = H(c)\}, \qquad (4)$$

де:

r – двійковий вектор розміром l_r від 8 до 16 байт, називаний запитом активації (або, просто, запитом);

c – двійковий вектор розміром l_c від 8 до 16 байт, називаний підтвердженням активації (або, просто, підтвердженням), вибирається довільно, можливо, випадковим образом.

Вибір діапазону значень параметрів l_r і l_c обмежений знизу міркуваннями колізійної стійкості, а зверху – міркуваннями зручності й зменшення ймовірності помилки при диктуванні криптограми по голосовому каналі зв'язку.

$H(x)$ – криптографічна хеш-функція, значення якої усікається до необхідного розміру.

Параметри $\{c\}$ пар таблиці будуть секретними параметрами.

На клієнті зберігається таблиця запитів активації, що є підмножиною першого стовпця таблиці пар на сервері. Розмір таблиці – 2^{16} записів, тобто 0.5 – 1 Мб. Вибір підмножини здійснюється довільно.

Активація що захищається ПЗ здійснюється в такий спосіб. Клієнт здійснює збір прив'язочних значень обчислювального середовища E, зберігає його й потім хеширует цей набір значень у двійковий вектор $e = h(E)$ довжиною 16 біт.

Для цього рекомендується використовувати криптографічний алгоритм хешування. Даний хеш буде індексом у таблиці запитів активації клієнта. Обраний запит $r[e]$ активації відправляється клієнтом на сервер. Сервер знаходить у своїй таблиці пар запит-підтвердження $c = H^{-1}(r[e])$ необхідне підтвердження й відправляє його клієнтові. Клієнт шляхом обчислення хеш-функції від підтвердження й порівняння результату із запитом активації $H(c) = r[e]$ переконується в автентичності підтвердження. Після чого клієнт зберігає підтвердження у своїй області даних.

Для перевірки прив'язки в обчислювальному середовищі, клієнт, аналогічно вищеописаному, здійснює збір прив'язочних значень обчислювального середовища *E*' і робить порівняння векторів *E* і *E*'. У випадку успішного порівняння, клієнт переконується у відповідності підтвердження активації збереженому еталону, тобто перевіряє рівність $H(c) = r[h(E)]$.

Розробка структурної схеми

Структурна схема розробленої системи зображена на рисунку 3. На ній показано структурні блоки, з яких складається система, та структурні взаємозв'язки між цими блоками.

Структурна схема складається з трьох основних блоків:

– Інтелектуальна мікропроцесорна карта з EEPROM.
– Програмне забезпечення на серверній частині.
– Банкомат (Картрідер).

Розглянемо ці блоки більш детально.

Інтелектуальна мікропроцесорна карта з EEPROM являє собою пластикову картку, яку можливо використовувати для операцій з грошима. EEPROM – (Electrically Erasable Programmable Read-Only Memory, електрично стираємий перепрограмувальний постійний запам'ятовувальний пристрій ЕСППЗП). Пам'ять такого типу може стиратися й заповнюватися даними кілька десятків тисяч разів. Використовується у твердотільних накопичувачах. Однією з різновидів EEPROM є флеш-память.

Структурно вона включає в себе наступні блоки:

– Блок автентифікації користувача.
– Блок захисту даних на пластиковій карті.
– Блок здійснення операцій над рахунком.

Блок автентифікації користувача включає в себе наступні дані:

– Дані про користувача – прізвище, ім'я та по батькові.
– PIN-код користувача.

Блок захисту даних на пластиковій карті включає в себе наступні складові:

– Номер банківського рахунку користувача.
– Кількість грошей на рахунку.

– Криптоалгоритм ДСТ 28147:2009, яким зашифровані перераховані вище дані користувача.

Блок здійснення операцій над рахунком включає в себе наступні операції:

– читання про стан рахунку;
– зняття грошей з рахунку;
– поповнення рахунку;
– переведення грошей на інший рахунок.

Розглянувши структурний склад інтелектуальної мікропроцесорної карти з EEPROM, перейдемо до розгляду іншої складової – програмного забезпечення на серверній частині.

