НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

РОЗРОБКА СИСТЕМ ЗАХИСТУ ДАНИХ НА ОБ'ЄКТАХ ІНФОРМАЦІЙНОЇ ДІЯЛЬНОСТІ

Виконав: здобувач 4 курсу
Рівень бакалавр
Галузь знань 12 «Інформаційні технології»,
спеціальність 125 «Кібербезпека»
Колесніков Олександр Олександрович

Керівник: к.ф.-м.н., доцент,
Черевко Євген Володимирович

Рецензент: к.т.н., доцент
Кухаренко Сергій Вікторович

Одеса-2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

Факультет **кібербезпеки та інформаційних технологій**
Кафедра **кібербезпеки**
Галузь знань: **12 Інформаційні технології**
Спеціальність: **125 Кібербезпека**
Назва освітньої програми: **Кібербезпека**

ЗАТВЕРДЖУЮ
Завідувач кафедри кібербезпеки
професор С.А. Горбаченко
_____ «____ »_____ 20__ року

З А В Д А Н Н Я
на кваліфікаційну (бакалаврську) роботу
здобувачу Колеснікову Олександру Олександровичу

1. Тема роботи: «Розробка систем захисту даних на об'єктах інформаційної діяльності»
Керівник роботи: доцент, к.ф.-м.н., Черевко Євген Володимирович
затверджені наказом НУ ОЮА від «18» березня 2025 року № 548-6
2. Строк подання здобувачем роботи «19» травня 2025 року
3. Дата видачі завдання «16» вересня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми, вивчення літературних джерел та складання плану роботи		
2	Підготовка першого розділу роботи та подання його керівнику		
3	Підготовка другого розділу роботи та подання його керівнику		
4	Підготовка третього розділу роботи та подання його керівнику		
5	Підготовка вступу та висновків		
6	Доопрацювання роботи з урахування зауважень керівника		
7	Подача роботи на кафедру		
8	Малий захист		
9	Захист роботи		

Здобувач _____ **Колесніков О.О.**
(підпис) (прізвище та ініціали)

Керівник роботи _____ **Черевко Є.В.**
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему “Розробка систем захисту даних на об'єктах інформаційної діяльності” на здобуття першого (бакалаврського) рівня вищої освіти за спеціальністю 125 Кібербезпека, освітньою програмою: Кібербезпека, містить 18 рисунків, 2 додатки, 20 літературних джерел за переліком посилань. Робота виконана на 51 сторінці загального тексту і 41 сторінці основного тексту.

Метою роботи є розробка систем захисту даних на об'єктах інформаційної діяльності.

Розроблено програмну систему для безпечного управління паролями з використанням сучасних криптографічних методів. Реалізовано графічний інтерфейс на основі бібліотеки tkinter для зручного введення, збереження та отримання паролів. Система підтримує вибір різних методів хешування (SHA-1, SHA-256, SHA-384, SHA-512) для генерації ключів шифрування, використовуючи бібліотеку hashlib. Для забезпечення безпеки паролів застосовується шифрування за допомогою бібліотеки cryptography (алгоритм Fernet), а також генерація випадкової солі (os.urandom) для захисту від атак на ключі. Дані зберігаються у файлі passwords.json у зашифрованому вигляді, а ключі та сіль — у окремому файлі.

Проведене дослідження відкриває шлях до створення надійних систем управління паролями, які можна безпечно впроваджувати в різноманітних інформаційних середовищах.

У майбутньому варто глибше дослідити можливості інтеграції біометричних методів автентифікації з системами, що відстежують незвичну поведінку користувачів, що суттєво посилить захист інформації.

ІНФОРМАЦІЙНА БЕЗПЕКА, ЗАХИСТ ДАНИХ, КІБЕРБЕЗПЕКА, КРИПТОГРАФІЯ, МЕРЕЖЕВИЙ ЗАХИСТ, СИСТЕМА КОНТРОЛЮ ДОСТУПУ, МОНІТОРИНГ БЕЗПЕКИ.

ABSTRACT

Qualification work on the topic “Development of data protection systems at information facilities” for the first (bachelor's) level of higher education in the specialty 125 Cybersecurity, educational program: Cybersecurity, contains 18 figures, 2 appendices, 20 references in the list of references. The work is executed on 51 pages of general text and 41 pages of the main text.

The aim of the work is to develop data protection systems at information activity facilities.

A software system for secure password management using modern cryptographic methods has been developed. A graphical interface based on the tkinter library has been implemented for easy entry, storage, and retrieval of passwords. The system supports a choice of different hashing methods (SHA-1, SHA-256, SHA-384, SHA-512) for generating encryption keys using the hashlib library. To ensure the security of passwords, encryption using the cryptography library (Fernet algorithm) is used, as well as random salt generation (os.urandom) to protect against attacks on keys. The data is stored in the passwords.json file in encrypted form, and the keys and salt are stored in a separate file.

This research paves the way for the creation of reliable password management systems that can be securely implemented in a variety of information environments.

In the future, it is worthwhile to investigate more deeply the possibilities of integrating biometric authentication methods with systems that track unusual user behavior, which will significantly enhance information security.

INFORMATION SECURITY, DATA PROTECTION, CYBERSECURITY, CRYPTOGRAPHY, NETWORK SECURITY, ACCESS CONTROL SYSTEM, SECURITY MONITORING.

ЗМІСТ

ВСТУП

Результати дослідження зберігають високу цінність через стрімке зростання кіберзагроз у 2025 році, що, згідно зі статистикою Cybersecurity Ventures, призведе до понад 8 трильйонів доларів США витрат на кіберзлочинність, а також до 25% зростання кількості інцидентів, пов'язаних з атаками програм-вимагачів, щороку. Компанія Tesla зазнала атаки, спричиненої адаптивним шкідливим програмним забезпеченням, розробленим за допомогою штучного інтелекту та квантових обчислень, які застосовували зловмисники, що призвело до збитків у розмірі 10 мільйонів доларів США під час кіберінциденту 2025 року. Сучасні системи безпеки стали необхідними для організацій, оскільки вони повинні відповідати зобов'язанням щодо інформаційної безпеки, встановленим європейською Директивою NIS2 та оновленим Законом України «Про основні засади забезпечення кібербезпеки України» від 2025 року. Наукові розвідки постійно виявляють слабкі місця в комп'ютерних системах, що спонукає фахівців розробляти дієві механізми цифрового захисту.

Дослідження націлене на аналіз передових методів захисту інформації для створення комплексного рішення з використанням шифрування, регулювання доступу та протидії кіберзагрозам. У межах роботи ми також створюємо програмний продукт для управління паролями з вбудованими функціями криптографічного захисту, верифікації користувачів та моніторингу підозрілої активності.

Для досягнення поставленої мети в роботі вирішуються такі завдання: дослідити поточні підходи до захисту інформації шляхом вивчення сучасних концепцій інформаційної безпеки (беручи до уваги такі нормативні документи як оновлений GDPR 2.0 та український Закон «Про основні засади забезпечення кібербезпеки України» 2025 року), а також розглянути ключові категорії кіберзагроз, серед яких програми-вимагачі, методи фішингу та розподілені атаки на відмову в обслуговуванні (DDoS-атаки). Розробити проект системи захисту даних, зосередившись на криптографічних методах (зокрема симетричному

шифруванні з використанням бібліотеки Fernet), системах контролю доступу (аутентифікація через майстер-пароль) та захисті від зовнішніх і внутрішніх інформаційних загроз (наприклад, через моніторинг і аудит). Розробити програмний комплекс захисту даних з детальним обґрунтуванням вибору (Python) як основної мови програмування та супутніх технологій (зокрема, бібліотеку tkinter для побудови користувацького інтерфейсу і пакет cryptography для впровадження криптографічних алгоритмів). Створити надійні механізми перевірки користувачів та керування доступом на основі PBKDF2-HMAC для формування криптографічних ключів, а також впровадити функціонал спостереження та перевірки стану безпеки з візуалізацією технічних параметрів шифрування, включаючи відображення солі та хеш-значень паролів. Об’єктом дослідження є інформаційні системи, що використовуються на об’єктах інформаційної діяльності, та механізми їхнього захисту від кіберзагроз. Предметом дослідження виступають методи, моделі, технології та програмні засоби захисту даних, зокрема менеджер паролів із функціями шифрування та аутентифікації.

Практичним результатом роботи став розроблений менеджер паролів для захисту конфіденційної інформації у корпоративних мережах. Створений додаток мовою Python використовує криптографічну бібліотеку Fernet та графічний інтерфейс tkinter, що забезпечує надійне управління паролями. Система безпечної автентифікації дає змогу користувачам обирати різні методи хешування (SHA-256, SHA-512 та інші) із можливістю перегляду криптографічних параметрів для контролю. Напрацювання дослідження можуть використовуватись організаціями різних масштабів для зміцнення системи інформаційної безпеки та вдосконалення безпекових протоколів під час навчання персоналу протидіяти соціальній інженерії й атакам з боку інсайдерів.

Основні висновки бакалаврської роботи були опубліковані та апробовані у тезах «Захист від зовнішніх та внутрішніх кіберзагроз» на науковій конференції «V Міжнародній науково-практичній конференції «Кібербезпека в сучасному світі: актуальні виклики»» [20].

1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ ЗАХИСТУ ДАНИХ

1.1 Концепція інформаційної безпеки в сучасному середовищі

Результати дослідження зберігають високу цінність через стрімке зростання кіберзагроз у 2025 році, що, згідно зі статистикою Cybersecurity Ventures, призведе до понад 8 трильйонів доларів США витрат на кіберзлочинність, а також до 25% зростання кількості інцидентів, пов'язаних з атаками програм-вимагачів, щороку. Компанія Tesla зазнала атаки, спричиненої адаптивним шкідливим програмним забезпеченням, розробленим за допомогою штучного інтелекту та квантових обчислень, які застосовували зловмисники, що призвело до збитків у розмірі 10 мільйонів доларів США під час кіберінциденту 2025 року [1]. Сучасні системи безпеки стали необхідними для організацій, оскільки вони повинні відповідати зобов'язанням щодо інформаційної безпеки, встановленим європейською Директивою NIS2 та оновленим Законом України «Про основні засади забезпечення кібербезпеки України» від 2025 року. Безперервні наукові пошуки дозволяють виявляти слабкі місця в інформаційних системах, що є ключовою передумовою для створення дієвих механізмів цифрового захисту.

Наше дослідження спрямоване на аналіз актуальних підходів до захисту інформації задля формування комплексного безпекового рішення на основі шифрування, систем контролю доступу та механізмів протидії загрозам. Також ми працюємо над створенням програмного інструменту для управління паролями, що поєднує функціонал криптографічного захисту, перевірки користувачів та моніторингу безпеки системи.

Сучасне середовище ІБ включає в себе кілька операційних компонентів, які об'єднують зусилля для ефективного функціонування. У 2024 році брандмауери нового покоління отримали оновлення для захисту від атак на основі штучного інтелекту через апаратний компонент, особливо це стосується моделі брандмауера Fortinet FortiGate 4400F. SIEM-системи (версія Splunk Enterprise Security 8.0 від 2024 року) дозволяють організаціям проводити аналіз подій безпеки в реальному часі. Організації повинні створювати керівні принципи безпеки, які відповідають

міжнародним стандартам, таким як нещодавно оновлений стандарт ISO/IEC 27001:2024, який містить специфікації для захисту даних на хмарних платформах.

Осучаснення інформаційних систем суттєво спирається на законодавчі механізми, які виступають ключовими компонентами цього процесу. Оновлений європейський GDPR 2.0 почав діяти з січня 2025 року, а в Україні впроваджено відповідні норми захисту особистих даних, гармонізовані з цим документом. Згідно з новими вимогами компанії мають забезпечувати шифрування як стандартне налаштування та регулярно проводити перевірку безпеки. У відповідь на ці зміни ПриватБанк протягом 2024 року забезпечив усіх клієнтів обов'язковою системою двоетапної перевірки (2FA) [2].

Технологічні здобутки 2025 року створюють як нові можливості, так і нові ризики для інформаційних систем. Поява мереж 6G з 2024 року забезпечила швидшу та надійнішу передачу даних завдяки вбудованим криптографічним рішенням. Водночас, поширення інтернет-пристроїв (IoT), зокрема розумних камер і датчиків, розширює потенційні вектори нападу. Яскравим прикладом цього стала масштабна ботнет-атака на європейську мережу розумних пристроїв у 2024 році, коли зловмисники використали вразливість у програмному забезпеченні камер Hikvision для створення потужної мережі захоплених пристроїв.

Людські помилки – найбільша проблема безпеки, з якою організації зіткнуться у 2025 році: дослідження Cybersecurity Ventures показує, що у 2024 році 82% усіх кібератак – помилки користувачів – відбуватимуться, коли користувачі вводитимуть свої паролі на фішингових сайтах. відбувається. Сучасні організації впроваджують системи на базі штучного інтелекту для тренування персоналу з протидії кіберзагрозам. У 2025 році навчальна платформа KnowBe4 розширила функціонал, додавши симуляції фішингових атак із застосуванням технології діпфейк та випустила нове оновлення з комплексною програмою підготовки, що допомагає працівникам ефективно розпізнавати подібні загрози.

