Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАСТОСУНКУ ЗАХИСТУ ДАНИХ ВІД НЕСАНКЦІОНОВАНОГО ВИКОРИСТАННЯ ІЗ ЗАСТОСУВАННЯМ АПАРАТНИХ ЗАКЛАДНИХ ЗАСОБІВ

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-22-1
Деркач А. Ю.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник ст. викл. Бродецький Ф. А.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Деркачу Андрію Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення програмного застосунку захисту даних від несанкціонованого використання із застосуванням апаратних закладних засобів

затверджена наказом університету від 26 травня 2026 року № 435Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 22 червня 2026 р.
3. Вихідні дані до роботи: науково-методична та науково-технічна література з питань цифрового захисту даних, документація мови програмування Python, документація бібліотек і фреймворків, використаних під час розроблення застосунку, матеріали наукових конференцій, електронні інформаційні ресурси мережі Інтернет, результати аналізу сучасних програмних засобів захисту даних.

4. Перелік питань, що потрібно опрацювати в роботі
1. Огляд предметної області захисту даних та аналіз сучасних методів інформаційної безпеки.
2. Проєктування системи захисту даних із використанням апаратних закладних засобів.
3. Програмна реалізація та тестування застосунку захисту даних.

5. Перелік графічного матеріалу: схема архітектури програмного застосунку, схема взаємодії модулів системи, схема алгоритму автентифікації користувача, схема алгоритму шифрування та дешифрування даних, схема структури бази даних, фрагменти програмної реалізації основних модулів, скріншоти інтерфейсу програмного застосунку, скріншоти процесу автентифікації користувача, роботи із захищеними даними та результатів тестування. _____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	13.04.2026	
2	Аналіз завдання, підбір літератури	14.04.26-16.04.26	
3	Аналіз літератури з проблеми, що вирішується	17.04.26-19.04.26	
4	Аналіз існуючих методів захисту даних	20.04.26-22.04.26	
5	Формування кроків вирішення проблеми	23.04.26-05.05.26	
6	Програмна реалізація	26.04.26-20.05.26	
7	Аналіз отриманих результатів	21.05.26-04.06.26	
8	Оформлення пояснювальної записки	05.06.26-09.06.26	
9	Перевірка на нормоконтроль	12.06.26-16.06.26	
10	Перевірка на плагіат	16.06.26-17.06.26	
11	Рецензування	17.06.26-30.06.26	
12	Підготовка презентації та доповіді	20.06.26-30.06.26	
13	Занесення роботи в електронний архів	01.07.2026	
14	Попередній захист кваліфікаційної роботи	01.07.2026	

Дата видачі завдання 13 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ ст. викл. Бродецький Ф. А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 69 с., 6 табл., 11 рис., 21 джерело.

АПАРАТНА АВТЕНТИФІКАЦІЯ, ЗАХИСТ ДАНИХ, ІНФОРМАЦІЙНА БЕЗПЕКА, ПРОГРАМНИЙ ЗАСТОСУНОК, USB-КЛЮЧ, PYTHON, SQLITE, AES-256-GCM.

Об’єкт роботи – процес забезпечення захисту даних у сучасних інформаційних системах.

Мета роботи – розроблення програмного застосунку захисту даних із використанням апаратних закладних засобів.

Під час роботи використано методи аналізу предметної області, проєктування програмного забезпечення, криптографічні алгоритми, мову програмування Python, бібліотеку PyQt6, базу даних SQLite та алгоритм AES-256-GCM. Розроблено програмний застосунок Secure Vault, який реалізує автентифікацію користувача за допомогою USB-пристрою, шифрування даних, журналювання подій та автоматичне блокування доступу. Проведене тестування підтвердило працездатність і надійність розробленої системи.

HARDWARE AUTHENTICATION, DATA PROTECTION, INFORMATION SECURITY, SOFTWARE APPLICATION, USB SECURITY KEY, PYTHON, SQLITE, AES-256-GCM.

The object of the study is the protection of confidential data in information systems.

The purpose of the work is to develop a software application for data protection using hardware authentication devices.

Python, PyQt6, SQLite, and AES-256-GCM were used to implement the Secure Vault application. The software provides USB-based authentication, data encryption, event logging, and automatic system locking. Testing confirmed the reliability and correct operation of the developed application.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 3
Вступ 5
1 Постановка задачі та аналіз предметної області 8
1.1 Проблема захисту даних у сучасних інформаційних системах 8
1.2 Методи та засоби захисту інформації 11
1.3 Апаратні закладні засоби: сутність та принцип роботи 14
1.4 Аналіз існуючих рішень у сфері захисту даних 18
1.5 Постановка задачі 21
2 Аналіз та проєктування системи захисту даних 25
2.1 Вибір підходу до реалізації захисту 25
2.2 Розробка архітектури програмного застосунку 28
2.3 Моделювання процесів роботи системи 34
2.4 Вибір технологій та інструментів реалізації 39
2.5 Проєктування бази даних та структури зберігання інформації 45
3 Програмна реалізація та тестування системи 46
3.1 Реалізація основного функціоналу програмного застосунку 46
3.2 Реалізація користувацького інтерфейсу 52
3.3 Реалізація механізмів захисту даних та апаратної автентифікації 54
3.4 Тестування та оцінка ефективності програмногозастосунку 58
Висновки 63
Перелік джерел посилання 67

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Автентифікація – процес підтвердження особи користувача шляхом перевірки його облікових даних та апаратного ключа

Дешифрування – процес відновлення відкритих даних із зашифрованого вигляду за допомогою відповідного криптографічного ключа

Конфіденційність інформації – властивість інформації бути доступною лише авторизованим користувачам

Криптографічний ключ – послідовність символів або байтів, що використовується алгоритмом шифрування для захисту інформації

Шифрування – процес перетворення відкритих даних у зашифрований вигляд за допомогою криптографічного алгоритму

AES (Advanced Encryption Standard) – симетричний алгоритм шифрування даних

AES-256-GCM (Advanced Encryption Standard 256-bit Galois/Counter Mode) – алгоритм симетричного шифрування з довжиною ключа 256 біт, що забезпечує конфіденційність та контроль цілісності інформації

API (Application Programming Interface) – програмний інтерфейс взаємодії між програмними компонентами

GUI (Graphical User Interface) – графічний інтерфейс користувача

PBKDF2 (Password-Based Key Derivation Function 2) – алгоритм генерації криптографічного ключа на основі пароля або PIN-коду

PIN (Personal Identification Number) – персональний ідентифікаційний код користувача, який використовується для автентифікації

PyQt6 – бібліотека для створення графічного інтерфейсу користувача мовою програмування Python

Python – високорівнева мова програмування, використана для реалізації програмного застосунку

pytest – програмний фреймворк для автоматизованого тестування Python-застосунків

SQLite – вбудована реляційна система керування базами даних

USB (Universal Serial Bus) – універсальний послідовний інтерфейс підключення периферійних пристроїв

USB-ключ – апаратний засіб автентифікації, що використовується для підтвердження права доступу до програмного застосунку

Secure Vault – програмний застосунок, розроблений у межах кваліфікаційної роботи для захисту конфіденційних даних

ВСТУП

У сучасних умовах розвитку інформаційних технологій питання захисту даних набуває особливої актуальності. Постійне збільшення обсягів цифрової інформації, активне використання комп'ютерних систем у професійній діяльності, поширення хмарних сервісів та мережевих технологій створюють нові виклики у сфері інформаційної безпеки. Значна кількість конфіденційних даних сьогодні зберігається в електронному вигляді, що робить їх потенційною ціллю для несанкціонованого доступу, викрадення або модифікації. У зв'язку з цим виникає необхідність створення ефективних засобів захисту інформації, здатних забезпечити належний рівень безпеки даних.

Традиційні методи захисту, засновані виключно на використанні логінів і паролів, уже не гарантують достатнього рівня безпеки. Сучасні методи атак дозволяють отримувати доступ до облікових записів шляхом підбору паролів, використання шкідливого програмного забезпечення, фішингових ресурсів або методів соціальної інженерії. Однією з найбільш ефективних тенденцій розвитку систем інформаційної безпеки є впровадження багаторівневих механізмів автентифікації, зокрема із застосуванням апаратних засобів захисту.

Використання апаратних закладних засобів дозволяє значно підвищити рівень захисту інформації завдяки поєднанню програмних механізмів контролю доступу з фізичними пристроями автентифікації. Подібний підхід забезпечує додатковий рівень безпеки, оскільки для отримання доступу до системи недостатньо лише знати пароль – необхідно також мати відповідний апаратний ключ або пристрій. Така технологія активно застосовується у корпоративних інформаційних системах, банківському секторі, державних установах та інших сферах, де особливо важливим є захист конфіденційної інформації.

Актуальність теми кваліфікаційної роботи полягає у необхідності створення програмного застосунку, який забезпечуватиме захист даних від несанкціонованого використання шляхом використання апаратних засобів автентифікації та криптографічних механізмів захисту інформації. Розробка подібних систем є важливим напрямом розвитку сучасних засобів інформаційної безпеки та дозволяє підвищити надійність зберігання й обробки даних.

Об’єктом роботи є процес забезпечення захисту даних у сучасних інформаційних системах.

Предметом роботи є методи та засоби програмної реалізації системи захисту даних із застосуванням апаратних закладних засобів.

Метою кваліфікаційної роботи є розроблення програмного застосунку захисту даних із використанням апаратних закладних засобів.

Для досягнення поставленої мети необхідно виконати такі завдання:

– проаналізувати сучасні загрози інформаційній безпеці;

– дослідити методи та засоби захисту інформації;

– виконати аналіз існуючих рішень у сфері захисту даних;

– розробити архітектуру програмного застосунку;

– реалізувати механізми автентифікації користувача та контролю доступу;

– реалізувати механізми шифрування та дешифрування інформації;

– провести тестування розробленого програмного застосунку та оцінити ефективність його роботи.

Під час виконання кваліфікаційної роботи використано методи аналізу інформаційних систем, методи криптографічного захисту даних, методи програмного проєктування, а також технології взаємодії програмного забезпечення з апаратними пристроями.

Практичне значення отриманих результатів полягає у можливості використання розробленого програмного застосунку для забезпечення

захисту конфіденційної інформації, обмеження доступу до даних та підвищення рівня інформаційної безпеки користувачів.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі розглянуто теоретичні аспекти захисту інформації та виконано аналіз предметної області. У другому розділі здійснено проєктування системи захисту даних та вибір технологій реалізації. У третьому розділі наведено практичну реалізацію програмного застосунку, результати тестування та оцінку ефективності розробленого рішення.

1 ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Проблема захисту даних у сучасних інформаційних системах

У сучасних умовах розвитку цифрових технологій інформація є одним із найцінніших ресурсів як для окремих користувачів, так і для підприємств, державних установ та організацій. Значна частина професійної діяльності, фінансових операцій, документообігу та комунікацій сьогодні здійснюється із використанням інформаційних систем і комп'ютерних мереж. У зв'язку з цим питання захисту даних набуває особливої актуальності, оскільки будь-яке порушення безпеки інформації може призвести до фінансових втрат, витоку конфіденційних відомостей або порушення функціонування інформаційної системи [1].

Інформаційна безпека являє собою комплекс заходів, методів і засобів, спрямованих на забезпечення конфіденційності, цілісності та доступності інформації. Конфіденційність передбачає захист даних від доступу сторонніх осіб, цілісність забезпечує незмінність інформації під час її зберігання та передавання, а доступність гарантує можливість отримання інформації авторизованими користувачами у необхідний момент часу [2].

Сучасні інформаційні системи функціонують в умовах постійного зростання кількості кіберзагроз. Одними з найбільш поширених загроз є несанкціонований доступ до інформації, викрадення даних, шкідливе програмне забезпечення, фішингові атаки, перехоплення мережевого трафіку та соціальна інженерія. У більшості випадків основною метою зловмисників є отримання доступу до конфіденційної інформації або порушення роботи інформаційної системи [1].

Однією з основних проблем сучасних систем захисту є недостатня ефективність традиційних механізмів автентифікації. Найпоширенішим способом підтвердження особи користувача залишається використання

логіна та пароля. Проте такий підхід має значну кількість недоліків. Користувачі часто створюють прості паролі, використовують однакові комбінації для різних сервісів або зберігають облікові дані у незахищеному вигляді. Це значно підвищує ризик компрометації облікових записів [1].

