Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів”

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-24М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Плужник В.О.
«____» _____ 2025 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Марченко К.М.
«____» _____ 2025 р.
Рецензент _____
_____

м. Кропивницький

АНОТАЦІЯ

Плужник В.О. Дослідження та програмна реалізація системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

Метою розробки є дослідження та програмна реалізація системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

Об'єктом дослідження є процес управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

Предметом дослідження є методи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

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

Результат роботи – програмна реалізація системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, ідентифікація, доступом, мережеві інформаційні ресурси

ABSTRACT

Pluzhnyk V.O. Research and software implementation of the identification and access management system for network information resources. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the identification and access management system for network information resources.

The purpose of the development is the research and software implementation of the identification and access management system for network information resources.

The object of the research is the process of managing identification and access to network information resources.

The subject of the research is the methods of managing identification and access to network information resources.

The research methods are based on information protection methods, methods of the theory of building computer networks, methods of mathematical statistics, methods of software development.

The result of the work is a software implementation of the identification and access management system for network information resources.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with the software are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Python environment.

Keywords: computer engineering, identification, access, network information resources

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЕЦП	–	електронний цифровий підпис
ІТ	–	інформаційні технології
КУМЗ	–	класи уніфікованих математичних завдань
ОМЗ	–	основні математичні завдання
ПЗ	–	програмне забезпечення
СКЗІ	–	система контролю та захисту інформації
СКУД	–	система контролю й управління доступом у приміщення
ОАОР	–	загальний річний ріст
OTP	–	OneTime Password
PKI	–	інфрастуктура відкритих ключів
USB	–	universal serial bus

ВСТУП

Актуальність теми. Інструменти мережевої безпеки спрямовані на запобігання несанкціонованому доступу до даних, крадіжці особистих даних та кіберзагрозам до пристроїв, технологій та процесів.

Мережева безпека запобігає несанкціонованому доступу до інформації або зловживанню мережею організації. Вона включає апаратні та програмні технології, розроблені для захисту безпеки та надійності мережі та даних.

Інструменти мережевої безпеки є важливими для захисту мережі вашої організації та запобігання кільком загрозам, які можуть пошкодити систему та мережу. Вони допомагають контролювати мережу та запобігати витокам даних.

Інструмент мережевої безпеки може аналізувати весь трафік у мережі. Моніторинг трафіку допомагає організації проактивно виявляти проблеми та загрози, перш ніж вони завдадуть їй значної шкоди. Інструменти мережевої безпеки надсилають сповіщення в режимі реального часу про будь-яку незвичайну поведінку, щоб запобігти будь-яким порушенням.

Деякі переваги інструментів мережевої безпеки:

– Інструменти мережевої безпеки мінімізують бізнес- та фінансовий вплив будь-якого порушення, оскільки вони допомагають вам дотримуватися правил та запобігати порушенням.

– Мережева безпека допомагає вашому бізнесу дотримуватися вимог і забезпечує кілька рівнів безпеки, щоб розширити масштаби вашого бізнесу та запропонувати краще робоче місце для ваших співробітників.

– Це забезпечує захист будь-якої конфіденційної інформації та даних, що передаються по мережі.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

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

– Огляд існуючих систем управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

– Дослідження системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

– Програмна реалізація системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

Об'єктом дослідження є процес управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

Предметом дослідження є методи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

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

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

– Розроблено вітчизняний продукт управління ідентифікацією та доступом до мережевих інформаційних ресурсів, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2025 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Ви коли-небудь замислювалися над будь-яким із наступних питань:

1. Як ми можемо однозначно ідентифікувати різних користувачів системи?

2. Як користувач може підтвердити системі свою особу?

3. Як ми можемо запобігти зловмиснику, який видає себе за законного користувача?

4. Як ми можемо вирішити, до чого користувач повинен мати доступ? Як ми можемо забезпечити виконання такого рішення?

5. Як ми можемо дізнатися, що робить користувач після входу в систему, щоб ми могли притягнути його до відповідальності за свої дії?

У даній роботі відповімо на вищезазначені та інші запитання, використовуючи формальні технічні терміни. Відповіді на вищезазначені запитання криються в наступних концепціях та процесах:

1. Ідентифікація.

2. Автентифікація.

3. Надійні паролі та багатофакторна автентифікація (MFA).

4. Авторизація та контроль доступу.

5. Ведення журналу та аудит.

У цій роботі ознайомлемо із ключовими концепціями, пов’язаними з управлінням ідентифікаторами та авторизаціями. Вона починається з ідентифікації та автентифікації, переходить до підзвітності, а завершується моделями контролю доступу.

Ідентифікація, автентифікація, авторизація та підзвітність (IAAA) – це чотири основи інформаційної безпеки. Кожен із цих елементів відіграє важливу

роль у забезпеченні конфіденційності, цілісності та доступності конфіденційної інформації та ресурсів.

Чотири етапи моделі IAAA:

1. Ідентифікація – це процес перевірки особи користувача. Він починається з того, що користувач заявляє про певну особу. Ідентичність може бути представлена унікальним ідентифікатором, таким як адреса електронної пошти, ім'я користувача або ідентифікаційний номер. Будь-який ідентифікатор, унікальний у відповідному середовищі, є допустимим варіантом; отже, багато веб-сайтів покладаються на адресу електронної пошти для ідентифікації, замість того, щоб просити користувача створити унікальне ім'я користувача.

2. Автентифікація – це процес підтвердження того, що користувач є тим, за кого він себе видає. Іншими словами, цей крок полягає у підтвердженні заявленої особи. Одним із способів автентифікації є надання правильного пароля. Через потенційні слабкі місця паролів набирають популярності багато інших методів, таких як прохання до користувачів ввести код, надісланий на їхню електронну пошту.

3. Авторизація визначає, до чого користувач має доступ. Іншими словами, він буде уповноважений виконувати певні операції на основі привілеїв свого облікового запису. Цей процес зазвичай здійснюється шляхом призначення ролей та дозволів на основі посадової функції користувача або рівня допуску. Ризик несанкціонованого доступу або витоку даних зменшується шляхом обмеження доступу лише до ресурсів, необхідних користувачеві для виконання своїх обов'язків.

4. Підзвітність відстежує активність користувачів, щоб забезпечити їхню відповідальність за свої дії. Після того, як користувачеві надається доступ до системи, важливо мати механізми, які притягують кожного до відповідальності за його дії. Цей процес досягається шляхом реєстрації всієї активності користувачів та її зберігання в централізованому місці. У разі інциденту безпеки цю інформацію можна використовувати для визначення джерела проблеми та вжиття

відповідних заходів.

ІААА допомагає запобігти несанкціонованому доступу, витокам даних та іншим інцидентам безпеки. Впроваджуючи ці найкращі практики, організації можуть захистити свою конфіденційну інформацію та ресурси від внутрішніх та зовнішніх загроз.

1.2 Область застосування

Ідентифікація – це спосіб, яким користувач (або процес, чи система) заявляє про свою особу. Розглянемо кілька прикладів з нашого повсякденного життя. Припустимо, вас запросили на вечірку, і ви почали спілкуватися, а потім хтось запитує вас: «Як вас звати?». І, найімовірніше, ви скажете їм своє справжнє ім’я. Зрештою, ви тут, щоб добре провести час і завести нових друзів. Зачекайте хвилинку, ви також можете придумати ім’я або вибрати щось зі свого улюбленого фільму та відповісти чимось на кшталт «Томас Андерсон»! У будь-якому випадку, інша людина не проситиме вашого посвідчення особи, щоб підтвердити вашу особу; це просто спілкування, а не доступ до зони високого рівня безпеки.

Ідентифікація може здійснюватися за допомогою імені користувача. Ім'я користувача може мати різні форми. Наприклад, Thomas Anderson може бути tanderson, thomasa, thomas01або ta001навіть neo. Все залежить від організації та платформи.

Ідентифікацію також можна здійснити за допомогою такого номера, як:

– Ідентифікаційний номер.

– Номер студентського квитка.

– Номер паспорта.

– Номер мобільного телефону.

Для ідентифікації може бути використано будь-який унікальний номер користувача. Багато веб-сайтів запитують адресу електронної пошти для

реєстрації, оскільки електронна адреса користувача гарантовано унікальна; це позбавляє користувача необхідності намагатися знайти унікальне ім'я користувача та запам'ятовувати його.

Розглянемо інший приклад поза комп'ютерами. Ви щойно почали ходити до спортзалу, і адміністратор запитав ваше ім'я. Ви можете відповісти ім'ям зі свого улюбленого фільму; проте цього разу адміністратор може попросити вас пред'явити посвідчення особи. Навіщо їм це робити? Щоб підтвердити, що ви той, за кого себе видаєте, це і є автентифікація. Вони не хочуть дозволити якомусь випадковому хлопцеві зайти до їхнього спортзалу та назвати ім'я одного зі своїх платних передплатників. Якщо будь-хто може зайти та назвати себе передплатником, власники спортзалів не зможуть мати прибутковий бізнес.

Без належної автентифікації можна завдати серйозної шкоди; розглянемо випадок, коли хтось видає себе за фальшиву особу під час отримання кредиту в банку. У світі ІТ без автентифікації будь-хто може отримати доступ до вашої електронної пошти, якщо знає вашу адресу електронної пошти. Більшість систем не можуть належним чином функціонувати без належної автентифікації; системи не обмежуються комп’ютерними системами та включають банківські системи, системи бронювання готелів та системи авіарейсів, серед багатьох інших.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Розглянемо 10 найкращих інструментів мережевої безпеки.

1. Wireshark

Wireshark – це аналізатор мережевих протоколів з відкритим кодом, який допомагає організаціям збирати дані в режимі реального часу, а також відстежувати, керувати та аналізувати мережевий трафік навіть з найдрібнішими деталями.

Це дозволяє користувачам переглядати потоки, відновлені після сеансу TCP. Це допомагає аналізувати вхідний та вихідний трафік для усунення проблем із мережею.

Особливості:

– Глибока перевірка сотень протоколів.

– Збір даних у режимі реального часу та офлайн-аналіз.

– Він працює на кількох операційних системах, таких як Windows, Linux, macOS тощо.

– Він надає кольорові коди кожному пакету для швидкого аналізу.

Переваги:

– Підтримує кілька операційних систем, таких як Windows, Linux тощо.

– Легко інтегрується зі сторонніми програмами.

Недоліки:

– Крута крива навчання.

– Важко читати зашифрований мережевий трафік.

– Відсутність підтримки.

2. Nexpose

Nexpose – це програмне забезпечення для мережевої безпеки, яке надає інформацію про вразливості в режимі реального часу та зменшує загрози в мережі. Крім того, Nexpose дозволяє користувачам присвоювати виявленим вразливостям оцінку ризику, щоб вони могли бути пріоритетизовані відповідно до рівнів безпеки.

Nexpose допомагає IT-командам сканувати мережу в режимі реального часу та виявляти вразливості мережі. Він також постійно оновлює та адаптується до нових загроз у програмному забезпеченні та даних.

Особливості:

– Nexpose надає дані про мережевий трафік у режимі реального часу.

– Він надає оцінку ризику та допомагає IT-командам пріоритезувати ризики відповідно до рівнів безпеки.

– Це показує IT-командам різні дії, які вони можуть негайно вжити для зменшення ризику.

Переваги:

– Легкий у використанні

– Поглиблене сканування мережевих вразливостей.

Недоліки:

– Немає автентифікації на основі домену для пристроїв Linux

– Відсутність підтримки клієнтів

3. Splunk

Splunk використовується для моніторингу безпеки мережі. Він забезпечує як аналіз даних у режимі реального часу, так і пошук історичних даних.

Це хмарна платформа, яка надає аналітику даних у петабайтному масштабі в гібридній хмарі.

Функція пошуку Splunk робить моніторинг програм простим та зручним для користувача.

Він містить інтерфейс користувача для збору, індексації та збору даних, а

також створення сповіщень, звітів, інформаційних панелей та графіків у режимі реального часу.

Особливості:

– Splunk приписує ризик користувачам і системам, зіставляє сповіщення з системами кібербезпеки та запускає сповіщення, коли ризик перевищує поріг.

– Це допомагає встановлювати пріоритети сповіщень та пришвидшувати розслідування завдяки вбудованій аналітиці загроз.

– Це допомагає отримувати автоматичні оновлення контенту безпеки, щоб бути в курсі нових загроз.

Переваги:

– Індексація даних проста.

– Легкий у використанні.

Недоліки:

– Крута крива навчання.

4. Nagios

Nagios – це інструмент мережевої безпеки, який допомагає контролювати хости, системи та мережі. Він надсилає сповіщення в режимі реального часу. Ви можете вибрати, які саме сповіщення ви хочете отримувати.

Він може відстежувати мережеві ресурси, такі як HTTP, NNTP, ICMP, POP3 та SMTP. Це безкоштовний інструмент.

Особливості:

– Nagios допомагає контролювати компоненти IT-інфраструктури, включаючи системні метрики, мережеві протоколи, служби додатків, сервери та мережеву інфраструктуру.

– Він надсилає сповіщення, коли виявляється неавторизована мережа, та повідомляє IT-адміністратора про важливі події.

– Він надає звіти, які відображають історію подій, сповіщень та відповідей на сповіщення для подальшого перегляду.

Переваги:

– Чудовий інструмент для моніторингу в реальному часі.

– Зручний для користувача.

– Моніторинг даних можна легко відстежувати.

Недоліки:

– Обмежені можливості звітності.

– Система сповільнюється під час моніторингу даних.

5. Tor

Tor – це інструмент мережевої безпеки, який забезпечує конфіденційність користувачів під час користування Інтернетом. Він допомагає запобігати кіберзагрозам та корисний для захисту інформаційної безпеки.

Tor працює за концепцією цибулевої маршрутизації, а шари накладаються один на один, подібно до цибулевої маршрутизації. Усі шари функціонують розумно, тому немає потреби розкривати IP-адресу та географічне розташування користувача. Таким чином, обмежується видимість будь-яких сайтів, які ви відвідуєте.

Особливості:

– Програмне забезпечення Tor доступне для Linux, Windows, а також Mac

– Це допомагає блокувати сторонні трекери, і реклама не може вас стежити

– Це запобігає тому, щоб сторонні спостерігачі за вашим з'єднанням знали, які вебсайти ви відвідуєте

– Це має на меті зробити всіх користувачів однаковими, і це складно для трекерів.

Переваги:

– Це захищає онлайн-ідентичність.

– Забезпечує високий рівень конфіденційності.

– Зручний інтерфейс.

Недоліки:

– Система стає повільнішою під час навігації.

– Час запуску та перегляду сторінок високий.

6. Nessus Professional

Nessus professional – це програмне забезпечення для мережевої безпеки, яке може виявляти вразливості, такі як помилки програмного забезпечення та загальні проблеми безпеки в програмних застосунках, IT-пристроях та операційних системах, а також відповідним чином керувати ними.

Користувачі можуть отримати доступ до різноманітних плагінів безпеки, а також розробляти власні та сканувати окремі комп'ютери та мережі.

Особливості:

– Він забезпечує налаштування звітів за вразливістю або хостами та створює зведення для користувачів.

– Надсилає сповіщення електронною поштою про результати сканування.

– Це допомагає виконувати державні, регуляторні та корпоративні вимоги.

– Він сканує хмарні програми та захищає вашу організацію від кіберзагроз.

Переваги:

– Він пропонує гнучкість для розробки індивідуальних рішень.

– Сканування Nessus VA охоплює всі стандартні мережеві пристрої, такі як кінцеві точки, сервери, мережеві пристрої тощо.

– Надайте плагіни для багатьох вразливостей.

Недоліки:

– Програмне забезпечення сповільнюється під час сканування великого обсягу.

– Погана підтримка клієнтів.

7. Metasploit

Metasploit – це програмне забезпечення безпеки, яке містить різні інструменти для виконання послуг тестування на проникнення. IT-фахівці

використовують цей інструмент для досягнення цілей безпеки, таких як виявлення вразливостей у системі, покращення безпеки комп'ютерної системи, розробка стратегій кіберзахисту та проведення повних оцінок безпеки.

Інструменти тестування на проникнення можуть досліджувати різні системи безпеки, включаючи веб-додатки, сервери, мережі тощо.

Це дозволяє організації проводити оцінки безпеки, покращувати загальний захист мережі та робити його більш чутливим.

Особливості:

– Ці інструменти використовуються для використання слабких місць системи.

– Модульні енкодери використовуються для перетворення кодів або інформації.

– Metasploit дозволяє чистий вихід з цільової системи. Він скомпрометував.

Переваги:

– Гарна підтримка тестування на проникнення.

– Корисно для вивчення та розуміння вразливостей, які існують у системі.

– Безкоштовно доступний та включає всі інструменти для тестування на проникнення.

Недоліки:

– Оновлення програмного забезпечення відбуваються рідше.

– Крута крива навчання.

8. Kali Linux

Kali Linux – це інструмент для тестування на проникнення, який використовується для сканування IT-систем та мережевих вразливостей. Організація може контролювати та обслуговувати свої системи мережевої безпеки лише на одній платформі.