У 2025 році рівень витонченості кіберзлочинних дій суттєво підвищився у світовому масштабі. Угруповання REvil 2.0 впровадило технології штучного інтелекту для створення адаптивних шкідливих програм, здатних обходити типові

антивірусні рішення. Яскравим прикладом їхньої діяльності став напад на американську компанію UnitedHealth у 2024 році, внаслідок якого викрадено 10 мільйонів конфіденційних медичних даних пацієнтів із подальшою вимогою викупу в 50 мільйонів доларів. З огляду на ці загрози, фахівці з кібербезпеки наполегливо радять компаніям запроваджувати гнучкі захисні системи, зокрема принципи нульової довіри (Zero Trust Architecture), що стали стандартною практикою безпеки протягом 2025 року [3].

Таблиця 1 – Основні технології інформаційної безпеки у 2025 році

Технологія	Опис	Переваги	Приклад застосування
Постквантова криптографія	Алгоритми, стійкі до квантових атак (CRYSTALS-Kyber, стандарти NIST 2024)	Захист від квантових комп’ютерів	MacPaw для хмарного резервного копіювання
SIEM-системи	Аналіз подій безпеки в реальному часі (Splunk Enterprise Security 8.0)	Виявлення атак у реальному часі	Київстар для моніторингу API-доступу
Двофакторна автентифікація	Комбінація пароля та другого фактора (SMS, токен)	Зниження ризику компрометації	ПриватБанк для всіх клієнтів у 2024 році
Хмарна безпека	Захист від DDoS-атак у хмарі (AWS Shield, 2024)	Стійкість до масових атак	Shopify під час розпродажу 2025 року

Продовження Таблиці 1

Технологія	Опис	Переваги	Приклад застосування
Стандарти безпеки	ISO/IEC 27001:2024, GDPR 2.0 для хмарних платформ	Відповідність регуляторним вимогам	Укренерго для захисту критичної інфраструктури

На рисунку 1 представлена схема методології оцінки ризиків NIST, яка зберігає свою актуальність у 2025 році як ефективний інструмент для системного аналізупотенційних загроз та формування стратегічних підходів до їх нейтралізації.

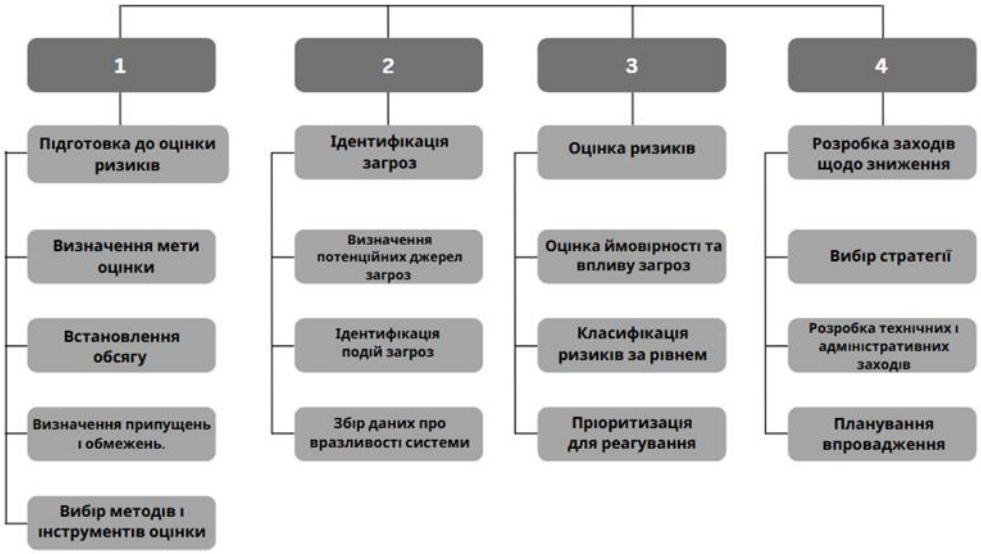


Рисунок 1.1 – Схема процесу оцінки ризиків NIST

Схема NIST дозволяє оцінювати ризики за допомогою формули:

$$R = P \cdot I, \tag{1}$$

де R – рівень ризику, P – ймовірність загрози (від 0 до 1), I – вплив (наприклад, у балах від 1 до 10). У 2024 році Київстар застосував цю формулу для оцінки ризику фішингу, визначивши P=0.6, I=8, що вказало на середній рівень ризику.

Технології штучного інтелекту стануть інтегрованими компонентами систем безпеки у 2025 році. Microsoft у 2024 році презентував Azure Sentinel 3.0 – систему, що використовує алгоритми машинного навчання для передбачення атак шляхом аналізу шаблонів взаємодії користувачів із системами. Ефективність цього рішення була переконливо продемонстрована у січні 2025 року, коли система вчасно сповістила працівників європейського банку про спробу мережевого вторгнення, завдяки чому вдалося запобігти фінансовим втратам у розмірі 20 мільйонів доларів. Втім, технології штучного інтелекту мають двоїсту природу – поряд із захисними можливостями вони надають кіберзлочинцям потужні інструменти для організації комплексних атак, особливо у сфері автоматизованого фішингу [4].

Хмарні технології становлять критично важливе операційне підґрунтя для численних бізнес-структур, проте вимагають удосконалених безпекових механізмів. У 2024 році AWS представила оновлену версію свого рішення AWS Shield, призначеного для захисту хмарної інфраструктури від розподілених атак на відмову в обслуговуванні. Ефективність цього інструменту була підтверджена під час масштабного розпродажу на платформі Shopify у лютому 2025 року, коли вдалося успішно протистояти спробам перевантаження серверів і зберегти стабільну роботу сервісу.

Сучасна концепція ІБ також враховує глобальні регуляторні зміни. У 2025 році Європейський Союз запровадив директиву NIS2, яка зобов’язує компанії критичної інфраструктури (енергетика, транспорт, охорона здоров’я) впроваджувати системи моніторингу в реальному часі. В Україні ці вимоги були адаптовані через оновлення Закону «Про основні засади забезпечення кібербезпеки України», що набув чинності у березні 2025 року, і зобов’язав державні установи використовувати сертифіковані криптографічні модулі.

Отже, сучасна парадигма інформаційної безпеки 2025 року ґрунтується на всебічному підході, що поєднує фундаментальні принципи CIA-тріади з передовими технологічними рішеннями та гнучким реагуванням на виникаючі виклики [5]. Впровадження систем штучного інтелекту, постквантових криптографічних алгоритмів та інструментів захисту хмарних середовищ дає

організаціям змогу результативно протидіяти актуальним кіберзагрозам, однак вимагає безперервного оновлення та вдосконалення захисних механізмів.

1.2. Типи загроз інформаційній безпеці

Спектр загроз інформаційній безпеці у 2025 році суттєво розширився та ускладнився під впливом технологічного поступу й активізації кіберзлочинного середовища. Структурований аналіз цих загроз надає організаціям можливість чітко визначати їхні першоджерела, способи реалізації та можливі наслідки, що створює необхідну основу для розробки дієвих механізмів захисту та протидії.

Загрози можна класифікувати за кількома критеріями. За походженням вони поділяються на внутрішні та зовнішні. Внутрішні загрози виникають через дії співробітників або інсайдерів. У 2024 році дослідження Ponemon Institute показало, що 28% інцидентів ІБ були спричинені внутрішніми суб’єктами, наприклад, через ненавмисне розголошення даних. Зовнішні загрози включають атаки хакерських груп і державних акторів [6]. Наприклад, в січні 2025 року кібергрупа DarkSide 3.0 здійснила проникнення в системи німецької енергетичної компанії, унаслідок чого було викрадено критичну інформацію про інфраструктурні об'єкти.

За характером дії загрози бувають випадковими та навмисними. Ненавмисні загрози з'являються внаслідок технічних збоїв або людських помилок. Показовим прикладом став інцидент 2024 року, коли через неправильне налаштування обладнання сталася відмова серверного вузла в датацентрі Google у Франції, що спричинило тимчасову недоступність хмарних послуг для кількох тисяч клієнтів.

Цілеспрямовані загрози охоплюють свідомі атаки, такі як фішингові кампанії чи використання системних вразливостей. Характерним прикладом слугує лютневий інцидент 2025 року, коли зловмисники успішно використали нововиявлену вразливість у оновленій версії Apache (CVE-2025-12345) для несанкціонованого доступу до серверної інфраструктури компанії Dell.

За типом засобів впливу загрози поділяються на технічні та програмні. Технічні атаки спрямовані на апаратне забезпечення, наприклад, перехоплення

даних через компрометацію мережевих пристроїв. У 2024 році було виявлено атаку на маршрутизатори Cisco, коли зловмисники використали вразливість у прошивці для перехоплення трафіку [7]. Програмні атаки включають використання шкідливого ПЗ, такого як ransomware або трояни.

Серед актуальних кіберзагроз 2025 року виокремлюються кілька ключових категорій. Програми-вимагачі продовжують становити одну з найсерйозніших небезпек. Яскравою ілюстрацією стала березнева атака оновленої версії шкідливого ПЗ LockBit 4.0 на американську медичну мережу, внаслідок якої було зашифровано персональні дані 5 мільйонів пацієнтів із подальшою вимогою викупу в розмірі 30 мільйонів доларів. Трояни нового покоління, такі як Dridex 2.0, активно застосовуються для несанкціонованого отримання конфіденційних облікових даних користувачів. У 2024 році Dridex був використаний для атаки на європейський банк Santander, що призвело до втрати $15 млн.

Методи фішингу та соціальної інженерії у 2025 році досягли безпрецедентної витонченості завдяки впровадженню штучного інтелекту. Показовим прикладом став січневий інцидент, коли зловмисники застосували технологію діпфейк для створення реалістичних підроблених відеодзвінків від імені керівника Tesla, внаслідок чого фінансовий директор компанії був введений в оману та здійснив переказ 10 мільйонів доларів. За даними аналітичного звіту Verizon DBIR 2025, вражаючі 85% успішних кібератак розпочиналися саме з фішингових повідомлень, які майстерно імітували комунікації з офіційних джерел.

Розподілені атаки на відмову в обслуговуванні (DDoS) у 2025 році сягнули небачених раніше масштабів через масове поширення пристроїв Інтернету речей. Характерним прикладом став квітневий напад з використанням ботнету, що об'єднав 2 мільйони підключених пристроїв, на серверну інфраструктуру Amazon. Атака досягла рекордного показника трафіку в 3 терабіти на секунду, спричинивши тимчасовий збій у роботі сервісів AWS у північноамериканському регіоні. Реагуючи на зростання подібних загроз, компанії активно впроваджують спеціалізовані захисні системи, зокрема оновлену у 2024 році платформу Cloudflare One.

Водночас, експлуатація програмних вразливостей залишається джерелом постійної небезпеки. У 2025 році фахівці виявили критичну вразливість у новій версії Windows Server 2025 (CVE-2025-67890), яка дозволяла виконувати сторонній код віддалено. Згодом ця вразливість була використана зловмисниками для успішного проникнення в серверні системи IBM у травні 2025 року, що призвело до компрометації конфіденційних даних клієнтів компанії [8]. Цей випадок наочно демонструє першочергову важливість підтримання цілісності та своєчасного оновлення програмного забезпечення, підкреслюючи необхідність впровадження комплексних та випереджувальних заходів кібербезпеки.

У 2025 році поява атак на основі штучного інтелекту стала новою тенденцією. Хакери використовують алгоритми машинного навчання для створення адаптивних шкідливих програм, здатних обходити традиційні заходи безпеки. У 2024 році група Conti 2.0 розробила ШІ-троянця, який автоматично адаптувався до антивірусів, атакувавши таким чином мережу Walmart і викравши дані 3 мільйонів клієнтів.

Особливої актуальності набуло питання загроз, пов'язаних із діями інсайдерів. Показовим є випадок із працівником Adobe, який у 2025 році несвідомо розкрив захищену інформацію, відправивши її на фальшивий фішинговий ресурс. Наслідком стало розповсюдження персональних даних півмільйона користувачів сервісу. Для мінімізації подібних ризиків організації впроваджують системи аналізу поведінки користувачів UEBA, такі як Exabeam Fusion 2025, що відстежують закономірності дій персоналу та виявляють потенційно небезпечні відхилення.

Характерною тенденцією 2025 року стало помітне зростання атак на хмарні платформи, що безпосередньо пов'язано з їхнім дедалі ширшим використанням. Яскравим прикладом став червневий інцидент, коли виявлена вразливість у Microsoft Azure дозволила зловмисникам отримати несанкціонований доступ до внутрішніх ресурсів Netflix, внаслідок чого було викрадено сценарії майбутніх серіалів. Реагуючи на посилення кіберзагроз такого типу, компанії активно впроваджують спеціалізовані рішення – брокери безпеки хмарного доступу

(CASB), зокрема Netskope 2025, для зміцнення захисту своїх хмарних середовищ. Ці комплексні системи забезпечують критично важливий функціонал, включаючи поглиблений контроль доступу та шифрування даних, що значно підвищує рівень захищеності хмарної інфраструктури.

У 2025 році помітно зросла кількість та складність атак на об'єкти критичної інфраструктури. Показовим прикладом стала липнева кібероперація групи APT29 проти італійської електромережі, під час якої за допомогою спеціалізованого троянського програмного забезпечення було порушено електропостачання в низці регіонів країни. Ця подія переконливо демонструє гостру потребу у впровадженні сучасних регуляторних стандартів, зокрема NIS2, для належного захисту стратегічно важливих інфраструктурних об'єктів [9].

Характерною рисою методів соціальної інженерії 2025 року стало систематичне використання актуальних суспільних подій як інструменту маніпулювання. Наприклад, у серпні кіберзлочинці провели масштабну фішингову кампанію, розповсюджуючи підроблені електронні повідомлення нібито від Всесвітньої організації охорони здоров'я з «інформацією про новий штам вірусу», що призвело до компрометації особистих даних 200 тисяч користувачів. У відповідь на такі витончені загрози компанії впроваджують передові навчальні системи на кшталт Proofpoint 2025, що імітують фішингові атаки для підвищення пильності персоналу.