Крім того, сучасні засоби автоматизованого підбору паролів дозволяють здійснювати тисячі спроб входу за короткий проміжок часу. У поєднанні з фішинговими атаками та шкідливим програмним забезпеченням це створює серйозну загрозу для безпеки інформаційних систем. У багатьох випадках навіть складні паролі не можуть гарантувати належного рівня захисту без використання додаткових механізмів безпеки [1].

У зв’язку з цим у сучасних інформаційних системах дедалі більшого поширення набуває багатофакторна автентифікація. Її принцип полягає у використанні декількох способів підтвердження особи користувача. Для отримання доступу до системи недостатньо лише знати пароль – необхідно також підтвердити особу за допомогою фізичного пристрою або біометричних даних. Такий підхід дозволяє значно знизити ризик несанкціонованого доступу навіть у разі компрометації одного з факторів автентифікації [2].

Одним із перспективних напрямів забезпечення інформаційної безпеки є використання апаратних засобів захисту інформації. Апаратні засоби дозволяють реалізувати додатковий рівень безпеки шляхом використання фізичних пристроїв для автентифікації користувача або зберігання криптографічних ключів. До таких засобів належать USB-токени, смарт-карти, TPM-модулі та інші спеціалізовані пристрої [2].

Особливістю апаратних засобів автентифікації є те, що вони поєднують програмний і фізичний рівні захисту. Для отримання доступу до інформації користувач повинен не лише знати пароль або PIN-код, але й мати відповідний пристрій. Це значно ускладнює процес несанкціонованого проникнення до системи та підвищує рівень безпеки даних [1, 2].

Важливу роль у забезпеченні захисту інформації відіграють також криптографічні методи. Шифрування дозволяє перетворити дані у форму, недоступну для читання без спеціального ключа дешифрування. Навіть у разі отримання доступу до файлів стороння особа не зможе використати інформацію без відповідного криптографічного ключа. Сучасні системи захисту активно використовують алгоритми AES, RSA та інші криптографічні механізми для забезпечення конфіденційності інформації [3].

Окремої уваги потребує проблема захисту локальних даних на персональних комп'ютерах. У багатьох випадках конфіденційна інформація зберігається безпосередньо на робочих пристроях користувачів. У разі фізичного доступу до комп'ютера або викрадення пристрою існує ризик втрати або компрометації даних. Саме тому використання апаратних засобів автентифікації та шифрування інформації є важливим елементом забезпечення безпеки локальних інформаційних систем [3].

Крім технічних факторів, значний вплив на рівень інформаційної безпеки має людський фактор. Помилки користувачів, ігнорування правил безпеки, встановлення невідомого програмного забезпечення або передавання облікових даних стороннім особам часто стають причиною інцидентів інформаційної безпеки. У зв'язку з цим сучасні системи захисту повинні бути не лише ефективними, але й зручними для використання, щоб мінімізувати ймовірність помилок користувача [1, 3].

Таким чином, проблема захисту даних у сучасних інформаційних системах є надзвичайно актуальною та потребує використання комплексних підходів до забезпечення інформаційної безпеки. Традиційні методи автентифікації вже не забезпечують достатнього рівня захисту, тому дедалі більшого поширення набувають рішення, що поєднують програмні механізми безпеки з апаратними засобами контролю доступу. Використання таких технологій дозволяє підвищити рівень захисту конфіденційної інформації та знизити ризики несанкціонованого використання даних.

1.2 Методи та засоби захисту інформації

У сучасних інформаційних системах захист інформації є одним із ключових елементів забезпечення стабільної та безпечної роботи програмного забезпечення. Постійне зростання кількості кіберзагроз, розвиток мережевих технологій та збільшення обсягів цифрових даних створюють необхідність використання ефективних методів і засобів захисту інформації. Основною метою таких методів є забезпечення конфіденційності, цілісності та доступності даних під час їх зберігання, обробки та передавання [2].

Під методами захисту інформації розуміють сукупність способів, принципів і технологій, які використовуються для запобігання несанкціонованому доступу до інформації, її модифікації або знищення. Засоби захисту інформації являють собою програмні, апаратні або організаційні рішення, за допомогою яких реалізуються відповідні механізми безпеки [4].

Одним із найбільш поширених методів захисту інформації є автентифікація користувача. Автентифікація використовується для підтвердження особи користувача перед наданням доступу до інформаційної системи. Традиційно для цього застосовуються логіни та паролі. Незважаючи на простоту реалізації та широке використання, парольний захист має значну кількість недоліків. Користувачі часто створюють прості паролі, повторно використовують однакові комбінації або зберігають облікові дані у незахищеному вигляді, що значно знижує рівень безпеки системи [1].

Для оцінювання стійкості парольного захисту використовується поняття ентропії пароля. Ентропія характеризує кількість інформації, необхідної для повного перебору всіх можливих комбінацій символів. Чим більшим є значення ентропії, тим складніше здійснити підбір пароля методом повного підбору:

$$H=L\cdot log_2 N, \quad (1.1)$$

де H – ентропія пароля, біт;

L – довжина пароля;

N – кількість допустимих символів.

Збільшення довжини пароля або розширення набору допустимих символів призводить до експоненційного зростання кількості можливих комбінацій, що суттєво підвищує рівень захисту інформації.

У сучасних умовах дедалі більшого поширення набуває багатофакторна автентифікація. Її основною особливістю є використання декількох способів підтвердження особи користувача. Найчастіше застосовуються такі фактори автентифікації:

– пароль або PIN-код;

– фізичний пристрій;

– біометричні дані користувача.

Використання багатофакторної автентифікації дозволяє значно підвищити рівень захисту інформаційної системи, оскільки навіть у разі компрометації одного з факторів доступ до системи залишається обмеженим [4].

Важливе місце серед методів захисту інформації займають криптографічні методи. Криптографія використовується для забезпечення конфіденційності даних шляхом їх шифрування. Під час шифрування інформація перетворюється у спеціальний формат, який неможливо прочитати без відповідного ключа дешифрування. Навіть у випадку отримання доступу до файлів стороння особа не зможе використати інформацію без криптографічного ключа [3].

Сучасні системи захисту інформації використовують симетричні та асиметричні алгоритми шифрування. У симетричному шифруванні один ключ використовується як для шифрування, так і для дешифрування

інформації. До найбільш відомих алгоритмів такого типу належать AES, DES та Blowfish. Основною перевагою симетричного шифрування є висока швидкість обробки даних, що робить його ефективним для роботи з великими обсягами інформації [5].

Асиметричне шифрування передбачає використання двох ключів – відкритого та закритого. Відкритий ключ використовується для шифрування інформації, а закритий – для її дешифрування. Найбільш поширеним алгоритмом асиметричного шифрування є RSA. Основною перевагою такого підходу є підвищений рівень безпеки під час передавання даних через мережу [5].

Для підвищення ефективності захисту інформації у сучасних інформаційних системах широко використовуються апаратні засоби безпеки. Їх головною особливістю є фізичне зберігання криптографічних ключів або ідентифікаційних даних на окремому пристрої. Використання апаратних засобів дозволяє значно ускладнити процес отримання несанкціонованого доступу до інформації [5].

До найбільш поширених апаратних засобів захисту інформації належать USB-токени, смарт-карти та TPM-модулі. USB-токени використовуються для підтвердження особи користувача та можуть містити унікальні ідентифікатори або криптографічні ключі. Смарт-карти оснащені вбудованими мікропроцесорами та здатні виконувати криптографічні операції безпосередньо на пристрої. TPM-модулі забезпечують апаратний захист ключів шифрування та використовуються у сучасних комп’ютерних системах для підвищення рівня інформаційної безпеки [4].

Окрему роль у забезпеченні захисту інформації відіграють методи контролю доступу. Контроль доступу дозволяє визначати права користувачів щодо виконання певних дій у системі. Серед найбільш поширених моделей контролю доступу виділяють дискреційну, мандатну та рольову моделі. Рольова модель є однією з найбільш популярних у сучасних інформаційних

системах, оскільки дозволяє ефективно розподіляти права доступу відповідно до ролі користувача [1].

Для контролю роботи системи та виявлення підозрілої активності використовуються механізми журналювання подій. Логування дозволяє фіксувати інформацію про входи користувачів, помилки системи, спроби несанкціонованого доступу та інші події, пов’язані з роботою програмного забезпечення. Аналіз журналів подій дає можливість своєчасно виявляти потенційні загрози інформаційній безпеці [5].

Важливим елементом захисту інформації є також резервне копіювання даних. Навіть за наявності ефективних механізмів безпеки існує ризик втрати інформації внаслідок технічних несправностей або дій шкідливого програмного забезпечення. Резервне копіювання дозволяє відновити дані у разі їх пошкодження або втрати та забезпечує стабільність роботи інформаційної системи [2].

У сучасних умовах найбільш ефективними є комплексні системи захисту, які поєднують програмні та апаратні методи безпеки. Використання багатофакторної автентифікації, криптографічного захисту, апаратних засобів контролю доступу та механізмів журналювання подій дозволяє значно підвищити рівень захисту інформаційних систем і мінімізувати ризик несанкціонованого використання даних.

Таким чином, методи та засоби захисту інформації є важливою складовою сучасних інформаційних систем. Використання сучасних технологій безпеки дозволяє забезпечити захист конфіденційної інформації, підвищити надійність системи та зменшити ризики втрати або компрометації даних.

1.3 Апаратні закладні засоби: сутність та принцип роботи

У сучасних інформаційних системах одним із найбільш ефективних способів забезпечення захисту даних є використання апаратних засобів безпеки. На відміну від виключно програмних методів захисту, апаратні засоби дозволяють реалізувати додатковий фізичний рівень контролю доступу до інформації. Особливе місце серед таких засобів займають апаратні закладні засоби, які використовуються для автентифікації користувачів, зберігання криптографічних ключів та забезпечення безпечної роботи інформаційних систем [6].

Під апаратними закладними засобами розуміють спеціалізовані фізичні пристрої або апаратні модулі, які використовуються для реалізації механізмів захисту інформації. Основною особливістю таких засобів є те, що критично важливі дані або функції безпеки зберігаються не у програмному середовищі, а безпосередньо на фізичному пристрої. Це дозволяє значно ускладнити процес отримання несанкціонованого доступу до інформації [2].

Апаратні засоби захисту широко використовуються у сучасних системах інформаційної безпеки завдяки високому рівню надійності. У більшості випадків програмні механізми безпеки можуть бути скомпрометовані за допомогою шкідливого програмного забезпечення, перехоплення даних або експлуатації вразливостей операційної системи. Використання фізичного пристрою як елемента системи автентифікації дозволяє знизити ризик компрометації облікових даних користувача [1].

Одним із найбільш поширених видів апаратних закладних засобів є USB-токени. USB-токен являє собою компактний пристрій, який підключається до комп’ютера через USB-інтерфейс і використовується для підтвердження особи користувача. Такі пристрої можуть містити криптографічні ключі, цифрові сертифікати або унікальні ідентифікатори. Для отримання доступу до інформаційної системи користувач повинен не

лише ввести пароль, але й підключити відповідний пристрій до комп’ютера [5].

Принцип роботи USB-токена полягає у перевірці унікального ідентифікатора пристрою або криптографічного ключа, який зберігається у пам’яті токена. Після підключення пристрою програмний застосунок виконує перевірку його справжності. Якщо перевірка успішна, користувач отримує доступ до системи або захищених даних. У разі відсутності пристрою або невідповідності ідентифікаційних параметрів доступ блокується [4].

Ще одним видом апаратних засобів захисту є смарт-карти. Смарт-карта являє собою пластикову картку з вбудованим мікропроцесором та пам’яттю. На відміну від звичайних карток доступу, смарт-карти здатні виконувати криптографічні операції безпосередньо на пристрої. Це дозволяє використовувати їх для автентифікації користувачів, цифрового підпису та захищеного зберігання інформації [3].