Він пропонує операційну систему аудиту безпеки та інструменти з більш ніж 300 методами, щоб забезпечити безпеку ваших сайтів та серверів Linux.

Kali Linux використовується професійними тестувальниками на проникнення, етичними хакерами, експертами з кібербезпеки та людьми, які розуміють використання та цінність цього програмного забезпечення.

Особливості:

– Kali Linux постачається з попередньо встановленими інструментами, такими як Nmap, Aircrack-ng, Wireshark тощо, які допомагають у виконанні завдань інформаційної безпеки.

– Він забезпечує багатомовну підтримку.

– Це допомагає створити кастомізовану версію Kali Linux.

Переваги:

– Попередньо встановлені інструменти готові до використання.

– Простий та зручний інтерфейс.

Недоліки:

– Обмежена налаштування.

– Процес встановлення складний.

9. Snort

Snort – це інструмент мережевої безпеки з відкритим кодом, який використовується для сканування мереж та запобігання будь-якій несанкціонованій активності в мережі. IT-фахівці використовують його для відстеження, моніторингу та аналізу мережевого трафіку. Він допомагає виявляти будь-які ознаки крадіжки, несанкціонованого доступу тощо. Після виявлення інструмент допомагає надсилати сповіщення користувачам.

Крім того, Snort використовується для аналізу протоколів, виявлення частих атак на систему, пошуку даних, отриманих з трафіку тощо.

Особливості:

– Snort забезпечує моніторинг трафіку в режимі реального часу.

– Він забезпечує аналіз протоколів.

– Його можна встановити в будь-якому мережевому середовищі.

Переваги:

– Добре підходить для моніторингу мережевого трафіку.

– Добре підходить для виявлення будь-яких мережевих вторгнень.

Недоліки:

– Складні налаштування та конфігурація.

– Крута крива навчання.

10. Forcepoint

Forcepoint – це хмарне рішення безпеки, яке використовується для визначення безпеки мережі, обмеження доступу користувачів до певного контенту та блокування різних спроб злому або отримання інформації вашої організації.

IT-адміністратор може налаштувати Forcepoint для моніторингу та виявлення будь-яких несанкціонованих дій у мережі та вжити відповідних заходів. Це додає додатковий рівень безпеки для критичних загроз.

Forcepoint призначений переважно для організацій, що працюють у хмарі, і він зможе блокувати або надавати попередження про будь-які ризиковані хмарні сервери.

Особливості:

– Forcepoint допомагає відстежувати будь-яку незвичайну хмарну активність.

– Він забезпечує відстеження будь-якої підозрілої поведінки та надсилає сповіщення IT-адміністраторам.

– Він захищає та забезпечує безпеку даних.

– Це допомагає обмежити доступ ваших співробітників у межах вашої організації.

Переваги:

– Гарна підтримка.

– Легке налаштування та зручний інтерфейс.

Недоліки:

– Створення звітів складне.

– Менша гнучкість моніторингу екрана в режимі реального часу.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – високорівнева мова програмування, яку називають другою за популярністю в світі. Її використовують для розробки вебзастосунків, програмного забезпечення, машинного навчання. Python застосовують для вирішення робочих завдань у компаніях Google, Instagram, Facebook, IBM, NASA, Dropbox, Netflix та інших. Розробники цінують цю мову програмування за простоту у вивченні, ефективність та мультиплатформність.

Python – скриптова мова програмування з досить простим синтаксисом. Для розуміння достатньо порівняти принципи написання найпростішої програми, яка виводить на екран текстове повідомлення. Саме тому мова програмування Python більш доступна для новачків, а професіонали встигли адаптувати її для вирішення великої кількості завдань. Це мультиплатформне рішення, тому знання Python дає можливість працювати у різних сферах: від розробки мобільних застосунків до ігрової індустрії та штучного інтелекту.

У мови програмування динамічна типізація: є можливість передавати до функцій будь-який тип даних без попереднього вказання. Інтерпретованість дозволяє знаходити помилки у коді ще до повної збірки у робочий застосунок. При цьому Python дуже чітко дає зрозуміти, де та через що виникла помилка.

Це мова об'єктно-орієнтованого програмування (ООП). Програмне забезпечення на Python оформлене у вигляді моделей, які можуть бути зібраними у пакети. Тип та структуру кожного об’єкта можна запитати під час виконання програми. Для кожного з об’єктів можна отримати всю інформацію щодо його внутрішньої структури. Окрім того:

– у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;

– гнучкість та масштабованість Python дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність;

– розробка на Python у більшості випадків проходить швидше, ніж на інших мовах програмування;

– Python – інтерпретована мова програмування. Це значить, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запустити;

– у Python – колосальна спільнота однодумців. Тож будь-які складнощі конкретних розробників вирішуються колективно.

Проте є декілька особливостей, які можна віднести до недоліків. Це повільність (ця мова програмування хоч і універсальна, проте повільніша за інші), велика кількість ресурсів, необхідних для роботи та «прив'язаність» до системних бібліотек.

Мова програмування Python використовується у наступних сферах:

1. Розробка програмних застосунків будь-якого напрямку.

2. Розробка серверної частини мобільних застосунків (найпопулярніший напрямок).

3. Ігри. Багато сучасних ігор для комп'ютерів (наприклад, World of Tanks) частково чи повністю написані на Python.

4. Вбудовані системи для різних пристроїв. Дуже часто Python використовують для написання внутрішніх платформ управління банкоматами.

5. Скрипти та плагіни до уже реалізованих програм для автоматизації процесів чи створення інших рішень.

6. Тестування (автоматизація цього процесу).

7. Машинне навчання. – основна мова для написання алгоритмів і аналітичних застосунків у сфері Machine Learning.

Бібліотеки Python

Різні бібліотеки Python використовують для виконання конкретних завдань. Наприклад, Matplotlib підходить для відображення даних у двовимірній та тривимірній графіці. Pandas підходить для зручної роботи з даними. NumPy дозволяє створювати масиви та керувати ними. Requests використовується для веброзробки. OpenCV-Python відкриває можливості для обробки зображень з метою оптимізації систем «машинного зору».

Найвідоміші фреймворки для мови програмування Python

Фреймворки Python допомагають створити зручне та функціональне середовище для розробки. У них міститься набір інструментів, модулів та бібліотек, корисних для виконання конкретних завдань. Це значно полегшує роботу: наприклад, дає змогу не витрачати час на розписування дій, які повторюються, а використати релевантний інструмент. Тож є можливість позбутися рутинних процесів та сконцентруватися на логіці проєкту.

Серед найпопулярніших фреймворків для Python:

– Django – найстаріший та найвідоміший. Створений для реалізації великих інтерактивних проєктів;

– Pyramid – зручний у налаштуваннях, і дає можливість реалізувати складні нестандартні ідеї;

– Web2py – підходить в першу чергу для вебзастосунків і може використовуватись на будь-яких архітектурах.

Популярні Python IDE

IDE або інтегровані середовища розробки – це програмне забезпечення, яке надає розробникам необхідні інструменти для написання, редагування, тестування та налаштування коду. Для розробки на Python найчастіше використовують IDE PyCharm, IDLE, Spyder та Atom.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Автентифікація – це процес перевірки особи користувача або системи. Повернімося до нашого прикладу зі спортзалом. Адміністратор у спортзалі приймає лише передплатників. Як це досягається в аналоговому світі? Вони можуть попросити вас надати вашу картку члена спортзалу (якщо припустити, що спортзал надає її кожному передплатнику). У вас повинна бути картка члена спортзалу з вашою фотографією та відповідними даними про передплату. Адміністратор може підтвердити вашу особу, перевіривши вашу картку; проблема автентифікації вирішена (якщо хтось не знайде спосіб підробити картку, але це вже інша проблема).

Автентифікація та ідентифікація є основними компонентами будь-якої інформаційної системи та мережі. Важливо розуміти різницю між автентифікацією та ідентифікацією.

Під час ідентифікації користувач (або система чи процес) заявляє про певну (унікальну) особу у відповідних налаштуваннях. Автентифікація – це підтвердження особи користувача (або системи чи процесу). Цей процес зазвичай здійснюється одним із таких способів:

– Щось тобі відомо.

– Щось у тебе є.

– Щось, що ти є.

Використовуються ще два методи, хоча й у меншій мірі:

– Десь ви знаходитесь (логічне/фізичне місцезнаходження).

– Щось, що ви робите (поведінка).

Опишемо кожен із трьох основних механізмів автентифікації.

Щось, що ти знаєш

«Щось, що ви знаєте», стосується чогось, що ви знаєте або запам’ятали. Приклади включають наступне:

– Паролі, такі як 4SNoPawKkdFiCdnmi%WAdWi-;4,mxRMQB.

– Парольні фрази, такі як «Judge Battle Advise Pain 9» та «Baggage Protection Dissatisfy Barrel 8».

– PIN-код (персональний ідентифікаційний номер), такий як 25063та6285.

Більшість мобільних телефонів автоматично блокуються протягом кількох хвилин бездіяльності. Залежно від початкової конфігурації, користувач може розблокувати їх, ввівши правильний PIN-код, пароль або графічний ключ. Хоча графічний ключ і намальований, він нічим не відрізняється від PIN-коду, тобто чогось, що запам'ятовується.

Розглянемо випадок, коли ви входите в TryHackMe. Ви повинні ідентифікувати себе за допомогою імені користувача або електронної пошти та автентифікувати свою особу за допомогою пароля. (Якщо ви входите в TryHackMe через Google, ви надаєте свої облікові дані для входу Google, і Google підтвердить вашу особу TryHackMe.) Ім'я користувача та електронна пошта є унікальними для вас; отже, ідентифікація може бути здійснена без жодної двозначності. Вважається, що пароль відомий лише вам, що доводить, що ви є власником облікового запису.

Щось у тебе є

«Щось, що у вас є» стосується об’єкта, зазвичай фізичного, який у вас є. Це може бути як телефон, так і ключ безпеки.

Наприклад, коли ви хочете зареєструватися в деяких додатках для обміну миттєвими повідомленнями, вас просять надати номер телефону, зазвичай номер мобільного телефону. Цей номер телефону є вашою ідентифікатором у цьому додатку. Як ви можете довести, що це справді ваш номер телефону? Один із способів – надіслати вам код через SMS або зателефонувати вам на цей номер і повідомити код. Прочитання SMS протягом кількох хвилин або отримання

дзвінка на цей номер доведе, що у вас є цей номер телефону. У випадку номера мобільного телефону, цього буде достатньо, щоб підтвердити, що у вас є SIM-картка (або eSIM).

Апаратний ключ безпеки достатньо малий, щоб його можна було носити на зв'язці ключів або в гаманці. Ви можете використовувати ключ безпеки для автентифікації, підключивши його до порту USB або USB C або піднісши його до зчитувача NFC (Near-Field Communication). Прикладами апаратних ключів безпеки є, наприклад , Yubico , Titan Security Key , Nitrokey та Thetis.

Щось, що ти є

«Щось, чим ви є», стосується біометричних зчитувачів. Прикладами є зчитувачі відбитків пальців, сканери розпізнавання обличчя, сканери сітківки ока та розпізнавання голосу.

Ви, найімовірніше, стикалися з автентифікацією за допомогою зчитувача відбитків пальців під час спроби розблокувати телефон. Багато сучасних мобільних телефонів дозволяють користувачеві автентифікуватися за допомогою відбитка пальця, зберігаючи пароль/PIN-код/графічний ключ як резервний варіант на випадок, якщо автентифікація за відбитком пальця не вдасться.

Розпізнавання облич також стає популярним у сучасних смартфонах. З роками біометричні зчитувачі та сканери стають не тільки надійнішими, але й доступнішими. Ця технологія вигідна як компаніям, які вимагають високого рівня безпеки, так і споживачам.

Багатофакторна автентифікація (MFA)

Багатофакторна автентифікація (MFA) означає використання двох або більше з перерахованих вище механізмів (щось, що ви знаєте/маєте/є). Мета полягає в тому, щоб забезпечити додатковий захист у разі порушення одного з механізмів автентифікації.

Якщо ви хочете скористатися банкоматом банку, вставте свою кредитну/дебетову картку та введіть свій PIN-код. Ця процедура є одним із найперших прикладів двофакторної автентифікації (2FA). Очевидна корисність

полягає в тому, що зловмиснику недостатньо просто отримати вашу картку, оскільки йому також потрібно знати ваш PIN-код.

Багато домашніх сейфів, доступних сьогодні, використовують 2FA, не обов'язково рекламуючи це. Вони вимагають від власника вставити ключ і ввести правильний PIN-код, щоб сейф відкрився. Вам потрібно знати PIN-код і мати ключ, щоб його відкрити. Отже, якщо комусь іншому вдасться отримати ключ від сейфа, він все одно не зможе відкрити сейф, не знаючи PIN-коду.

У другому випадку двофакторна автентифікація (2FA) вимагає двох механізмів автентифікації та підпадає під більш загальну багатофакторну автентифікацію (MFA), яка вимагає двох або більше факторів автентифікації. Ця вимога може значно покращити безпеку та захистити від різних атак, таких як ті, що використовують слабкі паролі.

3.2 Розробка структурної схеми

Після автентифікації користувачеві має бути надано належний рівень доступу. Авторизація визначає, до чого має бути дозволено автентифікованому користувачеві отримувати доступ та що робити. Повертаючись до нашого прикладу зі спортзалом, ми очікуємо, що платний учасник спортзалу матиме право доступу до будь-якого тренувального обладнання протягом встановленого робочого часу. Абонент не може позичати та брати бігову доріжку додому протягом місяця. Спортзал – це не бібліотека! Механізми контролю доступу забезпечать належне виконання авторизації. Якщо в спортзалі є належний механізм контролю доступу, хтось помітить, як ви намагаєтеся перенести та винести бігову доріжку.

Розглянемо ще один нетехнічний приклад: Віка забронювала номер у готелі на один тиждень. Завдяки цьому бронюванню Віка повинна мати право доступу до свого готельного номера, серед інших громадських приміщень, протягом усього перебування. Хоча їй дозволено доступ до призначеного їй

номера, їй не дозволено, наприклад, доступ до інших номерів. Як це можна забезпечити? Знову ж таки, авторизація забезпечується механізмами контролю доступу.

У прикладі з готелем Віці надають ключ, який надає їй доступ до призначеного їй номера. Відповідно, контроль доступу здійснюється за допомогою замків та ключів. Більш вишуканий готель може використовувати смарт-картки та електронні зчитувачі карток для відмикання дверей номера. У будь-якому випадку існує механізм, який би забезпечив авторизацію.

Розглянемо технічний приклад. Як частина команди з продажу, Ігор повинен мати доступ до всіх файлів, пов'язаних із продажами, які необхідні для ефективного та результативного виконання його роботи. Наприклад, команді з продажу не потрібно мати доступ до документів, що стосуються, наприклад, управління персоналом та бухгалтерського обліку. У цьому випадку контроль доступу можна забезпечити, встановивши відповідні дозволи доступу до файлів та бази знань компанії.

Коротко кажучи, авторизація визначає, до чого користувач повинен мати доступ, тоді як контроль доступу забезпечує дотримання встановленої політики. Наприклад, після входу до свого облікового запису електронної пошти ви повинні мати можливість читати свої електронні листи та надсилати нові. Однак за замовчуванням ви не повинні мати доступу до поштової скриньки жодного зі своїх колег. Поштовий сервер має бути розроблений таким чином, щоб дозволяти користувачеві доступ до своєї поштової скриньки та забороняти йому доступ до поштових скриньок інших користувачів.

Авторизація проти контролю доступу

– Авторизація – етап прийняття рішення – визначає, який доступ слід надавати кому, на основі ролей або політик.

Вирішення, що секретар може надсилати електронні листи від імені менеджера.

– Контроль доступу – механізм забезпечення дотримання правил –

реалізує правила, визначені на етапі авторизації.

Запобігання зміненню документа користувачами шляхом забезпечення доступу лише для читання.

Підзвітність та ведення журналу

Підзвітність гарантує, що користувачі несуть відповідальність за дії, які вони виконують у системі. Іншими словами, після автентифікації своєї особи та отримання дозволу на доступ до системи вони можуть нести відповідальність за свої дії. Підзвітність можлива, якщо у нас є можливості аудиту , які зазвичай вимагають належного функціонування журналювання.

Почнемо з нетехнічного прикладу: спортзал. У вас є абонемент у спортзал, і ви відвідуєте його тричі на тиждень. Тепер, коли ви стали його постійним клієнтом, адміністратор впізнає вас і не просить пред'являти вашу картку члена. Це ніби ви завжди «залогінені» в спортзалі! Ви помічаєте, що всі, хто має «доступ» до спортзалу, дотримуються певних правил. Наприклад, ніхто не розбиває настінне дзеркало, якщо незадоволений швидкістю свого прогресу. Якщо вони це зроблять, їхнє членство буде анульовано, і вони сплатять усі збитки. Іншими словами, кожен несе *відповідальність* за свої дії. Ця модель забезпечує зручність для всіх для безпечних тренувань.