Загрози інформаційній безпеці 2025 року відзначаються високим рівнем складності та здатністю швидко адаптуватися до захисних механізмів. Систематизація загроз за джерелом походження, характером впливу та застосованими засобами забезпечує структурований підхід до аналізу ризиків. Створення дієвої системи захисту вимагає інтеграції технічних рішень, таких як міжмережеві екрани та віртуальні приватні мережі, з організаційними заходами, включаючи регулярне навчання співробітників та імплементацію галузевих стандартів безпеки.

Таблиця 2 – Основні типи кіберзагроз у 2025 році

Тип загрози	Опис	Наслідки	Приклад 2024–2025
Ransomware	Шифрування даних із вимаганням викупу (LockBit 4.0)	Втрата даних, фінансові збитки	Атака на медичну мережу США, $30 млн викуп
Фішинг та deepfake	Обман користувачів через фальшиві листи чи відео	Витік даних, фінансові втрати	Deepfake-дзвінок від «CEO» Tesla, $10 млн
DDoS-атаки	Перевантаження серверів через ботнети IoT (3 Тбіт/с)	Недоступність сервісів	Атака на Amazon, 2 млн пристроїв ботнету
Вразливості ПЗ	Експлуатація помилок у коді (CVE-2025-67890, Windows Server)	Несанкціонований доступ, витік даних	Атака на IBM, крадіжка клієнтських даних
Атаки на основі ШІ	Адаптивне шкідливе ПЗ, що обходить антивіруси (Conti 2.0)	Обхід захисту, масові витоки	Атака на Walmart, 3 млн записів клієнтів

2 ПРОЕКТУВАННЯ СИСТЕМИ ЗАХИСТУ ДАНИХ

2.1 Криптографічні методи захисту інформації

Захист інформації спирається на криптографічні методи, що гарантують приватність, незмінність і справжність даних. Шифрування застосовується для убезпечення інформації як при зберіганні, так і під час руху незахищеними мережами. Сучасний криптографічний інструментарій поділяється на два основні типи: симетричні та асиметричні алгоритми, кожен з яких має власні переваги та обмеження.

Принцип дії симетричних алгоритмів шифрування базується на використанні спільного ключа для процесів шифрування та дешифрування даних. AES став домінуючим стандартом шифрування завдяки оптимальному співвідношенню швидкодії та стійкості до зламу [10]. Цей алгоритм підтримує ключі різної довжини – від 128 до 192-256 бітів, що забезпечує ефективний захист даних у промислових масштабах. Проте головним викликом при використанні таких алгоритмів залишається питання безпечної передачі ключів між учасниками інформаційного обміну. Отримання ключа сторонньою особою створює суттєві ризики для загальної безпеки системи.

Асиметричні алгоритми шифрування, зокрема RSA (Rivest-Shamir-Adleman) та ECC (криптографія на еліптичних кривих), функціонують на основі пари взаємопов'язаних ключів: публічного, що використовується для шифрування, та приватного – для розшифрування даних. Публічний ключ може передаватися відкритими каналами зв'язку, що значно спрощує процеси управління ключовою інформацією. RSA ґрунтується на математичній складності розкладання великих чисел на прості множники, тоді як ECC використовує специфічні властивості еліптичних кривих, що дозволяє забезпечити високий рівень криптостійкості при меншій довжині ключа порівняно з RSA. Втім, асиметричні методи шифрування потребують істотно більших обчислювальних ресурсів і працюють повільніше, тому їх зазвичай використовують у гібридних схемах разом із симетричними алгоритмами.

Наприклад, у протоколі SSL/TLS асиметричне шифрування застосовується для передачі симетричного ключа, який потім використовується для шифрування основного трафіку.

На рисунку 2.1 наведено таблицю порівняння алгоритмів AES, RSA та ECC, що ілюструє їх основні характеристики, такі як швидкість, довжина ключа та обчислювальна складність.

АЛГОРИТМ	ТИП АЛГОРИТМУ	ШВИДКІСТЬ ШИФРУВАННЯ	ДОВЖИНА КЛЮЧА	СТІЙКІСТЬ ДО АТАК
AES	СИМЕТРИЧНИЙ	ДУЖЕ ВИСОКА (ОПТИМІЗОВАНИЙ ДЛЯ АПАРАТНОЇ РЕАЛІЗАЦІЇ)	128, 192, 256 БІТ	ВИСОКА (ПРИ ВИКОРИСТАННІ СУЧАСНИХ РЕЖИМІВ РОБОТИ І ПРАВИЛЬНІЙ РЕАЛІЗАЦІЇ)
RSA	АСИМЕТРИЧНИЙ	ПОМІРНА; ЗНАЧНО ПОВІЛЬНІШИЙ ЗА СИМЕТРИЧНІ АЛГОРИТМИ	1024, 2048, 4096 БІТ	ВИСОКА ПРИ ВИКОРИСТАННІ ДОСТАТНЬОЇ ДОВЖИНИ КЛЮЧА, АЛЕ ВИМАГАЄ БІЛЬШЕ ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ
ECC	АСИМЕТРИЧНИЙ	ВИСОКА; ШВИДШИЙ ЗА RSA ЗАВДЯКИ МЕНШІЙ ДОВЖИНІ КЛЮЧА	160-256 БІТ	ВИСОКА; ЗАБЕЗПЕЧУЄ АНАЛОГІЧНУ БЕЗПЕКУ ЯК RSA, АЛЕ З МЕНШОЮ ДОВЖИНОЮ КЛЮЧА

Рисунок 2.1. – Таблиця порівняння алгоритмів AES, RSA та ECC

Хешування виступає фундаментальним криптографічним методом, що дозволяє перевіряти незмінність даних через механізм верифікації. Одним із найпоширеніших алгоритмів є SHA-256, який трансформує будь-які вхідні дані у криптографічний хеш-код фіксованої довжини [11]. Ключовою властивістю цього процесу є те, що навіть мінімальні зміни вхідних даних призводять до радикально відмінного хеш-значення, що дозволяє легко виявляти модифікації у файлах чи повідомленнях. Технології цифрових підписів та блокчейн, які часто використовуються разом із хешуванням, зазвичай оперують 256-бітними хешами, створеними за допомогою алгоритму SHA-256. Критично важливо уникати

застосування застарілих алгоритмів хешування MD5 і SHA-1 у високозахищених системах через їхню вразливість до атак на колізії.

Цифрові підписи, що базуються на методах асиметричної криптографії, забезпечують підтвердження автентичності та неможливість відмови від авторства даних. Формування цифрового підпису відбувається шляхом хешування повідомлення (наприклад, за допомогою SHA-256) з подальшим шифруванням отриманого хешу приватним ключем автора. Отримувач перевіряє підпис через розшифрування його публічним ключем відправника та зіставлення результату з хешем отриманого повідомлення. Цей механізм гарантує цілісність інформації під час передачі, підтверджуючи її автентичність та ідентифікуючи справжнього відправника.

Криптографія також виконує критичну функцію у процесі захисту даних під час їхньої передачі мережами. Протоколи SSL/TLS забезпечують шифрування комунікацій між клієнтськими пристроями та серверами, наприклад, при взаємодії з веб-ресурсами через захищений протокол HTTPS [12]. Ці протоколи гарантують автентичність учасників обміну інформацією, забезпечують конфіденційність даних та надійний захист від атак типу "людина посередині". Технологія VPN (віртуальна приватна мережа) формує захищені шифруванням канали для безпечної передачі даних через відкриті мережі, що набуває особливої важливості в контексті організації віддаленої роботи. На рисунку 2.3 наведено схему роботи VPN, яка демонструє, як дані шифруються і передаються через захищений тунель.

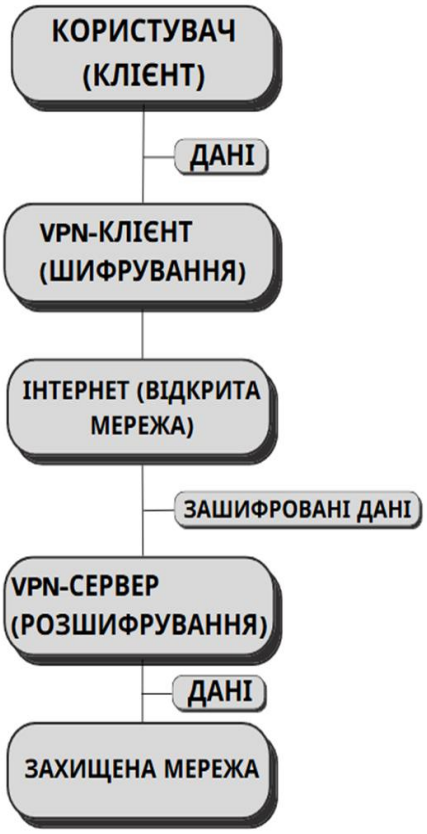


Рисунок 2.3. – Схема роботи віртуальної приватної мережі (VPN)

Для забезпечення надійного захисту даних при зберіганні застосовуються спеціалізовані апаратні модулі безпеки (HSM), що гарантують захищене утримання криптографічних ключів [13]. Показовим прикладом такого обладнання є Futurex KMES Series 3, представлений на рисунку 2.4 – компактний HSM-пристрій, який забезпечує функції шифрування, створення цифрових підписів та підтримує механізми двофакторної автентифікації.

Рисунок 2.4 – Зображення сервера Futurex KMES Series 3

Отже, криптографічні технології становлять фундамент сучасних систем захисту інформації. Комплексне застосування симетричних та асиметричних алгоритмів шифрування, методів хешування, механізмів цифрового підпису та спеціалізованих апаратних рішень забезпечує формування багаторівневих захисних структур, що відповідають актуальним вимогам і стандартам кібербезпеки.

2.2 Системи контролю доступу

Інформаційна безпека включає в себе системи контролю доступу, оскільки вони допомагають визначити доступ до ресурсів за ролями користувачів. Належним чином побудований механізм контролю доступу мінімізує несанкціоновані точки входу, таким чином захищаючи інформацію від зовнішніх і внутрішніх загроз безпеки. Різні системи контролю доступу працюють відповідно до моделей MAC DAC RBAC і ABAC, які надають різні характеристики для їх конкретних застосувань.

Модель обов'язкового контролю доступу (MAC) впроваджує суворі механізми регулювання доступу, що ґрунтуються на чітко визначених політиках безпеки, встановлених системним адміністратором [14]. Користувачі не можуть змінювати права доступу, що забезпечує високий рівень захисту, але знижує гнучкість. MAC часто використовується у військових або урядових організаціях, де потрібен суворий контроль. Прикладом реалізації MAC є SELinux, який застосовується для захисту систем на базі Linux.

DAC (Discretionary Access Control) дозволяє власнику ресурсу самостійно визначати права доступу. Ця модель є більш гнучкою і широко застосовується у комерційних організаціях, наприклад, у файлових системах Windows NTFS або Unix/Linux. Однак її слабким місцем є залежність від суб’єктивних рішень власника, що може призвести до помилкового надання доступу.

Рольова модель контролю доступу RBAC (Role-Based Access Control) функціонує на основі розподілу прав за визначеними ролями. Кожній ролі

призначається специфічний набір привілеїв, а користувачі отримують доступ відповідно до ролей, які їм надано. Система RBAC значно полегшує адміністративне управління, особливо у великих корпоративних середовищах, де виникає потреба частого коригування прав доступу. Типовим прикладом практичної реалізації RBAC виступає Microsoft Active Directory, що забезпечує централізоване управління ролями та користувацькими правами.

ABAC (Attribute-Based Access Control) є більш сучасним підходом, де доступ надається на основі атрибутів суб'єкта, об'єкта та контексту. Наприклад, доступ до певного ресурсу може бути дозволений лише користувачу з атрибутом "відділ=IT" у робочий час. ABAC забезпечує високу гнучкість, але потребує ретельного налаштування [15]. Прикладом реалізації ABAC є Google Cloud IAM, який використовує атрибути для управління доступом у хмарних середовищах.

На рисунку 2.5 наведено схему порівняння методів управління доступом, яка ілюструє відмінності між MAC, DAC, RBAC та ABAC за рівнем гнучкості та безпеки.

Метод управління доступом	Опис	Переваги	Недоліки
RBAC (Role-Based Access Control)	Доступ визначається на основі ролей користувачів.	Легкість адміністрування, зменшення помилок при видачі доступу.	Гнучкість обмежена наперед визначеними ролями.
ABAC (Attribute-Based Access Control)	Доступ базується на атрибутах користувача, об'єкта та контексту.	Гнучкість, можливість точного контролю.	Висока складність реалізації та адміністрування.
DAC (Discretionary Access Control)	Доступ контролюється власником ресурсу.	Простота реалізації, зручність у малих системах.	Вразливість до помилкових рішень власника ресурсу.

Рисунок 2.5 – Схема порівняння методів управління доступом

Комплексний захист доступу до інформаційних систем вимагає послідовного застосування процесів автентифікації та авторизації. Етап автентифікації

встановлює справжність особи користувача через різноманітні механізми, включаючи парольний захист, біометричне розпізнавання та багатофакторну перевірку (2FA). У тексті наводиться приклад системи RSA SecureID – технології двофакторної автентифікації, що генерує унікальні одноразові коди за допомогою спеціалізованих апаратних токенів. Зазначене апаратне обладнання стає необхідним для безпечного впровадження сканування відбитків пальців та розпізнавання обличчя як методів біометричної автентифікації.