Смарт-карти широко застосовуються у банківських системах, корпоративних мережах та державних інформаційних системах. Основною перевагою таких пристроїв є високий рівень захисту даних, оскільки криптографічні ключі не залишають межі фізичного пристрою. Навіть у разі зараження комп’ютера шкідливим програмним забезпеченням отримати доступ до ключів шифрування значно складніше [7].

Окреме місце серед апаратних засобів захисту займають TPM-модулі. TPM (Trusted Platform Module) – це спеціалізований мікрочип, який встановлюється на материнській платі комп’ютера та використовується для безпечного зберігання криптографічних ключів, паролів і сертифікатів. TPM-модулі забезпечують апаратний захист критично важливої інформації та використовуються для підвищення рівня безпеки операційних систем і програмного забезпечення [6].

Принцип роботи TPM-модуля полягає у створенні ізольованого середовища для виконання криптографічних операцій. Ключі шифрування генеруються та зберігаються безпосередньо всередині модуля, що значно

знижує ризик їх викрадення. TPM активно використовується у сучасних системах захисту даних, зокрема у технологіях шифрування дисків та системах багатофакторної автентифікації [1].

Важливою перевагою апаратних закладних засобів є можливість поєднання їх із програмними методами захисту інформації. У сучасних системах безпеки апаратні пристрої зазвичай використовуються разом із криптографічними алгоритмами, системами контролю доступу та механізмами журналювання подій. Такий комплексний підхід дозволяє значно підвищити ефективність захисту інформації та знизити ризик несанкціонованого доступу [7].

Незважаючи на значні переваги, апаратні засоби захисту мають і певні недоліки. Одним із головних недоліків є необхідність використання додаткового фізичного пристрою, який може бути втрачений або пошкоджений. Крім того, впровадження апаратних засобів безпеки часто потребує додаткових фінансових витрат та спеціалізованого програмного забезпечення для роботи з пристроями [8].

Ще однією особливістю апаратних засобів захисту є необхідність забезпечення сумісності між пристроями та програмним забезпеченням. Для коректної роботи системи необхідно реалізувати механізми взаємодії між застосунком і апаратним пристроєм, а також забезпечити стабільну обробку помилок у разі відключення або пошкодження пристрою.

У рамках даної кваліфікаційної роботи апаратний закладний засіб використовується як елемент системи автентифікації користувача. Основною функцією пристрою є підтвердження права доступу до захищених даних. У разі відсутності пристрою або невідповідності його ідентифікаційних параметрів доступ до інформації блокується. Такий підхід дозволяє реалізувати додатковий рівень захисту інформації та знизити ризик несанкціонованого використання даних.

Таким чином, апаратні закладні засоби є важливою складовою сучасних систем інформаційної безпеки. Використання фізичних пристроїв

для автентифікації та зберігання криптографічної інформації дозволяє значно підвищити рівень захисту даних та забезпечити більш надійний контроль доступу до інформаційних ресурсів.

1.4 Аналіз існуючих рішень у сфері захисту даних

У сучасних умовах розвитку інформаційних технологій існує значна кількість програмних рішень, призначених для захисту даних та забезпечення контролю доступу до інформації. Більшість сучасних систем безпеки поєднують програмні та апаратні механізми захисту, використовуючи криптографічні алгоритми, багатофакторну автентифікацію та фізичні пристрої підтвердження особи користувача. Аналіз існуючих рішень дозволяє визначити найбільш ефективні підходи до реалізації системи захисту даних та врахувати переваги й недоліки сучасних технологій під час розробки власного програмного застосунку [9].

Одним із найбільш відомих програмних рішень у сфері захисту інформації є система BitLocker, яка використовується в операційних системах Windows. Основною функцією BitLocker є шифрування даних на жорсткому диску з метою запобігання несанкціонованому доступу до інформації у разі втрати або викрадення пристрою. Для підвищення рівня безпеки система підтримує використання TPM-модуля, який забезпечує безпечне зберігання криптографічних ключів [9].

Перевагою BitLocker є інтеграція з операційною системою та високий рівень захисту даних. Система автоматично виконує шифрування диска та дозволяє реалізувати багатофакторну автентифікацію. Однак використання BitLocker потребує наявності відповідної версії операційної системи Windows та TPM-модуля. Крім того, система орієнтована переважно на захист диска в цілому, а не окремих даних або програмних застосунків.

Ще одним популярним рішенням є VeraCrypt – програмний засіб для шифрування даних із відкритим вихідним кодом. VeraCrypt дозволяє створювати захищені контейнери для зберігання інформації або виконувати повне шифрування диска. Система підтримує використання різних криптографічних алгоритмів, зокрема AES, Serpent та Twofish [10].

Основною перевагою VeraCrypt є високий рівень криптографічного захисту та гнучкість налаштувань. Користувач може самостійно визначати алгоритми шифрування, рівень захисту та параметри доступу до даних. Водночас система не використовує апаратні засоби автентифікації за замовчуванням, а процес налаштування може бути складним для недосвідчених користувачів.

У корпоративному середовищі широко використовуються USB-токени та системи апаратної автентифікації компанії Yubico. Пристрої YubiKey застосовуються для багатофакторної автентифікації та підтвердження особи користувача під час входу до інформаційних систем. Принцип роботи таких пристроїв базується на генерації одноразових кодів або використанні криптографічних ключів, які зберігаються безпосередньо на фізичному пристрої [11].

Основною перевагою YubiKey є високий рівень захисту від фішингових атак та компрометації паролів. Для отримання доступу користувач повинен фізично підключити пристрій до комп’ютера або підтвердити автентифікацію через NFC-інтерфейс. Проте використання таких рішень часто потребує інтеграції з онлайн-сервісами або корпоративними системами безпеки, що може ускладнювати їх застосування у локальних програмних застосунках.

Ще одним прикладом сучасної системи захисту інформації є Google Titan Security Key. Даний пристрій використовується для багатофакторної автентифікації та забезпечує додатковий рівень безпеки під час роботи з обліковими записами користувачів. Titan Security Key підтримує стандарти

FIDO U2F та FIDO2, які використовуються для безпечної автентифікації без необхідності передавання паролів через мережу [12].

Перевагою апаратних ключів типу Titan Security Key та YubiKey є високий рівень захисту від віддалених атак. Навіть у разі компрометації пароля отримати доступ до системи без фізичного пристрою практично неможливо. Однак такі системи здебільшого орієнтовані на роботу з вебсервісами та хмарними платформами, тоді як у рамках даної кваліфікаційної роботи основна увага приділяється локальному програмному застосунку для захисту даних.

Для локального захисту інформації також використовуються спеціалізовані програмні засоби з підтримкою USB-автентифікації. Одним із прикладів є USB Raptor – програма, що автоматично блокує комп'ютер після відключення визначеного USB-пристрою. Принцип роботи системи полягає у перевірці унікального ідентифікатора USB-носія. У разі відключення пристрою система автоматично обмежує доступ до комп'ютера [13].

Перевагою такого підходу є простота реалізації та можливість використання звичайного USB-накопичувача як засобу автентифікації. Проте подібні системи зазвичай не забезпечують криптографічного захисту інформації та не підтримують розширені механізми контролю доступу.

Аналіз існуючих рішень показує, що більшість сучасних систем захисту інформації використовують комплексний підхід, який поєднує:

- автентифікацію користувача;
- криптографічний захист даних;
- використання апаратних засобів безпеки;
- контроль доступу до інформації;
- журналювання подій системи.

Саме такий підхід був покладений в основу технічного завдання розроблюваного програмного застосунку. У межах кваліфікаційної роботи передбачається створення локального програмного застосунку для

операційної системи Windows із підтримкою апаратної автентифікації через USB-пристрій, механізмів шифрування інформації, контролю доступу та автоматичного блокування системи у разі відключення апаратного ключа.

На відміну від існуючих рішень, розроблюваний застосунок орієнтований на поєднання відносно простої реалізації з достатнім рівнем захисту даних. Для реалізації системи планується використання програмних засобів шифрування, локальної бази даних та механізму перевірки унікального ідентифікатора USB-пристрою. Такий підхід дозволить забезпечити захист інформації без необхідності використання дорогого спеціалізованого обладнання.

Таким чином, аналіз сучасних рішень у сфері захисту даних показав, що найбільш ефективними є системи, які поєднують програмні та апаратні методи безпеки. Використання апаратних засобів автентифікації, криптографічного захисту та механізмів контролю доступу дозволяє значно підвищити рівень інформаційної безпеки та знизити ризики несанкціонованого використання даних.

1.5 Постановка задачі

У результаті проведеного аналізу сучасних загроз інформаційній безпеці, методів захисту інформації та існуючих програмних рішень було встановлено, що традиційні механізми автентифікації не забезпечують достатнього рівня захисту даних. Використання лише логіна та пароля не дозволяє ефективно протидіяти сучасним методам несанкціонованого доступу, таким як підбір паролів, фішингові атаки або використання шкідливого програмного забезпечення. У зв’язку з цим виникає необхідність розробки програмного застосунку, який поєднуватиме програмні механізми захисту інформації з апаратними засобами автентифікації.

Основною метою кваліфікайіної роботи є розробка програмного застосунку захисту даних від несанкціонованого використання із застосуванням апаратних закладних засобів. Розроблювана система повинна забезпечувати контроль доступу до інформації шляхом використання фізичного пристрою автентифікації та механізмів криптографічного захисту даних.

У межах кваліфікаційної роботи передбачається створення локального програмного застосунку для операційної системи Windows. Основним принципом роботи системи є перевірка наявності визначеного USB-пристрою, який використовується як апаратний ключ доступу до інформації. Доступ до функціоналу застосунку та захищених даних повинен надаватися лише після успішної перевірки апаратного ключа та проходження процедури автентифікації користувача.

Однією з ключових задач є реалізація механізму ідентифікації USB-пристрою за його унікальними параметрами. Після запуску програмного застосунку система повинна автоматично виконувати перевірку наявності відповідного пристрою. У разі відсутності ключа або невідповідності його ідентифікаційних характеристик доступ до інформації має бути заблокований.

Крім апаратної автентифікації, система повинна забезпечувати додатковий рівень захисту інформації за допомогою криптографічних методів. Для цього передбачається реалізація механізмів шифрування та дешифрування даних із використанням сучасних криптографічних алгоритмів. Захищена інформація повинна зберігатися у зашифрованому вигляді, що дозволить унеможливити її використання сторонніми особами навіть у разі отримання фізичного доступу до файлів.

У рамках кваліфікаційної роботи також необхідно реалізувати механізм автоматичного блокування доступу до системи у разі відключення апаратного ключа під час роботи застосунку. Такий підхід дозволить

забезпечити додатковий рівень безпеки та запобігти несанкціонованому використанню даних у разі залишення робочого місця користувачем.

Для забезпечення контролю роботи системи передбачається реалізація журналювання подій. Програмний застосунок повинен фіксувати інформацію про входи користувачів, спроби несанкціонованого доступу, помилки автентифікації та інші події, пов’язані з роботою системи. Аналіз журналів подій дозволить контролювати роботу застосунку та своєчасно виявляти потенційні загрози безпеці.

Під час розробки програмного застосунку необхідно забезпечити:

- реалізацію механізму автентифікації користувача;
- перевірку наявності апаратного ключа;
- шифрування та дешифрування інформації;
- контроль доступу до захищених даних;
- автоматичне блокування системи у разі відключення USB-пристрою;

Для реалізації програмного застосунку планується використання мови програмування PYTHON та локальної бази даних SQLite. Вибір PYTHON обумовлений простотою реалізації програмної логіки, наявністю бібліотек для роботи з криптографією та підтримкою взаємодії з USB-пристроями. SQLite використовується як компактна локальна база даних для зберігання службової інформації та параметрів системи.

У результаті виконання кваліфікайної роботи має бути створений працездатний програмний застосунок, який забезпечуватиме захист конфіденційної інформації від несанкціонованого використання шляхом поєднання програмних механізмів безпеки з апаратними засобами автентифікації.

Таким чином, постановка задачі кваліфікаційної роботи полягає у створенні програмного рішення для забезпечення захисту даних із використанням апаратного USB-ключа, криптографічного захисту інформації

та механізмів контролю доступу, що дозволить підвищити рівень інформаційної безпеки локальних інформаційних систем.