Розглянемо більш технічний приклад, наприклад, касира банку. Такий працівник може переглядати та проводити різні операції на рахунку клієнта. Як ми можемо гарантувати, що недобросовісний працівник не зловживатиме такими повноваженнями? Нам потрібно безпечно реєструвати всі транзакції та відповідні деталі. Ми повинні мати можливість перевіряти всі проведені транзакції та перевіряти, хто що зробив. Без такої можливості ми не можемо ні покладатися на таку систему, ні довіряти їй.

Ведення журналу

Критичним аспектом підзвітності є ведення журналу. Ведення журналу – це процес запису подій, що відбуваються в системі. Цей процес включає дії користувачів, системні події та помилки. Реєструючи дії користувачів,

організація може вести облік того, хто і коли отримував доступ до якої інформації. Цей запис життєво важливий для дотримання нормативних вимог, реагування на інциденти та судово-медичних розслідувань.

Завдяки комплексній системі реєстрації, організація може відстежувати дії будь-якого користувача, виявляти будь-які аномалії або несанкціонований доступ і вживати відповідних заходів. Наприклад, якщо неавторизований користувач намагається отримати доступ до конфіденційних даних, система реєстрації може генерувати сповіщення для сповіщення персоналу служби безпеки.

Ведення журналів також може допомогти організаціям виявляти інциденти безпеки та реагувати на них. Аналізуючи дані журналів, команди безпеки можуть виявляти закономірності підозрілої активності, такі як повторні невдалі спроби входу або незвичайні моделі доступу. Цю інформацію потім можна використовувати для розслідування потенційних загроз безпеці та реагування на них.

Оскільки підзвітність є ключовим компонентом будь-якої безпечної інфраструктури, слід належним чином подбати про те, щоб ведення журналу відбувалося належним чином та безпечно. Крім того, залежно від вимог безпеки, журнали повинні бути захищеними від несанкціонованого доступу. Причина полягає в тому, що ви не хочете, щоб зловмисник видаляв або змінював журнали та приховував свої дії в мережі. Ось чому гарною практикою є налаштування окремого сервера журналювання з одним завданням: безпечне отримання та зберігання журналів. Звідси й переадресація журналів.

Пересилання журналів – це процес надсилання даних журналів з однієї системи до іншої. Цей процес часто об’єднує дані журналів з кількох джерел у централізоване місце для кращого аналізу та керування. Пересилання журналів також може використовуватися для надсилання даних журналів до хмарного сервісу для зберігання та аналізу.

Пересилання журналів має кілька переваг. Централізуючи дані журналів, організації можуть легше аналізувати та співвідносити події журналів з різних

систем для виявлення потенційних загроз безпеці. Це підводить нас до управління інформацією та подіями безпеки (SIEM).

Ведення журналу та SIEM

Управління інформацією та подіями безпеки (SIEM) – це технологія, яка об'єднує дані журналів з кількох джерел та аналізує їх на наявність ознак загроз безпеці. Рішення SIEM можуть допомогти організаціям виявляти аномалії, потенційні інциденти безпеки та надавати сповіщення командам безпеки.

Інтегруючи логування та SIEM, організації можуть краще розуміти активність своїх систем та мереж, а також контролювати потенційні загрози. Ця інтеграція дозволяє організаціям ефективніше виявляти загрози безпеці та реагувати на них.

Крім того, інтеграція ведення журналу та SIEM забезпечує додаткові переваги, такі як звітність про відповідність вимогам та судово-медичні розслідування. Звітність про відповідність вимогам є важливою частиною системи безпеки будь-якої організації, а ведення журналу допомагає організаціям виконувати вимоги до звітності, збираючи дані, необхідні для аудитів. Судово-медичні розслідування мають вирішальне значення для визначення джерела та причини інциденту безпеки. Рішення для ведення журналу та SIEM дозволяють організаціям проводити судово-медичні розслідування, надаючи детальну історію активності системи та мережі.

Управління ідентифікацією

Керування ідентифікацією (IdM) включає всі необхідні політики та технології для ідентифікації, автентифікації та авторизації. IdM має на меті забезпечити, щоб уповноважені особи мали доступ до активів та ресурсів, необхідних для їхньої роботи, тоді як неавторизованим особам доступ заборонено. IdM вимагає, щоб кожному користувачеві або пристрою було призначено цифрову ідентифікацію.

IdM допомагає організаціям захищати конфіденційні дані та підтримувати дотримання нормативних вимог. Це також дозволяє організаціям оптимізувати

процеси доступу користувачів, зменшувати витрати, пов'язані з управлінням ідентифікацією, та покращувати взаємодію з користувачами. Впроваджуючи ефективну стратегію IdM, організації можуть гарантувати, що їхні користувачі автентифіковані та авторизовані для безпечного доступу до необхідних їм ресурсів.

Деякі джерела використовують IdM та керування ідентифікацією та доступом (IAM) як взаємозамінні. Інші джерела вважають, що IdM більше зосереджений на питаннях безпеки, пов'язаних з ідентифікацією користувача, таких як автентифікація та дозволи. Вони стверджують, що IdM займається управлінням атрибутами та дозволами користувачів, пристроїв та груп, тоді як IAM більше займається оцінкою атрибутів та дозволів і наданням або забороною доступу відповідно до політики компанії. У цьому завданні ми представляємо їх як різні, хоча межа між ними, як правило, розпливчаста.

Управління ідентифікацією (IdM)

Управління цифровими ідентифікаторами (IdM) – це важливий компонент кібербезпеки, що стосується процесу управління та контролю цифрових ідентифікацій. Він включає управління ідентифікаторами користувачів, їхню автентифікацію, авторизацію та контроль доступу. Головна мета IdM – забезпечити доступ до певних ресурсів та інформації лише уповноваженим особам. Системи IdM використовуються для керування ідентифікаторами користувачів у мережі організації.

Системи IdM використовують централізовану базу даних для зберігання ідентифікаційних даних користувачів та прав доступу. Вони також надають функціональні можливості для керування та моніторингу доступу користувачів до ресурсів. Системи IdM зазвичай включають такі функції, як надання користувачів, автентифікація та авторизація. Надання користувачів стосується процесу створення та керування обліковими записами користувачів, тоді як автентифікація та авторизація стосуються перевірки особи користувача та надання доступу до певних ресурсів.

Системи IdM є критично важливими в організаціях, де існує кілька систем і програм, що потребують контролю доступу. Вони допомагають спростити управління ідентифікацією користувачів, зменшуючи ризик несанкціонованого доступу до ресурсів. Крім того, системи IdM забезпечують єдину точку відліку для управління ідентифікацією користувачів, що спрощує для організацій управління правами доступу користувачів.

Керування ідентифікацією та доступом (IAM)

IAM – це більш комплексне поняття, ніж IdM. Воно охоплює всі процеси та технології для управління цифровими ідентифікаторами та правами доступу, а також їх захисту. Системи IAM включають різноманітні функції, такі як надання користувачам доступу, контроль доступу, управління ідентифікаторами та управління відповідністю. Системи IAM гарантують, що лише авторизовані користувачі мають доступ до певних ресурсів і даних, а їхній доступ моніториться та контролюється.

Системи IAM пропонують комплексне рішення для керування та захисту доступу до ресурсів в організації. Вони інтегруються з кількома системами та програмами, забезпечуючи централізоване уявлення про ідентифікаційні дані користувачів та права доступу. Системи IAM використовують різні технології для керування доступом, включаючи контроль доступу на основі ролей, багатофакторну автентифікацію та єдиний вхід.

Системи IAM допомагають організаціям дотримуватися нормативних вимог, таких як HIPAA, GDPR та PCI DSS. Вони надають функції для керування життєвим циклом ідентифікації користувачів, включаючи адаптацію, видалення та скасування доступу. Крім того, системи IAM дозволяють організаціям відстежувати та перевіряти активність користувачів, що допомагає запобігти порушенням безпеки та забезпечити дотримання галузевих норм.

IdM та IAM є важливими компонентами кібербезпеки. Вони гарантують, що лише уповноважені особи мають доступ до певних ресурсів та інформації. Системи IdM керують ідентифікаторами користувачів, тоді як системи IAM

охоплюють ширші функції для управління та захисту цифрових ідентифікаторів та прав доступу.

Атаки проти автентифікації

Це завдання охоплюватиме приклади атак на наївний протокол автентифікації. Мета полягає в тому, щоб дати уявлення про важливість використання існуючих та перевірених протоколів замість створення протоколу та його використання без ретельного тестування одноранговими користувачами.

Автентифікація в аналоговому світі

Припустимо, ви належите до кінного клубу. Клуб резервує місцевий ресторан для щотижневих зустрічей. Ви можете поспілкуватися про свої пригоди, насолоджуючись улюбленою стравою. Охоронець біля входу знає не всіх членів клубу. Тож ви розробляєте схему автентифікації, щоб охоронець міг вирішити, чи відчиняти двері.

Одна з найпростіших ідей, яка спадає на думку, – це використання загальної секретної фрази. Тож кожен, хто хоче увійти, повинен сказати секретну парольну фразу; нікому не буде дозволено вхід, якщо він не скаже « *сім коней* » на запитання «Скільки?». Цей механізм автентифікації працює чудово, доки зловмисник, що стоїть поруч, не підслухає та не дізнається вашу парольну фразу. Тепер він отримає доступ до вашої приватної зустрічі, ніби він один із вас. Було б корисно, якби у вас було щось складніше.

Ви можете спланувати десять запитань з десятьма різними секретними відповідями замість одного запитання та однієї відповіді; однак зловмисник, який перебуває досить близько, зрештою вивчить їх усі. Використання безпечного механізму автентифікації без застосування криптографії може бути практично неможливим. На щастя, у сценарії з охоронцем біля дверей легко помітити будь-яких підозрілих осіб, які бездіяльно гуляють; інакше вся ваша група буде скомпрометована.

Автентифікація в цифровому світі

Ситуацію в мережі ще складніше захистити. Якщо користувач надсилає

своє ім'я користувача та пароль у відкритому тексті, будь-хто, хто перехоплює трафік у мережі, може дізнатися це ім'я користувача та пароль. Як ми можемо запобігти отриманню облікових даних для входу?

Сервер і користувач можуть домовитися про фіксований секретний ключ. Замість того, щоб надсилати пароль у відкритому вигляді, користувач шифрує його за допомогою вибраного секретного ключа. Щоразу, коли користувачі хочуть увійти, вони надсилають своє ім'я користувача та пароль у зашифрованому вигляді за допомогою призначеного їм секретного ключа. Тепер зловмисник ніколи не повинен мати змоги дізнатися пароль, чи не так? На жаль, хоча вони не зможуть дізнатися пароль, вони все одно зможуть пройти автентифікацію.

Хоча зловмисник не знає пароля, він все одно може автентифікуватися, відтворивши ту саму відповідь. Ця атака вважається **атакою повторного відтворення**. Чи можемо ми щось зробити, щоб виправити це?

Зробіть відповідь на виклик унікальною

Зашифрований пароль, який завжди має одне й те саме значення, легко обійти. Нам потрібен певний механізм, який гарантуватиме, що відповідь не буде використана повторно. Один із підходів – використовувати поточний час і дату як частину відповіді. Іншими словами, користувач надсилатиме зашифрований поточний час (і дату) разом із паролем. Хоча це вимагає синхронізації годинників обох сторін, це гарантує, що відповідь буде дійсною лише протягом короткого часу, зазвичай у мілісекундах.

Це завдання має на меті пролити світло на деякі проблеми, пов'язані з автентифікацією та протоколами автентифікації. Багато інших вразливостей можуть потрапити в протоколи автентифікації; однак це виходить за рамки цієї кімнати.

Атака повторного відтворення відбувається , коли зловмисник перехоплює дійсні дані автентифікації (наприклад, зашифрований пароль або токен сеансу) та повторно надсилає (або «відтворює») їх, щоб видати себе за

оригінального користувача, навіть не знаючи пароля та не розшифровуючи повідомлення.

– Користувач шифрує свій пароль за допомогою спільного секретного ключа та надсилає його на сервер.

– Зловмисник перехоплює це зашифроване повідомлення під час його передачі мережею.

– Пізніше зловмисник відтворює точне повідомлення на сервері.

– Сервер приймає його як дійсний, оскільки він відповідає очікуванням, навіть якщо реального входу не відбулося.

Система не має можливості визначити, чи:

– Цей запит є свіжим (від легітимного користувача), або

– Це повторне повідомлення з попереднього сеансу.

Чому це трапляється:

– Статична відповідь: Зашифрований пароль виглядає однаково щоразу.

– Відсутність актуальності: Сервер не перевіряє, коли було створено повідомлення.

– Без одноразового номера/позначки часу: Немає унікального значення для кожного сеансу для розрізнення запитів.

Як запобігти атакам повторного відтворення

Щоб уникнути атак повторного відтворення, протоколи автентифікації повинні гарантувати, що кожна спроба автентифікації є унікальною.

– Мітки часу – Включити поточний час у зашифроване повідомлення. Сервер відхиляє старі позначки часу.

– Nonces Використовуйте одноразове випадкове число (nonce) для кожного сеансу або завдання.

– Токени сесії – генерувати унікальні токени сесії після автентифікації.

– Протоколи виклику-відповіді – сервер надсилає виклик; клієнт підписує його своїм секретним ключем. Запобігає повторному використанню.

Сучасні протоколи, що пом'якшують це:

– Kerberos: використовує позначки часу та симетричне шифрування для запобігання атакам повторного відтворення.

– OAuth 2.0: видає короткочасні токени та використовує токени оновлення для тривалих сеансів.

– TLS/SSL: використовує унікальні ключі сеансу та рукостискання з одноразовими числами для забезпечення актуальності повідомлень.

Моделі контролю доступу

Система контролює доступ до різних ресурсів на основі обраної моделі. Деякі з поширених моделей контролю доступу:

– Дискреційний контроль доступу (DAC).

– Контроль доступу на основі ролей (RBAC).

– Обов'язковий контроль доступу (MAC).

Дискреційний контроль доступу

Багато хто вже використовував дискреційний контроль доступу (DAC) під час обміну файлами або папками з друзями та колегами. Під час використання DAC власник ресурсу явно додасть користувачів із відповідними дозволами.

Розглянемо наступний приклад. Ви зберігаєте свої фотографії на одній з онлайн-платформ для зберігання. Щоб поділитися всіма зображеннями, пов’язаними з вашим випускним, з родиною, ви додаєте їхні облікові записи окремо та надаєте їм доступ до відповідного альбому. Зрештою, дозволи альбому відображатимуть кілька облікових записів із правами перегляду.

Весь процес простий і повністю контролюється власником даних. Він дуже добре працює для обміну даними з членами родини або кількома користувачами компанії. Однак це може стати складним, коли ви намагаєтеся масштабувати обмін з багатьма користувачами, особливо якщо роль користувача змінюється з часом. Ця ситуація підводить нас до обміну на основі ролей користувачів.

Контроль доступу на основі ролей

Контроль доступу на основі ролей (RBAC) використовує дуже інтуїтивно зрозумілий підхід. Кожен користувач має одну або кілька ролей або функціональних посад; крім того, вони мають право доступу до різних ресурсів на основі своїх ролей.

Бухгалтеру потрібен доступ до бухгалтерських книг компанії, але не потрібен доступ до дослідницьких та розробницьких лабораторій чи документів. Відповідно, користувачі розподіляються на різні групи залежно від їхніх ролей. Авторизація та доступ надаються залежно від групи, до якої належить користувач.

Класифікація користувачів на основі їхніх ролей має багато переваг. Наприклад, якщо користувачеві призначено нову роль, все, що потрібно, це додати його до відповідної нової групи. Більше того, якщо користувачі відмовилися від певної ролі, нам потрібно лише видалити їх зі старої групи. Такий підхід робить обслуговування більш керованим та ефективним.

Обов'язковий контроль доступу

Операційна система, що використовує обов'язковий контроль доступу (MAC), надасть пріоритет безпеці та значно обмежить можливості користувачів. Такі системи використовуються для певних цілей або для обробки високосекретних даних. Отже, користувачам не потрібно виконувати завдання, що перевищують суворо необхідні. Іншими словами, користувачі не зможуть встановлювати нове програмне забезпечення або змінювати дозволи на доступ до файлів.

AppArmor надає можливість використовувати MAC на дистрибутиві Linux. Він вже постачається з різними дистрибутивами Linux, такими як Debian та Ubuntu.

Проект **SELinux** забезпечує гнучку MAC-адресу для систем Linux. Вона є стандартною для кількох дистрибутивів Linux, таких як Red Hat та Fedora.

Єдиний вхід

Користувачам потрібен доступ до різних джерел для виконання своїх щоденних робочих завдань. Наприклад, їм потрібен доступ до електронної пошти, спільних файлів, принтерів тощо. Доступ до цих ресурсів вимагає від користувача облікових даних для успішної автентифікації. Кількість різних імен користувачів та паролів робить це досить складним, особливо якщо користувачі не мають підстав використовувати один і той самий пароль у кількох системах.