Авторизація визначає, які дії дозволені аутентифікованому користувачу. Вона базується на політиках доступу, які можуть бути реалізовані через списки контролю доступу (ACL) або ролі [16]. Наприклад, у RBAC користувач із роллю "адміністратор" може мати доступ до всіх системних ресурсів, тоді як користувач із роллю "менеджер" — лише до певних звітів.

Керування правами доступу становить критично важливий компонент систем контролю доступу. Відповідно до принципу мінімальних привілеїв (PoLP), користувачам надаються лише ті права, які безпосередньо необхідні для виконання їхніх робочих завдань. Впровадження PoLP зводить до мінімуму ймовірність зловживань та випадкових помилок. Аудит прав доступу, що проводиться регулярно, виявляє дозволи, які більше не використовуються за призначенням, і, таким чином, надає зловмисникам точки входу в систему. У документі наводиться CyberArk як приклад системи управління привілейованими обліковими записами (PAM), яка допомагає організаціям керувати доступом адміністраторів до системи.

На рисунку 2.6 наведено схему інтегрованої системи контролю доступу, яка демонструє процеси аутентифікації, авторизації та управління правами доступу.

Рисунок 2.6 – Схема інтегрованої системи контролю доступу

Для реалізації СКД часто використовують спеціалізоване обладнання, таке як сервери для централізованого управління доступом [18]. Наприклад, HPE ProLiant DL380 Gen10 Plus, зображений на рисунку 2.7, є високопродуктивним сервером, який може використовуватися для розміщення систем управління доступом, таких як Active Directory.

Рисунок 2.7 – Зображення сервера HPE ProLiant DL380 Gen10 Plus

Отже, системи контролю доступу виступають невіддільним елементом комплексного захисту інформації. Інтеграція різноманітних підходів (MAC, DAC, RBAC, ABAC), впровадження новітніх технологій автентифікації та впровадження регулярних перевірок наданих прав забезпечує побудову адаптивних та надійних механізмів безпеки, що відповідають конкретним вимогам і потребам організації.

2.3 Захист від зовнішніх та внутрішніх інформаційних загроз

Розробка ефективних систем захисту інформації вимагає від фахівців чіткого визначення пріоритетності зовнішніх та внутрішніх інформаційних загроз, що формують основний напрямок їхньої професійної діяльності. Цілісна стратегія інформаційного захисту охоплює як зовнішні загрози – кібератаки, фішингові кампанії та розподілені атаки на відмову в обслуговуванні, так і внутрішні ризики, пов'язані з ненавмисними діями співробітників та зловмисними вчинками інсайдерів. Побудова всеосяжної системи безпеки потребує гармонійного поєднання технологічних рішень з організаційними процедурами та культурою безпечної поведінки персоналу.

Антивірусне програмне забезпечення формує базовий рівень захисту від зовнішніх загроз безпеці. Подібні програми спроектовані для виявлення та знешкодження шкідливого ПЗ – вірусів, троянських програм, програм-вимагачів – запобігаючи тим самим пошкодженню системних компонентів. Сучасні антивірусні рішення, такі як Kaspersky та Norton, використовують комбіновані методи сигнатурного аналізу та поведінкової аналітики для виявлення як відомих загроз, так і потенційних безпекових ризиків [19]. Ефективність такого захисту безпосередньо залежить від актуальності вірусних баз даних, інтегрованих у системи безпеки.

Протидія фішинговим атакам через електронну пошту здійснюється за допомогою розвинутих антиспам-засобів. Алгоритми на основі машинного навчання ідентифікують підозрілі посилання та вкладення, фільтруючи небажані повідомлення та блокуючи потенційно небезпечну кореспонденцію. Вбудовані системи фільтрації небажаної пошти в корпоративних рішеннях Microsoft Exchange та сервісі Gmail суттєво знижують ризик успішних фішингових кампаній завдяки глибокій інтеграції з цими платформами.

Людський фактор робить атаки соціальної інженерії настільки частими як зовнішню загрозу. Користувачі зазнають кібератак шляхом фішингу, коли за допомогою методів обману з них виманюють паролі та фінансові дані. Захист від

соціальної інженерії вимагає як технологічних підходів до безпеки, так і організаційної політики захисту.

Технічні засоби захисту включають впровадження спеціалізованих систем моніторингу підозрілої активності, таких як Wazuh, що інтегрується з SIEM-платформами для комплексного відстеження безпекових подій. Організаційні заходи охоплюють систематичне навчання персоналу – проведення тренінгів з виявлення фішингових повідомлень, розпізнавання штучно створених ситуацій терміновості (на кшталт вимоги "негайно оновити пароль") та інших маніпулятивних прийомів, що сприяє підвищенню пильності працівників.

Внутрішні загрози часто виникають через необґрунтовано розширені права доступу або через неуважність працівників. Для мінімізації таких ризиків критично важливо запроваджувати надійні механізми управління доступом [20]. Принцип мінімальних привілеїв (PoLP) гарантує, що користувачі отримують доступ виключно до тих ресурсів, які безпосередньо необхідні для виконання їхніх професійних обов'язків. Наприклад, менеджер із продажів не повинен мати доступ до серверів розробки.

Технологія багатофакторної автентифікації (MFA) забезпечує підвищений рівень захисту інформаційних ресурсів організації. Для успішної ідентифікації користувачі мають пройти щонайменше два різні типи перевірки, використовуючи, наприклад, коди, надіслані через SMS, та спеціалізовані апаратні пристрої автентифікації. У документації щодо інформаційної системи наводиться приклад рішення Duo Security – платформи MFA, що відзначається легкістю інтеграції з різноманітними корпоративними застосунками.

Завдяки впровадженню регулярних аудитів прав доступу можна виявити надлишкові дозволи, що залишилися від колишніх працівників або неактуальних систем. Організаціям необхідно оперативно скасовувати права доступу для співробітників при їхньому переведенні на інші посади. Платформа CyberArk функціонує як комплексне PAM-рішення, що забезпечує суворий контроль над привілейованим доступом адміністраторів до критично важливих системних компонентів.

Захист організації від зовнішніх загроз значною мірою залежить від мережевих технологій. Система мережевої безпеки Cisco ASA разом з pfSense працює як міжмережеві екрани, які встановлюють правила для контролю вхідних і вихідних з'єднань. Мережевий пристрій pfSense, зображений на рисунку 2.8, ілюструє, як pfSense блокує інтернет-трафік від Cisco, демонструючи таким чином роботу брандмауерів.

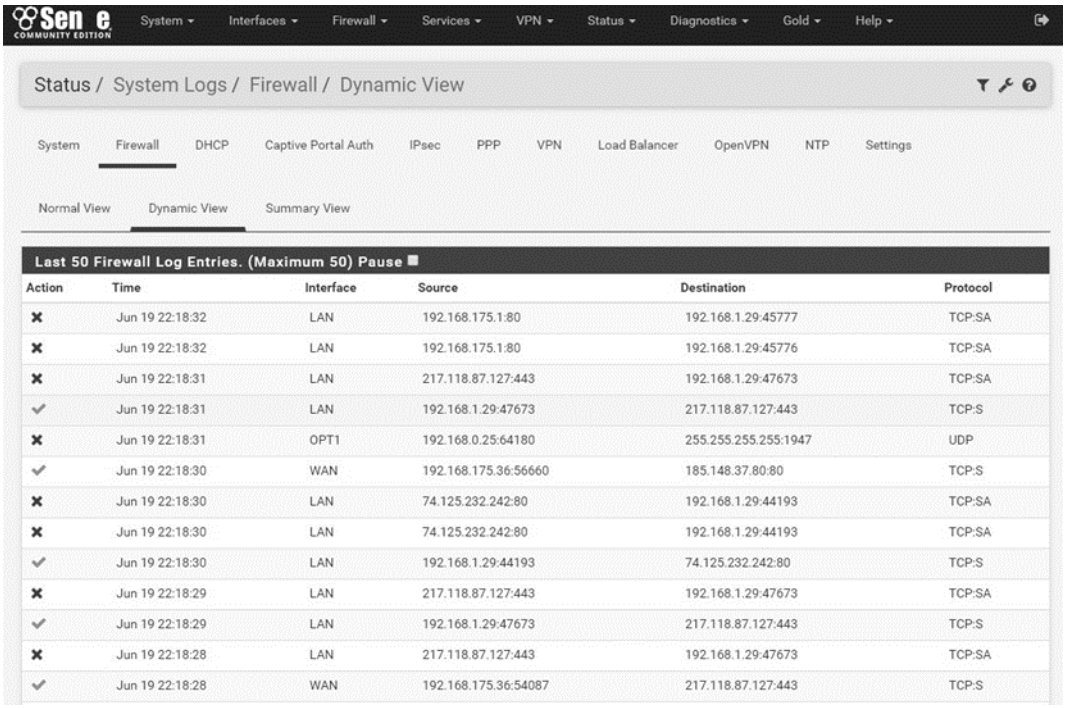


Рисунок 2.8 – Зображення pfSense, який блокує трафік від Cisco

Комплекс захищеного обміну даними включає технологію VPN та криптографічні протоколи SSH і TLS. Віртуальна приватна мережа створює зашифровані канали віддаленого доступу, що забезпечують конфіденційну передачу інформації – особливо важливий аспект для організації безпечної роботи з домашнього офісу. Безпечне керування серверами здійснюється за допомогою протоколу SSH, тоді як веб-ресурси, до яких користувачі звертаються через HTTPS-з'єднання, захищені протоколом TLS.

Забезпечення безпеки бездротових мереж становить важливий аспект загальної системи захисту. Стандарти WPA2 та WPA3 реалізують шифрування з'єднань Wi-Fi, ефективно блокуючи спроби зловмисників перехоплювати

мережевий трафік. Сучасний стандарт WPA3 пропонує вдосконалені алгоритми шифрування та механізми захисту від спроб підбору паролів. Для максимальної безпеки рекомендується використовувати складні паролі, вимикати публічну трансляцію ідентифікатора мережі (SSID) та створювати окремий гостьовий доступ для відвідувачів.

Системи виявлення (IDS) та попередження (IPS) вторгнень допомагають виявляти та блокувати підозрілу активність. Наприклад, Snort, зображений на рисунку 2.9, аналізує мережевий трафік у реальному часі, використовуючи набір правил для виявлення атак.

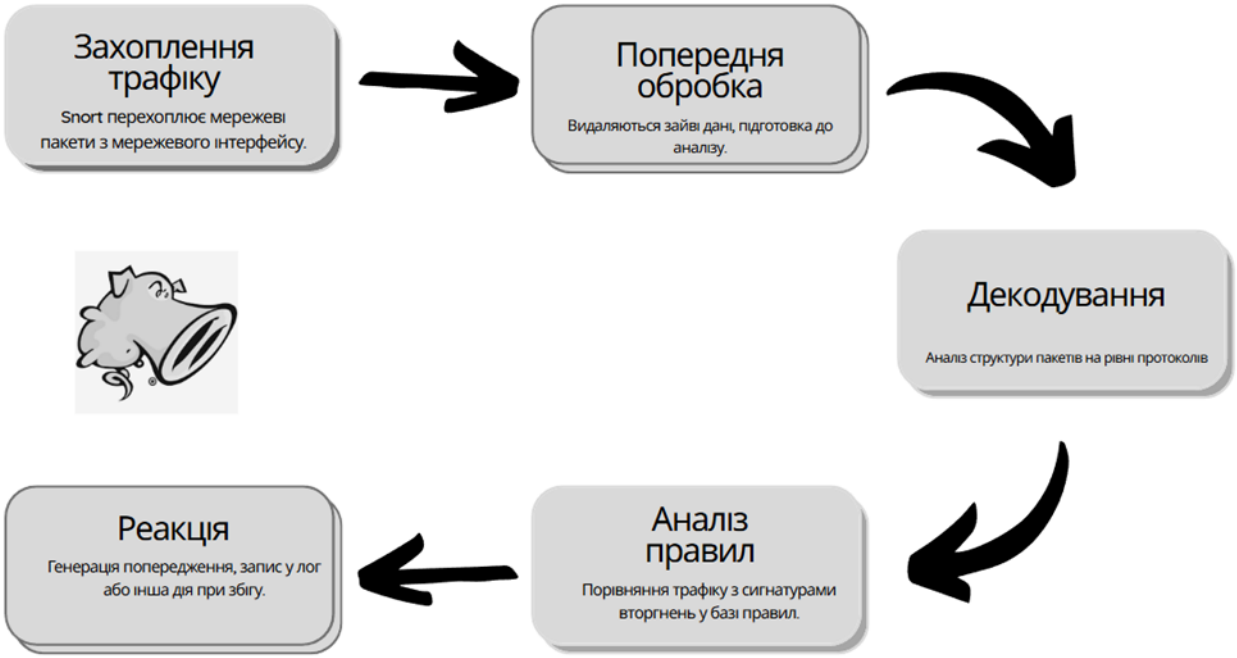


Рисунок 2.9 – Демонстрація принципу дії Snort

Suricata представляє собою прогресивний інструмент безпеки, що суттєво перевершує Snort завдяки високій продуктивності та підтримці багатопотокової обробки даних. Система Wazuh забезпечує функціонал моніторингу через інтеграцію з SIEM-платформами, надаючи користувачам можливість аналізувати журнали подій для виявлення аномальної активності.

Особливо ефективним методом моніторингу виступає аналіз поведінки користувачів (UEBA). Ця технологія ідентифікує небажані відхилення від типових

операційних шаблонів шляхом виявлення нестандартних спроб доступу чи незвичної поведінки системних компонентів. UEBA відіграє вирішальну роль у розпізнаванні інсайдерських загроз, що виникають через компрометацію облікових записів.