2 АНАЛІЗ ТА ПРОЄКТУВАННЯ СИСТЕМИ ЗАХИСТУ ДАНИХ

2.1 Вибір підходу до реалізації захисту

Під час розробки системи захисту даних одним із найважливіших етапів є вибір підходу до реалізації механізмів безпеки. Ефективність захисту інформації значною мірою залежить від правильного поєднання програмних та апаратних засобів, методів автентифікації, криптографічних алгоритмів і способів контролю доступу до даних. У рамках даної кваліфікаційної роботи основна увага приділяється створенню локального програмного застосунку, який забезпечуватиме захист інформації від несанкціонованого використання за допомогою апаратного USB-ключа та механізмів шифрування даних.

На початковому етапі проєктування було проаналізовано декілька можливих підходів до реалізації системи захисту інформації. Першим варіантом було використання виключно парольної автентифікації. Такий підхід є найпростішим у реалізації та не потребує використання додаткових пристроїв. Проте парольний захист має значну кількість недоліків, пов’язаних із ризиком компрометації облікових даних користувача. Паролі можуть бути викрадені за допомогою шкідливого програмного забезпечення, фішингових атак або методів соціальної інженерії, що значно знижує рівень безпеки системи [14].

Іншим варіантом було використання виключно криптографічного захисту інформації без додаткових механізмів автентифікації. У такому випадку дані зберігалися б у зашифрованому вигляді, а доступ до них здійснювався б за допомогою ключа дешифрування. Незважаючи на високий рівень захисту інформації, такий підхід також має певні недоліки. У разі отримання зловмисником доступу до ключа дешифрування система втрачає ефективність захисту [5].

У результаті аналізу було визначено, що найбільш ефективним рішенням є комплексний підхід, який поєднує:

– апаратну автентифікацію користувача;

– криптографічний захист даних;

– контроль доступу до інформації;

– журналювання подій системи.

Саме такий підхід дозволяє реалізувати багаторівневу систему безпеки, у якій компрометація одного механізму захисту не призводить до повного порушення безпеки системи.

У межах кваліфікаційної роботи як апаратний засіб автентифікації обрано USB-пристрій. Вибір саме такого рішення пояснюється простотою реалізації, доступністю обладнання та можливістю взаємодії з операційною системою без використання дорогих спеціалізованих пристроїв. USB-пристрій використовується як фізичний ключ доступу до програмного застосунку. Для отримання доступу до інформації користувач повинен підключити відповідний пристрій до комп’ютера та пройти процедуру автентифікації.

Основним принципом роботи системи є перевірка унікального ідентифікатора USB-пристрою. Після запуску застосунку система автоматично визначає наявність пристрою та виконує його ідентифікацію. Якщо пристрій відповідає збереженим параметрам, користувач отримує доступ до функціоналу системи. У разі відсутності пристрою або невідповідності його характеристик доступ до інформації блокується.

Такий підхід дозволяє значно підвищити рівень безпеки порівняно зі звичайною парольної автентифікацією. Навіть у разі компрометації пароля стороння особа не зможе отримати доступ до інформації без фізичного USB-ключа. Крім того, використання фізичного пристрою ускладнює реалізацію віддалених атак на систему [7].

Для забезпечення додаткового рівня захисту у системі передбачено використання криптографічних алгоритмів шифрування даних. У рамках проєкту планується використання симетричного алгоритму AES, який є одним із найбільш поширених сучасних алгоритмів шифрування інформації. Основними перевагами AES є висока швидкість роботи, ефективність та достатній рівень криптографічної стійкості [4].

Основою симетричного шифрування є математичне перетворення відкритого тексту у зашифровані дані із використанням секретного ключа.

У розробленому програмному застосунку саме така схема використовується під час роботи алгоритму AES-256-GCM, що забезпечує конфіденційність інформації, яка зберігається у локальній базі даних.

Захищені дані у системі повинні зберігатися лише у зашифрованому вигляді.

Відновлення інформації після успішної автентифікації користувача здійснюється відповідно до формули:

$$P = D_k(C), \tag{2.1}$$

де D – функція дешифрування;

K – секретний ключ;

C – зашифровані дані;

P – відновлений відкритий текст.

Таким чином, навіть якщо стороння особа отримає доступ до бази даних, вона не зможе прочитати її вміст без відповідного криптографічного ключа. Дешифрування виконується лише після успішної автентифікації користувача та підтвердження наявності зареєстрованого USB-пристрою.

Дешифрування інформації виконується тільки після успішної перевірки USB-пристрою та проходження автентифікації користувача. Такий підхід дозволяє забезпечити захист інформації навіть у разі отримання фізичного доступу до файлів системи.

Однією з важливих особливостей обраного підходу є реалізація автоматичного блокування системи у разі відключення USB-пристрою під час роботи застосунку. Програмний застосунок постійно контролює наявність апаратного ключа. Якщо пристрій відключається, система автоматично завершує доступ до захищених даних та блокує роботу користувача. Це дозволяє запобігти несанкціонованому використанню інформації у випадках, коли користувач залишає робоче місце без завершення роботи із системою.

У процесі вибору підходу до реалізації захисту також було враховано необхідність забезпечення простоти використання системи. Надмірно складні механізми безпеки можуть створювати труднощі для користувачів та знижувати ефективність роботи із застосунком. Саме тому у межах кваліфікаційної роботи обрано рішення, яке поєднує достатній рівень безпеки з відносно простою реалізацією та зручністю використання.

Для реалізації програмного застосунку було обрано мову програмування PYTHON. Основними причинами такого вибору є наявність бібліотек для роботи з криптографією, підтримка взаємодії з USB-пристроями та простота реалізації програмної логіки. Для зберігання службових даних та параметрів системи планується використання локальної бази даних SQLite.

Таким чином, у результаті аналізу можливих підходів до реалізації системи захисту було обрано комплексне рішення, яке поєднує апаратну автентифікацію через USB-пристрій, криптографічний захист інформації та механізми контролю доступу. Обраний підхід дозволяє забезпечити достатній рівень захисту даних, простоту реалізації та зручність використання програмного застосунку.

2.2 Розробка архітектури програмного застосунку

Під час розробки програмного застосунку захисту даних важливим етапом є проєктування архітектури системи. Архітектура визначає структуру програмного забезпечення, взаємодію окремих компонентів, логіку роботи системи та способи обробки інформації. Правильно спроєктована архітектура забезпечує стабільність роботи застосунку, зручність подальшого супроводу та можливість розширення функціоналу в майбутньому [14].

У межах кваліфікаційної роботи було прийнято рішення реалізувати локальний програмний застосунок для операційної системи Windows. Основною причиною вибору локальної архітектури є необхідність забезпечення роботи системи без постійного підключення до мережі та мінімізація ризиків, пов’язаних із передаванням конфіденційної інформації через інтернет. Усі основні операції із захисту даних, автентифікації користувача та шифрування інформації виконуються безпосередньо на комп’ютері користувача.

Архітектура програмного застосунку побудована за модульним принципом. Такий підхід дозволяє розділити функціонал системи на окремі логічні компоненти, кожен із яких відповідає за виконання визначених задач. Модульна структура спрощує процес розробки програмного забезпечення, полегшує тестування окремих компонентів та дозволяє вносити зміни до системи без необхідності повного перепроєктування застосунку [15].

У структурі програмного застосунку можна виділити декілька основних модулів:

- модуль автентифікації;
- модуль роботи з USB-пристроєм;
- модуль шифрування та дешифрування даних;
- модуль управління даними;
- модуль журналювання подій;

– модуль користувацького інтерфейсу.

Основним елементом системи є модуль автентифікації користувача. Його основною функцією є перевірка права доступу до програмного застосунку. Після запуску системи модуль виконує перевірку наявності USB-пристрою та здійснює його ідентифікацію за унікальними параметрами. У разі успішної перевірки користувачу надається доступ до функціоналу системи. Якщо пристрій відсутній або його параметри не відповідають збереженим значенням, доступ до інформації блокується.

Для реалізації взаємодії із фізичним пристроєм використовується окремий модуль роботи з USB-пристроєм. Основним завданням цього модуля є визначення підключених USB-носіїв, зчитування їхніх ідентифікаційних параметрів та передавання отриманих даних до модуля автентифікації. Крім цього, модуль здійснює постійний контроль наявності пристрою під час роботи системи.

Однією з важливих функцій системи є автоматичне блокування доступу у разі відключення USB-пристрою. Для реалізації цього механізму програмний застосунок у фоновому режимі перевіряє наявність апаратного ключа. Якщо пристрій відключається під час роботи користувача із захищеними даними, система автоматично завершує доступ до інформації та переводить застосунок у заблокований стан.

Для забезпечення захисту інформації в архітектурі системи передбачено модуль шифрування та дешифрування даних. Основною функцією цього модуля є перетворення інформації у зашифрований вигляд перед її збереженням та відновлення даних після успішної автентифікації користувача. У межах кваліфікаційної роботи планується використання симетричного алгоритму AES, який забезпечує достатній рівень криптографічного захисту та високу швидкість роботи [4].

Модуль управління даними відповідає за створення, редагування, перегляд та видалення інформації у системі. Доступ до функцій модуля можливий лише після проходження процедури автентифікації та

підтвердження наявності USB-ключа. Усі дані, що зберігаються у системі, перед записом проходять процедуру шифрування.

Для контролю роботи системи та фіксації дій користувачів використовується модуль журналювання подій. У журналі подій зберігається інформація про успішні входи до системи, спроби несанкціонованого доступу, помилки автентифікації, відключення USB-пристрою та інші події, пов’язані з роботою застосунку. Наявність журналу дозволяє контролювати роботу системи та аналізувати можливі загрози інформаційній безпеці [1].

Окрему увагу під час проєктування архітектури було приділено користувацькому інтерфейсу. Інтерфейс програмного застосунку повинен бути простим, зрозумілим та зручним для користувача. Основними елементами інтерфейсу є форма авторизації, головне вікно застосунку, індикатор підключення USB-пристрою та повідомлення про помилки або блокування системи.

У рамках кваліфікаційної роботи для реалізації програмного застосунку обрано мову програмування Python. Вибір даної мови зумовлений простотою розробки, високою продуктивністю, ефективною роботою з системними ресурсами, а також наявністю сучасних бібліотек для реалізації криптографічного захисту інформації, взаємодії з файловою системою та USB-пристроями. Крім того, Python забезпечує можливість створення кросплатформних програмних продуктів, що дає змогу використовувати розроблений застосунок на різних операційних системах без суттєвих змін програмного коду. Для реалізації графічного інтерфейсу може бути використано бібліотеки Tkinter або PyQt, а для зберігання службових даних і параметрів системи планується застосування локальної бази даних SQLite.

З метою зберігання службової інформації, параметрів конфігурації, облікових записів користувачів, журналу подій та зашифрованих записів обрано вбудовану реляційну систему керування базами даних SQLite. Використання SQLite не потребує встановлення окремого серверного програмного забезпечення, характеризується високою швидкістю роботи,

надійністю та мінімальними вимогами до обчислювальних ресурсів, що робить її оптимальним рішенням для автономного програмного застосунку.

Для реалізації криптографічного захисту даних використовуються стандартні криптографічні засоби мови Python, які забезпечують підтримку сучасних алгоритмів симетричного шифрування, генерації криптографічно стійких ключів та безпечного керування ними. Застосування вбудованих бібліотек дозволяє уникнути використання сторонніх програмних компонентів та забезпечує відповідність сучасним вимогам інформаційної безпеки.

Взаємодія програмного застосунку з апаратним USB-ключем реалізується шляхом отримання унікального ідентифікатора підключеного пристрою, його порівняння із зареєстрованим у базі даних значенням та постійного контролю стану підключення протягом роботи системи. У разі відключення або заміни USB-пристрою застосунок автоматично завершує активний сеанс користувача, блокує доступ до конфіденційних даних та переводить систему у режим очікування повторної автентифікації.

Таким чином, розроблена архітектура програмного застосунку забезпечує реалізацію основних функцій захисту даних, підтримує взаємодію з апаратним USB-ключем та дозволяє реалізувати механізми контролю доступу, шифрування інформації та журналювання подій. Модульний принцип побудови системи забезпечує гнучкість архітектури, спрощує процес розробки та створює можливість подальшого розширення функціоналу програмного застосунку.