Єдиний вхід (SSO) вирішує цю проблему. Замість того, щоб користувачеві доводилося запам’ятовувати кілька імен користувачів та паролів, йому потрібно запам’ятати лише один набір облікових даних для входу. Він може автентифікувати себе в одній системі, що надає йому доступ до інших систем, необхідних для його роботи.

Традиційно користувач повинен створити кілька паролів, таких як пароль для входу на комп’ютер, ще один пароль для перевірки електронної пошти та третій пароль для доступу до спільного файлового ресурсу. Запам’ятовування такої кількості паролів може бути складним завданням, особливо тому, що, в ідеалі, пароль не слід використовувати повторно. Кращим підходом було б вимагати від користувача одноразового входу в систему та надавати йому доступ до всіх необхідних служб; саме це робить SSO.

SSO дозволяє організаціям автентифікувати користувачів один раз, перш ніж надати їм доступ до ресурсів, необхідних для їхньої роботи. Це може дати нам багато переваг. Ми розглянемо деякі з них.

– Один надійний пароль: очікувати, що користувач запам'ятає один надійний пароль, прийнятніше, ніж просити його запам'ятати десять різних надійних паролів.

– Простіша багатофакторна автентифікація (MFA): додавання MFA до кожної окремої служби є надзвичайно складним завданням для виконання та підтримки. За допомогою єдиного входу (SSO) MFA потрібно вмикати та налаштовувати один раз.

– Простіша підтримка: запити на підтримку, такі як скидання пароля, стали простішими, оскільки тепер вони обмежені одним обліковим записом.

– Ефективність: Користувачеві не потрібно входити в систему щоразу, коли йому потрібен доступ до нової послуги.

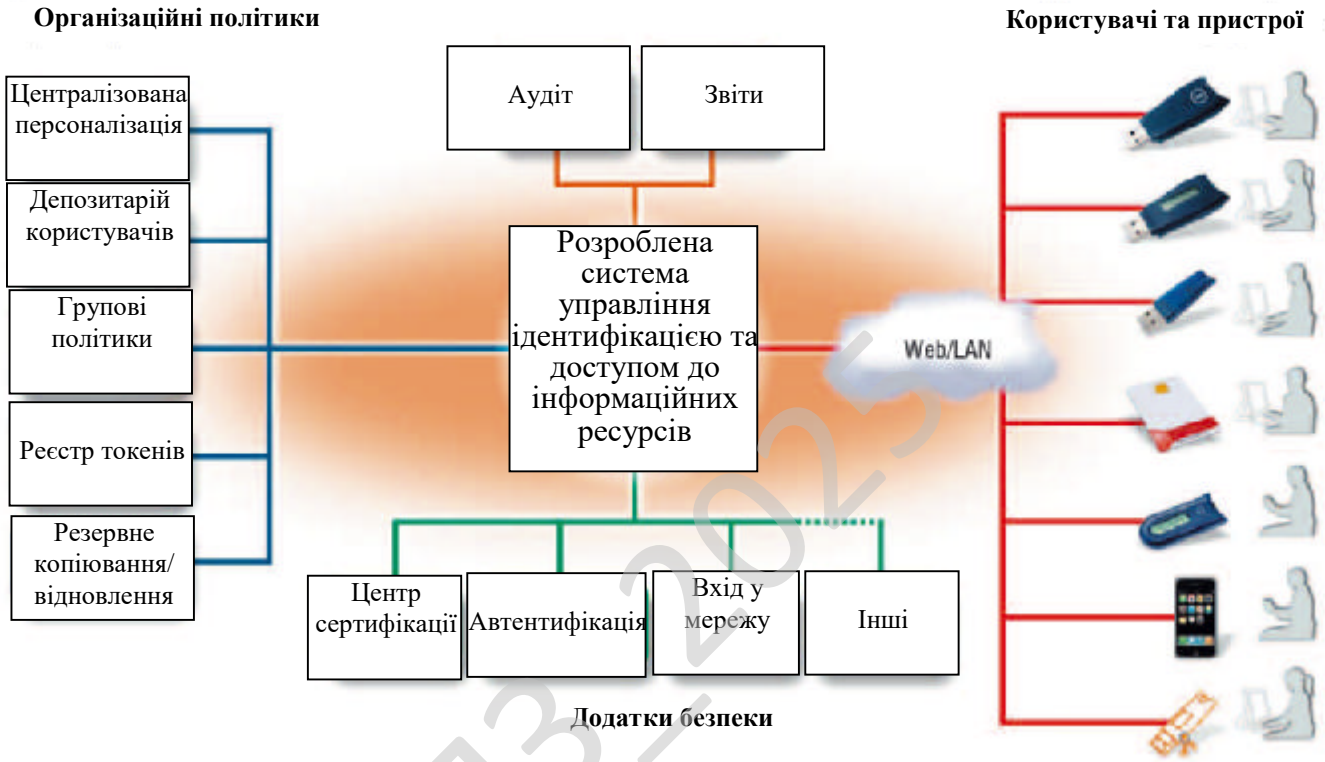

Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

Функціональна схема розробленої системи зображена на рисунку 3.2. З рисунку видно, що розроблена система складається з наступних частин, які реалізують типи прав доступу до USB-ключа:

1. Адміністраторський – надає наступні можливості:

– право міняти PIN-код користувача, не знаючи його;

– право зміни пароля адміністратора;

– право налаштовувати параметри кешування змісту закритої області пам'яті й додаткового захисту закритих ключів паролем, а також можливість робити ці настроювання доступними в користувальницькому режимі.

2. Користувальницький – надає наступні можливості:

– право переглядати, змінювати й видаляти об'єкти в закритих, відкритих і вільній областях пам'яті;

– можливість одержання загальної інформації відносно USB-ключа;

– право міняти PIN-код і перейменовувати USB-ключ;

– право налаштовувати параметри кешування змісту закритої області пам'яті й додаткового захисту закритих ключів паролем (при відсутності пароля адміністратора або з дозволу адміністратора)

– право перегляду й видалення сертифікатів у сховищі USB-ключа і ключових контейнерах RSA.

3. Гостьовий – надає наступні можливості:

– можливість переглядати об'єкти у відкритій області пам'яті;

– можливість одержання із системної області пам'яті загальної інформації відносно USB-ключа, що включає ім'я USB-ключа, ідентифікатори й деякі інші параметри.

При гостьовому доступі знання PIN-коду не обов'язково.

4. Ініціалізаційний – право форматувати USB-ключ.

Для одержання доступу до даних, що зберігається в пам'яті USB-ключа, необхідно ввести PIN-код (Personal Identification Number). В PIN-коді не рекомендується використовувати пробіли й кирилицю. При цьому PIN-код повинен задовольняти критеріям якості, заданим у файлі %systemroot%\system32\etcpass.ini.

Редагування цього файлу, що містить критерії якості PIN-коду, здійснюється за допомогою утиліти USB-ключа.

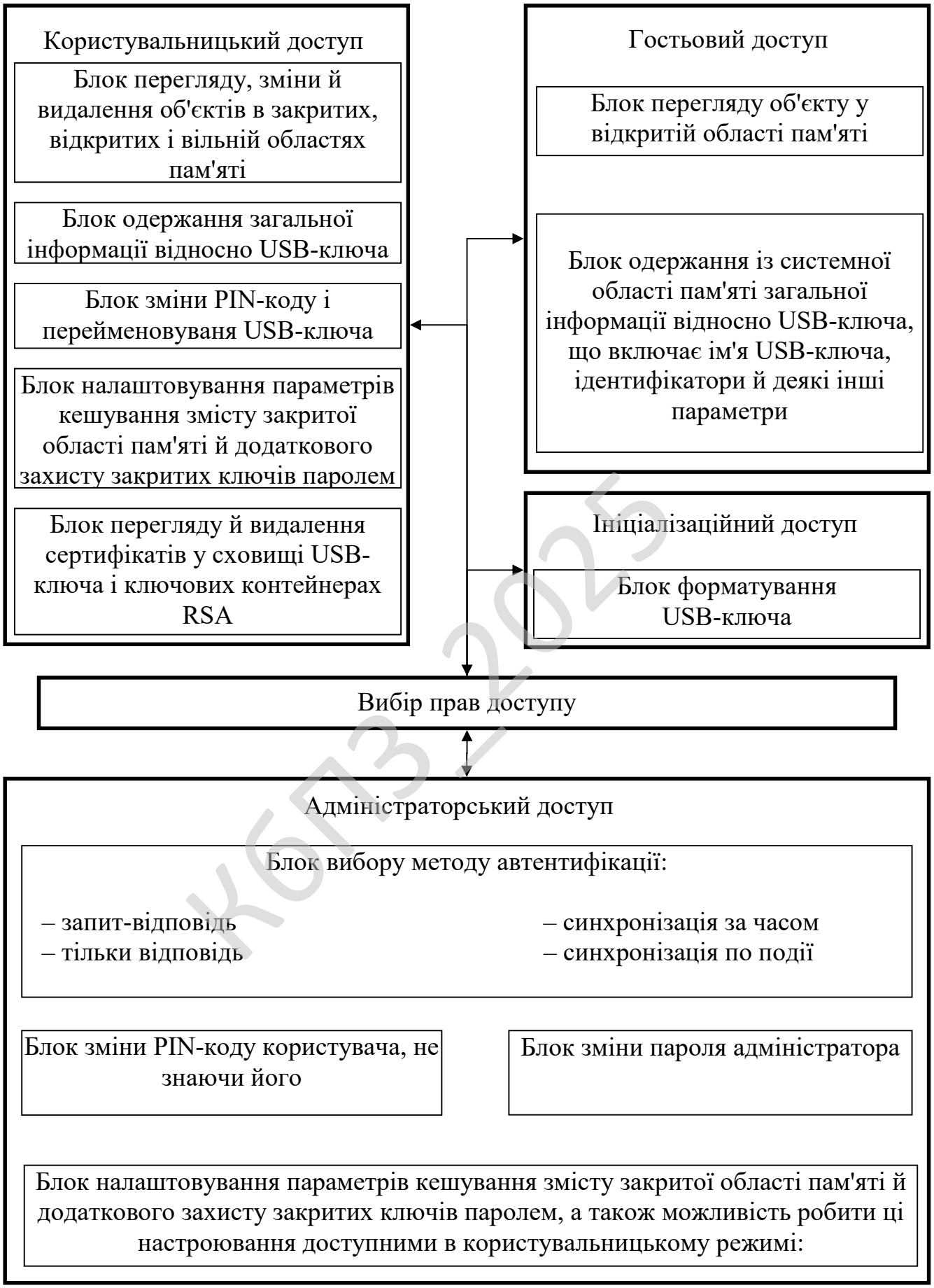

Рисунок 3.2 – Функціональна схема системи

Адміністраторський доступ до USB-ключа може бути зроблений тільки після правильного введення пароля адміністратора. Якщо ж у процесі форматування пароль адміністратора не заданий, то звернутися із правами адміністратора не можна.

За допомогою розробленого програмного забезпечення можна:

– налаштовувати параметри USB-ключа і його драйверів;

– переглядати загальну інформацію відносно USB-ключа;

– імпортувати, переглядати й видаляти сертифікати (за винятком сертифікатів зі сховища USB-ключів) і ключові контейнери RSA;

– форматувати USB-ключ;

– налаштовувати критерії якості PIN-кодів.

Для установки даного програмного забезпечення необхідні права локального адміністратора. Варто пам'ятати, що до установки розробленого програмного забезпечення не можна підключати USB-ключ.

Якщо на комп'ютері встановлене програмне забезпечення, підключите USB-ключ до порту USB або до подовжувального кабелю. Після цього почнеться процес обробки нового обладнання, що може зайняти якийсь час. По завершенні процесу обробки нового обладнання на USB-ключ засвітиться світловий індикатор.

Утиліта "Властивості USB-ключа" дозволяє виконувати основні операції по керуванню токенами, такі як зміна паролів, перегляд інформації й сертифікатів, розташованих у пам'яті USB-ключа. Крім того, за допомогою утиліти "Властивості USB-ключа" можна швидко й легко переносити сертифікати між комп'ютером і USB-ключем, а також імпортувати ключі до пам’яті USB-ключа.

Кнопка "Розблокувати" необхідна, якщо користувач забув свій PIN-код, і не може прийти до адміністратора USB-ключа (наприклад, користувач перебуває у відрядженні). Звернувшись до адміністратора по e-mail, користувач зможе одержати для цього USB-ключа від адміністратора шестнадцятирічний запит,

сформований на підставі даних, що зберігаються в базі TMS, занесши який у поле "відповідь" користувач одержить доступ на зміну PIN-коду.

При зміні PIN-коду необхідно щоб новий PIN-код відповідав вимогам якості введеного пароля. Якість пароля перевіряється відповідно до введених критеріїв. Для того щоб перевірити відповідність пароля обраним критеріям, введіть пароль у рядок. Під цим рядком виводиться інформація про причини невідповідності введеного пароля обраним критеріям у відсотках, а також графічно й у відсотках умовно відображається якість введеного пароля відповідно до обраних критеріїв.

Для того щоб задати список неприпустимих або небажаних паролів, створіть текстовий файл. Або можливо скористатися так званими частотними словниками, які використовуються для підбора паролів. Файли таких словників можна взяти на сайті www.passwords.com.

Приклад такого словника:

– anna

– annette

– bill

– password

– william

Призначте критерію "Dictionary" шлях до створеного файлу. При цьому шлях до файлу словника на кожному комп'ютері повинен збігатися зі значенням критерію "Dictionary".

Вхід у систему за допомогою USB-ключа

При автентифікації в Windows використовуються ім'я користувача й пароль, що зберігаються в пам'яті USB-ключ. Це дає можливість застосовувати строгу автентифікацію на основі токенів.

Разом з тим хотілося б додати, що у великих компаніях, що використовують доменну структуру, необхідно подумати про впровадження PKI і централізованому застосуванні SmartCardLogon.

При використанні USB-ключа можуть застосовуватися нікому не відомі випадкові складні паролі. Крім того, передбачена можливість використання сертифікатів, що зберігаються в пам'яті USB-ключа, для реєстрації на основі смарт-карт, що підвищує безпека входу в Windows.

Це стало можливим завдяки тому, що система Windows XP/7/8/10 дозволяє використовувати різні механізми доступу, що заміняють метод автентифікації за замовчуванням. Механізми ідентифікації й автентифікації служби входу в Windows (winlogon), що забезпечує інтерактивну реєстрацію в системі, убудовані в замінну бібліотеку, що приєднується динамічно (DLL), іменовану GINA (Graphical Identification and Authentication, робочий стіл автентифікації). Коли система має потребу в іншому методі автентифікації, який би замінив механізм "ім'я користувача/пароль" (використовуваний за замовчуванням) стандартну msgina.dll заміняють новою бібліотекою.

При установці USB-ключа заміняється бібліотека робочого стола автентифікації й створюються нові параметри реєстру. GINA відповідає за політику інтерактивного підключення й здійснює ідентифікацію й діалог з користувачем. Заміна бібліотеки робочого стола автентифікації робить USB-ключ основним механізмом перевірки дійсності, що розширює можливості стандартної автентифікації Windows XP/7/8/10, заснованої на застосуванні ім'я користувача й пароля.

Користувачі можуть самостійно записувати до пам’яті USB-ключ інформацію, необхідну для входу в Windows (профілі), якщо це дозволено політикою безпеки підприємства. Профілі можна створювати за допомогою майстра створення профілів USB-ключа Windows Logon.

USB-ключ SecurLogon автентифікує користувача Windows XP/7/8/10 с допомогою USB-ключ, використовуючи або сертифікат користувача зі смарт-картою, або ім'я користувача й пароль, які зберігаються в пам'яті USB-ключа. USB-ключ містить у собі всі необхідні файли й драйвери, що забезпечують підтримку USB-ключа в Windows Logon.

Всі нові USB-ключ мають той самий PIN-код, установлений за замовчуванням при виробництві. Цей PIN-код 1234567890. Для забезпечення строгої, двофакторної автентифікації й повної функціональності користувач обов'язково повинен замінити PIN-код за замовчуванням власним PIN-кодом відразу після одержання нового USB-ключ.

Важливо: PIN-код не слід плутати з паролем користувача Windows.

Блокування робочої станції

Можливо забезпечувати безпеку вашого комп'ютера, не виходячи із системи, шляхом блокування комп'ютера. При від'єднанні USB-ключа від порту USB або кабелю (після вдалої реєстрації) операційна система автоматично заблокує ваш комп'ютер.

Розблокування ваш комп'ютера

Коли ваш комп'ютер заблокований, з'являється вікно "Блокування комп'ютера Computer Locked". Підключите USB-ключ до порту USB або кабелю. У вікні, що з'явилося, введіть PIN-код у поле " USB-ключ Password" і натисніть кнопку "OK" – комп'ютер розблокований. У випадку натискання "CTRL+ALT+DEL" і введення пароля комп'ютер буде розблокований без використання USB-ключа.