Для створення всеосяжної системи захисту необхідна інтеграція різних захисних методів у єдиний злагоджений комплекс. Організації впроваджують SIEM-системи, такі як Splunk чи ELK Stack, для агрегації безпекових подій, що забезпечує централізований моніторинг та виявлення аномальних паттернів у даних. SIEM-платформи підвищують загальний рівень захищеності завдяки здатності обробляти інформацію з міжмережевих екранів, систем виявлення/запобігання вторгненням та системних журналів, формуючи єдиний масив даних про безпекові події.

Виявлення вразливостей та вдосконалення захисних механізмів здійснюється через регулярні аудити у поєднанні з тестуванням на проникнення. Міжмережеві екрани, системи контролю доступу та мережеві конфігурації перевіряються на наявність слабких місць за допомогою пентестів, що моделюють реальні атаки кіберзловмисників.

Захист від ризиків вимагає, щоб зовнішні та внутрішні заходи безпеки функціонували як додатковий метод захисту. Сучасна система захисту успішно бореться з кіберзагрозами за допомогою антивірусних і антиспамових технологій у поєднанні із захистом від соціальної інженерії та безпечним контролем доступу і мережевими технологіями разом із системами виявлення вторгнень.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗАХИСТУ ДАНИХ

3.1 Обґрунтування середовища програмування та технологій

Для розробки програмного комплексу захисту даних у середовищі Visual Studio Code було використано мову Python – одну з найпоширеніших мов програмування, що видрізняється своєю доступністю, адаптивністю та багатим набором бібліотек для реалізації криптографічних алгоритмів і побудови графічних інтерфейсів. Python забезпечує швидкий цикл розробки, що надзвичайно важливо при створенні прототипів та проведенні тестування механізмів інформаційного захисту.

Основні технології та бібліотеки, що використовуються в цьому проекті, включають в себе наступні:

Основним середовищем виконання розробленого додатку виступає Python 3.11 – версія, що забезпечує стабільну роботу, підтримку актуальних стандартів та оптимізовану продуктивність системи.

Бібліотека tkinter, що входить до стандартного пакету Python, використовується для візуальної розробки користувацького інтерфейсу. Вибір на користь tkinter зумовлений її нативною інтеграцією з Python та здатністю оперативно створювати інтерактивні вікна, що суттєво спрощує процеси введення та відображення інформації.

Бібліотека hashlib дозволяє користувачам створювати криптографічні хеш-висновки. Алгоритм PBKDF2-HMAC з методами хешування SHA-1, SHA-256, SHA-384 та SHA-512 генерує ключі шифрування за допомогою програми цього проекту. Зазначені криптографічні методи перешкоджають зловмисникам здійснювати атаки на основі словників.

Бібліотека криптографії (Fernet) забезпечує реалізацію симетричного шифрування. Механізм Fernet використовує алгоритм AES-128 у режимі CBC для гарантування надійного захисту конфіденційності даних, а також підтримує забезпечення операційної цілісності системи за допомогою автентифікації HMAC.

Цей вибір був зроблений з огляду на потужні безпекові можливості у поєднанні з інтуїтивно зрозумілим інтерфейсом користувача.

Завдяки використанню стандартних бібліотек Python – os, json та base64 – додаток забезпечує функціонал роботи з файловою системою, можливість серіалізації даних у форматі JSON та конвертацію ключів у форматі base64.

Вибір Python та відповідних бібліотек обумовлений їхньою доведеною надійністю, детальною документацією та потужною спільнотою користувачів. Швидке впровадження рішення забезпечується завдяки готовим механізмам захисту інформації, які також створюють основу для подальшого вдосконалення програмного продукту.

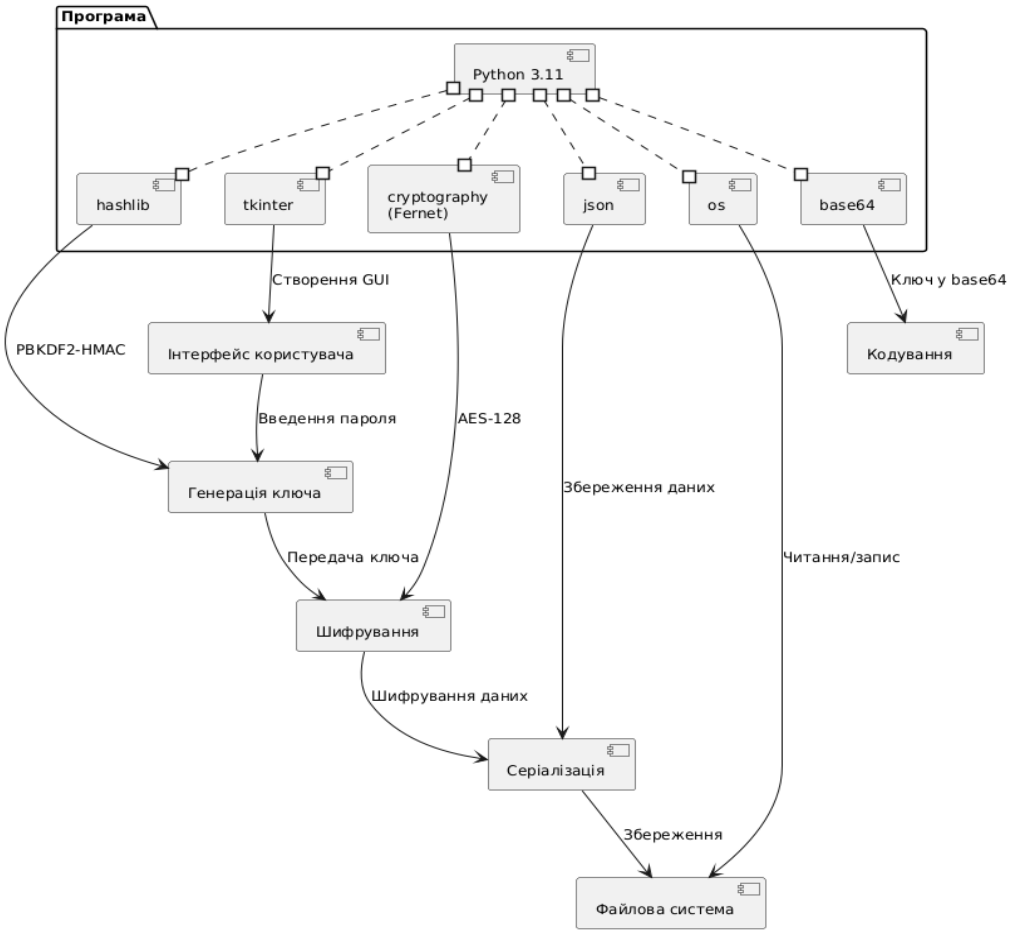


Рисунок 3.10 – Архітектура програми на Python 3.11 з модулями шифрування та обробки даних

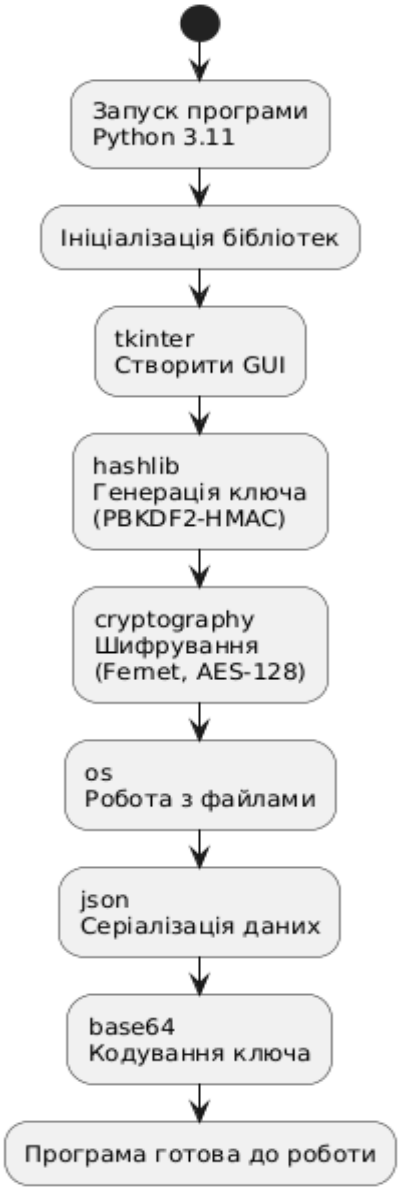


Рисунок 3.11 – Архітектура програми на Python 3.11 з використанням бібліотек для шифрування та обробки даних

3.2 Реалізація механізмів аутентифікації та контролю доступу

Програмне рішення впроваджує механізми автентифікації та контролю доступу для запобігання неавторизованому використанню даних. Майстер-пароль функціонує як первинний спосіб відновлення зашифрованих паролів через систему криптографічного захисту.

Процес аутентифікації реалізовано через метод authenticate() у класі PasswordManagerApp. Користувач вводить майстер-пароль і обирає метод

хешування (SHA-1, SHA-256, SHA-384 або SHA-512) через графічний інтерфейс. Далі викликається метод init_cipher(), який виконує такі дії:

Генерація ключа шифрування:

Метод derive_key() використовує бібліотеку hashlib для створення ключа шифрування за допомогою алгоритму PBKDF2-HMAC. Вхідними параметрами є майстер-пароль, сіль (генерується через os.urandom(16), якщо не існує) і обраний метод хешування.

Кількість ітерацій для PBKDF2 установлено на 100000, що забезпечує стійкість до атак грубої сили.

Сіль зберігається у спеціальному файлі key_<method>.dat для подальшого використання при наступних процедурах автентифікації.

Ініціалізація шифрування:

Створений криптографічний ключ передається до модуля Fernet (з бібліотеки cryptography) для ініціалізації об'єкта, що забезпечує функції шифрування (self.cipher).

При успішному завершенні процесу в графічному інтерфейсі відображаються значення солі (у шістнадцятковому форматі) та ключа (у форматі base64).

Якщо ініціалізація не вдалася (наприклад, через неправильний майстер-пароль), користувачу відображається повідомлення про помилку.

Перехід до головного меню:

Після вдалого проходження процедури автентифікації вікно входу (auth_frame) зникає, а користувачеві відкривається доступ до головного меню програми (main_frame), в якому можна виконувати операції зі збереження та отримання паролів.

Контроль доступу забезпечується через шифрування всіх паролів, які зберігаються у файлі passwords.json. Доступ до паролів можливий лише після успішної ініціалізації об’єкта Fernet із правильним ключем, який залежить від майстер-пароля. Це означає, що без правильного майстер-пароля розшифрувати дані неможливо.

Збереження пароля (save_password): Пароль шифрується за допомогою Fernet перед збереженням у passwords.json. Існуючі дані (якщо є) розшифровуються, додається новий запис, і весь набір даних знову шифрується.

Отримання пароля (get_password): Дані з passwords.json розшифровуються, і користувачу відображається лише той пароль, який відповідає введеному сервісу.

Отже, впроваджені механізми автентифікації та регулювання доступу забезпечують надійний захист інформації від неавторизованого використання, а застосування майстер-пароля як фундаментального елемента для криптографічних операцій створює додатковий рівень захисту системи.

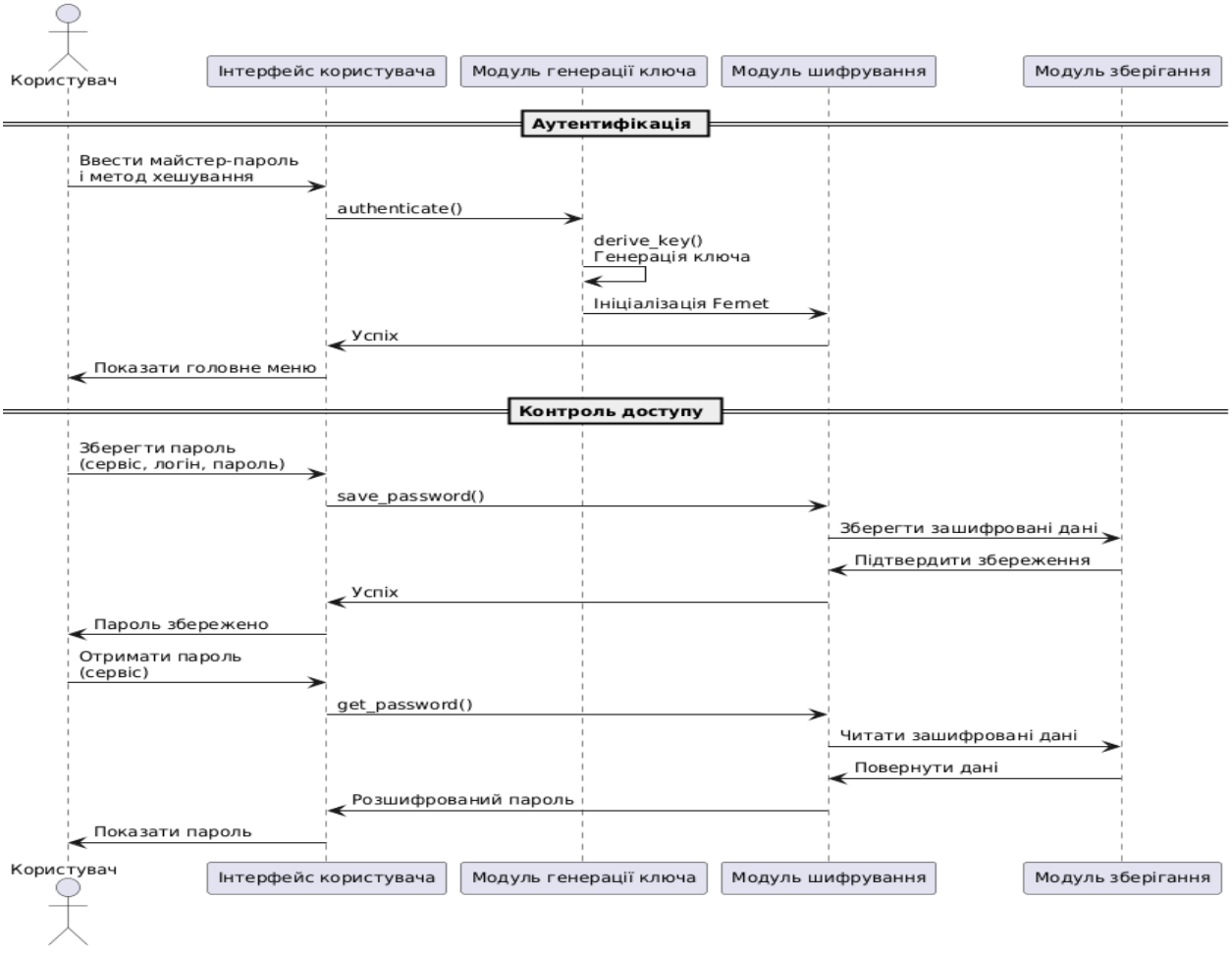


Рисунок 3.12 – Схема взаємодії між користувачем, модулями аутентифікації, генерації ключа, шифрування та контролем доступу