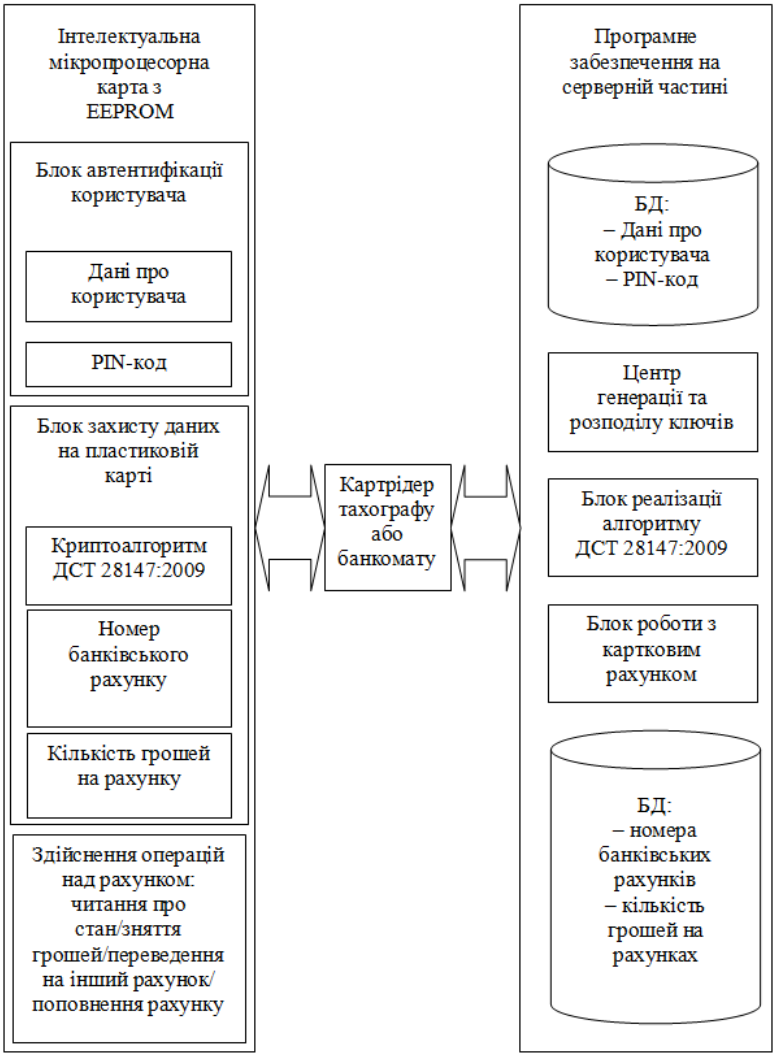


Рисунок 3 – Структурна схема системи

Програмне забезпечення на серверній частині складається з наступних структурних блоків:

– База даних про користувача.
– Центр генерації та розподілу ключів.
– Блок реалізації алгоритму ДСТ 28147:2009.
– Блок роботи з картковим рахунком.
– База даних про рахунок.

База даних про користувача включає в себе наступні дані:

– Дані про користувача – прізвище, ім’я та по батькові.
– PIN-код користувача.

База даних про рахунок включає в себе наступні дані:

– Номер банківського рахунку користувача.

– Кількість грошей на рахунку.

Опишемо алгоритм ДСТ 28147:2009. Для захисту даних на мікропроцесорній карті запропоновано використовувати алгоритм ДСТ 28147:2009, що є класичним алгоритмом симетричного шифрування на основі мережі Фейстеля (Рисунок 4). Даний алгоритм шифрує інформацію блоками по 64 біта (такі алгоритми називаються "блоковими"). Зміст мережі Фейстеля полягає в тому, що блок шифруемої інформації розбивається на два або більше субблоків, частина яких обробляється за певним законом, після чого результат цієї обробки накладається (операцією побітового додавання за модулем 2) на необроблювані субблоки. Потім субблоки міняються місцями, після чого обробляються знову й т.д. певне для кожного алгоритму число раз – раундів.

$L_i = R_{i-1}$

$R_i = L_i \oplus f\ (R_{i-1}, K_i)$

Функція F проста. Спочатку права половина й і-ий підключ складаються за модулем 2^{32}. Потім результат розбивається на вісім 4-бітових значень, кожне з яких подається на вхід S-box. ДЕРЖСТАНДАРТ 28147 використовує вісім різних S-boxes, кожний з яких має 4-бітовий вхід і 4-бітовий вихід.

Виходи всіх S-boxes поєднуються в 32-бітне слово, що потім циклічно зсувається на 11 бітів вліво. Нарешті, за допомогою XOR результат поєднується з лівою половиною, у результаті чого виходить нова права половина.

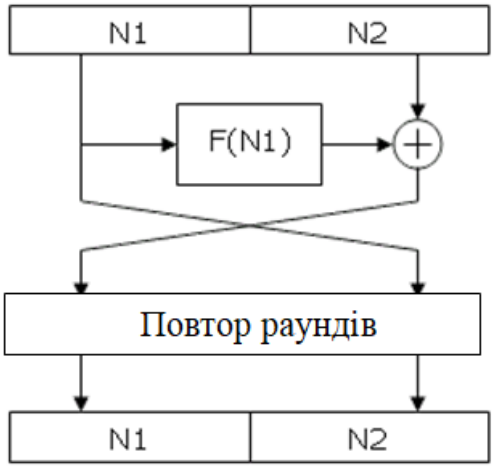


Рисунок 4 – Мережа Фейстеля

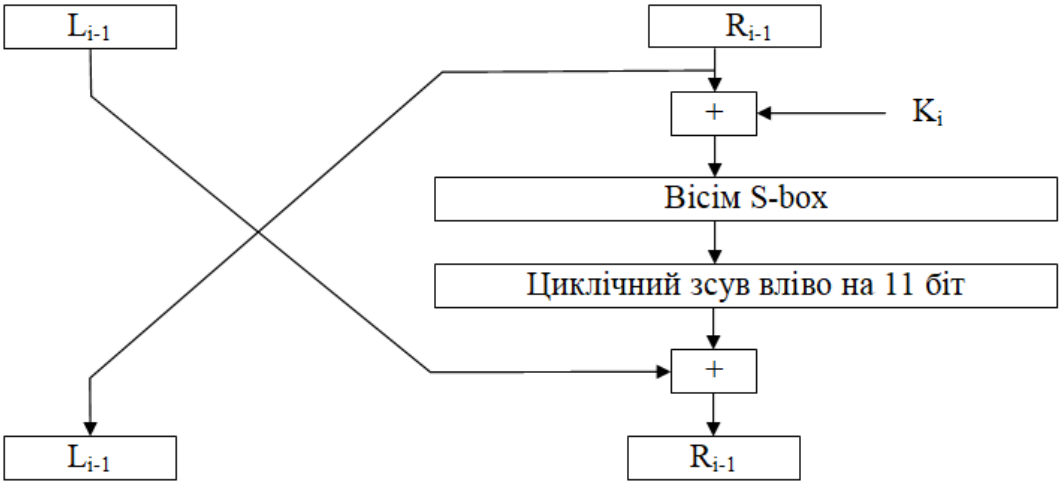


Рисунок 5 – і-ий раунд ГОСТ 28147:2009

Основна відмінність алгоритмів симетричного шифрування друг від друга складається саме в різних функціях обробки субблоків.