Архітектура програмного застосунку захисту даних наведена на рисунку 2.1.

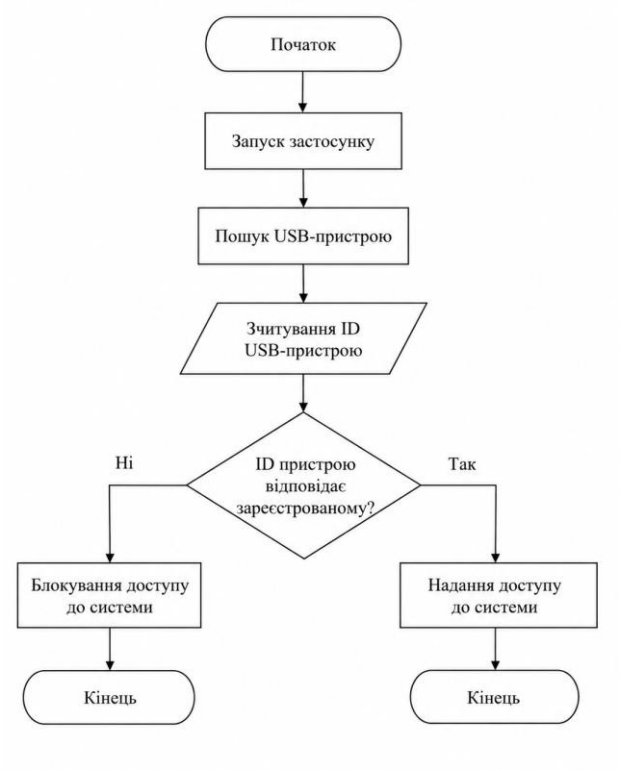


Рисунок 2.1 – Архітектура програмного застосунку захисту даних

2.3 Моделювання процесів роботи системи

Під час розробки програмного застосунку захисту даних важливим етапом є моделювання процесів роботи системи. Моделювання дозволяє визначити послідовність виконання основних операцій, взаємодію окремих компонентів застосунку та логіку роботи механізмів захисту інформації. Використання моделей і алгоритмів дає можливість ще на етапі проєктування виявити можливі недоліки архітектури системи та оптимізувати роботу програмного забезпечення [3].

У межах кваліфікаційної роботи основна увага приділяється моделюванню процесів автентифікації користувача, перевірки USB-пристрою, доступу до захищених даних та автоматичного блокування

системи у разі відключення апаратного ключа. Усі процеси реалізуються у межах локального програмного застосунку для операційної системи Windows.

Після запуску програмного застосунку система переходить до початкового етапу перевірки. Основним завданням на цьому етапі є визначення наявності підключеного USB-пристрою, який використовується як апаратний ключ доступу до інформації. Програмний застосунок виконує пошук USB-пристроїв, підключених до комп’ютера, та отримує їхні ідентифікаційні параметри.

Після отримання інформації про пристрій система порівнює його параметри із даними, збереженими у конфігурації застосунку. У разі успішного збігу система дозволяє перейти до наступного етапу автентифікації. Якщо пристрій відсутній або його ідентифікатор не відповідає зареєстрованому значенню, доступ до функціоналу застосунку блокується.

Після успішної перевірки USB-пристрою система переходить до етапу автентифікації користувача. У межах проєкту передбачено використання пароля як додаткового рівня захисту. Користувач вводить пароль доступу до системи, після чого програмний застосунок виконує перевірку правильності введених даних.

У разіуспішної автентифікації користувачу відкривається доступ до основного функціоналу системи. Якщо пароль введено неправильно, система виводить повідомлення про помилку та блокує доступ до інформації. Для підвищення рівня безпеки може бути реалізоване обмеження кількості спроб введення пароля.

Алгоритм перевірки USB-пристрою наведено на рисунку 2.2.

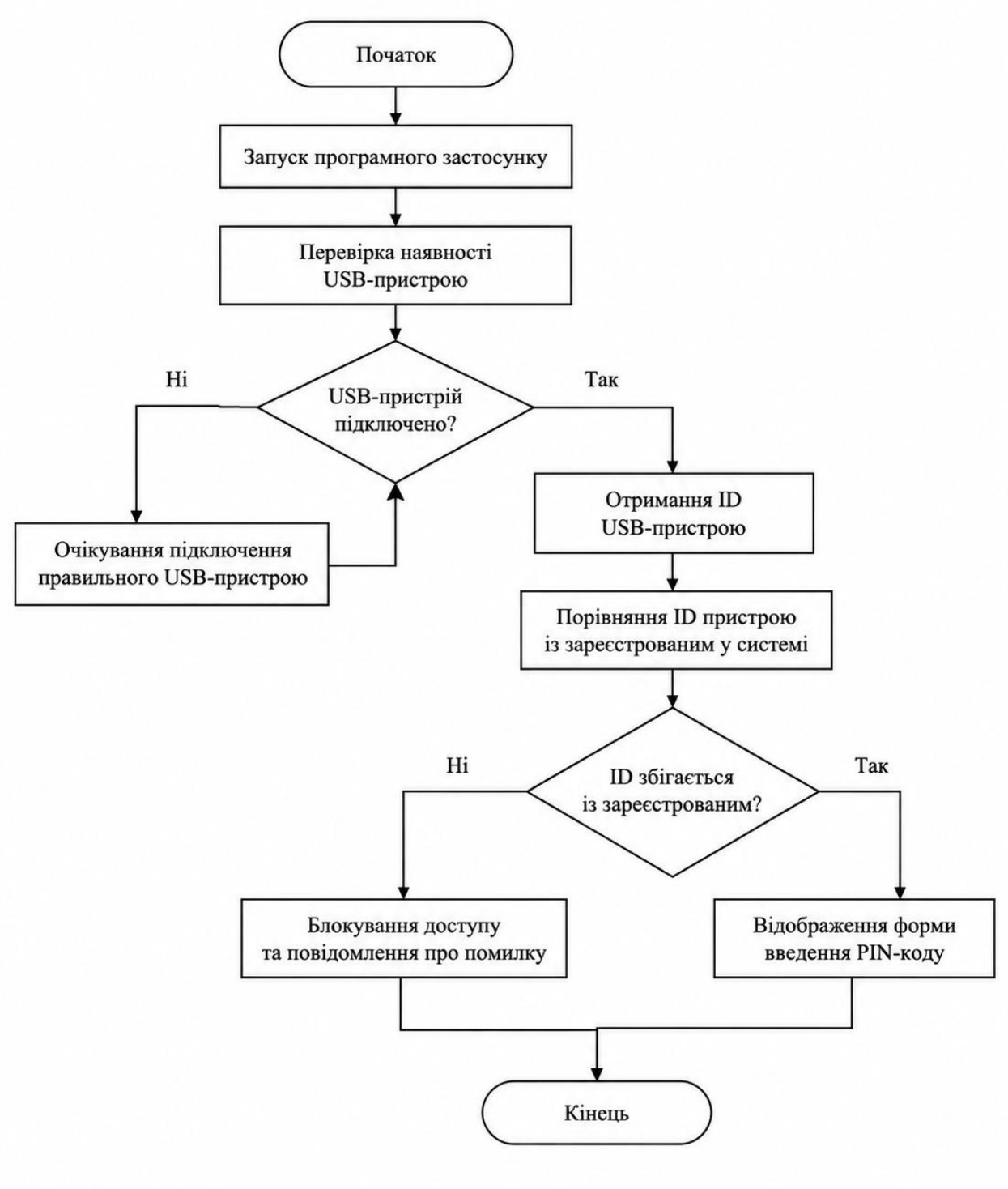


Рисунок 2.2 – Алгоритм перевірки USB-пристрою

Однією з ключових функцій системи є захист інформації шляхом шифрування даних. Усі файли або записи, що зберігаються у системі, перед

записом проходять процедуру шифрування. Для цього використовується симетричний алгоритм AES, який забезпечує достатній рівень криптографічного захисту та високу швидкість обробки інформації [6].

Процес роботи із захищеними даними включає такі етапи:

– отримання даних від користувача;
– шифрування інформації;
– збереження зашифрованих даних;
– дешифрування інформації після успішної автентифікації;
– відображення інформації користувачу.

Важливим елементом системи є механізм постійного контролю підключення USB-пристрою. Після успішної автентифікації застосунок у фоновому режимі продовжує перевіряти наявність апаратного ключа. Якщо пристрій відключається під час роботи користувача, система автоматично завершує доступ до інформації та переводить застосунок у заблокований режим.

Такий механізм дозволяє забезпечити додатковий рівень захисту інформації у випадках, коли користувач залишає робоче місце без завершення роботи із системою. Навіть у разі фізичного доступу до комп’ютера стороння особа не зможе отримати доступ до захищених даних без повторного підключення USB-пристрою та проходження автентифікації.

Для контролю роботи системи передбачено механізм журналювання подій. Усі основні дії користувача та системні події фіксуються у журналі. До таких подій належать:

– запуск застосунку;
– успішна автентифікація;
– помилки введення пароля;
– відключення USB-пристрою;
– спроби несанкціонованого доступу;
– завершення роботи системи.

Алгоритм автоматичного блокування системи наведено на рисунку 2.3.

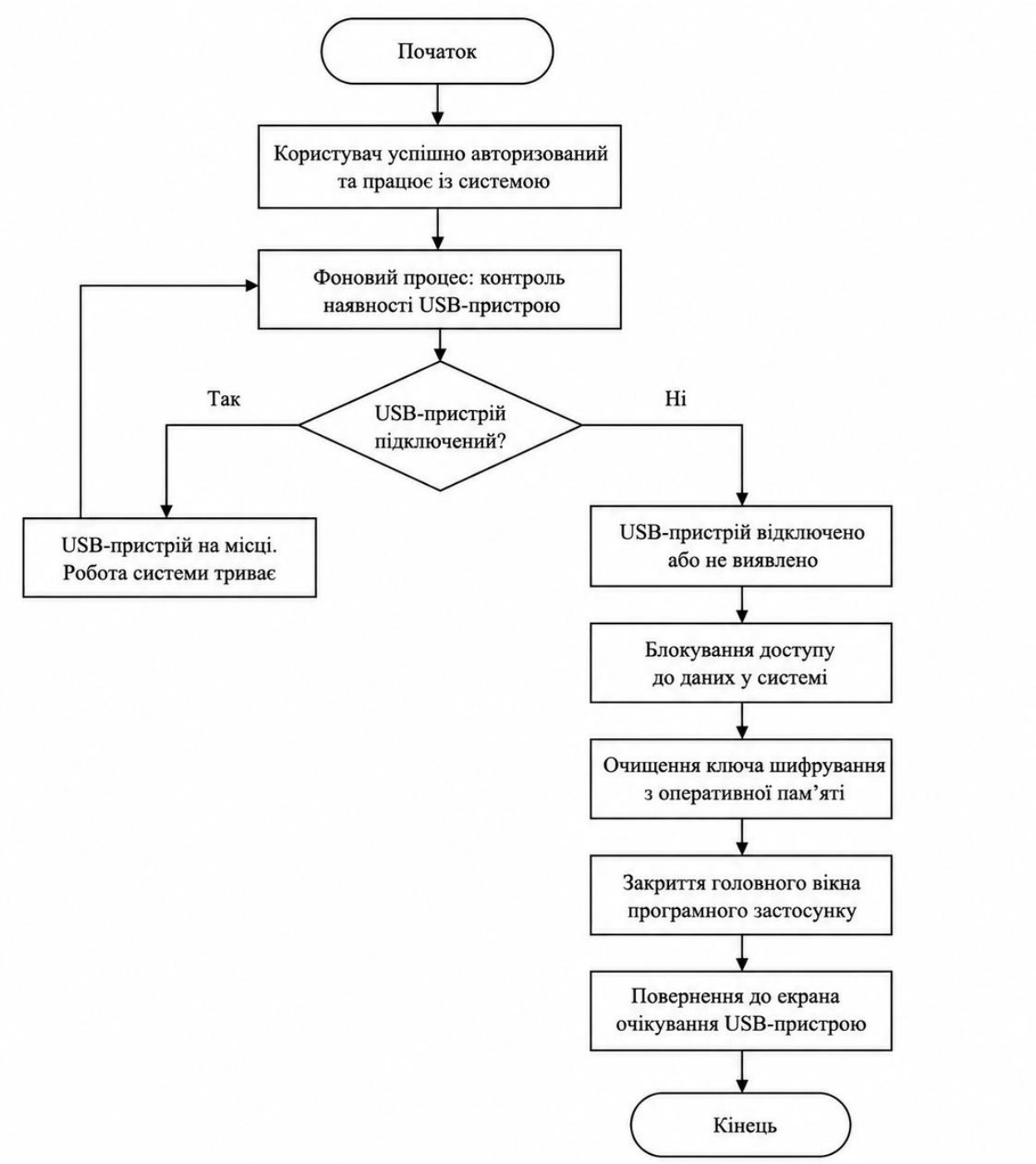


Рисунок 2.3 – Алгоритм автоматичного блокування системи

Журналювання подій дозволяє контролювати роботу програмного застосунку та своєчасно виявляти можливі порушення безпеки. Для реалізації журналу подій у PYTHON може використовуватися стандартний

модуль logging, який забезпечує запис інформації у текстові файли або базу даних.