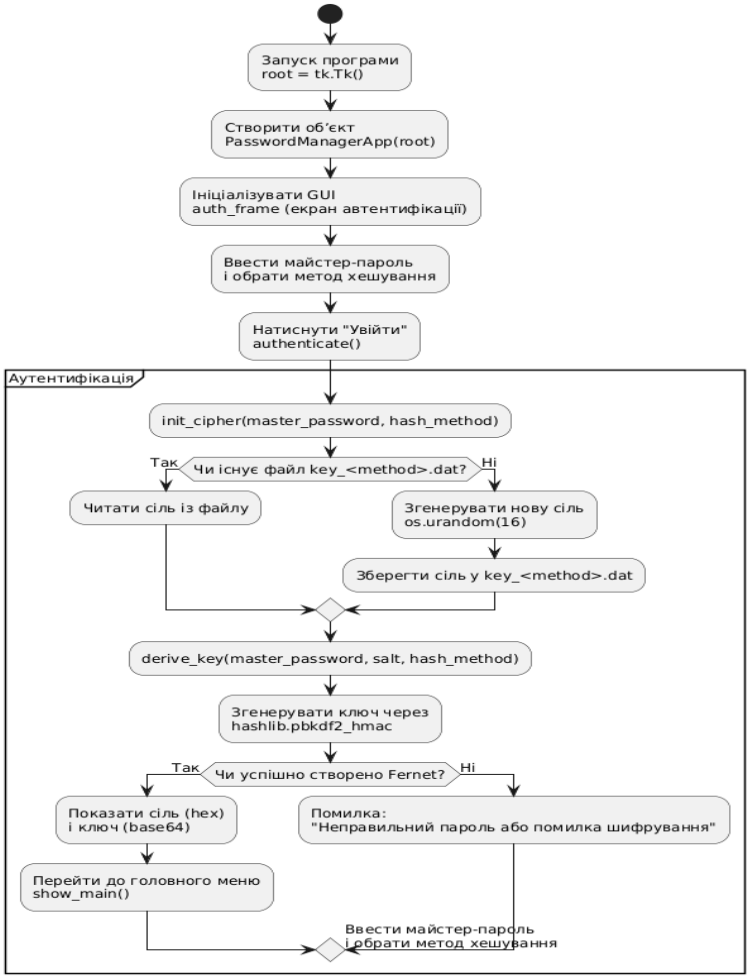


Рисунок 3.13 – Блок-схема процесу аутентифікації і процесу збереження пароля

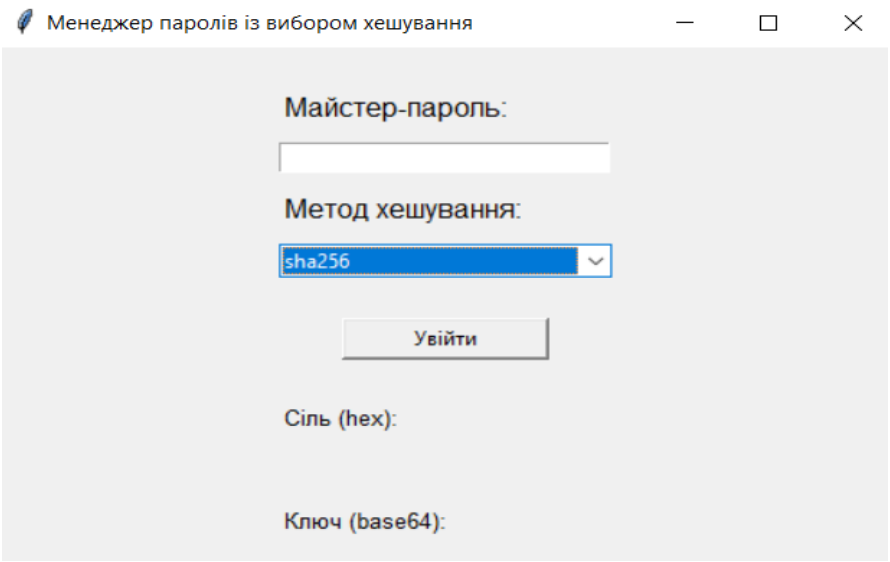


Рисунок 3.14 – Інтерфейс програми менеджера паролів із вибором хешування

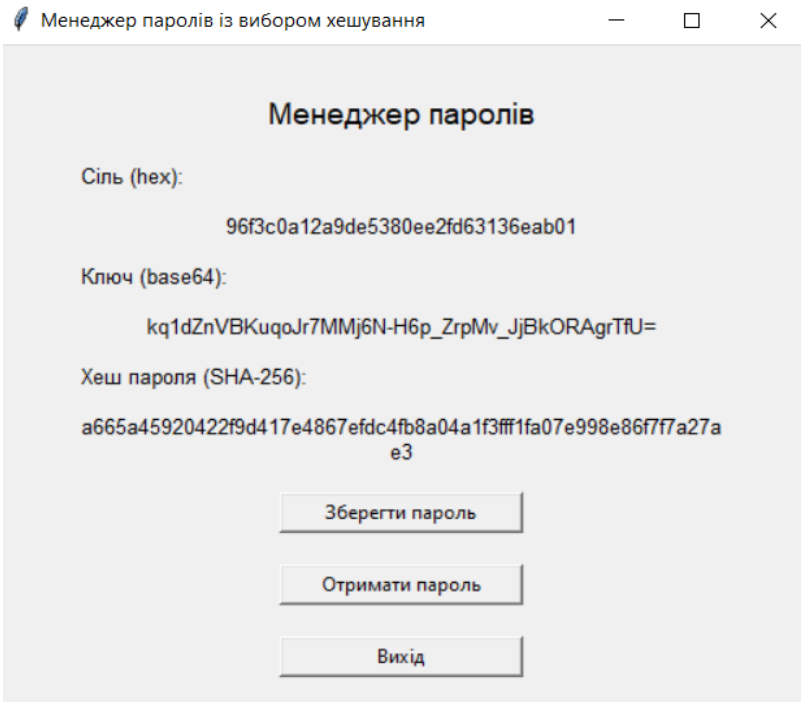


Рисунок 3.15 – Вікно солі і хешу

3.3 Моніторинг та аудит інформаційної безпеки

В програмі реалізовано фундаментальні функції моніторингу та аудиту, які забезпечують відстеження користувацьких дій та своєчасне виявлення потенційних загроз безпеці.

Моніторинг реалізовано через відображення ключових параметрів шифрування у графічному інтерфейсі:

Після введення майстер-пароля та ініціалізації шифрування (init_cipher) користувачу відображаються:

Сіль у шістнадцятковому форматі (self.salt.hex()).

Ключ шифрування у форматі base64 (self.key.decode()).

Це надає користувачам можливість перевірити правильність формування ключа та солі, а також пересвідчитись у коректності ініціалізації криптографічних механізмів системи.

Аудит інформаційної безпеки реалізовано через обробку та відображення помилок:

Помилки аутентифікації: Якщо ініціалізація Fernet не вдалася (наприклад, через неправильний майстер-пароль), користувачу відображається повідомлення про помилку з описом проблеми (messagebox.showerror).

Помилки дешифрування: Під час збереження або отримання пароля, якщо дешифрування даних із passwords.json не вдалося, користувачу повідомляється про проблему (наприклад, "Помилка дешифрування: Перевірте майстер-пароль і метод хешування").

Відсутність даних: Якщо файл passwords.json не існує або порожній, користувачу відображається відповідне повідомлення ("Жодних паролів не знайдено" або "Файл паролів порожній").

Наразі програма не веде журнал подій (логування), що є обмеженням для повноцінного аудиту. Для покращення моніторингу та аудиту рекомендується:

Впровадити систему реєстрації всіх користувацьких операцій (включаючи введення паролів, спроби входу до системи, збереження та запити паролів) з відображенням цієї інформації у спеціальному журнальному файлі.

Реалізувати механізм сповіщень про підозрілі дії (наприклад, кілька невдалих спроб аутентифікації).

Додати тимчасові мітки до подій для аналізу історії доступу.

Попри наявнім обмеженням, поточна версія програми забезпечує необхідний базовий рівень моніторингу та аудиту, що дозволяє користувачам контролювати процеси шифрування та вчасно виявляти можливі проблеми з доступом до збережених даних.

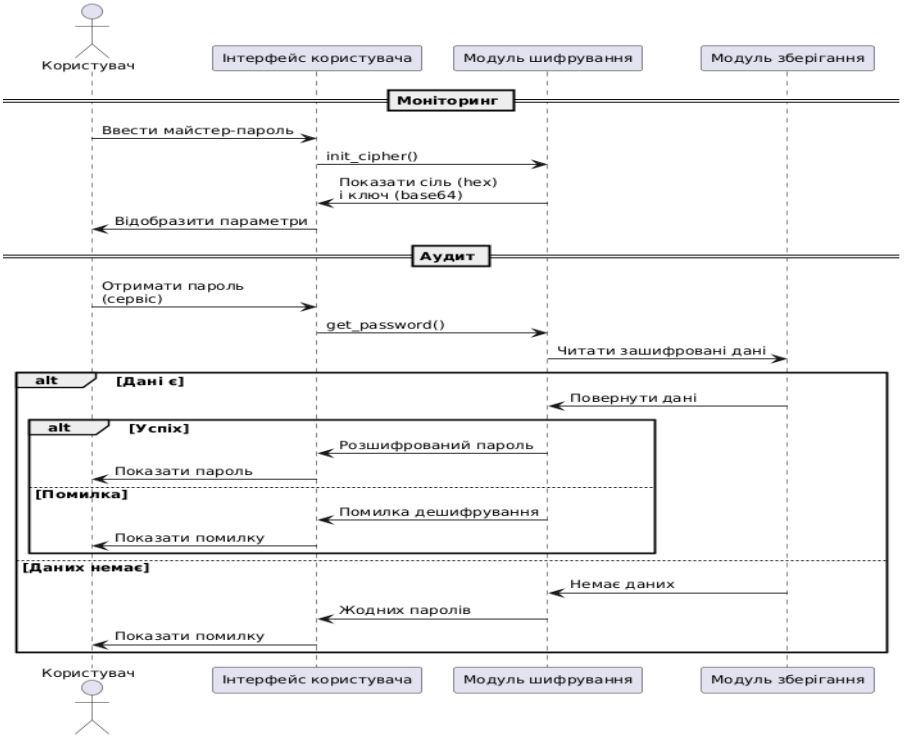


Рисунок 3.16 – Схема взаємодії між користувачем, інтерфейсом, модулями шифрування та аудитом обробки паролів