Установка

Для того щоб установити USB-ключ Windows Logon:

– увійдіть у систему як користувач із правами адміністратора;

– двічі клацніть SecurLogon.msi;

– з'явиться вікно майстра установки USB-ключ SecurLogon;

– натисніть кнопку "Next", з'явиться ліцензійна угода USB-ключ Enterprise;

– прочитайте угоду, натисніть кнопку "I accept" (Приймаю), а потім кнопку "Next";

– наприкінці установки виробляється перезавантаження.

Використання USB-ключ SecurLogon

USB-ключ SecurLogon дозволяє користувачам реєструватися в Windows XP/7/8/10 за допомогою USB-ключа із записаним у пам'яті паролем.

Зміна пароля

Можливо перемінити пароль Windows після входу в систему за допомогою USB-ключ. Для того щоб перемінити пароль після входу в систему за допомогою USB-ключ:

– увійдіть у систему, використовуючи USB-ключ;

– натисніть "CTRL+ALT+DEL", з'явиться вікно "Безпека Windows / Windows Security";

– Клацніть кнопку "Зміна пароля / Change Password", якщо поточний пароль був створений вручну, те з'явиться вікно "Зміна пароля / Change Password", але якщо поточний пароль був створений випадковим образом, то переходимо до пункту 5;

– Уведіть новий пароль у полях "Новий пароль / New Password" і "Підтвердження / Confirm New Password" і натисніть кнопку "OK";

– Якщо поточний пароль був створений випадковим образом, то й новому паролі буде створений автоматично;

– у діалоговому вікні, що з'явилося, введіть PIN-код USB-ключ і натисніть кнопку "OK"

– з'явиться вікно з підтверджувальним повідомленням.

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходе взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки

чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

Рисунок 3.3 – Діаграма взаємодії процесів.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи управління ідентифікацією та доступом до інформаційних ресурсів.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

Під час роботи над магістерською роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз’яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елементу. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елементу. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому

місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Також при розробці магістерської роботи було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів; Діаграма компонент.

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням

тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

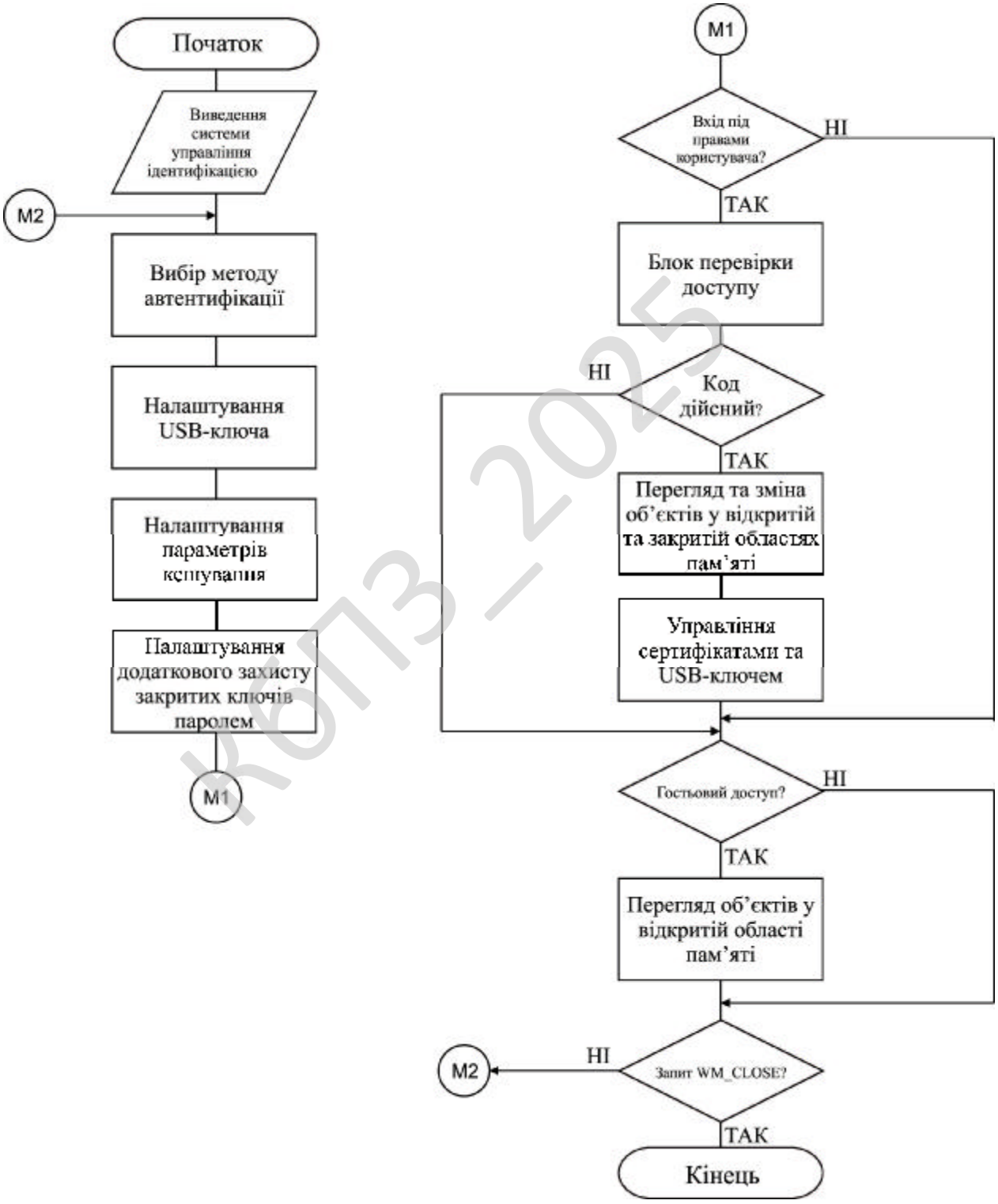

Рисунок 4.1 – Блок-схема основної програми

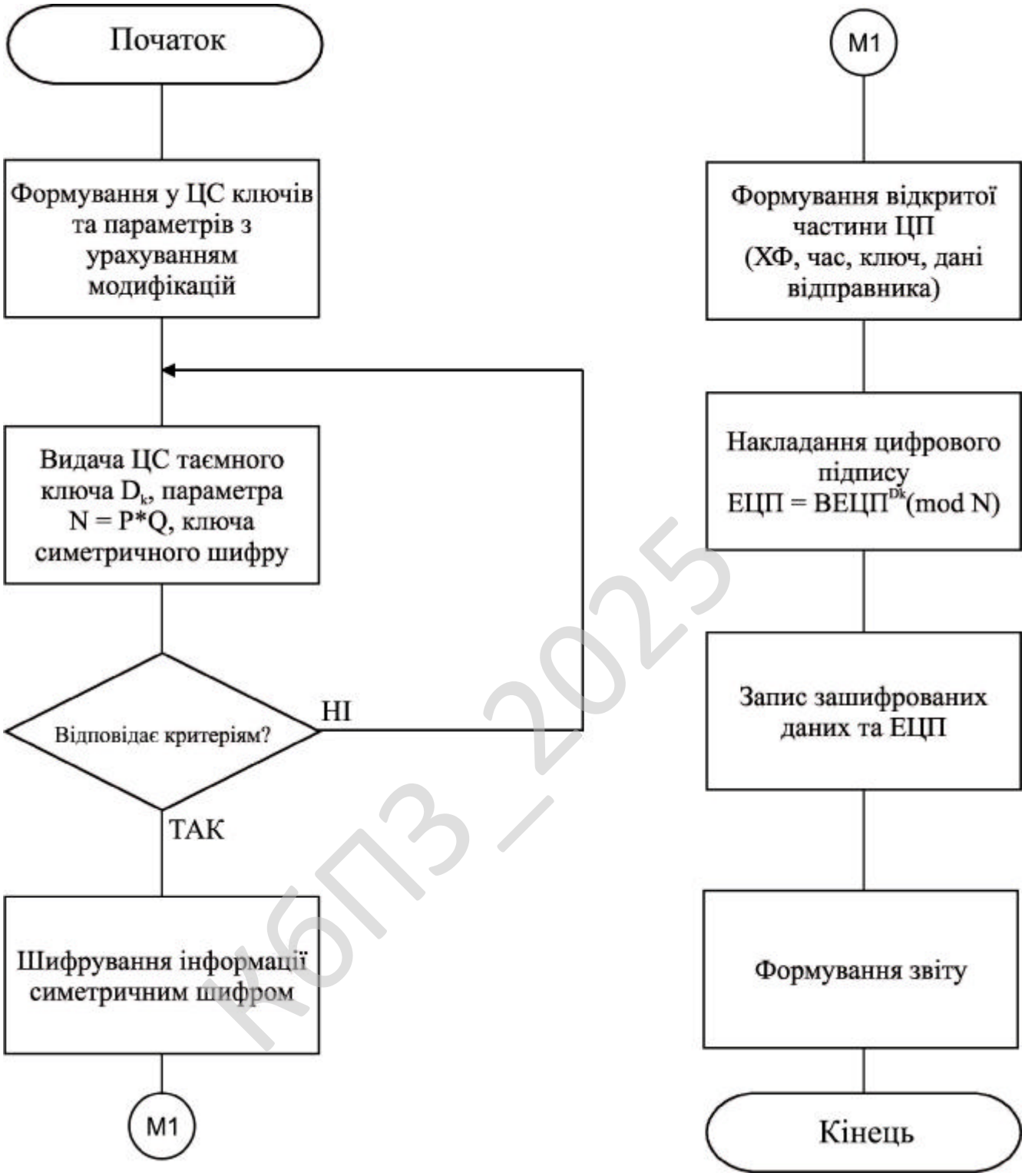

Рисунок 4.2 – Блок-схема роботи підпрограми

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

– асоціації (association relationship);

– включення (include relationship);

– розширення (extend relationship);

– узагальнення (generalization relationship).

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

Діаграма класів це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

В UML існують наступні типи зв'язків які використовуються у діаграмі класів: Асоціації; Агрегація; Композиція.

Асоціації це якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарної. Можна, хоча це рідко

буває необхідним, створювати асоціації, що зв'язують відразу кілька класів. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами.

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає необхідність посилатися на них і відрізняти один від одного. Ім'я буде особливо корисним, якщо між одними і тими ж класами існує кілька різних асоціацій.

Клас, що бере участь в асоціації, грає в ній деяку роль. По суті, це "обличчя", яким клас, що знаходиться на одній стороні асоціації, звернений до класу з іншого її боку. Можна явно позначити роль, яку клас грає в асоціації.

Часто при моделюванні буває важливо вказати, скільки об'єктів може бути пов'язано допомогою одного примірника асоціації. Це число називається кратністю (Multiplicity) ролі асоціації та записується або як вираз, значенням якого є діапазон значень, або в явному вигляді.

Вказуючи кратність на одному кінці асоціації, ви тим самим говорите, що на цьому кінці саме стільки об'єктів повинно відповідати кожному об'єкту на протилежному кінці. Кратність можна задати рівною одиниці (1), можна вказати діапазон: "нуль або одиниця" (0..1), "багато" (0.. *), "одиниця або більше" (1.. *). Дозволяється також вказувати певне число (наприклад, 3). За допомогою списку можна задати і більш складні кратності, наприклад 0.. 1, 3..4, 6.. *, що означає "будь-яке число об'єктів, крім 2 і 5".

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого». Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовані ромбиком.

Діаграма компонент в UML це діаграма, на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись.

Модуль програмного забезпечення може бути представлено в якості компоненти. Деякі компоненти існують під час компіляції, деякі – під час компонування, а деякі під час роботи програми.

Діаграма компонент відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортування.

Компоненти об'єднуються разом використовуючи структурні зв'язки (assembly connector) щоб об'єднати інтерфейси двох компонент. Це ілюструє зв'язок типу «клієнт-сервер».

Структурна взаємодія – «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту».

При використанні діаграми компонент щоб показати внутрішню структуру компонента, клієнтські та серверні інтерфейси можуть утворювати пряме з'єднання з внутрішніми. Таке з'єднання називається з'єднанням делегації.

Розгортання служб сертифікації

Для впровадження системи сертифікації й керування відкритими ключами в системі автентифікації користувачів пропонується використовувати дворівневу ієрархію центрів сертифікації: ізольований ЦС (Stand-alone Root CA) як центр сертифікації й ізольований підлеглий ЦС (Stand-alone subordinate CA) як центр реєстрації.

Розгортання системи сертифікації й керування відкритими ключами виконується в наступній послідовності:

– установка центра сертифікації;
– конфігурування центра сертифікації;
– установка й настроювання центра реєстрації;
– установка й настроювання сховища сертифікатів;
– установка й настроювання допоміжних систем і додатків.

Центр сертифікації поєднує людей, процеси, програмні й апаратні засоби, залучені в безпечне зв'язування імен користувачів і їхніх відкритих ключів. Центр сертифікації відомий суб'єктам інфраструктура відкритих ключів алгоритму RSA по двох атрибутах: назві й відкритому ключу. Центр сертифікації включає своє ім'я в кожний випущений їм сертифікат і в **список анульованих сертифікатів** (CAC) і підписує їх за допомогою власного секретного ключа. Користувачі можуть легко ідентифікувати сертифікати по ім'ю центр сертифікації і переконатися в їхній дійсності, використовуючи його відкритий ключ. Центр сертифікації – головний керуючий компонент інфраструктури відкритих ключів алгоритму RSA – виконує наступні основні функції:

– формує власний секретний ключ; якщо є головним центр сертифікації, то видає й підписує свій сертифікат, який називається **самовиданим** або **самопідписаним**;

– випускає (тобто створює й підписує) сертифікати відкритих ключів підлеглих центрів, що засвідчують, і кінцевих суб'єктів інфраструктури відкритих ключів алгоритму RSA; може випускати крос-сертифікати, якщо зв'язано відносинами довіри з іншими інфраструктурами відкритих ключів алгоритму RSA;

– підтримує реєстр сертифікатів (базу всіх виданих сертифікатів) і формує списки CAC із регулярністю, певної регламентом сертифікаційного центра;

– публікує інформацію про статус сертифікатів і списків CAC.

Реєстраційний центр (РЦ) є необов'язковим компонентом інфраструктури відкритих ключів алгоритму RSA. Звичайно РЦ одержує від центра, що засвідчує, повноваження реєструвати користувачів, забезпечувати їхня взаємодія із центр сертифікації і перевіряти інформацію, що заноситься в сертифікат. Сертифікат може містити інформацію, що надана суб'єктом, що подає заявку на сертифікат і пред'являє документ (паспорт, права водія, чекову книжку й т.п.) або третьою стороною (наприклад, кредитним агентством – про кредитний ліміт пластикової карти). Іноді в сертифікат включається інформація з відділу кадрів або дані, що характеризують повноваження суб'єкта в компанії (наприклад, право підпису документів певної категорії). РЦ агрегує цю інформацію й надає її центр сертифікації.

Репозиторій – спеціальний об'єкт інфраструктури відкритих ключів, база даних, у якій зберігається реєстр сертифікатів. Репозиторій значно спрощує керування системою й доступ до ресурсів. Він надає інформацію про статус сертифікатів, забезпечує зберігання й поширення сертифікатів і CAC, управляє внесеннями змін у сертифікати. До репозиторію пред'являються наступні вимоги:

– простота й стандартність доступу;

– регулярність відновлення інформації;

– вбудована захищеність;

– простота керування;

– сумісність із іншими сховищами (необов'язкова вимога).

Архів сертифікатів виконує функцію довгострокового зберігання (від імені сертифікаційного центра) і захисту інформації про всі видані сертифікати. Архів підтримує базу даних, використовувану при виникненні споровши із приводу надійності електронних цифрових підписів, якими в минулому засвідчувалися документи. Архів підтверджує якість інформації в момент її одержання й забезпечує цілісність даних під час зберігання. Інформація, надавана центр сертифікації архіву, повинна бути достатньої для визначення статусу сертифікатів і їхнього видавця. Архів повинен бути захищений відповідними технічними засобами й процедурами.

Кінцеві суб'єкти, або користувачі, інфраструктури відкритих ключів алгоритму RSA діляться на дві категорії: власники сертифікатів і сторони, що довіряють. Вони використовують деякі сервіси й функції, щоб одержати сертифікати або перевірити сертифікати інших суб'єктів. Власником сертифіката може бути фізична або юридична особа, додаток, сервер і т.д. сторони, Що Довіряють, запитують і покладаються на інформацію про статус сертифікатів і відкритих ключів підпису своїх партнерів по діловому спілкуванню.

Операційні протоколи – це протоколи для доставки сертифікатів (або інформації про їхній статус) і списків анульованих сертифікатів до клієнтських систем, що використовують сертифікати.