Під час моделювання процесів роботи системи також було враховано необхідність забезпечення стабільної роботи застосунку у випадку виникнення помилок. У разі некоректного відключення USB-пристрою, помилки дешифрування або збою під час роботи програми система повинна коректно обробляти помилки та повідомляти користувача про причини блокування доступу.

Таким чином, моделювання процесів роботи системи дозволило визначити основні алгоритми функціонування програмного застосунку, логіку взаємодії між компонентами системи та послідовність виконання операцій захисту інформації. Реалізація описаних алгоритмів забезпечить контроль доступу до даних, підтримку апаратної автентифікації та захист інформації від несанкціонованого використання.

2.4 Вибір технологій та інструментів реалізації

Одним із важливих етапів розробки програмного застосунку є вибір технологій та програмних інструментів, які будуть використовуватися для реалізації функціоналу системи. Від правильності вибору мови програмування, середовища розробки, бібліотек та допоміжних технологій залежить стабільність роботи програмного забезпечення, швидкість розробки та можливість подальшого вдосконалення системи [8].

У межах кваліфікаційної роботи для реалізації програмного застосунку було обрано мову програмування PYTHON. Дане рішення обумовлене рядом переваг, серед яких простота синтаксису, велика кількість готових бібліотек, підтримка роботи з криптографією та можливість взаємодії з апаратними USB-пристроями. PYTHON є однією з найбільш популярних мов

програмування у сфері розробки програмного забезпечення, автоматизації процесів та створення систем інформаційної безпеки [15].

Однією з головних переваг PYTHON є наявність великої кількості бібліотек для реалізації криптографічного захисту інформації. Для шифрування та дешифрування даних у межах кваліфікаційної роботи планується використання бібліотеки PyCryptodome. Дана бібліотека підтримує сучасні криптографічні алгоритми, зокрема AES, RSA та інші методи захисту інформації. Використання готових криптографічних бібліотек дозволяє значно спростити реалізацію системи шифрування та підвищити надійність програмного застосунку [4].

Для реалізації графічного інтерфейсу користувача було обрано бібліотеку Tkinter. Tkinter входить до стандартного набору бібліотек PYTHON та дозволяє створювати desktop-застосунки для операційної системи Windows без використання додаткового програмного забезпечення. Основними перевагами Tkinter є простота використання, достатній набір елементів інтерфейсу та сумісність із різними версіями PYTHON.

Графічний інтерфейс застосунку повинен забезпечувати:

– форму авторизації користувача;
– відображення статусу USB-пристрою;
– доступ до функцій роботи із захищеними даними;
– повідомлення про помилки та блокування системи;
– перегляд журналу подій.

Для роботи з USB-пристроями у PYTHON можуть використовуватися бібліотеки pywin32, psutil або засоби Windows API. Основним завданням відповідного модуля є визначення підключених USB-пристроїв, отримання їхніх ідентифікаційних параметрів та контроль наявності апаратного ключа під час роботи системи.

У межах кваліфікаційної роботи USB-пристрій використовується як апаратний ключ автентифікації. Після запуску програмного застосунку

система виконує перевірку підключених USB-носіїв та порівнює отримані параметри із зареєстрованими даними. Такий підхід дозволяє реалізувати додатковий рівень захисту інформації без використання складних спеціалізованих пристроїв.

Для зберігання службової інформації та параметрів системи було обрано локальну базу даних SQLite. SQLite є компактною вбудованою системою керування базами даних, яка не потребує встановлення окремого серверного програмного забезпечення. Дана база даних добре підходить для локальних desktop-застосунків та забезпечує достатню швидкість роботи для потреб проєкту [16].

У базі даних планується зберігати:

– інформацію про зареєстрований USB-пристрій;
– параметри системи;
– журнал подій;
– службову інформацію користувача.

Однією з переваг SQLite є простота інтеграції з PYTHON через стандартний модуль sqlite3, який входить до базового набору бібліотек мови програмування.

Для реалізації механізму журналювання подій використовується стандартний модуль logging. Даний модуль дозволяє автоматично фіксувати системні події у текстових файлах або базі даних. Журналювання необхідне для контролю роботи системи та виявлення спроб несанкціонованого доступу до інформації [1].

Під час розробки програмного застосунку також планується використання інтегрованого середовища розробки Visual Studio Code. Дане середовище підтримує роботу з PYTHON, має вбудовані інструменти налагодження програмного коду та забезпечує зручну організацію структури проєкту.

Таблиця 2.1 – Технології та інструменти реалізації програмного застосунку

Технологія / інструмент	**Призначення**	**Використання у проєкті**
1	**2**	**3**
PYTHON	Основна мова програмування	Реалізація логіки програмного застосунку
Tkinter	Бібліотека для створення графічного інтерфейсу	Реалізація вікон авторизації та роботи із системою
SQLite	Локальна система керування базами даних	Зберігання службової інформації та журналу подій
PyCryptodome	Криптографічна бібліотека	Реалізація шифрування та дешифрування даних
logging	Стандартний модуль PYTHON	Журналювання системних подій
Visual Studio Code	Середовище розробки	Написання та налагодження програмного коду
USB-пристрій	Апаратний засіб автентифікації	Підтвердження доступу до системи
Windows 10/11	Операційна система	Середовище виконання програмного застосунку
AES	Алгоритм симетричного шифрування	Захист конфіденційної інформації
pywin32	Бібліотека для взаємодії з Windows API	Робота з USB-пристроями та системними функціями

Під час вибору технологій також враховувалась необхідність забезпечення простоти реалізації та відповідності функціоналу

кваліфікаційної роботи. Усі обрані інструменти є достатньо поширеними, мають офіційну документацію та активно використовуються у сучасній розробці програмного забезпечення.

Таким чином, для реалізації програмного застосунку було обрано набір технологій та інструментів, які дозволяють забезпечити реалізацію основного функціоналу системи, підтримку роботи з USB-пристроями, шифрування інформації та створення зручного графічного інтерфейсу користувача. Використання PYTHON, SQLite та сучасних бібліотек забезпечує достатню гнучкість системи, спрощує процес розробки та дозволяє реалізувати всі функції, передбачені технічним завданням кваліфікаційної роботи.

2.5 Проєктування бази даних та структури зберігання інформації

Одним із важливих етапів розробки програмного застосунку захисту даних є проєктування структури бази даних. Від правильності організації зберігання інформації залежить швидкодія системи, надійність зберігання даних, можливість контролю доступу та ефективність реалізації механізмів захисту інформації. У межах кваліфікаційної роботи для зберігання службових даних було обрано систему керування базами даних SQLite, яка забезпечує простоту інтеграції із застосунком, не потребує окремого серверного програмного забезпечення та добре підходить для локальних програмних систем [6].

База даних програмного застосунку Secure Vault використовується для зберігання інформації про користувачів системи, параметри автентифікації, захищені записи, журнал подій та дані про блокування доступу після невдалих спроб входу. Для реалізації цих функцій було створено чотири основні таблиці: users, records, event_log та lockout_state.

Структура бази даних програмного застосунку наведена на рисунку 2.4.

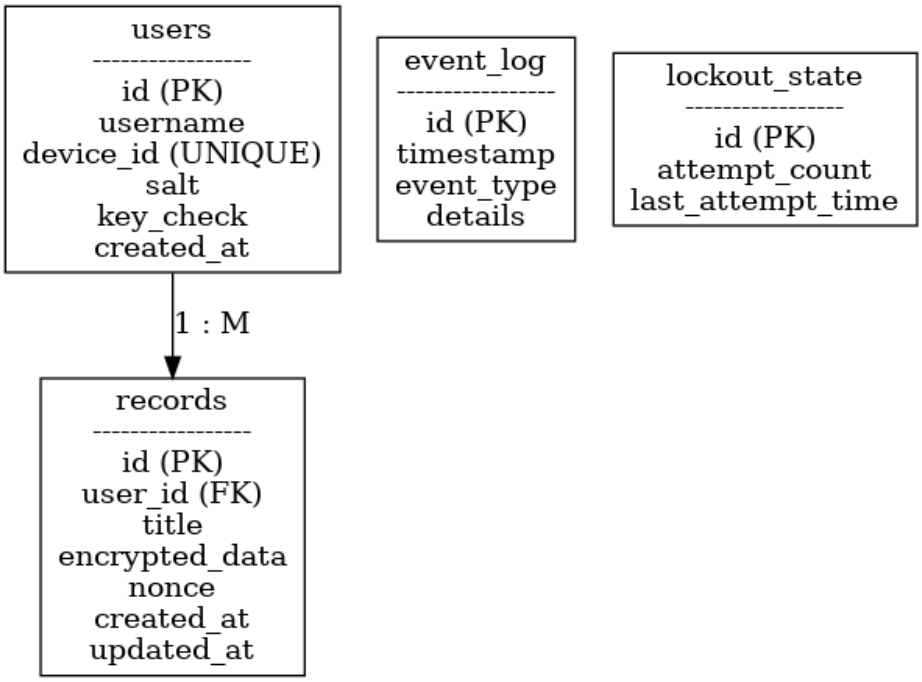


Рисунок 2.4 – Структура бази даних програмного застосунку Secure Vault

Центральним елементом бази даних є таблиця users, яка використовується для зберігання інформації про зареєстрованих користувачів системи та параметри їхньої автентифікації.

Таблиця 2.2 – Структура таблиці users

Поле	**Тип даних**	**Призначення**
id	INTEGER	Унікальний ідентифікатор користувача
username	TEXT	Ім’я користувача
device_id	TEXT	Унікальний ідентифікатор USB-пристрою
salt	BLOB	Випадкова послідовність для генерації ключа
key_check	BLOB	Контрольний маркер перевірки ключа
created_at	TEXT	Дата створення запису

На відміну від класичних систем автентифікації, у розробленому застосунку пароль користувача не зберігається у базі даних. Для перевірки правильності введеного PIN-коду використовується спеціальний контрольний маркер key_check та випадкова послідовність salt. Такий підхід підвищує рівень безпеки системи та унеможливлює отримання пароля навіть у разі компрометації бази даних [17].

Для зберігання захищених записів користувача використовується таблиця records. Саме в ній містяться всі зашифровані дані, які створює користувач під час роботи із застосунком.

Таблиця 2.3 – Структура таблиці records

Поле	**Тип даних**	**Призначення**
1	**2**	**3**
id	INTEGER	Унікальний номер запису
user_id	INTEGER	Посилання на користувача
title	TEXT	Назва запису
encrypted_data	BLOB	Зашифрований вміст запису
nonce	BLOB	Випадковий криптографічний параметр
created_at	TEXT	Дата створення запису
updated_at	TEXT	Дата останнього редагування

Поле encrypted_data містить інформацію у зашифрованому вигляді. Для захисту даних використовується алгоритм AES-256-GCM, який забезпечує конфіденційність та контроль цілісності інформації. Поле nonce використовується як додатковий параметр шифрування та необхідне для коректної роботи криптографічного алгоритму [12].

Для контролю роботи програмного застосунку використовується таблиця журналу подій event_log.