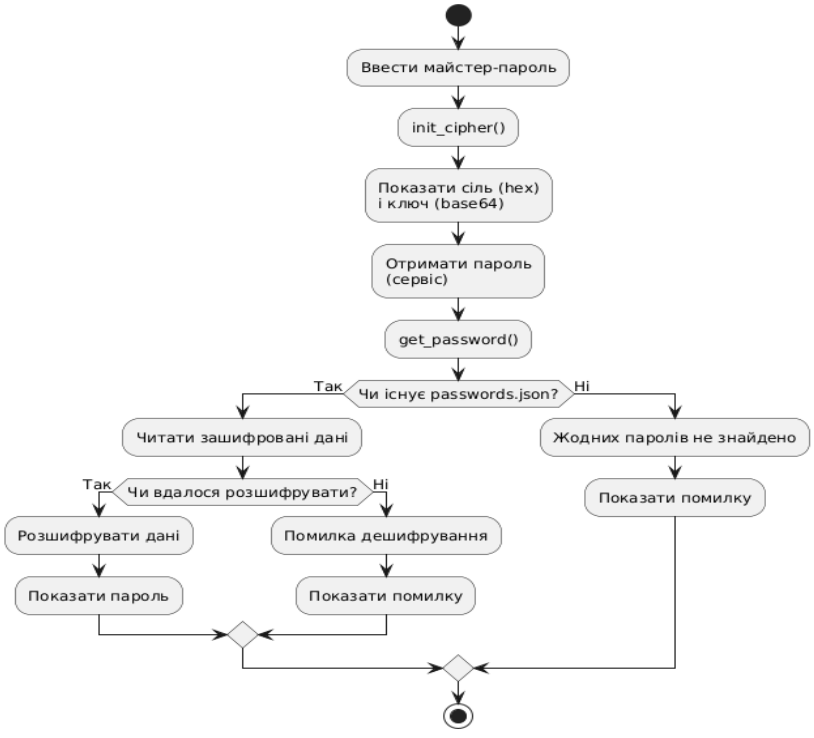


Рисунок 3.17 – Схема алгоритму введення та обробки паролів

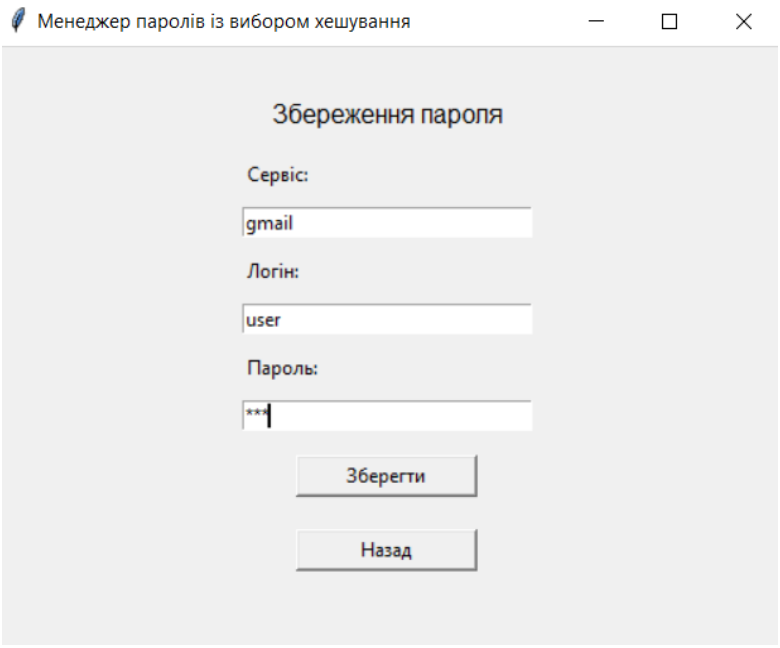


Рисунок 3.18 – Вікно збереження паролю

У розробленому програмному забезпеченні реалізовано сучасні методи криптографії, зокрема PBKDF2-HMAC та Fernet, які забезпечують надійний захист паролів – їхню конфіденційність та цілісність. Обрані інструменти, серед яких Python разом із бібліотеками tkinter, hashlib і cryptography, дають змогу створити зручний та ефективний додаток. Система також передбачає контроль параметрів шифрування через аудит і реалізує доступ на основі головного пароля для автентифікації користувача.

ВИСНОВКИ

Світ стрімко цифровізується, а разом із ним розвиваються й загрози – саме тому розробка сучасних засобів захисту даних сьогодні є не розкішшю, а необхідністю. Інформаційна безпека у 2025 році дедалі більше ґрунтується на поєднанні класичних принципів конфіденційності, цілісності та доступності з можливостями штучного інтелекту, постквантової криптографії та хмарних технологій. Такий підхід дозволяє не лише протистояти поширеним ризикам – зокрема кіберзлочинності та людському фактору – а й бути готовим до викликів, пов'язаних із новими технологіями. Події останніх років, коли компанії Tesla, Amazon і Netflix стали жертвами кібератак, наочно демонструють необхідність впровадження комплексних технічних, організаційних і правових заходів для захисту критично важливої інформації.

Організації повинні впровадити розгалужену систему протидії загрозам інформаційній безпеці, до яких належать атаки з використанням програм-вимагачів, фішингові схеми, а також розподілені атаки на відмову в обслуговуванні та атаки, що базуються на штучному інтелекті. Аналіз загроз безпеці через перевірку походження загроз (внутрішніх і зовнішніх) у поєднанні з оцінкою характеру загроз (випадкових і навмисних) та оцінкою впливу (апаратних і програмних засобів) створює структуровані методи для створення захисних заходів. Атака на медичну мережу LockBit 4.0 у США у 2025 році та витік даних Netflix через Microsoft Azure демонструють нагальну потребу підтримувати актуальні версії програмного забезпечення та впроваджувати системи моніторингу разом із навчанням працівників для зменшення потенційних загроз. Створення ефективних систем захисту даних є комплексним процесом, що передбачає чіткий методичний підхід. Основне завдання – забезпечити конфіденційність, цілісність та автентичність інформації. Для цього використовуються сучасні криптографічні засоби, зокрема симетричне та асиметричне шифрування, хеш-функції та цифрові підписи. Поєднання систем контролю доступу, що використовують моделі MAC, DAC, RBAC і ABAC, забезпечує сувору авторизацію ресурсів, що знижує рівень

несанкціонованої активності. Системи антивірусного захисту та IDS/IPS у поєднанні із захистом від соціальної інженерії, а також безпечний контроль доступу, міжмережеві екрани та VPN захищають як від внутрішніх, так і від зовнішніх загроз. Формування єдиної системи безпеки за допомогою цих методів разом з плановими роботами з моніторингу та аудиту є ефективним захистом від сучасної кіберзлочинності. У процесі розробки програмного забезпечення для захисту даних – зокрема менеджера паролів – було продемонстровано практичне впровадження криптографічних алгоритмів та механізмів аутентифікації. Для симетричного шифрування паролів було використано бібліотеку cryptography.fernet, що забезпечує надійний захист зберіганої інформації. Алгоритм PBKDF2-HMAC, із підтримкою хеш-функцій SHA-1, SHA-256, SHA-384 та SHA-512, дав змогу безпечно генерувати ключі на основі головного пароля та криптографічної солі. Інтерфейс, створений за допомогою tkinter, надає зручний доступ до функцій збереження та отримання паролів, а також відображення криптографічних параметрів, таких як сіль, ключ і хеш пароля. Функціонал аутентифікації через майстер-пароль із вибором методу хешування забезпечує гнучкість і безпеку, а збереження даних у зашифрованому файлі passwords.json гарантує їхню конфіденційність. Для посилення безпеки впроваджене програмне забезпечення потребує підтримки двофакторної автентифікації (2FA) та систем моніторингу підозрілої активності при інтеграції з ширшими платформами захисту даних, такими як управління паролями в корпоративній мережі. Створення надійної системи захисту даних для інформаційних активів вимагає всебічного підходу – від аналізу потенційних загроз до розробки ефективних механізмів безпеки та впровадження відповідного програмного забезпечення. Сучасні системи захисту активно вдосконалюються завдяки застосуванню поведінкового аналізу на основі штучного інтелекту, постквантових криптографічних рішень та масштабованих хмарних технологій. Ефективний захист цифрової інформації значною мірою залежить від регулярного оновлення систем, рівня підготовки персоналу та дотримання чинних правових норм, що разом забезпечує високий рівень операційної безпеки.

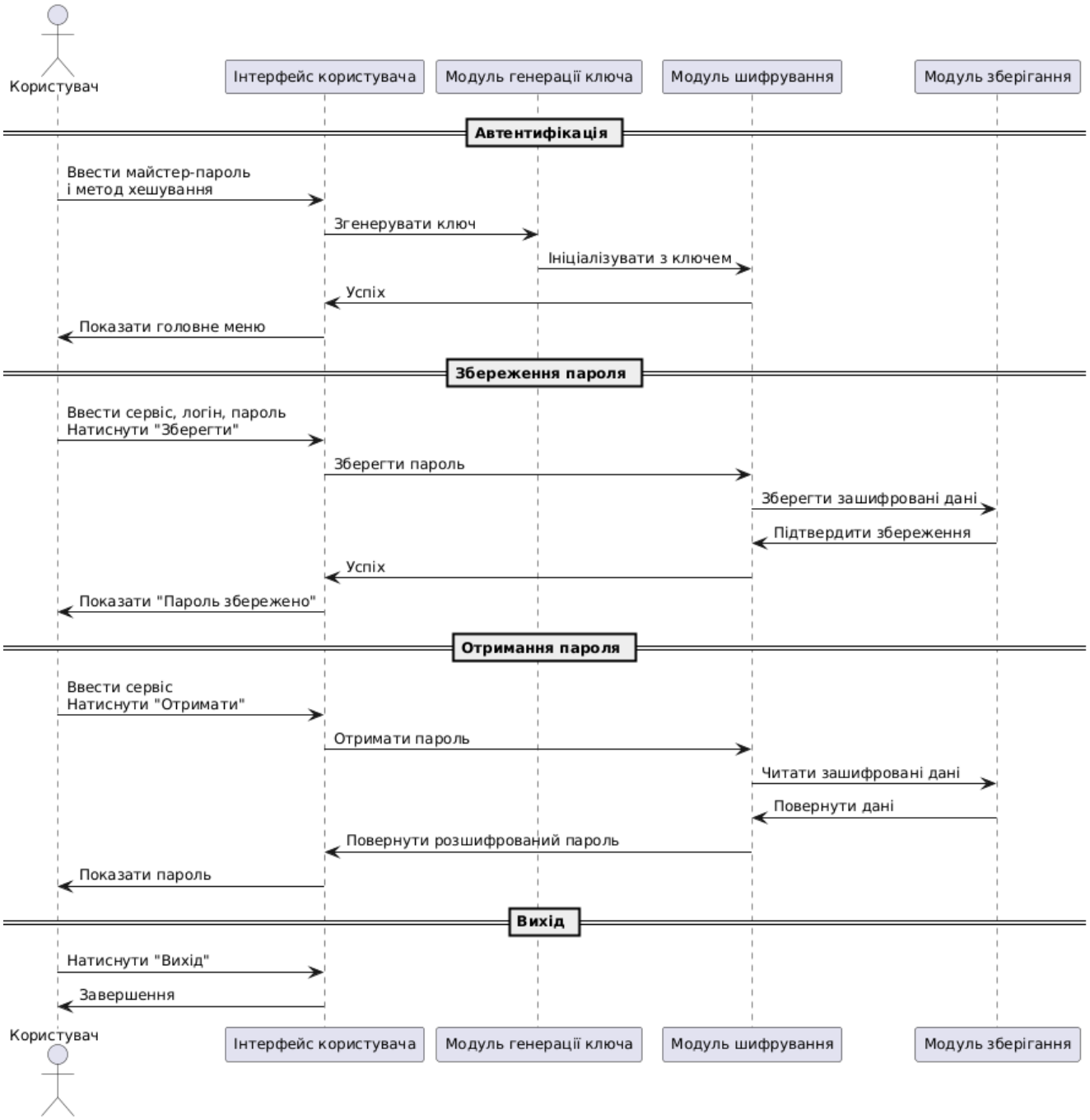
ПЕРЕЛІК ПОСИЛАНЬ

1. Chukwudi T.A., Patience E., Tawo O., Oluwatoyin A. Data security strategies to avoid data breaches in modern information systems // World Journal of Advanced Research and Reviews. – 2025. – Vol. 25, № 1. – P. 827–849. – DOI: https://doi.org/10.30574/wjarr.2025.25.1.3906 URL: https://journalwjarr.com/node/356
2. Andersson S., Bergström E., Lundgren M., Bernsmed K., Bour G. Information security risk management tools in the air traffic management domain: what are practitioners' needs? // Information Security Journal: A Global Perspective. – 2025. – P. 1–18. – DOI: https://doi.org/10.1080/19393555.2025.2498472 URL: https://is.gd/5jyMJh
3. Wang W., Jian S., Tan Y., Wu Q., Huang C. Robust unsupervised network intrusion detection with self-supervised masked context reconstruction // Computers & Security. – 2023. – Vol. 128. – P. 103131. – DOI: https://doi.org/10.1016/j.cose.2023.103131 URL: https://is.gd/SSWIkB
4. Cao Y. Research on the Application of Blockchain Technology in the Security Protection of Sensitive Data in Information Systems // Journal of Cyber Security and Mobility. – 2025. – DOI: https://doi.org/10.13052/jcsm2245-1439.1419 URL: https://is.gd/oaHPw7
5. Dubrovin A. S., Sumin V. I., Gromov Yu. Yu., Tyutyunnik V. M. A Systems Approach to Ensuring Security in a Special-Purpose Automated Information System // Automatic Documentation and Mathematical Linguistics. – 2025. – Vol. 59, № 1. – P. 33–40. – DOI: https://doi.org/10.3103/s0005105525700049 URL: https://is.gd/UwTVrf
6. Longo G., Lupia F., Merlo A., Pagano F., Russo E. A data anonymization methodology for security operations centers: Balancing data protection and security in industrial systems // Information Sciences. – 2025. – Vol. 690. – P. 121534. – DOI: https://doi.org/10.1016/j.ins.2024.121534 URL: https://is.gd/0a2TLv