Протоколи керування необхідні для підтримки взаємодій між користувачем інфраструктури відкритих ключів і суб'єктами керування. Протоколи керування підтримують:

– реєстрацію суб'єкта для одержання сертифіката;
– ініціалізацію (наприклад, генерації пари ключів);
– випуск сертифіката;
– відновлення пари ключів;
– відновлення пари ключів після закінчення терміну дії сертифіката;
– обіг із запитом про анулювання сертифіката;

– крос-сертифікацію, коли два сертифікаційних центри обмінюються інформацією для генерації крос-сертифіката.

Політика застосування сертифікатів і регламент центра сертифікації є в документах, де приведенні зобов'язання сторін і правила використання сертифікатів.

Загальна схема функціонування інфраструктури відкритих ключів алгоритму RSA працює наступним чином. Користувач відправляє запит на сертифікат у РЦ (транзакція керування). Якщо запит фактично схвалений, то направляється безпосередньо в центр сертифікації для завірення цифровим підписом. Центр сертифікації перевіряє запит на сертифікат, і якщо той проходить верифікацію, то підписується й випускається сертифікат. Сертифікат публікується в репозиторії; залежно від конкретної конфігурації інфраструктури відкритих ключів алгоритму RSA, ця функція може бути покладена на реєстраційний або центр, що засвідчує. Процес анулювання сертифіката аналогічний процесу його генерації. Кінцевий суб'єкт запитує центр сертифікації про анулювання свого сертифіката, РЦ приймає рішення й направляє запит про анулювання в центр сертифікації. Центр сертифікації вносить зміни в список анульованих сертифікатів і публікує його в репозиторії. Кінцеві суб'єкти можуть перевірити статус конкретного сертифіката через операційний протокол

Поточні завдання системи сертифікації й керування відкритими ключами

До завдань, постійно виконуваних системою сертифікації й керування відкритими ключами, відносяться:

– архівування й відновлення центрів сертифікації;
– відмова або схвалення запитів на сертифікати;
– відкликання сертифікатів;
– публікація списків відкликаних сертифікатів (CRL);
– відновлення сертифікатів центрів сертифікації;
– відновлення сертифікатів користувачів.

Відновлення після збою

Серйозні збої, такі, як відмова жорсткого диска або компрометація сертифіката ЦС, здатні повністю порушити роботу служб сертифікації. Існує кілька шляхів мінімізації негативних наслідків таких збоїв і забезпечення своєчасного відновлення. До операцій по зниженню ризику відмови або компрометації ЦС відносяться:

– захист закритих ключів центрів сертифікації;

– розробка планів відновлення.

Наступні операції дозволяють попередити ризик відмови ЦС і звести до мінімуму час простою служб ЦС:

– створення дублюючих ЦС;

– часте архівування ЦС, що дозволяє швидко й з мінімальними втратами даних відновити центр сертифікації;

– установка служб сертифікації на дискових масивах і масивах RAID-5;

– розробка планів відновлення й навчання адміністраторів виконанню цих планів;

– збереження дані конфігурації ЦС для безпроблемного й точного відновлення конфігурації у випадку збою.

Забезпечення безпеки центрів сертифікації

Комп'ютери, на яких працюють ЦС, – це найбільш імовірні мети для атаки зловмисників, що намагаються порушити роботу служб або скомпрометувати захист і інформаційних систем. Одержавши доступ до ЦС або скориставшись недоліками захисту, ті що атакують можуть одержати доступ до ресурсів і скомпрометувати безпеку цілого ланцюжка довіри. Тому ЦС мають потребу в більше надійному захисті, чим звичайні.

Ризик атаки на ЦС залежить від багатьох факторів, у тому числі від захищеності , цілей атаки й витрат на неї. До можливих негативних наслідків компрометації ЦС відносяться:

– втрата інформації, що становить інтелектуальну власність організації;

– відмова або простій служб;

– ушкодження або руйнування ресурсів;

– витрати на усунення наслідків компрометації ЦС і повторне розгортання центрів сертифікації й сертифікатів.

Компрометація кореневого ЦС обійдеться набагато дорожче, ніж проміжного або випускаючого ЦС. Щоб знизити потенційні збитки, варто створити в організації ієрархію з декількох ЦС.

Визначаючи адекватність мер безпеки, необхідно зважити й оцінити витрати на ці заходи щодо забезпечення безпеки й можливі збитки у випадку компрометації ЦС. До заходів щодо забезпечення безпеки ЦС відносяться:

– розміщення у захищених центрах даних і надання фізичного доступу до них тільки довіреним адміністраторам;

– використання апаратних центрів сертифікації або апаратних постачальників CSP для забезпечення максимального захисту закритих ключів ЦС;

– конфігурування параметрів безпеки для забезпечення високого рівня захисту, такого, як рівень High Security (Високий) захисту шаблонів;

– використання службової програми Windows 2000 System Key (SysKey) для забезпечення шифрування захищених сховищ ЦС;

– аудит безпеки для контролю й спостереження за можливими атаками на ЦС;

– обмеження надання прав користувачам шляхом надання прав тільки відповідній групі адміністраторів (іншим користувачам або групам треба заборонити перегляд або виконання будь-яких завдань на локальному комп'ютері зі ЦС);

– відключення непотрібних служб на х зі ЦС; працюючі служби являють собою додаткові «лазівки» для зловмисників;

– розгортання політики й виконання процедур безпеки при розгортанні ЦС на підприємстві.

При виборі мір безпеки ЦС варто зважити витрати на їхню реалізацію й підтримку, з одного боку, і ризик нападу на ЦС і можливі збитки від компрометації ЦС, з іншої сторони. У загальному випадку чим вище ризик нападу й збитки від компрометації, тим більше витрати на заходи щодо забезпечення безпеки ЦС. Максимальний захист варто забезпечити кореневим ЦС, а проміжні ЦС треба убезпечити надійніше, ніж випускаючі ЦС.

Допустимо, що в організації ухвалено рішення захистити великий обсяг надзвичайно коштовної й конфіденційної інформації, використавши рішення на основі відкритих ключів. Також вирішено придбати дорогий апаратний ЦС для виконання функцій кореневого ЦС і розмістити його для безпеки в захищеному сховищі, розташованому в головному офісі організації. Доступ до кореневого ЦС, що сертифікує всі проміжні ЦС у підрозділах, надається тільки довіреним адміністраторам. Проміжні ЦС – це ізольовані ЦС на комп'ютерах під керуванням Windows 10/11, від’єднаних від й розміщених у захищених центрах даних під наглядом адміністратора підрозділу. Проміжний ЦС використовується в міру необхідності для сертифікації випускаючих ЦС під керуванням Windows 10/11 відповідно до потреб кожного підрозділу. Випускаючі ЦС – це ЦС Windows 10/11 підприємства або ізольовані ЦС, розміщені в захищених центрах даних кожного підрозділу. Політика безпеки організації повинна мати на увазі самі строгі міри безпеки виконання запиту, авторизації й впровадження кореневого, проміжних і випускаючих ЦС на підприємстві й контроль за ними.

З іншого боку, якщо в організації рішення безпеки на базі відкритого ключа застосовуються для захисту не дуже коштовної інформації, цілком достатньо ізольованого кореневого ЦС Windows 10/11, розміщеного в центрі даних, а не згаданого в попередньому прикладі дорогого апаратного ЦС, розміщеного в захищеному сховищі. При цьому припустимо розмістити проміжні й випускаючі ЦС у підрозділах за межами центра даних. Також не потрібно таких строгих обмежень на впровадження, запити й авторизацію ЦС.

Служби сертифікації Windows 10/11 на базі постачальника Microsoft Base

CSP зможуть задовольнити більшість потреб у захисті ЦС. Для забезпечення найвищої безпеки ЦС варто використовувати апаратні ЦС.

Захист закритих ключів алгоритму RSA центрів сертифікації

Якщо в зловмисника є доступ до комп'ютера ЦС – безпосередньо або по , він у стані розшифрувати особистий ключ і потім виступати від імені ЦС і одержувати доступ до коштовних ресурсів. Він зможе красти інформацію, порушити роботу служб і знищити ресурси. Скомпрометований ключ алгоритму RSA ЦС підриває й зводить «на ні» всю систему безпеки, забезпечену цим ЦС, і всю його ієрархію ЦС. Необхідно регулярно проводити заходу щодо зниження атаки на ключі ЦС.

Необхідно забезпечити захист із центрами сертифікації, як описано раніше в цьому розділі. Забезпечення фізичної безпеки мінімізує ризик одержання атакуючого доступу до ЦС або захищеному сховищу (будь воно апаратним або програмним), де зберігається ключ алгоритму RSA ЦС. Забезпечення безпеки й (програмного забезпечення) знижує ризик того, що порушники одержать доступ до ЦС або скористаються недоліками додатків або служб цього для компрометації ключа ЦС. Необхідно забезпечити посилений захист ключів центра сертифікації. Якщо потрібна максимальна безпека закритих ключів, необхідно скористатися апаратними постачальниками CSP, тому що в цьому випадку ключі зберігаються на стійкі до злому апаратних пристроях і ніколи не надаються операційній системі. Необхідно використовувати службову програму SysKey для забезпечення додаткового захисту закритих ключів ЦС, збережених постачальниками Microsoft CSP.

Використання в центрах сертифікації ключів великої довжини знижує ризик атаки на ключ алгоритму RSA, однак довгі ключі вимагають більше дискового простору й обчислювальних потужностей для підписання сертифікатів. Варто вибрати максимальну можливу довжину ключа й урахувати обмеження на пам'ять і продуктивність ЦС.

Наприклад, 4096-бітний ключ алгоритму RSA ЦС забезпечує чудову

безпеку, але підписання сертифікатів таким довгим ключем займає занадто багато часу навіть при наявності плати криптоакселератора. Такий ключ алгоритму RSA цілком придатний для коренев або проміжного ЦС, які використовуються нечасто й тільки для сертифікації підлеглого ЦС. Для більшості випускаючих ЦС 4096-бітних ключів неприйнятний через неприпустиме зниження продуктивності роботи ЦС. На випускаючих ЦС варто використовувати ключі, які забезпечують задовільну безпеку, не сповільнюють роботу ЦС і відповідають довгостроковим цілям конкретної служби сертифікації. Для підвищення продуктивності випускаючого ЦС і використання більше довгих ключів рекомендується встановити плату криптоакселератора. Перш ніж розгортати ЦС на підприємстві, необхідно протестувати його продуктивність для різних довжин ключів у лабораторних умовах і на пілотних системах.

Необхідно визначити адекватні терміни дії ключів ЦС. Чим більше термін дії ключа, тим вище ризик його компрометації, тому що в атакуючих більше часу на злом. Не існує простого правила визначення максимальних термінів дії ключів. Однак у загальному випадку термін дії ключів у значній мірі залежать від якості їхнього захисту й довжини. У загальному випадку, терміни дії більше довгих ключів більше. Аналогічно більше захищені ключі служать довше. Наприклад, ключі, що зберігаються в стійкі до злому апаратних криптоустройствах надійніше, ніж ключі, розміщені на жорсткому диску локального комп'ютера. Тому для двох ключів однієї довжини термін дії ключа, збереженого на апаратному криптографічному пристрої, звичайно більше, ніж ключа, розміщеного в програмному постачальнику CSP на жорсткому диску.

Розробка планів відновлення

Розробка детального плану відновлення дозволяє швидко повернути ЦС у робочий стан у випадку збоїв служб сертифікації або компрометації. Рекомендується протестувати складений план, щоб переконатися в його коректності. Крім того варто провести навчання співробітників, щоб бути впевненим у тім, що вони знають, як діяти відповідно до цього плану.

План відновлення повинен передбачати:

– процедури відновлення й контрольні списки для адміністраторів;

– набори засобів і службових програм відновлення;

– план дій у непередбачених обставинах.

Існує безліч причин, по яких ЦС може потерпіти збій, у тому числі поломка жорсткого диска , відмова мережної карти або вихід з ладу системної (материнської) плати. Деякі збої усуваються швидко – шляхом локалізації й виправлення джерела неполадки ЦС. Наприклад, після заміни несправної мережної карти або системної плати для відновлення служби сертифікації досить запустити знову комп'ютер.

Жорсткий диск, що відмовив, заміняється, і ЦС відновлюється з останнього архіву. При ушкодженні або руйнуванні ЦС його відновлюють також з останнього архіву. Після цього ЦС установлюють у вихідній конфігурації й з вихідним особистим ключем і сертифікатом ЦС.

При виявленні факту компрометації ЦС треба негайно:

– відкликати сертифікат скомпрометованого ЦС;

– опублікувати новий список CRL з відкликаним сертифікатом ЦС;

– видалити скомпрометовані сертифікати ЦС зі сховища TRCA (Довірені кореневі центри сертифікації) і із всіх списків довіри (CTL);

– сповістити всіх зацікавлених користувачів і адміністраторів про компрометацію й відізвати сертифікати, які випущені скомпрометованим ЦС;

– усунути всі «лазівки», що стали причиною компрометації.

Для відновлення ієрархії центрів сертифікації варто повторно виконати розгортання нового ЦС, далі – повторно випустити всі сертифікати для користувачів, комп'ютерів і служб.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Twofish, який є симетричним алгоритмом блочного шифрування з розміром блоку 128 біт і довжиною ключа до 256 біт. Число раундів 16. Розроблено групою фахівців на чолі з Брюсом Шнайером. Був одним з п'яти фіналістів другого етапу конкурсу AES. Алгоритм розроблений на основі алгоритмів Blowfish, SAFER і Square.

Відмінними особливостями алгоритму є використання попередньо обчислюваних та залежних від ключа S-box'ів і складна схема розгортки підключення шифрування. Половина n-бітного ключа шифрування використовується як власне ключ шифрування, інша – для модифікації алгоритму (від неї залежать S-box'и).

Twofish розроблявся спеціально з урахуванням вимог та рекомендацій NIST для конкурсу AES [1]:

– 128-бітний блочний симетричний шифр.

– Довжина ключів 128, 192 і 256 біт.

– Відсутність слабких ключів.

– Ефективна програмна (в першу чергу на 32-бітних процесорах) та апаратна реалізація.

– Гнучкість (можливість використання додаткових довжин ключа, використання в поточному шифруванні, хеш-функціях і т.д.).

– Простота алгоритму – для можливості його ефективного аналізу.

Однак саме складність структури алгоритму і, відповідно, складність його аналізу на предмет слабких ключів або прихованих зв'язків, а також досить повільне час шифрування порівняно з Rijndael на більшості платформ, зіграло не на його користь.[2]

Алгоритм Twofish виник в результаті спроби модифіковані алгоритм Blowfish для 128-бітового вхідного блоку. Новий алгоритм повинен був бути

легко реалізованим апаратно (у тому числі використовувати таблиці меншого розміру), мати досконалішу систему розширення ключа (key schedule) і мати однозначну функцію F.

В результаті, алгоритм був реалізований у вигляді змішаної мережі Фейстеля з чотирма гілками, які модифікують один одну з використанням криптоперетворень Адамара (Pseudo-Hadamar Transform, PHT).

Можливість ефективної реалізації на сучасних (для того часу) 32-бітних процесорах (а також в смарт-картах і подібних пристроях) – один з ключових принципів, яким керувалися розробники Twofish.

Наприклад, у функції F при обчисленні PHT і складання з частиною ключа K навмисно використовується додавання, замість традиційного xor. Це дає можливість використовувати команду LEA сімейства процесорів Pentium, яка за один такт дозволяє обчислити перетворення Адамара. (Правда в такому випадку код доводиться компілювати під конкретне значення ключа).

Алгоритм Twofish не запатентований і може бути використаний ким завгодно без будь-якої плати або відрахувань. Він використовується в багатьох програмах шифрування, хоча і отримав менше поширення, ніж Blowfish.

Опис алгоритму

Twofish розбиває вхідний 128-бітний блок даних на чотири 32-бітних підблоки, над якими, після процедури вхідного відбілювання (input whitening), проводиться 16 раундів перетворень. Після останнього раунду виконується вихідна відбілювання (output whitening).

Відбілювання (whitening)

Відбілювання – це процедура xor'а даних з підключами перед першим раундом і після останнього раунду. Вперше ця техніка була використана в шифрі Khufu/Khare і, незалежно, Рональдом Рівестом (англ. Ron Rivest) в алгоритмі шифрування DESX. Joe Killian (NEC) і Phillip Rogaway (Каліфорнійський університет) показали, що відбілювання дійсно ускладнює завдання пошуку ключа повним перебором (exhaustive key-search) в DESX[3]. Розробники Twofish

стверджують[4], що в Twofish відбілювання також істотно ускладнює завдання підбору ключа, оскільки криптоаналітика не може дізнатися, які дані потрапляють на вхід функції F першого раунду.

Тим не менш, виявилися і негативні сторони. Цікаве дослідження провели фахівці дослідницького центру компанії IBM.[5] Вони виконали реалізацію алгоритму Twofish для типової смарт-карти з CMOS-архітектурою і проаналізували можливість атаки за допомогою диференціального аналізу споживаної потужності (DPA – Differential Power Analysis). Атаці піддавалася саме процедура вхідного вибілювання, оскільки вона безпосередньо використовує xor підключів з вхідними даними. У результаті дослідники показали, що можна повністю обчислити 128-бітовий ключ проаналізувавши всього 100 операцій шифрування довільних блоків.