Таблиця 2.4 – Структура таблиці event_log

Поле	**Тип даних**	**Призначення**
Id	INTEGER	Унікальний номер події
timestamp	TEXT	Час виникнення події
event_type	TEXT	Тип події
details	TEXT	Додаткова інформація про подію

У журналі подій автоматично фіксуються всі важливі дії користувача та системні події. До них належать успішні входи до системи, помилки автентифікації, створення та видалення записів, підключення або відключення USB-пристроїв, а також інші події, пов’язані з функціонуванням програмного забезпечення [20].

Для захисту системи від перебору PIN-кодів реалізовано окрему таблицю lockout_state.

Таблиця 2.5 – Структура таблиці lockout_state

Поле	**Тип даних**	**Призначення**
id	INTEGER	Ідентифікатор запису
attempt_count	INTEGER	Кількість невдалих спроб входу
last_attempt_time	TEXT	Час останньої невдалої спроби

Таблиця lockout_state містить інформацію про кількість помилкових спроб автентифікації. Після досягнення встановленого порогу система автоматично блокує можливість введення PIN-коду на визначений проміжок часу. Такий механізм дозволяє ефективно протидіяти атакам перебором паролів та підвищує рівень захисту інформації [18].

Під час проєктування бази даних було реалізовано зв’язок між таблицями users та records. Один користувач може мати необмежену

кількість захищених записів, тоді як кожен запис належить лише одному користувачеві. Такий зв’язок реалізовано за допомогою зовнішнього ключа user_id.

Для підвищення продуктивності роботи бази даних створено індекси для пошуку записів користувача та сортування інформації. Використання індексів дозволяє зменшити час виконання запитів та підвищити швидкодію застосунку під час роботи з великими обсягами інформації [19].

Таким чином, розроблена структура бази даних забезпечує зберігання всієї необхідної службової інформації, підтримує механізми автентифікації користувачів, захисту даних та журналювання подій. Використання SQLite дозволяє створити компактне та ефективне рішення для локального програмного застосунку захисту інформації, а застосування криптографічних механізмів підвищує рівень безпеки збережених даних.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Реалізація основного функціоналу програмного застосунку

У межах кваліфікаційної роботи було реалізовано програмний застосунок Secure Vault, призначений для захисту конфіденційних даних від несанкціонованого використання шляхом поєднання програмних механізмів безпеки та апаратної автентифікації користувача. Розробка виконувалась мовою програмування PYTHON із використанням бібліотеки PyQt6 для створення графічного інтерфейсу користувача, SQLite для локального зберігання інформації та криптографічних бібліотек для реалізації механізмів захисту даних [20].

Основною особливістю програмного застосунку є використання апаратного ключа доступу у вигляді USB-пристрою. Робота системи починається з перевірки наявності підключеного пристрою та його ідентифікації за унікальними параметрами. Лише після успішного підтвердження пристрою користувач отримує можливість пройти процедуру автентифікації та отримати доступ до захищених даних.

Розроблений застосунок реалізований за модульною архітектурою, що забезпечує незалежність окремих компонентів системи та спрощує їх подальшу модифікацію. До складу програмного забезпечення входять модулі автентифікації користувача, криптографічного захисту інформації, взаємодії з USB-пристроями, роботи з базою даних, журналювання подій та графічного інтерфейсу користувача.

Структура програмного проєкту представлена на рисунку 3.1.

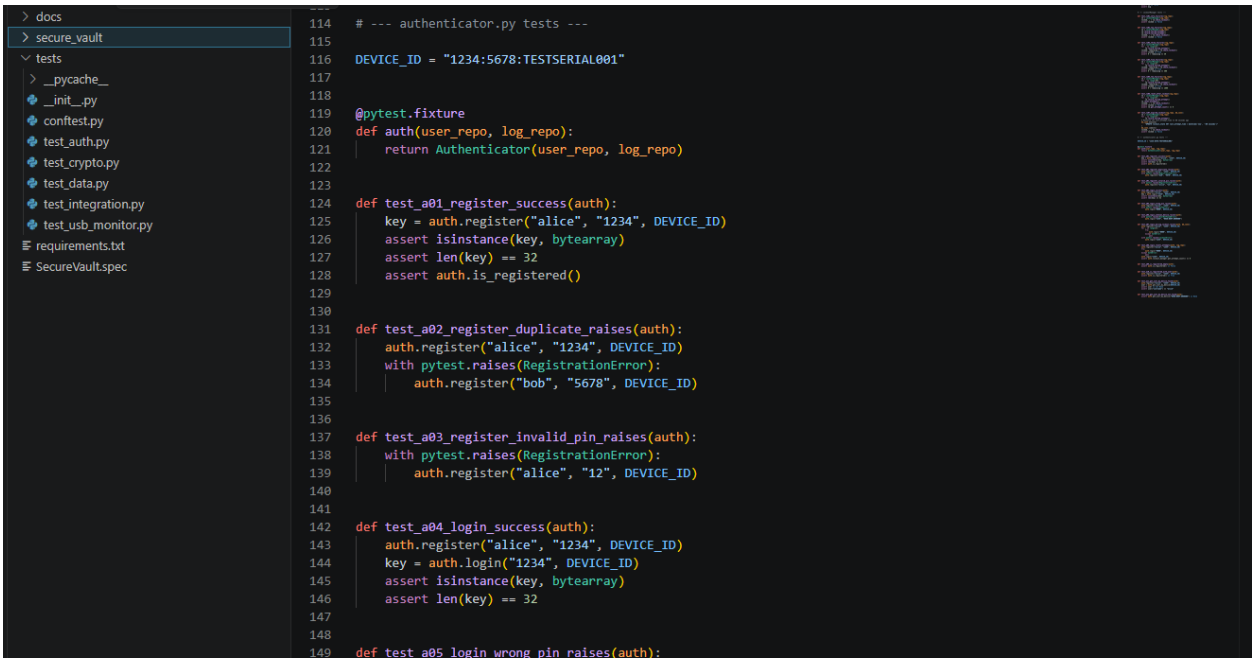


Рисунок 3.1 – Структура програмного проєкту Secure Vault

У процесі реалізації програмного забезпечення було створено окремі пакети для кожного функціонального компонента системи. Зокрема, пакет auth відповідає за автентифікацію користувачів, пакет crypto реалізує механізми шифрування та управління ключами, пакет data забезпечує взаємодію з базою даних SQLite, пакет hardware відповідає за роботу з USB-пристроями, а пакет ui реалізує графічний інтерфейс користувача. Такий підхід дозволив забезпечити високу структурованість програмного коду та спростити процес тестування окремих модулів.

Центральним елементом програмного забезпечення є модуль керування застосунком AppController. Саме цей компонент забезпечує взаємодію між усіма підсистемами, координує процес автентифікації, обробляє події підключення та відключення USB-пристроїв, а також виконує операції із захищеними записами.

Фрагмент програмного коду головного модуля наведено на рисунку 3.2

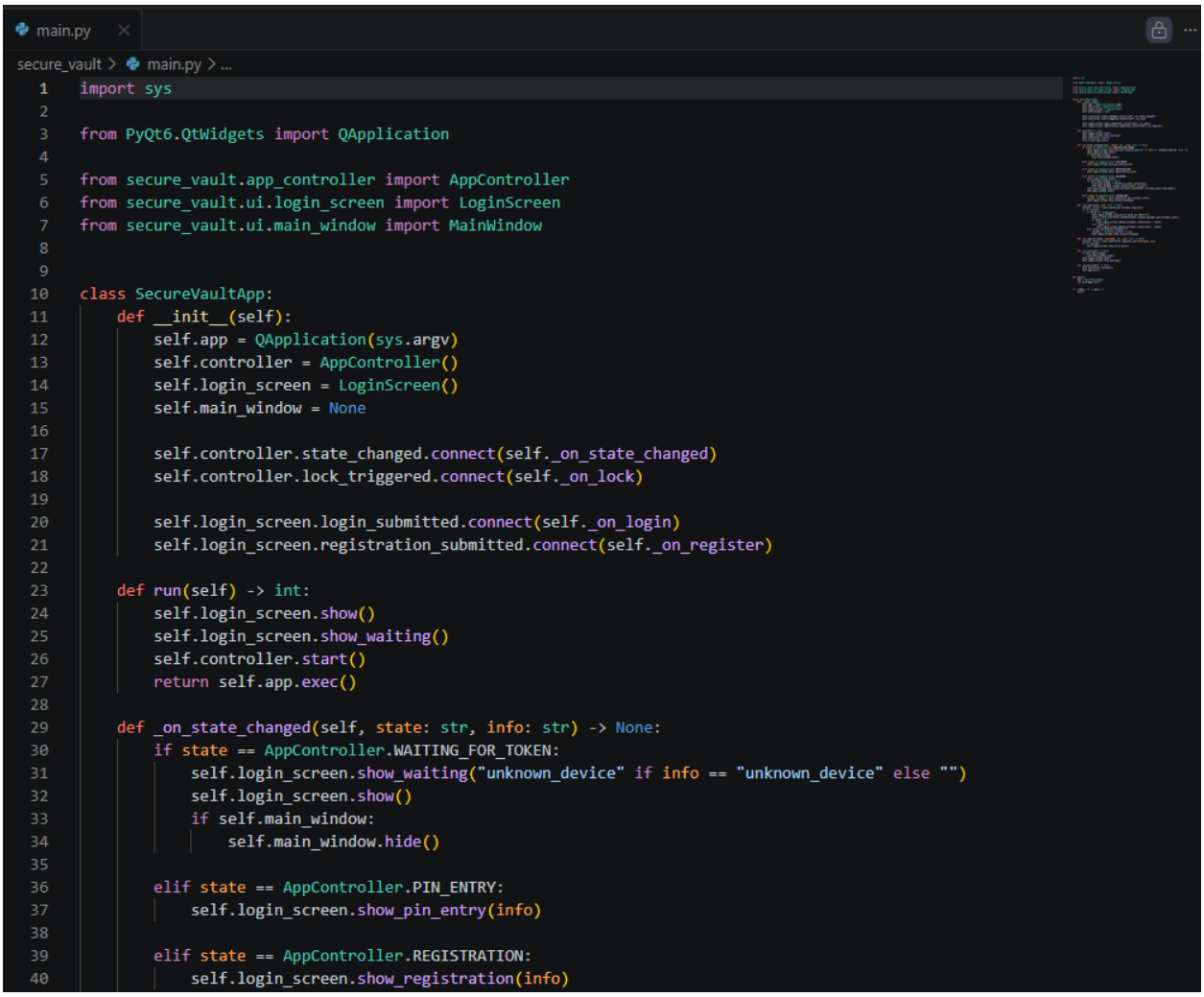


Рисунок 3.2 – Реалізація головного модуля застосунку

Для забезпечення захисту інформації у системі реалізовано механізм криптографічного шифрування даних на основі алгоритму AES-256-GCM. Даний алгоритм забезпечує не лише конфіденційність інформації, а й контроль цілісності даних. Перед записом інформації до бази даних усі користувацькі записи автоматично шифруються, а їх дешифрування виконується лише після успішної автентифікації користувача [21].

Особливістю реалізації є використання механізму генерації ключа на основі PIN-коду користувача та ідентифікатора USB-пристрою. Для формування криптографічного ключа застосовується алгоритм PBKDF2-

HMAC-SHA256 із 600 000 ітерацій, що суттєво ускладнює реалізацію атак перебором.

Для зберігання інформації використовується локальна база даних SQLite. У базі даних реалізовано таблиці користувачів, захищених записів, журналу подій та стану блокування системи. Використання SQLite дозволило створити автономний застосунок, який не потребує окремого серверного програмного забезпечення та може працювати на будь-якому персональному комп’ютері під керуванням операційної системи Windows [22].

Важливою функцією програмного застосунку є автоматичне блокування доступу до інформації у разі відключення USB-пристрою. У цьому випадку система негайно очищає криптографічний ключ із оперативної пам’яті та повертається до режиму очікування підключення апаратного ключа. Такий механізм дозволяє суттєво підвищити рівень безпеки та унеможливлює доступ до даних без фізичного носія автентифікації.