Дана функція часто називається "основним криптографічним перетворенням", оскільки саме вона несе основне навантаження при шифруванні інформації.

Основне перетворення алгоритму ДСТ 28147:2009 є досить простим, що забезпечує високу швидкодію алгоритму; у ньому виконуються наступні операції (Рисунок 6).

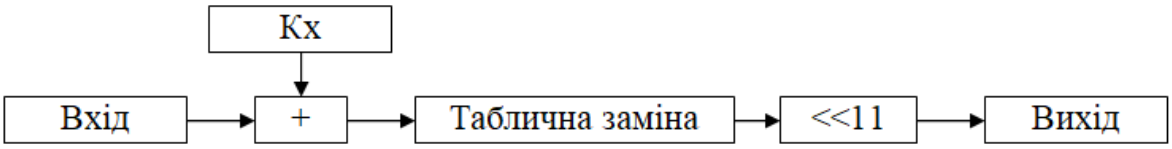


Рисунок 6 – Основне перетворення алгоритму ДСТ 28147:2009

1. Додавання субблоку з певним фрагменом ключа шифрування за модулем 2^{32}. Кх – це 32-бітна частина ("підключ") 256-бітного ключа шифрування, якому можна представити як конкатенацію 8 підключей: $К = К_0К_1К_2К_3К_4К_5К_6К_7$. Залежно від номера раунду й режиму роботи алгоритму (про їх – нижче), для даної операції вибирається один з підключей.

2. Таблична заміна. Для її виконання субблок розбивається на 8 4-бітних фрагментів, кожний з яких прогоняється через свою таблицю заміни. Таблиця заміни містить у певній послідовності значення від 0 до 15 (тобто всі варіанти значень 4-бітні фрагменти даних); на вхід таблиці подається блок даних, числове подання якого визначає номер вихідного значення. Наприклад, подається значення 5 на вхід наступної таблиці: "13 0 11 74 91 10 143 5 122 15 8 6". У результаті на виході виходить значення 9 (оскільки 0 заміняється на 13, 1 – на 0, 2 – на 11 і т.д.).

3. Побітове циклічне зрушення даних усередині субблока на 11 біт уліво.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів даних карти тахографа. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем даних карти тахографа. Досліджена система даних карти тахографа. На основі отриманих результатів досліджень створена програмна реалізація системи даних карти тахографа. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання даних карти тахографа. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Kovalenko Oleksandr Qualitative risk analysis of software development / Oleksandr Kovalenko, Jamil Al-Azzeh, Oleksii Smirnov, Anna Kovalenko, Serhii Smirnov // Asian Journal of Information Technology. – Volume 17 Issue 3. – Medwell Journals. – 2018. – P. 218-230. ISSN: 1682-3915. URL: http://medwelljournals.com/abstract/?doi=ajit.2018.218.230 Doi: ajit.2018.218.230

2. Kovalenko Oleksandr, The mathematical model of the testing technology for DOM XSS vulnerabilities / O. Kovalenko, O. Smirnov, A.Kovalenko, S. Smirnov, V. Vialkova // Scientific & practical cyber security journal (SPCSJ) Volume 2 Issue 1, P. 22-28. Georgia. Tbilisi. Scientific Cyber Security Association (SCSA), 2018 ISSN: 2587-4667.

3. Kovalenko O.V. Method of testing the dom xss vulnerability / Kovalenko Oleksandr, Kovalenko Anna, Smirnov Oleksii, Smirnov Serhii // International Conference «information technologies, systems and networks ITSN-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces "Alexandru cel Bun". – 2017. – P. 7.

4. Коваленко О.В. Метод тестування DOM XSS уразливості / О.В. Коваленко, О.А. Смірнов, А.С. Коваленко, С.А. Смірнов // Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті». м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.

5. Коваленко О.В. GERT-модель технології тестування DOM XSS уразливості / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов, С.А. Смірнов // Збірник наукових праць IV міжнародної науково-практичної

конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 21-24 лютого 2018 р. – Київ: Європейський університет. – 2018. – С. 65-70.
6. Коваленко О.В. Технології тестування уразливостей Web-застосунків з використанням GERT-моделі / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов, С.А. Смірнов // Збірник тез всеукраїнської науково-практичної конференції "Комп'ютерні інтелектуальні системи та мережі (КІСМ-2018)". м. Кривий Ріг. 21-23 березня 2018 р. – Кривий Ріг.: ДВНЗ КНУ – 2018. – С. 227-230.
7. Коваленко А.В. Тестирование уязвимости Web-приложений к атаке вида межсайтовый скриптинг / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // Збірник тез «Securitea informationala 2018». Conferenta internationala (editia a XIV-a). Chisinau. Moldova. 20-21 martie 2018. – Chisinau: ADSEM. – 2018. – P. 54-56.
8. Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // Збірник тез X міжнародної науково-практичної конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 19-20 квітня 2018 р. – Харків: ХНЕУ. – 2018. – С. 38.
9. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
10. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).
11. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14. (Scopus).
12. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
13. Smirnov O., Kuznetsov A., Pushkar’ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
14. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136. (Scopus).
15. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379. (Scopus).
16. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
17. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645. (Scopus).
18. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660., (Scopus).
19. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
20. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019. (Scopus).