Функція g

Функція g – основа алгоритму Twofish. На вхід функції подається 32-бітове число X, яке потім розбивається на чотири байти x0, x1, x2, x3. Кожен з вийшов байтів пропускається через свій S-box. (Слід зазначити, що S-box'и в алгоритмі не фіксовані, а залежать від ключа). Отримані 4 байти на виходах S-box'ов інтерпретуються як вектор з чотирма компонентами. Цей вектор множиться на фіксовану матрицю MDS (maximum distance separable) розміром 4x4, причому обчислення проводяться в скінченному полі по модулю непривідного многочлена

MDS матриця – це така матриця над кінцевим полем K, що якщо взяти її як матрицю лінійного перетворення з простору у простір , то будь-які два вектори з простору виду (x, f (x)) будуть мати як мінімум m+1 відмінностей в компонентах. Тобто набір векторів вигляду (x, f (x)) утворює код, що володіє властивістю максимального рознесення (maximum distance separable code). Таким кодом, наприклад, є код Ріда-Соломона.

В Twofish властивість максимальної рознесеність матриці MDS означає, що загальна кількість змінних байт вектора a і вектора не менше п'яти. Іншими

словами, будь-яка зміна тільки одного байта в a призводить до зміни всіх чотирьох байтів в b.

Криптоперетворення Адамара (Pseudo-Hadamar Transform, PHT)

Криптоперетворення Адамара – оборотне перетворення бітового рядка довжиною 2n. Рядок розбивається на дві частини a і b однакової довжини в n біт. Перетворення обчислюється таким чином:

)

)

Ця операція часто використовується для «розсіювання» коду (наприклад в шифрі SAFER).

В Twofish це перетворення використовується при змішуванні результатів двох g-функцій (n = 32).

Циклічний зсув на 1 біт

У кожному раунді два правих 32-бітових блоки, які xor-яться з результатами функції F, додатково циклічно зрушуються на один біт. Третій блок зрушується до операції xor, четвертий блок – після. Ці зрушення спеціально додані, щоб порушити вирівнювання по байтах, яке властиво S-box'ам та операції множення на MDS-матрицю. Проте шифр перестає бути повністю симетричним, так як при шифруванні й розшифровці зрушення слід здійснювати в протилежні сторони.

Генерація ключів

Twofish розрахований на роботу з ключами довжиною 128, 192 і 256 біт. З вихідного ключа генерується 40 32-бітних підключів, перші вісім з яких використовуються тільки в операціях вхідного і вихідного вибілювання, а решта 32 – в раундах шифрування, по два підключі на раунд. Особливістю Twofish є те, що вихідний ключ використовується також і для зміни самого алгоритму шифрування, так як використовуються у функції g S-box'и не фіксовані, а залежать від ключа.

Для формування раундових підключів вихідний ключ M розбивається з перестановкою байт на два однакові блоки M_o і M_e. Потім за допомогою блоку M_o і функції h шифрується значення 2 * i, а за допомогою блоку M_e шифрується значення 2*i+1, де i – номер поточного раунду (0 – 15). Отримані зашифровані блоки змішуються криптоперетвореням Адамара, і потім використовуються як раундові підключі.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання магістерської роботи. Розроблене програмне забезпечення системи управління ідентифікацією та доступом до інформаційних ресурсів складається з наступних функціональних блоків:

– Навігаційне меню: Система автентифікації; Сертифікати; Параметри; Журнал подій; Довідка.

– Вікна обрання профілю.

– Вікно виведення результату роботи системи.

– Функціональних кнопок ПЗ: Параметри; Про програму ; Змінити ПІН; Журнал; Додати; Видалити; Редагувати; Додати сертифікат.

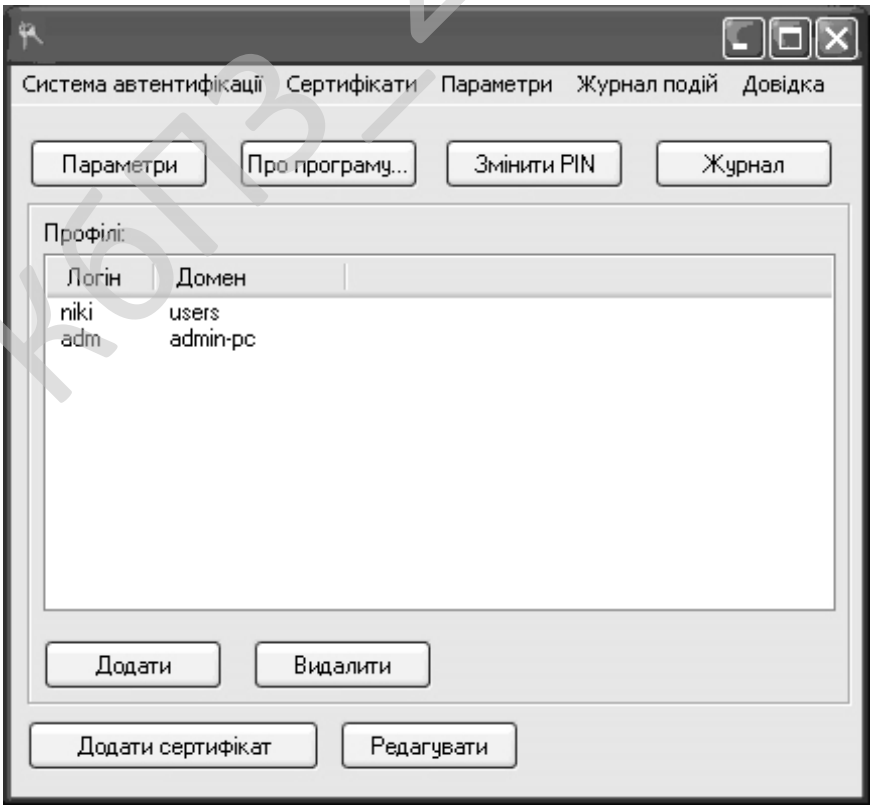

Рисунок 5.1 – Головне вікно розробленого ПЗ

Для перегляду короткої довідки про програму слід натиснути на основному вікні кнопку авторського права, після чого на екрані з’явиться вікно показане на рисунку 5.2.

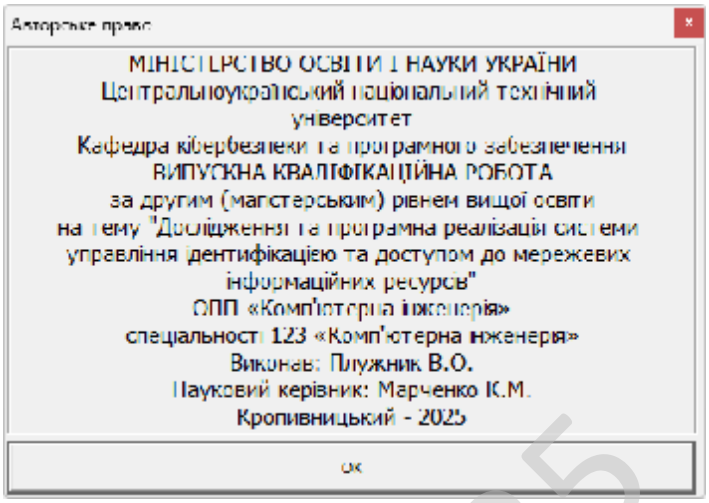

Рисунок 5.2 – Вікно розробника ПЗ

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

– відповідності поставленим вимогам;

– правильна відповідь для усіх можливих вхідних даних;

– виконання функцій за прийнятний час;

– практичність;

– сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

– Гарантується перевірка всіх незалежних маршрутів програми.

– Знаходяться гілки True, False для всіх логічних рішень.

– Виконуються всі цикли (у межах їхніх кордонів та діапазонів).

– Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

– Кількість незалежних маршрутів може бути дуже велика.

– Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

– У програмі можуть бути пропущені деякі маршрути.

– Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

– Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

– Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Обрано умови розповсюдження – commercial software.

Програмне забезпечення, створене комерційною організацією з метою отримання прибутку від його використання іншими, наприклад, шляхом продажу копій.

Найважливішою особливістю комерційний програмних продуктів є підтримка великих компаній, прямо зацікавлених у поширенні програм. Багато організацій надають виключно платну підтримку своїх продуктів, такий підхід, як правило, використовують організації надають відкриті вихідні коди. Для продуктів, що розповсюджуються на комерційній основі діють зазвичай безкоштовні служби підтримки, покликані збільшити рівень довіри у клієнтів і потенційних покупців.

Далеко не завжди, але як правило терміни критично важливих змін в комерційних продуктах значно менше, ніж у некомерційних проектів. Це пов'язано з тим, що над комерційним продуктом працюють цілі групи розробників і ця робота є їх основним заняттям. Розробникам-початківцям як правило доводиться шукати додаткові способи заробітку, і це збільшує час, що витрачається на доповнення і зміни програм. Так як основним рушійним фактором створення комерційного ПЗ є одержання прибутку, то комерційні програмні продукти першими заповнюють вільні ніші та пропонують варіанти вирішення завдань відразу по мірі виявлення вакууму в будь-якому секторі ринку.

Окремий вид комерційних програм, коли їх розробка оплачується безпосередньо замовником. Такі програми найчастіше позбавлені всіх переваг комерційних продуктів, оскільки мають обмежений бюджет, але більш адаптовані до вимог замовника, ніж аналоги.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

Метою розробки є дослідження та програмна реалізація системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

Об'єктом дослідження є процес управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

Предметом дослідження є методи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

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

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

– Розроблено вітчизняний продукт управління ідентифікацією та доступом до мережевих інформаційних ресурсів, який має більш широкі можливості, на відміну від існуючих аналогів.

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та впровадження системи управління ідентифікацією та доступом (IAM) можуть бути цікавими передусім великим корпораціям і державним структурам, які мають складну інформаційну інфраструктуру з великою кількістю користувачів і систем. Такі організації стикаються з проблемами контролю доступів, необхідністю швидко створювати та блокувати облікові записи, а також із вимогами кібербезпеки, що постійно зростають. Для них впровадження IAM є не лише технічним покращенням, а стратегічним кроком у напрямку цифрової безпеки.

Малий і середній бізнес також може знайти у цій системі практичну користь, особливо ті компанії, які працюють із конфіденційними даними клієнтів або партнерами у сфері IT, фінансів чи медицини. Для них IAM – це можливість уникнути витоків інформації, автоматизувати процеси керування правами доступу та спростити проходження аудитів.

Результати дослідження будуть цінними і для освітніх закладів, де в різних студентів, викладачів і адміністрації є потреба у доступі до окремих частин мережевих ресурсів. Впровадження такої системи дозволяє керувати цифровими ідентичностями всіх користувачів без хаосу, який зазвичай виникає у великих освітніх платформах.

Також розробка має потенціал бути цікавою для компаній-інтеграторів і постачальників IT-послуг, які прагнуть розширити свій портфель продуктів у сфері безпеки. IAM-системи сьогодні стають стандартом для організацій, що прагнуть відповідати міжнародним вимогам і забезпечувати безпечне середовище для обміну даними.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для визначення привабливості системи ІАМ доцільно використати метод експертних оцінок, що ґрунтується на думках спеціалістів із різних сфер – ІТ-безпеки, корпоративного управління, економіки та аналітики ризиків. Кожен експерт оцінює систему за кількома критеріями: економічна доцільність, технічна ефективність, рівень автоматизації, зручність використання та потенціал масштабування. За результатами узагальнення оцінок формується середній показник привабливості, який дає змогу об’єктивно визначити ринкову перспективність системи.

Наприклад, якщо десять експертів надали свої оцінки, то середній бал може становити 8,7 із 10, що свідчить про високу зацікавленість у впровадженні рішення. При цьому найбільше балів система отримала за критерій “зменшення людського фактору” та “економію робочого часу адміністраторів”. Це підтверджує актуальність системи для підприємств, які прагнуть автоматизувати процеси управління доступом і підвищити безпеку.

Такі оцінки також дозволяють побачити слабкі сторони продукту, наприклад, потребу у додаткових функціях аудиту або складність інтеграції з деякими старими системами. Тобто метод експертних оцінок не лише визначає рівень привабливості, а й допомагає розробникам удосконалити продукт.

У результаті ми отримуємо комплексну картину, яка враховує думки професіоналів і практиків, що працюють із подібними рішеннями. Це забезпечує обґрунтованість подальших рішень щодо просування ІАМ на ринку.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості системи ІАМ найдоцільніше застосувати витратно-функціональний метод, який дозволяє врахувати не лише прямі фінансові

інвестиції, але й вартість функціональної вигоди від автоматизації процесів. Цей підхід дає змогу оцінити проєкт із двох боків – скільки коштує його впровадження та яку цінність він приносить підприємству у вигляді зменшення витрат на ручне адміністрування, підвищення продуктивності персоналу і скорочення кількості інцидентів безпеки.

Додатково можна залучити елементи доходного методу, оскільки IAM сприяє підвищенню ефективності бізнес-процесів і непрямо впливає на прибуток компанії. Наприклад, завдяки швидшій роботі користувачів та зниженню простоїв, пов'язаних із блокуванням облікових записів, компанія зменшує втрати часу, а отже – підвищує продуктивність праці.

Також можна врахувати метод аналогій, коли порівнюється вартість впровадження подібних рішень у компаніях зі схожими масштабами діяльності. Це допомагає реалістично оцінити витрати і зрозуміти, які параметри найбільше впливають на кінцеву ціну (наприклад, кількість користувачів або рівень інтеграції з існуючими системами).

Застосування комбінованого підходу дозволяє сформувати обґрунтовану оцінку, яка буде враховувати як поточні витрати, так і довгострокові переваги від упровадження системи IAM.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

На підприємстві працює близько 800 співробітників, які користуються десятками корпоративних інформаційних систем – ERP, CRM, внутрішнім документообігом, базами даних і хмарними сервісами. До впровадження IAM кожен користувач мав окремі облікові записи, а адміністратори вручну створювали, змінювали й блокували доступи. Це займало багато часу, створювало ризики несанкціонованого доступу після звільнення працівників і ускладнювало контроль дотримання політик безпеки.

Система управління ідентифікацією та доступом (IAM) автоматизує процеси автентифікації, авторизації та управління правами користувачів. Вона забезпечує єдину точку входу до всіх ресурсів (Single Sign-On), автоматичне надання ролей при прийомі працівника на роботу та їх відкликання при звільненні, а також централізований аудит усіх дій користувачів. Це дозволяє зменшити адміністративні витрати, підвищити рівень безпеки і забезпечити відповідність вимогам внутрішніх регламентів та стандартів ISO/IEC 27001. Вхідні дані зафіксовано в таблиці 7.1.

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	**До впровадження**	**Після впровадження**	**Економічний ефект**
Кількість адміністраторів, що займаються управлінням доступами	4	2	–2
Середня заробітна плата адміністратора	45 000 грн/міс	—	—
Річні витрати на адміністрування доступів	2 160 000 грн	1 080 000 грн	–1 080 000 грн
Кількість інцидентів безпеки через неправильні права доступу	10	2	–8
Середня вартість ліквідації одного інциденту	150 000 грн	50 000 грн	–100 000 грн
Річні витрати на ліквідацію інцидентів	1 500 000 грн	100 000 грн	–1 400 000 грн
Витрати на впровадження системи IAM (ліцензії, сервери, інтеграція)	—	2 500 000 грн	—
Щорічні витрати на обслуговування IAM	—	300 000 грн	—

Розрахунок економічного ефекту демонструє наступне: економія на адмініструванні доступів – 1 080 000 грн/рік, економія за рахунок зниження кількості інцидентів – 1 400 000 грн/рік, сукупний річний економічний ефект – 2 480 000 грн/рік, чистий економічний ефект – 2 180 000 грн/рік, термін окупності – 1,15 року (~14 місяців), коефіцієнт рентабельності – 87 %.

Додаткові (немонетарні) переваги: підвищення рівня безпеки – IAM усуває людський фактор у наданні доступів і знижує ризики витоку даних, зручність для користувачів – завдяки Single Sign-On співробітники входять у всі системи через один обліковий запис, відповідність стандартам безпеки – система полегшує аудит і демонструє контроль над управлінням доступом, прозорість процесів – журнал дій користувачів дозволяє швидко виявити та локалізувати проблеми, гнучкість інтеграції – IAM легко підключається до ERP, CRM, Microsoft Active Directory, Google Workspace, SAP тощо.

Таким чином, система IAM є не лише інструментом оптимізації витрат, а й основою корпоративної стратегії кіберзахисту, яка гарантує стабільну роботу бізнесу в умовах зростаючих цифрових ризиків.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Просування проєкту IAM доцільно почати з визначення цільової аудиторії – компаній, які мають потребу у високому рівні контролю доступу до даних. Після цього варто створити демонстраційні кейси з описом реальних проблем, які система вирішує: надлишкові права доступу, складність аудиту, втрати даних після звільнення працівників тощо. Такі приклади допомагають потенційним клієнтам краще зрозуміти практичну користь системи.