7. Rodríguez-Correa P. A., Valencia-Arias A., Martínez Rojas E., Oré León A., Mellin Rubio R. H., Vásquez Coronado M. H., Jimenez Garcia J. A. Information security education: a thematic trend analysis // F1000Research. – 2025. – Vol. 14. – P. 5. – DOI: https://doi.org/10.12688/f1000research.159828.1 URL: https://is.gd/0YkHKN
8. Yuan S. Research on Anomaly Detection and Privacy Protection of Network Security Data Based on Machine Learning // Procedia Computer Science. – 2025. – Vol. 261. – P. 227–236. – DOI: https://doi.org/10.1016/j.procs.2025.04.193 URL: https://is.gd/SPcOVA
9. Tumma C., Azmeera R., Ayyamgari S., Thumma B. Y. R. Data Security and Privacy Protection in Artificial Intelligence Models: Challenges and Defense Mechanisms // International Journal of Scientific Research in Engineering and Management. – 2025. – Vol. 09, № 01. – P. 1–9. – DOI: https://doi.org/10.55041/ijsrem27335 URL: https://is.gd/XJa7Zx
10. Fasii B. National security and personal data protection in the era of digital technologies // Society and Security. – 2025. – Vol. 6, № 6. – P. 76–82. – DOI: https://doi.org/10.26642/sas-2024-6(6)-76-82 URL: https://is.gd/6TIASP
11. Song S., Lei X. Privacy Protection and Data Security Governance Strategies for Association Algorithms in the Digital Era of Enterprises // Procedia Computer Science. – 2025. – Vol. 261. – P. 589–597. – DOI: https://doi.org/10.1016/j.procs.2025.04.249 URL: https://is.gd/I9Rdzg
12. Błachut M., Świętochowski N. Security of information and communication systems in a digital society // Scientific Journal of the Military University of Land Forces. – 2025. – Vol. 215, № 1. – P. 1–23. – DOI: https://doi.org/10.5604/01.3001.0054.9201 URL: https://is.gd/31P2Kr
13. Dubynskyi G., Zubok V. The resilience of critical information infrastructure topology in the cyberspace // International Science Journal of Engineering & Agriculture. – 2025. – Vol. 4, № 1. – P. 103–110. – DOI: https://doi.org/10.46299/j.isjea.20250401.09 URL: https://isg-journal.com/isjea/article/view/943

14. Islam M. S., Xiangdong L. Guardians of the Web: The Evolution and Future of Website Information Security // arXiv. – 2025. – DOI: https://doi.org/10.48550/ARXIV.2505.04308 URL: https://arxiv.org/abs/2505.04308v1
15. Zhou J. Network Information Security System Based on Information Hiding Technology // Procedia Computer Science. – 2025. – Vol. 261. – P. 247–253. – DOI: https://doi.org/10.1016/j.procs.2025.04.195 URL: https://is.gd/EwBw58
16. Сумін П. Інформаційні технології як інструмент забезпечення національної безпеки на сучасному етапі розвитку: проблеми та перспективи // Society and Security. – 2025. – № 6(6). – С. 37–43. – DOI: https://doi.org/10.26642/sas-2024-6(6)-37-43 URL: https://sas.ztu.edu.ua/article/view/320727
17. Яремко С., Новицький Р. Розробка критеріїв захищеності інформаційних ресурсів в системах управління навчанням // Herald of Khmelnytskyi National University. Economic Sciences. – 2025. – № 338(1). – С. 252–258. – DOI: https://doi.org/10.31891/2307-5740-2025-338-37 URL: https://is.gd/cVF27a
18. Куперштейн Л. Інформаційна технологія моніторингу безпеки даних програмного забезпечення / Л.Куперштейн, Г.Луцишин, М.Кренцін // Кібербезпека : освіта, наука, техніка. – 2024. – № 3 (23). – С. 71–84. DOI 10.28925/2663-4023.2024.23.7184 URL: https://is.gd/flPdDX
19. hashlib — Secure hashes and message digests URL: https://docs.python.org/3/library/hashlib.html
20. Колесніков О.О. Захист від зовнішніх та внутрішніх кіберзагроз. 2024. V Міжнародна науково-практична конференція «Кібербезпека в сучасному світі: актуальні виклики. С. 85-87. URL: https://is.gd/g0nghi

Додаток А. Схема взаємодії модулів захисту інформації



Додаток Б. Лістинг програмного модуля для моніторингу безпеки

main.py

```
import os
import json
import base64
import hashlib
import tkinter as tk
from tkinter import messagebox, Entry, ttk
from cryptography.fernet import Fernet
ERROR_TITLE = "Помилка"
class PasswordManagerApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Менеджер паролів із вибором хешування")
        self.root.geometry("500x650")
        self.cipher = None
        self.salt = None
        self.key = None
        self.master_password = None
        self.hash_methods = ["sha1", "sha256", "sha384", "sha512"]
        self.auth_frame = tk.Frame(self.root)
        self.auth_frame.pack(pady=20)
        tk.Label(self.auth_frame, text="Майстер-пароль:",
font=("Arial", 12)).grid(row=0, column=0, padx=10, pady=5,
sticky="w")
        self.master_password_entry = Entry(self.auth_frame,
show="*", width=30)
        self.master_password_entry.grid(row=1, column=0, padx=10,
pady=5)
        tk.Label(self.auth_frame, text="Метод хешування:",
font=("Arial", 12)).grid(row=2, column=0, padx=10, pady=5,
sticky="w")
        self.hash_method_var = tk.StringVar(value="sha256")
        self.hash_method_menu = ttk.Combobox(self.auth_frame,
textvariable=self.hash_method_var, values=self.hash_methods,
state="readonly", width=27)
        self.hash_method_menu.grid(row=3, column=0, padx=10, pady=5)
        tk.Button(self.auth_frame, text="Увійти",
command=self.authenticate, width=15).grid(row=4, column=0, padx=10,
pady=20)
        self.main_frame = tk.Frame(self.root)
        tk.Label(self.main_frame, text="Менеджер паролів",
font=("Arial", 14)).grid(row=0, column=0, padx=10, pady=10)
        tk.Label(self.main_frame, text="Сіль (hex):", font=("Arial",
10)).grid(row=1, column=0, padx=10, pady=5, sticky="w")
        self.salt_display = tk.Label(self.main_frame, text="",
font=("Arial", 10), wraplength=400)
        self.salt_display.grid(row=2, column=0, padx=10, pady=5)
```

```
        tk.Label(self.main_frame, text="Ключ (base64):",
font=("Arial", 10)).grid(row=3, column=0, padx=10, pady=5,
sticky="w")
        self.key_display = tk.Label(self.main_frame, text="",
font=("Arial", 10), wraplength=400)
        self.key_display.grid(row=4, column=0, padx=10, pady=5)
        tk.Label(self.main_frame, text="Хеш пароля (SHA-256):",
font=("Arial", 10)).grid(row=5, column=0, padx=10, pady=5,
sticky="w")
        self.password_hash_display = tk.Label(self.main_frame,
text="", font=("Arial", 10), wraplength=400)
        self.password_hash_display.grid(row=6, column=0, padx=10,
pady=5)
        tk.Button(self.main_frame, text="Зберегти пароль",
command=self.show_save_password, width=20).grid(row=7, column=0,
padx=10, pady=10)
        tk.Button(self.main_frame, text="Отримати пароль",
command=self.show_get_password, width=20).grid(row=8, column=0,
padx=10, pady=10)
        tk.Button(self.main_frame, text="Вихід",
command=self.root.quit, width=20).grid(row=9, column=0, padx=10,
pady=10)
        self.save_frame = tk.Frame(self.root)
        tk.Label(self.save_frame, text="Збереження пароля",
font=("Arial", 12)).grid(row=0, column=0, padx=10, pady=10)
        tk.Label(self.save_frame, text="Сервіс:").grid(row=1,
column=0, padx=10, pady=5, sticky="w")
        self.service_save_entry = Entry(self.save_frame, width=30)
        self.service_save_entry.grid(row=2, column=0, padx=10,
pady=5)
        tk.Label(self.save_frame, text="Логін:").grid(row=3,
column=0, padx=10, pady=5, sticky="w")
        self.username_save_entry = Entry(self.save_frame, width=30)
        self.username_save_entry.grid(row=4, column=0, padx=10,
pady=5)
        tk.Label(self.save_frame, text="Пароль:").grid(row=5,
column=0, padx=10, pady=5, sticky="w")
        self.password_save_entry = Entry(self.save_frame, show="*",
width=30)
        self.password_save_entry.grid(row=6, column=0, padx=10,
pady=5)
        tk.Button(self.save_frame, text="Зберегти",
command=self.save_password_action, width=15).grid(row=7, column=0,
padx=10, pady=10)
        tk.Button(self.save_frame, text="Назад",
command=self.show_main, width=15).grid(row=8, column=0, padx=10,
pady=10)
        self.get_frame = tk.Frame(self.root)
        tk.Label(self.get_frame, text="Отримання пароля",
font=("Arial", 12)).grid(row=0, column=0, padx=10, pady=10)
```

```
        tk.Label(self.get_frame, text="Сервіс:").grid(row=1,
column=0, padx=10, pady=5, sticky="w")
        self.service_get_entry = Entry(self.get_frame, width=30)
        self.service_get_entry.grid(row=2, column=0, padx=10,
pady=5)
        tk.Button(self.get_frame, text="Отримати",
command=self.get_password_action, width=15).grid(row=3, column=0,
padx=10, pady=10)
        tk.Button(self.get_frame, text="Назад",
command=self.show_main, width=15).grid(row=4, column=0, padx=10,
pady=10)
    def derive_key(self, master_password: str, salt: bytes = None,
hash_method: str = "sha256") -> tuple:
        if salt is None:
            salt = os.urandom(16)
        key = hashlib.pbkdf2_hmac(
            hash_name=hash_method,
            password=master_password.encode(),
            salt=salt,
            iterations=100000,
            dklen=32
        )
        key = base64.urlsafe_b64encode(key)
        return key, salt
    def init_cipher(self, master_password: str, hash_method: str) ->
bool:
        key_file = f'key_{hash_method.lower()}.dat'
        if os.path.exists(key_file):
            with open(key_file, 'rb') as f:
                self.salt = f.read(16)
            self.key, _ = self.derive_key(master_password,
self.salt, hash_method)
        else:
            self.key, self.salt = self.derive_key(master_password,
hash_method=hash_method)
            with open(key_file, 'wb') as f:
                f.write(self.salt)
        try:
            self.cipher = Fernet(self.key)
            self.salt_display.config(text=self.salt.hex())
            self.key_display.config(text=self.key.decode())
            return True
        except Exception as e:
            messagebox.showerror(ERROR_TITLE, f"Неправильний
майстер-пароль або помилка шифрування: {e}")
            self.salt_display.config(text="")
            self.key_display.config(text="")
            return False
    def save_password(self, service: str, username: str, password:
str):
```

```
        data_file = 'passwords.json'
        passwords = {}
        if os.path.exists(data_file):
            with open(data_file, 'rb') as f:
                encrypted_data = f.read()
                if encrypted_data:
                    try:
                        decrypted_data =
self.cipher.decrypt(encrypted_data).decode()
                        passwords = json.loads(decrypted_data)
                    except Exception as e:
                        messagebox.showerror(ERROR_TITLE, f"Помилка
дешифрування: {e}. Перевірте майстер-пароль і метод хешування.")
                        return
        passwords[service] = {'username': username, 'password':
self.cipher.encrypt(password.encode()).decode()}

        with open(data_file, 'wb') as f:
            f.write(self.cipher.encrypt(json.dumps(passwords).encode
()))
        messagebox.showinfo("Успіх", f"Пароль для {service}
збережено.")
    def get_password(self, service: str):
        data_file = 'passwords.json'
        if not os.path.exists(data_file):
            messagebox.showerror(ERROR_TITLE, "Жодних паролів не
знайдено.")
            return
        with open(data_file, 'rb') as f:
            encrypted_data = f.read()
            if not encrypted_data:
                messagebox.showerror(ERROR_TITLE, "Файл паролів
порожній.")
                return
            try:
                decrypted_data =
self.cipher.decrypt(encrypted_data).decode()
                passwords = json.loads(decrypted_data)
                if service in passwords:
                    username = passwords[service]['username']
                    password =
self.cipher.decrypt(passwords[service]['password'].encode()).decode(
)
                    messagebox.showinfo("Пароль", f"Сервіс:
{service}\nЛогін: {username}\nПароль: {password}")
                else:
                    messagebox.showerror(ERROR_TITLE, f"Сервіс
{service} не знайдено.")
            except Exception as e:
```

```
                messagebox.showerror(ERROR_TITLE, f"Помилка
дешифрування: {e}. Перевірте майстер-пароль і метод хешування.")
    def authenticate(self):
        self.master_password = self.master_password_entry.get()
        hash_method = self.hash_method_var.get()
        if self.init_cipher(self.master_password, hash_method):
            password_hash =
hashlib.sha256(self.master_password.encode()).hexdigest()
            self.password_hash_display.config(text=password_hash)
            self.auth_frame.pack_forget()
            self.show_main()
    def show_main(self):
        self.save_frame.pack_forget()
        self.get_frame.pack_forget()
        self.main_frame.pack(pady=20)
    def show_save_password(self):
        self.main_frame.pack_forget()
        self.save_frame.pack(pady=20)
    def show_get_password(self):
        self.main_frame.pack_forget()
        self.get_frame.pack(pady=20)
    def save_password_action(self):
        service = self.service_save_entry.get()
        username = self.username_save_entry.get()
        password = self.password_save_entry.get()
        if service and username and password:
            self.save_password(service, username, password)
            self.service_save_entry.delete(0, tk.END)
            self.username_save_entry.delete(0, tk.END)
            self.password_save_entry.delete(0, tk.END)
        else:
            messagebox.showerror(ERROR_TITLE, "Заповніть усі поля.")
    def get_password_action(self):
        service = self.service_get_entry.get()
        if service:
            self.get_password(service)
            self.service_get_entry.delete(0, tk.END)
        else:
            messagebox.showerror(ERROR_TITLE, "Введіть назву
сервісу.")
if __name__ == "__main__":
    root = tk.Tk()
    app = PasswordManagerApp(root)
    root.mainloop()
```