УДК 004

Д.Ященко, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ФОРМУВАННЯ ПРОФІЛІВ ЗАХИСТУ ХМАРНИХ СЕРВІСІВ

У статті розроблено програмне забезпечення, яке призначено для системи формування профілів захисту хмарних сервісів. Метою розробки є дослідження та програмна реалізація системи формування профілів захисту хмарних сервісів. Об’єктом дослідження є процес формування профілів захисту хмарних сервісів. Предметом дослідження є методи формування профілів захисту хмарних сервісів. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи формування профілів захисту хмарних сервісів. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп’ютерна інженерія, профілі захисту, хмарні сервіси

Постановка проблеми. В останні роки в банківській діяльності загострилася проблема забезпечення безпеки даних. Вона містить у собі кілька аспектів. По-перше, це гнучка, багаторівнева й надійна регламентація повноважень користувачів – цінність банківської інформації висуває особливі вимоги до захисту даних від несанкціонованого доступу, у тому числі, до контролю керування процесами, що змінюють стан даних. По-друге, важливим аспектом є наявність засобів для підтримки цілісності й несуперечності даних; подібні засоби мають на увазі можливість здійснення контролю введення даних, підтримки й контролю зв'язків між даними, а також уведення й модифікації даних у режимі транзакцій – набір операцій, що забезпечують підтримку погодженості даних. По-третє, необхідна присутність у системі багатофункціональних процедур архівації, відновлення й моніторингу даних при програмних і апаратних збоях.

Забезпечення безпеки інформаційних банківських систем являє собою комплексну проблему, що вирішується в напрямках удосконалювання правового регулювання застосування інформаційних технологій, удосконалювання методів і засобів їхньої розробки, розвитку системи сертифікації, забезпечення відповідних організаційно-технічних умов експлуатації. Ключовим аспектом рішення проблеми безпеки є вироблення системи вимог, критеріїв і показників для оцінки рівня безпеки інформаційних технологій.

ДСТ ІСО/МЕК 15408 визначає критерії, за яких історично закріпилася назва "Загальні критерії" (ЗК). ЗК призначені для використання інформаційних технологій як основу при оцінці характеристик безпеки продуктів і систем. Установлюючи загальну базу критеріїв, ЗК роблять результати оцінки безпеки значимими для більше широкої аудиторії.

Сукупність вимог безпеки, узятих з ЗК або сформульованих у явному виді, представляється у вигляді профілю захисту, оцінка й обґрунтування якого виконується відповідно до критеріїв оцінки, що втримується в частині 3 ЗК. Метою такої оцінки є демонстрація того, що профіль повний, несуперечливий, технічно правильний і придатний для використання при викладі вимог до об'єкта оцінки, передбачуваному для оцінки.

Обґрунтування ж містить у собі наступне:

а) логічне обґрунтування цілей безпеки, що демонструє, що викладені цілі безпеки зіставлені з усіма аспектами середовища безпеки;

б) логічне обґрунтування вимог безпеки, що демонструє, що сукупність вимог безпеки придатна для досягнення цілей безпеки й порівнянна з ними.

Проте, відсутність методології економічного обґрунтування й оцінки профілю захисту в цей час приводить до відсутності прагнення до впровадження ЗК. Розробка моделі обґрунтування, що опирається на економічні показники діяльності, буде стимулювати впровадження ЗК у різні галузеві сфери, зокрема, у сферу банківських інформаційних технологій. Це дозволить значно підвищити рівень безпеки банківських інформаційних систем, збільшить довіру до них як з боку користувачів (банківських організацій), так і сторони кінцевих споживачів банківського продукту.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи формування профілів захисту хмарних сервісів.

Мета й завдання дослідження. роботи є дослідження та програмна реалізація системи формування профілів захисту хмарних сервісів.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем формування профілів захисту хмарних сервісів.
– Дослідження системи формування профілів захисту хмарних сервісів.
– Програмна реалізація системи формування профілів захисту хмарних сервісів.

Об'єктом дослідження є процес формування профілів захисту хмарних сервісів.

Предметом дослідження є методи формування профілів захисту хмарних сервісів.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Розглянемо основні принципи, методи формування й структуру профілю захисту, а також характеристику стану інформаційних систем банківської сфери

Представимо огляд сучасних інформаційних банківських систем, їхню структура. Основні функціональні модулі систем, що реалізують всі види банківських послуг:

– розрахунково-касове обслуговування юридичних осіб;
– обслуговування рахунків банків-кореспондентів;
– кредитні, депозитні й валютні операції;
– будь-які види внесків приватних осіб і операції по них;
– фондові операції; розрахунки за допомогою пластикових карток;
– бухгалтерські функції;
– аналіз, прийняття рішень, менеджмент, маркетинг і ін.

Освітимо найбільш перспективні напрямки розвитку банківських інформаційних технологій, такі як:

– інтернет-банкінг;
– системи дистанційного обслуговування: «Інтернет-банк», «Інтернет-клієнт», домашній банк, телебанк, мобільний банк або WAP-сервіс.

З їхньою допомогою задовольняються практично будь-які, крім касового обслуговування, вимоги клієнтів банку. Освоєння українськими кредитними організаціями нових напрямків розвитку брокерських послуг полягає в наданні фізичним особам доступу до українських і міжнародних валютних і фондових ринків (інтернет-трейдінг).

Розглянемо стандарти в області інформаційної безпеки:

– міжнародні й національні стандарти оцінки керування інформаційною безпекою;
– галузеві стандарти забезпечення безпеки в банківській сфері;
– стандарти й рекомендації в області стандартизації:

а) забезпечення інформаційної безпеки організацій банківської системи України. Загальні положення СТО БР ІББС-1.0-2006;

б) методика оцінки відповідності інформаційної безпеки організацій банківської системи України вимогам СТО БР ІББС-1.0-2006;

в) посібник із самооцінки відповідності інформаційної безпеки організацій банківської системи України вимогам СТО БР ІББС-1.0-2006;

– аудит інформаційної безпеки.