Далі важливо запустити інформаційну кампанію, яка включатиме презентації, вебінари та участь у спеціалізованих конференціях. Головна ідея – показати, що IAM не просто “ще один IT-продукт”, а рішення, яке зменшує ризики, економить кошти та підвищує прозорість бізнесу.

Важливу роль відіграє партнерство з інтеграторами та провайдерами кібербезпеки. Саме через них можна швидко розширити охоплення клієнтів без необхідності створення великої власної збутової структури.

Завершальним етапом має стати створення системи післяпродажної підтримки, яка забезпечить клієнтам довіру до продукту. Коли користувачі бачать, що компанія супроводжує їх навіть після впровадження, вони стають лояльними та рекомендують рішення своїм партнерам.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для ефективної реалізації проєкту IAM варто поєднувати кілька стратегій. По-перше, зробити акцент на прямих продажах великим підприємствам, адже саме вони мають найвищий рівень потреби в контролі доступу. Для цього можна використовувати персональні презентації або пілотні впровадження.

По-друге, необхідно розвивати партнерські канали збуту – співпрацю з компаніями, які вже впроваджують рішення у сфері IT-безпеки, документообігу чи управління інфраструктурою. Такі партнери зможуть включати IAM у свої портфелі продуктів, що допоможе вийти на нові ринки без значних витрат.

Також корисно створити онлайн-платформу з відкритою демонстраційною версією продукту, щоб клієнти могли самостійно оцінити його можливості. Такий інструмент допомагає збільшити довіру до системи та спрощує процес прийняття рішення.

Нарешті, важливо інвестувати у маркетинг через експертний контент – публікації, відео, аналітичні матеріали про кіберзагрози та управління доступами. Це створює імідж компанії як компетентного постачальника і зміцнює її позиції на ринку.

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключовими факторами успіху проєкту IAM є поєднання технологічної надійності, економічної доцільності та зручності для кінцевого користувача. Система має бути не лише безпечною, а й зрозумілою для адміністраторів і співробітників, які з нею працюють. Це дозволяє досягти високого рівня прийняття продукту в компанії.

Другим важливим фактором є інтеграційна гнучкість. Успішна IAM-система повинна легко поєднуватися з іншими корпоративними рішеннями – ERP, CRM, хмарними платформами, поштовими сервісами. Чим легше відбувається інтеграція, тим швидше компанія відчує ефект від впровадження.

Не менш значущою є система підтримки та оновлень, адже кіберзагрози постійно змінюються. Тому стабільна технічна підтримка та регулярні оновлення безпеки – запорука довготривалої ефективності системи.

І, звичайно, успіх визначається тим, наскільки IAM допомагає компанії досягати стратегічних цілей – підвищувати ефективність, зменшувати ризики й відповідати сучасним стандартам інформаційної безпеки. Саме це поєднання технології та користі робить систему справді успішною.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Сучасний розвиток технічного та технологічного стану виробництва передбачає постійну автоматизацію та оптимізацію виробничих процесів. Комп’ютер – невід’ємна складова сучасного життя, зокрема самих різноманітних галузей суспільної та виробничої діяльності людей. За допомогою обчислювальної техніки вирішують складні робочі задачі, ведуться наукові дослідження, створюються архітектурні креслення і твори мистецтва. Сьогодні, напевно, важко уявити компанію, господарська діяльність в якій здійснювалась би без використання комп’ютерної техніки.

Незважаючи на видиму безпеку та розвиток сучасних технологій, при роботі за комп'ютером є ряд чинників, які можуть вплинути на здоров'я людини. Через масовий характер робіт, що виконуються працівниками за допомогою комп’ютера, законодавством України чітко врегульовано норми та вимоги до використання комп’ютерної техніки на підприємстві й охорона праці при роботі за комп’ютером.

Законом України “Про охорону праці” [1] регламентуються загальні положення державної політики в галузі охорони праці, а реалізуються ці положення, зокрема, Вимогами щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями, затверджені наказом Мінсоцполітики від 14.02.2018р. № 207, зареєстровані в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 [2].

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи і достатньо великим навантаженням на м'язи рук при роботі з клавіатурою ЕОМ.

У даному розділі магістерської роботи висвітлюються основні питання охорони праці працівників, робота яких пов’язана з роботою за комп’ютером, планування робочого приміщення, де працюють користувачі ПК; параметри мікроклімату, освітленість робочих місць та виробничих приміщень; шумові завади.

Правильна організація і раціональне устаткування робочого місця дає можливість ефективно і з якнайменшими витратами праці виконувати свої функції. Велике значення має раціональна конструкція і розташовує елементів робочого місця, що важливе для підтримки оптимальної робочої пози людини-оператора, а також необхідно дотримувати правильний режим праці і відпочинку.

Що стосується питання охорони праці, його необхідно вирішувати на всіх стадіях трудового процесу незалежно від виду професійної діяльності.

Забезпечення безпечних і здорових умов праці в значній мірі залежить від правильної оцінки небезпечних, шкідливих виробничих факторів. Однакові по важкості зміни в організмі людини можуть бути викликані різними причинами. Це можуть бути фактори виробничого середовища, надмірне фізичне і розумове навантаження, нервово-емоційна напруга, а також різне сполучення цих причин.

Робота працівників пов’язана з роботою за комп’ютером, тому актуальною є розгляд саме умов праці та стану охорони праці працівників які постійно працюють із комп’ютерною технікою.

Щоб розробити якісний програмний продукт необхідно організувати безпеку на робочому місці програміста. Під час проектування безпеки на робочому місці з ПК необхідно домагатися високої якості та надійності технічного забезпечення, а також створювати комфортні параметри довкілля для працюючих.

8.2 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

У зазначеному приміщенні працюють 4 людей.

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	5,4
Довжина	6
Висота	2,75

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого·

Геометрична характеристика	Одиниця виміру	Нормативне значення *	Фактичне значення
Площа, S	$м^2$	не менше 6.0	8,1
Обсяг, V	$м^3$	не менше 20.0	24,3

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин) [2].

За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа та об’єм приміщення у розрахунку на одне робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони

праці під час експлуатації електронно-обчислювальних машин»). Таким чином, можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Вологість,%	Швидкість повітря, м/с	Температура,°С	Вологість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,1
Тепла	23-25	50-70	0,1	24-25	50-65	0,11

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер HP 1100, електродвигуни вентиляторів ЕОМ. Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці. З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [1], у яких прописані вимоги до використання всіх освітлювальних приладів, у тому числі світлодіодних.

Працю працівника, який постійно використовує комп’ютер, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В).

Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [1],

Крім того все поле зору повинне бути освітленим достатньо рівномірно – це основна гігієнічна вимога. Оскільки яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.3 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

– розміри приміщення, у розрахунку на одного працюючого, відповідають нормативам;

– мікроклімат відповідає нормативним значенням;

– акустичні умови роботи не перевищують нормативних значень;

Таким чином, можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга). Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В. Оскільки при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним. Для підвищення ефективності системи управління охороною праці дуже важлива роль належить формуванню і розвитку інформаційної культури фахівців ІТ-технологій.

8.4 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток 45х45х5 мм, довжиною L=2.5 м, та горизонтальний електрод – металева полоса з перетином 40х4 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – у ряд (рис. 8.1).

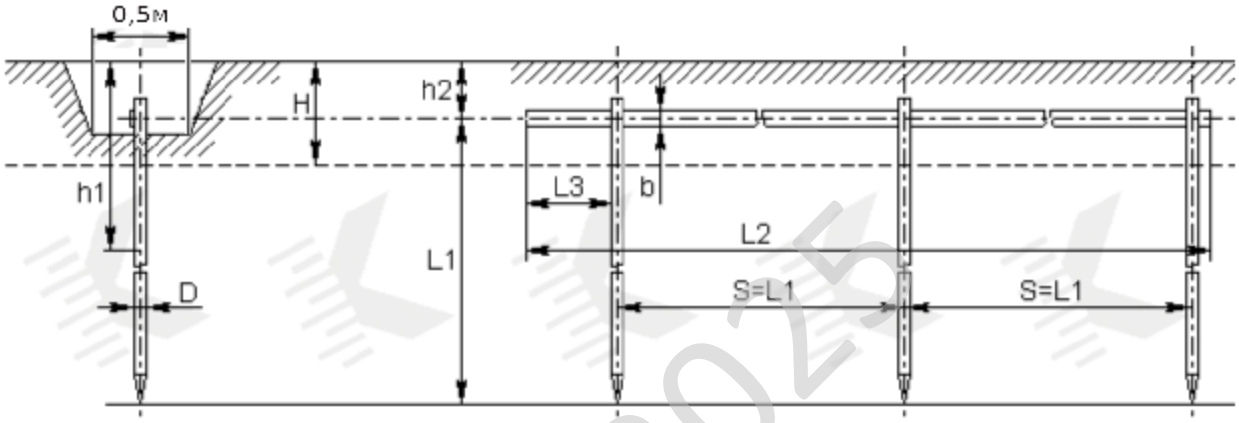

Рисунок 8.1 – Схема штучного заземлення.

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення:

– тип верхнього шару ґрунту – чорнозем, нижнього шару ґрунту – глина (питомий опір ρ_2 = 40 Ом·м);

– умовна товщина верхнього шару ґрунту: H=0.5 м;

– відстань між вертикальними заземлювачами (електродами) A=3 м;

– глибина закладення горизонтального контура заземлення t=0.8 м;

– опір заземлювача, який нормується: $R_{ЗН}$ = 4 Ом.

Необхідно визначити мінімальну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Виконаємо розрахунок захисного заземлення.

Відстань від центру вертикального заземлювача до поверхні землі:

$$T = t + L/2 = 0{,}8 + 2.5/2 = 2.05 \text{ м.}$$

Розрахунковий питомий опір ґрунту (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунту):

$$\rho = \psi \cdot \rho_2 = 1.36 \cdot 40 = 54.4 \text{ Ом·м.}$$

де $\psi = 1.36$ – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багатошаровому ґрунті [6]; $\rho = 40$ Ом·м. – табличне значення питомого опору нижнього шару ґрунту (глина) [11].

Еквівалентний діаметр вертикального електрода (кутка) [11]:

$$D_в = 0.95 \cdot К = 0.95 \cdot 45 = 0.043 \text{ м.}$$

де К= 45 мм – розмір металевого кутка (заданий).

Відношення A/L=3/2.5=1.2.

Опір розтіканню електричного струму одного електрода вертикального заземлювача з урахуванням заглиблення заземлювача [11]:

$$R_О = 0.366 \cdot (\rho/L) \cdot [\lg(2L/D_в) + (1/2)\lg((4 \cdot T + L)/(4 \cdot T - L))] =$$

$$= 0.366 \cdot (54.5/2.5) \cdot [(\lg(2 \cdot 2.5/0.0475) + (1/2) \cdot \lg((4 \cdot 2.3 + 2.5)/(4 \cdot 2.3 - 2.5))] = 17.55 \text{ Ом.}$$

Визначаємо коефіцієнт екранування вертикальних електродів Кев=0.8 при орієнтовній кількості вертикальних електродів, яка дорівнює 4 [11].

Визначаємо необхідну кількість вертикальних заземлювачів (без врахування горизонтального заземлювача), при $R_{ЗН} = 4$ Ом

$$N = R_О / (К_{ев} \cdot R_{ЗН}) = 17.55 / (0.8 \cdot 4) = 5.48 \approx 6 \text{ шт.}$$

Визначаємо довжину з’єднуючої полоси:

$$L_П = 1.05 \cdot A \cdot N = 1.05 \cdot 2.5 \cdot 5.48 = 17.28 \approx 18 \text{ м.}$$

Опір розтіканню електричного струму з’єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору ґрунту $К_П$ [11]:

$$R_П = 0.366 \cdot (\rho_2 \cdot К_П / L_П) \cdot \lg(2(L_П \cdot L_П)/(B \cdot t)) =$$

$$= 0.366 \cdot (40 \cdot 5/17.28) \cdot \lg((2 \cdot 17.28^2)/(0.04 \cdot 0.8)) = 18 \text{ Ом.}$$

де $К_П = 5$ – табличне значення кліматичного коефіцієнта питомого опору ґрунту для відповідної кліматичної зони для з’єднуючої полоси [11]:

B = 40 мм. = 0.04 м.- ширина з’єднуючої полоси (задана).

Загальний опір розтіканню електричного струму заземлювача [11]:

$$R = (R_О \cdot R_П)/(R_О \cdot \eta_П + N \cdot R_П \cdot Кев)=$$

$$=(17.55 \cdot 18)/(17.55 \cdot 0.75 + 5.48 \cdot 18 \cdot 0.8) = 3.43 \text{ Ом.}$$

де $\eta_П = 0.75$ – табличне значення коефіцієнта екранування з'єднуючої полоси [11].

Умова $R \leq R_{ЗН}$ виконується, $3.43 \leq 4$.

При необхідності можна зменшити кількість електродів заземлювача зменшивши загальний опір розтіканню електричного струму заземлювача методом зменшення питомого опору ґрунту, домішуючи у ґрунт безпосередньо навколо електродів заземлювача розчини солей NaCl, CaCl, сажу, соду, шлак, коксову дрібницю, або спеціальні суміші.

8.5 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста. Розроблено заходи з охорони праці.

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

– Досліджена система управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

– На основі отриманих результатів досліджень створена програмна реалізація системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання управління ідентифікацією та доступом до мережевих інформаційних ресурсів.

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.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Twofish.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Плужник В.О. Дослідження та програмна реалізація системи управління ідентифікацією та доступом до мережевих інформаційних ресурсів // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.

2. Awais Rashid, Howard Chivers, George Danezis, Emil Lupu, Andrew Martin. CyBOK The Cyber Security Body of Knowledge. The National Cyber Security Centre. 2019. 854 p.

3. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.

4. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.

5. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.

6. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.

7. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p

8. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.

9. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.

10. Alex Matrosov, Eugene Rodionov, Sergey Bratus. Rootkits and Bootkits. No Starch Press. 2019. 450 p.

11. Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед та ін. «Системи інформаційної зброї та технології інформаційної війни»: підручник / Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед, Жарков Я.М., Смірнов О.А, Буравченко К.О., Давидюк А.В., Кононович В.Г., Корчинский В.В., Кудирко В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.– К.: Видавничий центр “Кафедра”, 2025.– 320 с.

12. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев, О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка.* 2025. Том 1 № 29. С.704–716, 2025

13. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.

14. Kuznetsov, O., Atzeni, G., Arnesano, M., Randieri, C., Smirnov, O. «Secure IoT-based smart wheelchair system: From implementation to security enhancement strategy». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 225–257.

15. Kuznetsov, O., Smirnov, O., Kuznetsova, T., Shaikhanova, A., Svatowsky, I. «Privacy-utility trade-offs in IoT networks: A comparative analysis of differential privacy mechanisms for sensor data aggregation». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 589–622.

16. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*, 2025. vol 389. pp 377-389. Springer, Singapore.

17. Kuznetsov O., Frontoni E., Kuznetsova Y., Smirnov O., Moskovchenko I. «Trust-Based Security Architecture for Edge Computing: A Simulation Study of Dynamic Trust Evolution and Attack Detection». *CEUR Workshop Proceedings,* 2024, 3909, pp. 227–241.

18. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security

Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

19. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447.

20. Ткаченко, О., Ільєнко, А., Улічев, О., Мелешко, Є., Смірнов, О. «Правові засади поширення інформаційних впливів в соціальних мережах». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2024. № 2(26), С. 170–188.

21. Смірнова Т.В., Конопліцька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка.* 2024. №4(24), С. 6-27.

22. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка.* 2024. №3(23), С. 111-131.

23. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології,* 2024, № 13, с. 28-35.

24. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

25. Akhalaia, G., Iavich, M., Iashvili, G., Prysiazhnyy, D., Smirnova, T. «Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings,* 2023, 3550, pp. 313-320.

26. 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

27. 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.

28. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings,* Volume 3504, 2023, pp. 1-11.

29. 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.

30. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Сидоренко В.М., Жигаревич О.К., «Система корелювання подій та управління інцидентами кібербезпеки на об'єктах критичної інфраструктури». *Кібербезпека: освіта, наука, техніка*, №*3*(19), 2023, С. 176-196.

31. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *II Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи hозвитку інформаційних технологій (ІПШРІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

32. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення,* м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

33. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення,* м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

34. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

35. 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,

36. 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.

37. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління,* № 2(70). 2022. С. 28-37.

38. 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.

39. 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.

40. 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.

41. 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

42. 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.

43. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020,* Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

44. 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.*

45. 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.

46. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

47. 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.

48. 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.

49. 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.

50. 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.

51. 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.

52. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I.,Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT),* Ukraine, Kyiv, May 14-18. 2020. P. 166-171.