Розглянемо стандарти й методичні рекомендації, присвячені формуванню й оцінці профілів захисту й завдань по безпеці відповідно до ДСТ ІСО/МЕК 15408. Представлено існуючі на даному етапі способи формування профілів захисту й завдань по безпеці. Особлива увага приділена складеним об'єктам оцінки (складається із двох і більше компонентів), якими і є в деяких випадках автоматизовані банківські системи. На рисунку 1 представлені два види складених об'єктів оцінки (ОО), з єдиним і різним середовищем безпеки.

Запропонуємо конкурентну модель СЗІ автоматизованих банківських систем комерційних банків і метод обґрунтування профілю захисту на основі даної моделі.

Введемо поняття конкуренції стосовно до інформаційних систем. Інваріантність даного поняття дає підставу припускати можливість побудови деякої математичної моделі даного процесу. Конкуренція, одержавши широке поширення в теорії еволюції біологічних і економічних систем, а також інших сферах, таких як політика, історія науки, утворення, мистецтво, соціальна психологія й навіть фізика, дозволяє використовувати дане поняття й у сфері інформаційних технологій. Зокрема, в області інформаційної безпеки.

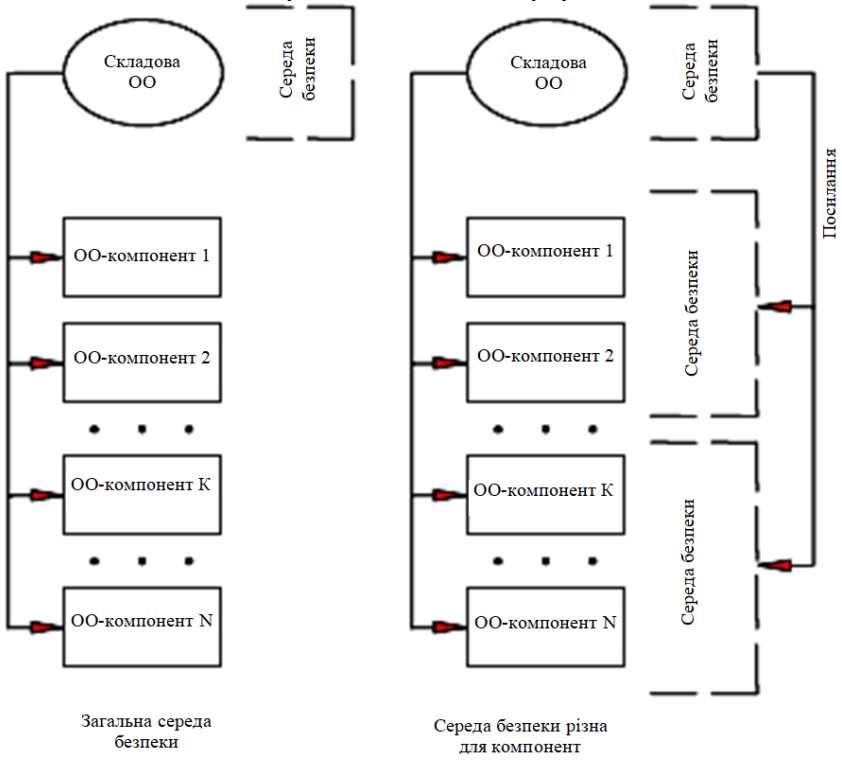


Рисунок 1 – Складені об'єкти оцінки ОО

Опишемо вимоги, запропоновані до моделі як до відбиття однієї зі сторін взаємодії, й не є повним функціональним аналогом реальної системи, а також вимоги до вибору економічного показника, що не повинен бути вузькоспеціалізованим параметром, характерним тільки для деяких систем, а повинен бути властивий всім учасникам взаємодії. Як такий показник запропонований використовувати чисті активи (нетто-активи), обумовлені як різниця між активами й пасивами:

$$A = AK - П, \quad (1)$$

де:

– A – чисті активи (грн.);
– AK – активи (грн.);
– $П$ – пасиви (грн.).

Для виключення впливу таких макроекономічних показників як середні доходи населення, середня заробітна плата, ціни, рівень інфляції, безробіття, зайнятість, продуктивність праці вводиться поняття нормалізованого активу:

$$AN_i = \frac{A_i}{\sum_{i=1}^{n} A_i}, \tag{2}$$

де:

– AN – нормалізований актив;
– A – чистий актив (грн.);
– n – число діючих кредитних організацій.

У графічному виді представлена й проаналізований взаємозв'язок між чистими й нормалізованими активами п'яти найбільших банків.

Введемо визначення конкурентоспроможності як суми показника захищеності ($З$) і показника росту активів ($Н$):

$$КС = З + Н, \tag{3}$$

де:

– $КС$ – нормалізований актив;
– $З$ – захищеність;
– $Н$ – приріст активу.

Формулювання конкурентної моделі взаємодії систем СЗІ автоматизованих банківських систем комерційних банків представлені на рисунку 2.

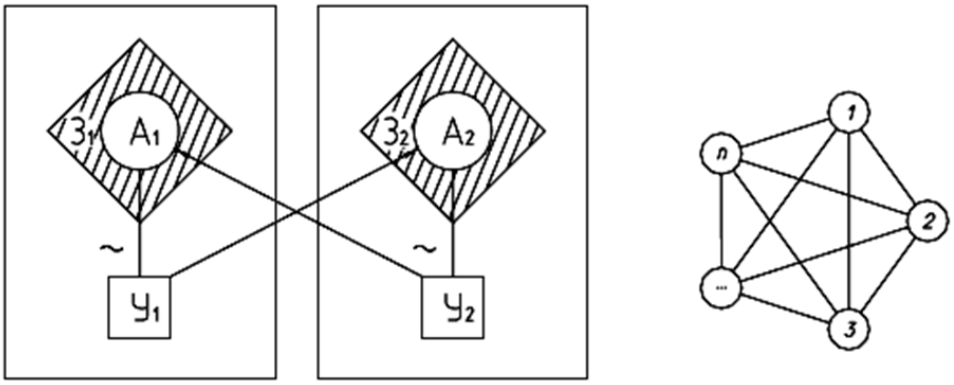


Рисунок 2 – Концептуальна схема конкурентної моделі СЗІ АБС КБ

Суб'єкт 1 має актив ($А_1$), що володіє тією або іншою захищеністю ($З_1$) від погроз, що виходять від другого ($В_2$). І, у той же час, сам формує погрозу ($В_1$), спрямовану на інший суб'єкт:

$$З_i(t) = \frac{\Delta AN_i(t)}{У_i}. \tag{4}$$

Захищеність суб'єкта ($З$) – це зміна нормалізованого активу AN від часу t-1 до часу t під впливом погрози ($В$):

$$\Delta AN_i(t) = AN_{i(t)} - AN_{i(t-1)}. \tag{5}$$

Погроза – імовірність втрати активу в одиницю часу. Як вихідне положення поле погроз для всіх суб'єктів системи єдино. Розходження погроз обумовлене як потужністю активу як з боку джерела, так і мети спрямованості погроз.

$$У_i = К_i \times \left(\sum_{i=1}^{n} К_i - К_i \right), \tag{6}$$

де:

– $У$ – погроза;
– $К$ – потужність активу.

Ріст своїх активів у результаті реалізації погроз, спрямованих на інші, у конкурентній моделі СЗІ автоматизованих банківських систем комерційних банків:

$$H = \frac{A_i^t - \sum_{i=1}^{n} A_i^t / \sum_{i=1}^{n} A_i^t \times A_i^{t-1}}{\sum_{i=1}^{n}\left(A_i^t - \sum_{i=1}^{n} A_i^t / \sum_{i=1}^{n} A_i^t \times A_i^{t-1} \right)}, \quad (7)$$

де:
H – приріст активу;
A – чистий актив (грн.).

У випадку, коли система не задовольняє вимозі конкурентоспроможності, за результатами експлуатації буде потрібно внесення розроблювачем виправлень в об'єкт оцінки, а також перевизначення вимог безпеки й/або припущень щодо середовища експлуатації, що спричинить перегляд профілю захисту.

Викладені вище положення не суперечать п.4.2.2 (Оцінка ОО) Державного стандарту України ДСТ ІСО/МЕК 15408-1-2002, у якому регламентоване, що процес оцінки може проводитися як паралельно з розробкою, так і слідом за нею.

Опишемо метод виконання оцінки на базі моделюючого комплексу. Метод містить у собі наступні етапи:

1. Одержання вихідних даних і прийняття ряду положень.

2. Розрахунок необхідних показників (нормалізований актив, захищеність, потужність активу, зміна активу за рахунок перерозподілу, конкурентоспроможність).

3. Ухвалення рішення про відповідність профілю захисту.

1-й етап: Збір статистичних даних, отриманих з відкритих джерел, за результатами діяльності банків і кредитних організацій.

Достатньою умовою є одержання даних по 500 самих великих організаціях. Вплив інших незначний, дані по них екстраполюються. Одержання свідчення про відповідність профілю захисту конкретної реалізації системи.

2-й етап: Обчислення нормалізованого активу за формулою (2): n – приймаємо рівним 1125, а потім і на часовому інтервалі від t-1 до t (5). Величина інтервалу приймається залежно від даних, отриманих на 1-му етапі (0,5 року). Визначаємо потужність активу кожного суб'єкта відповідно до його активу.

Ріст активів у результаті реалізації погроз, спрямованих на інші, являє собою відхилення від середнього приросту активів протягом часу від від t-1 до t. Обчислюється за формулою (7).

Обчислення захищеності виробляється за формулою (4), де B приймається за формулою (6). Таким чином, величина погрози B являє собою добуток потужності власного активу на суму потужностей активів інших організацій. Безпосереднє значення погрози, як імовірності втрати активу, не враховується, тому що є константою для всіх суб'єктів. При обчисленні значення захищеності вхідними параметрами виступають величина суб'єкта і його нормалізований актив.

Останнім значенням обчислюється параметр КС (конкурентоспроможність). Результатом є сума показників захищеності й показника росту активу.

3-й етап: Ухвалення рішення про відповідність профілю захисту на підставі показника КС. Для успішного розвитку в майбутньому показник КС повинен приймати, як мінімум, значення вище за середнє, тому що захищеність має тенденцію до зниження протягом часу, а разом з нею знизиться й показник КС .

Розробка структурної схеми

Структурна схема розробленої системи зображена на рисунку 3. В основному інтелектуальні засоби захисту інформації (СЗІ) знайшли своє застосування в системах

виявлення атак як інтелектуальний інструмент, у яких, як правило, використовуються нейронні мережі (НМ), системи нечіткої логіки (НЛ) і засновані на правилах експертні системи (ЕС) [1].

Схеми виявлення атак розділяють на дві категорії:

1) виявлення зловживань;

2) виявлення аномалій.

До першої відносять атаки, які використовують відомі уразливості інформаційної системи (ІС), а до других – невластиву користувачам ІС діяльність.

Для виявлення аномалій виявляється діяльність, що відрізняється від шаблонів, установлених для користувачів або груп користувачів. Виявлення аномалій, як правило, пов'язане зі створенням бази знань (БЗ), що містить профілі контрольованої діяльності [2], а виявлення зловживань – з порівнянням діяльності користувача з відомими шаблонами поводження хакера [3] і використовує методи на основі правил, що описують сценарії атак. Механізм виявлення ідентифікує потенційні атаки у випадку, якщо дії користувача не збігаються із установленими правилами.

Завдання класифікації в експертних системах

Експертні системи (рисунок 3) призначені для рішення класифікаційних завдань у вузькій предметній області виходячи з бази знань, сформованої шляхом опитування кваліфікованих фахівців і представленою системою класифікаційних правил *If-Then* (*Якщо – Тоді*) [4]. У системах забезпечення безпеки ІС експертні системи використовуються в інтелектуальних СЗІ на основі моделі [5] і містять у БЗ опис класифікаційних правил, що відповідають профілям легальних користувачів ІС, сценаріям атак на ІС [6].

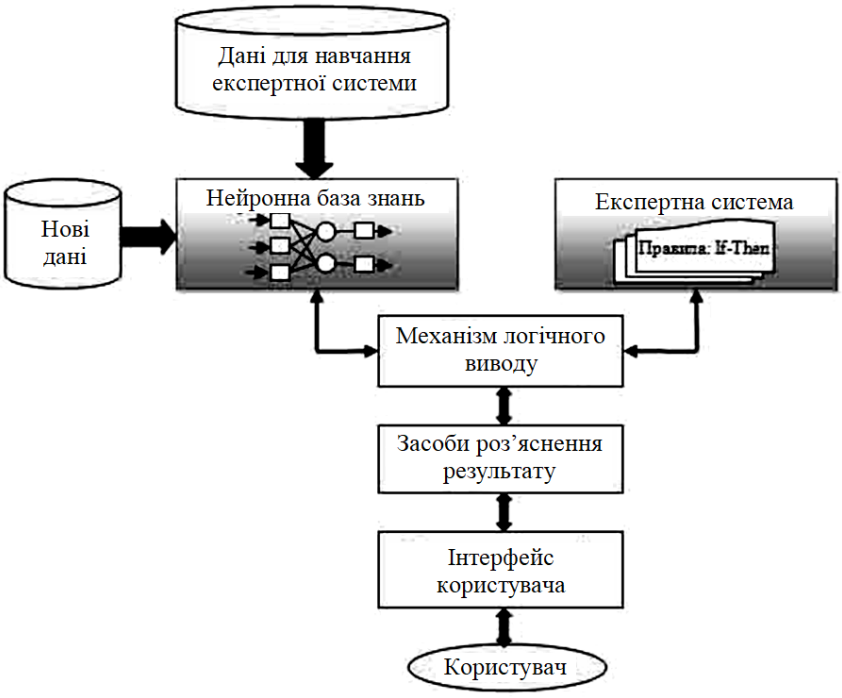


Рисунок 3 – Структурна схема системи

До недоліків ЕС, як засобів класифікації, відносять [7]:

– Непрозорість зв'язків між окремими правилами в базі знань. Хоча окремі правила відносно прості й логічно прозорі, наочність їхнього логічного взаємозв'язку в межах БЗ має бути досить низькою, тобто не просто визначити суперечливі правила в БЗ і їхню роль у рішенні завдання.

– Неефективна стратегія пошуку. ЕС із великою базою знань можуть виявитися недостатньо продуктивними для рішення оперативних завдань забезпечення безпеки ІС у реальному масштабі часу.

– Відсутність можливості адаптації. ЕС не мають здатність до автоматичного

навчання, тобто ЕС не може автоматично змінювати БЗ, коректувати існуючі правила або додавати нові правила *If-Then.*

Імовірнісні методи рішення завдання класифікації

Методи недостовірного керування й імовірнісних міркувань застосовуються в ЕС не тільки для формулювання класифікаційних висновків відповідно до правил *If-Then,* але й формування оцінок вірогідності проведеної класифікації у вигляді значень значення фактора впевненості або умовної ймовірності виникнення класифікуємої події.

Можливість оцінки вірогідності пророкування є достоїнством методу імовірнісних міркувань Байеса, тому що базується на математичному апараті теорії ймовірностей [4]. У вітчизняній практиці метод імовірнісних міркувань Байеса застосовують у ряді прикладних ЕС [8].

Практична значимість методу факторів упевненості для рішення завдання класифікації підтверджена розробкою ряду експертних систем [9]. Останній підхід до класифікації більше прийнятний з погляду обчислювальної ефективності, тому що не вимагає наявності більших обсягів статистичних даних і складних розрахунків умовних ймовірностей при великій розмірності простору вхідних посилок [7].

У всіх додатках забезпечення безпеки ІС на основі експертних систем може успішно застосовуватися підхід формування й підтримки класифікаційної бази знань відповідно до методів недостовірного керування й імовірнісних міркувань.

Чисельна оцінка класифікаційних висновків особливо важлива в умовах неповноти й низкою вірогідності вхідних ознак, використовуваних як посилки більшістю існуючих систем класифікації вторгнень у комп'ютерні системи.

Завдання нечіткої класифікації

Нечітка класифікація є подальшим розвитком підходу до рішення експертними системами завдань класифікації. Достоїнство нечіткої класифікації – можливість формулювати достовірні класифікаційні висновки виходячи з неповних і не цілком достовірних вхідних посилок [7].

При збереженні математичного апарата, розробленого для систем чіткої логіки, у нечітких логічних системах вирішене завдання перетворення чисельної і якісної інформації в ступінь приналежності значень конкретним нечітким безлічам [10]. Нечіткі безлічі описуються за допомогою функцій приналежності, що ставлять у відповідність безлічі значень із області визначення безперервної змінної безліч значень істинності з інтервалу [0, 1].

Етапи нечіткого логічного виводу безпосередньо пов'язані із процесом формування класифікаційних висновків [11]:

1) етап введення нечіткості (fuzzification) пов'язаний з перетворенням по засобом вхідних функцій приналежності (input membership functions) кожного із чітких вхідних значень x_i , $i = 1,\ldots,n$, де n – число вхідних значень класифікатора (crisp inputs) у ступінь істинності відповідної посилки μ_{xi}, $i= 1,\ldots,m$, де m-для кожного із класифікаційних правил (fuzzy rules);

2) етап нечіткого логічного виводу відповідає формуванню висновку (відповідні нечіткі підмножини) по кожному із правил μ_{Ri}, $i = 1,\ldots,m$, де m – кількість класифікаційних правил, виходячи зі ступеня істинності посилок μ_{xi}, $i= 1,\ldots,n$;

3) етап композиції нечітких підмножин по кожному із правил μ_{Ri}, $i = 1,\ldots,m$ за допомогою вихідних функцій приналежності (output membership functions) з метою формування нечітких підмножин класифікаційних висновків μ_{Ci}, $i = 1,\ldots,p$, де p – число виходів класифікатора;

4) етап об'єднання (aggregation) нечітких підмножин μ_{Ci}, $i = 1,\ldots,p$ і приведення до чіткості (defuzzification) приводить до формування вихідного чіткого значення $в$.

Нечіткі логічні системи зберігають у своєму составі базу знань кваліфікованих фахівців ІБ у вигляді системи правил *If-Then,* однак розширюють область застосування ЕС за рахунок рішення завдання класифікації виходячи з неповної й не цілком достовірної

інформації.

Системи НЛ мають обмежені можливості до адаптації, тому що можуть навчатися шляхом зміни параметрів функцій приналежності під реальні значення вхідних даних і бажаних класифікаційних висновків. Варто відзначити, що процес навчання функцій приналежності нечіткої ЕС із досить великою базою знань (понад 100 правил) трудомісткий і вимагає значних витрат часу [7].

Застосування НМ у завданнях класифікації й кластеризації

Нейронні мережі найбільше часто використовують для рішення завдань класифікації. Доведено, що НМ є універсальним аппроксіматором, тобто будь-яка функція може бути представлена у вигляді багатошарової НМ із формальних нейронів з нелінійною функцією активації. Формально підтверджена верхня границя складності НМ, що реалізує довільну безперервну функцію від декількох аргументів. Нейронною мережею з одним схованим шаром і прямими повними зв'язками можна представити будь-яку безперервну функцію, для чого досить у випадку n-мірного вхідного вектора $2n+1$ ФН схованого шару із заздалегідь застереженими обмеженими функціями активації [10].

Відомі численні застосування нейромережних засобів для забезпечення безпеки ІС, причому більшість випадків пов'язане з рішенням завдань класифікації й кластеризації [10].

Варто врахувати, що із всіх розглянутих раніше інтелектуальних засобів тільки НМ обдають властивістю самоорганізації, вирішує використовувати їх для рішення завдання кластеризації.

Можливість самоорганізації розглядається як одне з найбільш важливих якостей нейромережних СЗІ, вирішує адаптуватися до зміни вхідної інформації. Навчальним фактором виступають присутні в даних сховані закономірності й надмірність вхідної інформації. Інформаційна надмірність дозволяє фіксувати в інформаційному полі НМ вхідні дані, представляючи їх у більше компактній формі. Зменшення ступеня надмірності інформації в адаптивних СЗІ дозволяє виділяти істотні незалежні ознаки в даних.

Самоорганізація НМ реалізується за рахунок механізму кластеризації: подібні вхідні дані групуються нейронною мережею відповідно до взаємної кореляції й представляються конкретним ФН-прототипом. НМ, здійснюючи кластеризацію нечітких даних, знаходить такі усереднені по кластеру значення ваг ФН-прототипів, які мінімізують помилку подання згрупованих у кластер даних.

– Розглянуті механізми класифікації й кластеризації вхідних даних у СЗІ дозволяють не тільки відносити класифікуємий об'єкт (вектор вхідних даних) до одного з відомих класів, але й реалізувати еволюційні процеси самоорганізації, адаптації, розвитку в інтелектуальних засобах забезпечення інформаційної безпеки ІС. Причому кращі функціональні характеристики виходять при сполученні різних інтелектуальних засобів у гібридній системі захисту інформації.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів формування профілів захисту хмарних сервісів.

Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем формування профілів захисту хмарних сервісів; Досліджена система формування профілів захисту хмарних сервісів; На основі отриманих результатів досліджень створена програмна реалізація системи формування профілів захисту хмарних сервісів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання формування профілів захисту хмарних сервісів.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Kovalenko Oleksandr Qualitative risk analysis of software development / Oleksandr Kovalenko, Jamil Al-Azzeh, Oleksii Smirnov, Anna Kovalenko, Serhii Smirnov // Asian Journal of Information Technology. – Volume 17 Issue 3. – Medwell Journals. – 2018. – P. 218-230. ISSN: 1682-3915.
2. Kovalenko Oleksandr, The mathematical model of the testing technology for DOM XSS vulnerabilities / O. Kovalenko, O. Smirnov, A.Kovalenko, S. Smirnov, V. Vialkova // Scientific & practical cyber security journal (SPCSJ) Volume 2 Issue 1, P. 22-28. Georgia. Tbilisi. Scientific Cyber Security Association (SCSA), 2018 ISSN: 2587-4667.
3. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.
4. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12..
5. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022..
6. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34..
7. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477..
8. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w.
9. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143.
10. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207..
11. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58..
12. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256..
13. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114..
14. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346..
15. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131..
16. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14..
17. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». Lecture Notes in Networks and Systems, vol 152. Springer, Cham. 2021, pp 66-84..
18. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587..
19. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». CEUR Workshop Proceedings Volume 2616, 2020, Pages 125-136..
20. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379..

ЗМІСТ