Для реєстрації всіх важливих подій у системі реалізовано модуль журналювання. Журнал містить інформацію про запуск застосунку, успішні входи до системи, помилки автентифікації, створення та видалення записів, а також події підключення або відключення USB-пристроїв. Це дозволяє здійснювати аудит роботи системи та виявляти спроби несанкціонованого доступу [21].

Розроблений графічний інтерфейс (рисунок 3.3) забезпечує просту та зрозумілу взаємодію користувача із системою. Головне вікно застосунку містить список захищених записів, область редагування інформації та панель інструментів для виконання основних операцій.

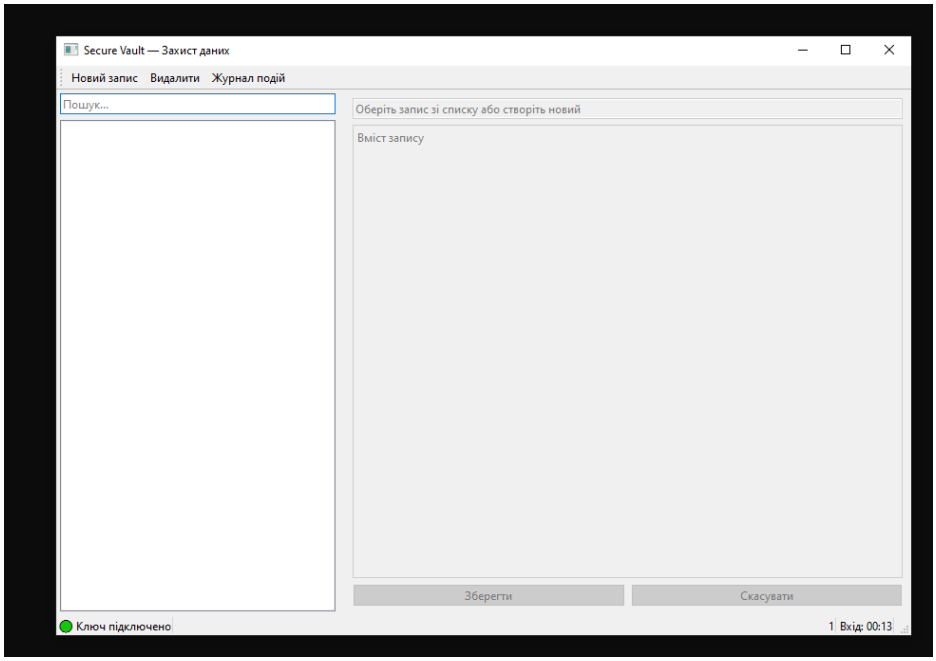


Рисунок 3.3 – Головне вікно програмного застосунку Secure Vault

Таким чином, у результаті реалізації було створено працездатний програмний застосунок Secure Vault, який забезпечує захист конфіденційної інформації за допомогою апаратної автентифікації, криптографічного захисту даних, журналювання подій та контролю доступу. Реалізований функціонал повністю відповідає вимогам технічного завдання та забезпечує можливість безпечного зберігання інформації в локальному середовищі [15, 21].

3.2 Реалізація користувацького інтерфейсу

Одним із важливих етапів розробки програмного застосунку Secure Vault стала реалізація графічного інтерфейсу користувача. Інтерфейс забезпечує взаємодію користувача із системою захисту даних, дозволяє виконувати автентифікацію, працювати із захищеними записами та переглядати журнал подій. Під час розробки особлива увага приділялася простоті використання, зрозумілості навігації та відповідності принципам безпечної роботи з конфіденційною інформацією.

Для створення графічного інтерфейсу було використано бібліотеку PyQt6, яка забезпечує широкі можливості для розробки настільних застосунків під операційну систему Windows. Використання PyQt6 дозволило реалізувати сучасний інтерфейс із підтримкою подій, діалогових вікон та динамічного оновлення інформації [21].

Після запуску програмного застосунку користувач переходить до вікна авторизації. На цьому етапі система перевіряє наявність підключеного USB-пристрою та пропонує ввести персональний PIN-код для підтвердження особи користувача. Такий підхід забезпечує двофакторну автентифікацію, що поєднує фізичний носій та секретний код доступу.

Як видно з рисунка 3.4, вікно авторизації містить інформацію про користувача, поле введення PIN-коду та кнопку входу до системи. Після успішної перевірки введених даних користувач отримує доступ до основного функціоналу програмного застосунку.

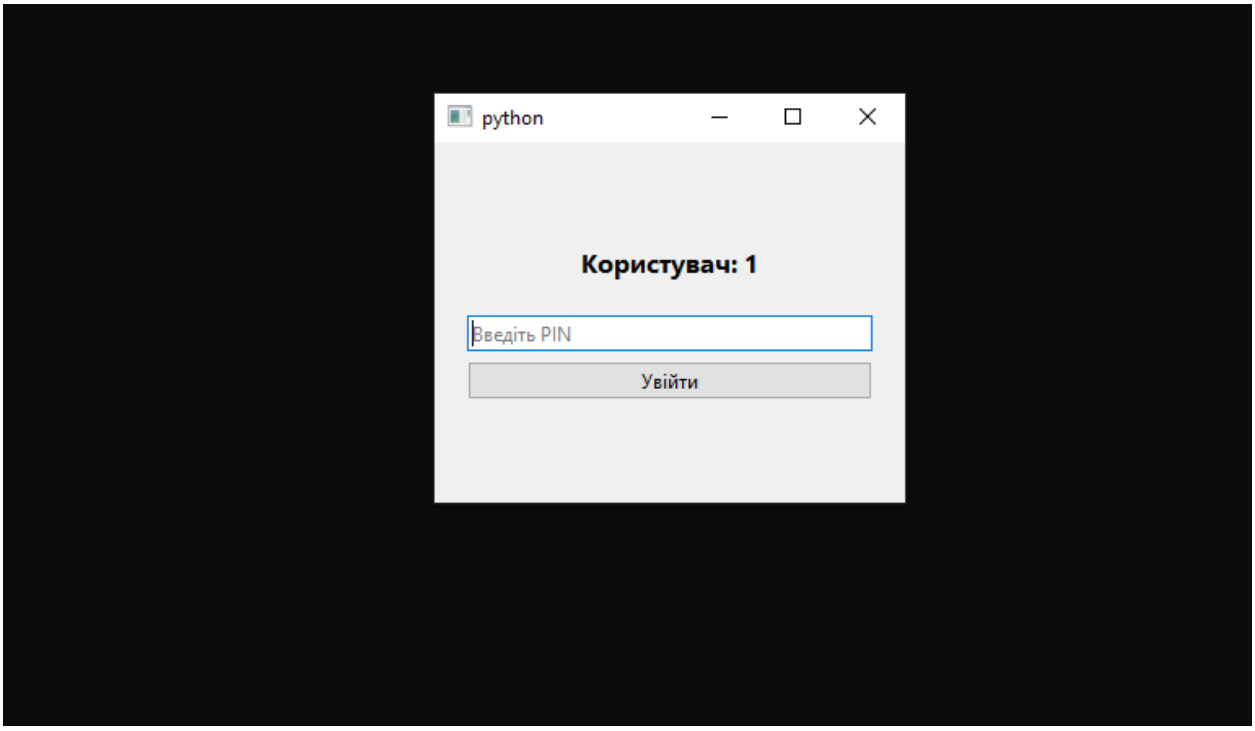


Рисунок 3.4 – Вікно автентифікації користувача

Після завершення процедури автентифікації відкривається головне вікно системи (рисунок 3.5). Саме через нього здійснюється робота із захищеними записами та управління інформацією.

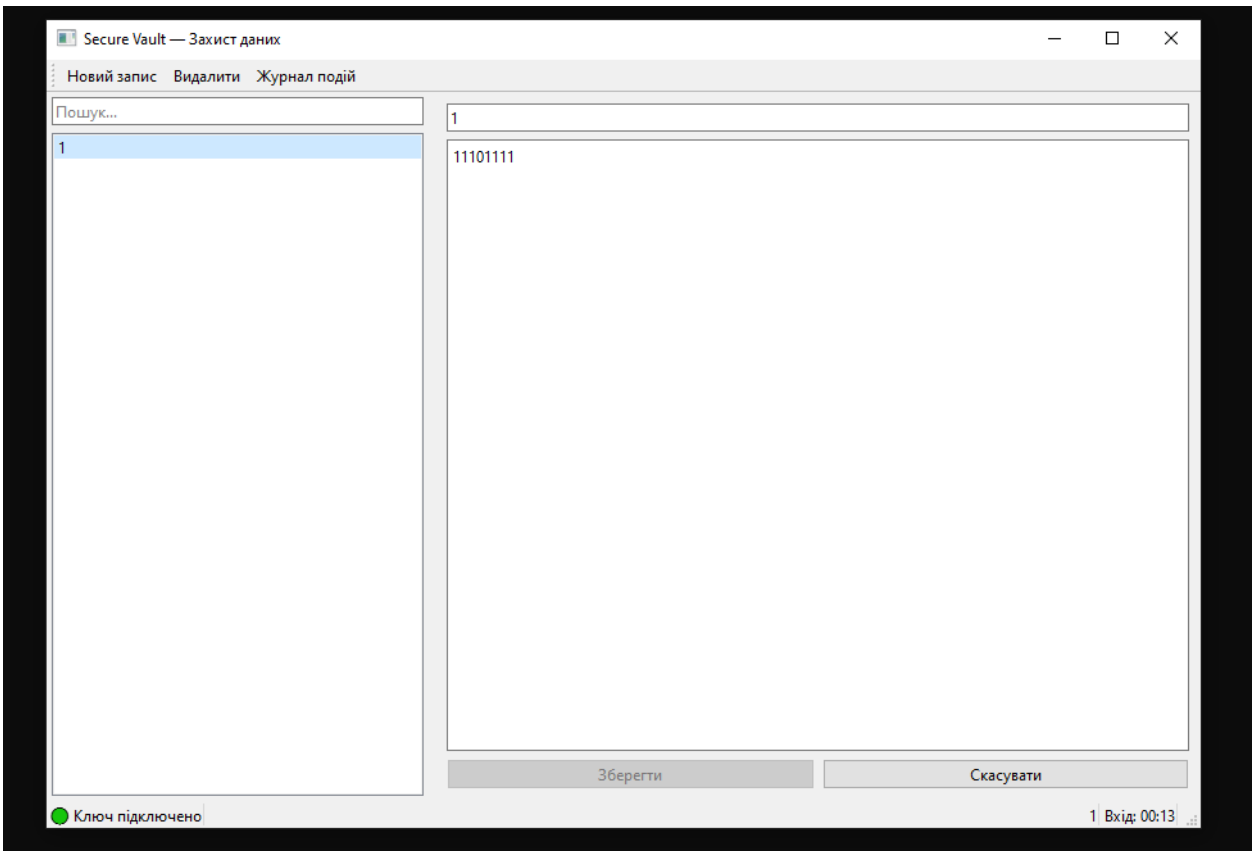


Рисунок 3.5 – Головне вікно програмного застосунку

Головне вікно програмного застосунку складається з декількох функціональних областей. Ліва частина призначена для відображення списку захищених записів та поля пошуку. У правій частині розміщується редактор даних, який дозволяє переглядати та змінювати інформацію. Верхня панель містить основні команди керування записами, зокрема створення нового запису, видалення інформації та відкриття журналу подій.

У нижній частині інтерфейсу розташовано рядок стану. Він відображає інформацію про підключення апаратного ключа та поточний стан системи. Наявність індикатора підключення дозволяє користувачеві постійно

контролювати доступність USB-пристрою, який використовується як засіб автентифікації.

Для створення нових записів або редагування існуючих користувач використовує вбудований редактор інформації. Інтерфейс редактора забезпечує можливість введення заголовка запису та його текстового вмісту.

Після натискання кнопки збереження система автоматично виконує шифрування введених даних та записує їх до бази даних у захищеному вигляді. Завдяки цьому конфіденційна інформація ніколи не зберігається у відкритому форматі, що підвищує рівень інформаційної безпеки [12].

Важливою складовою інтерфейсу є система журналювання подій(рисунок 3.6). Для перегляду журналу реалізовано окреме діалогове вікно, яке відображає інформацію про всі дії користувача та системні події.

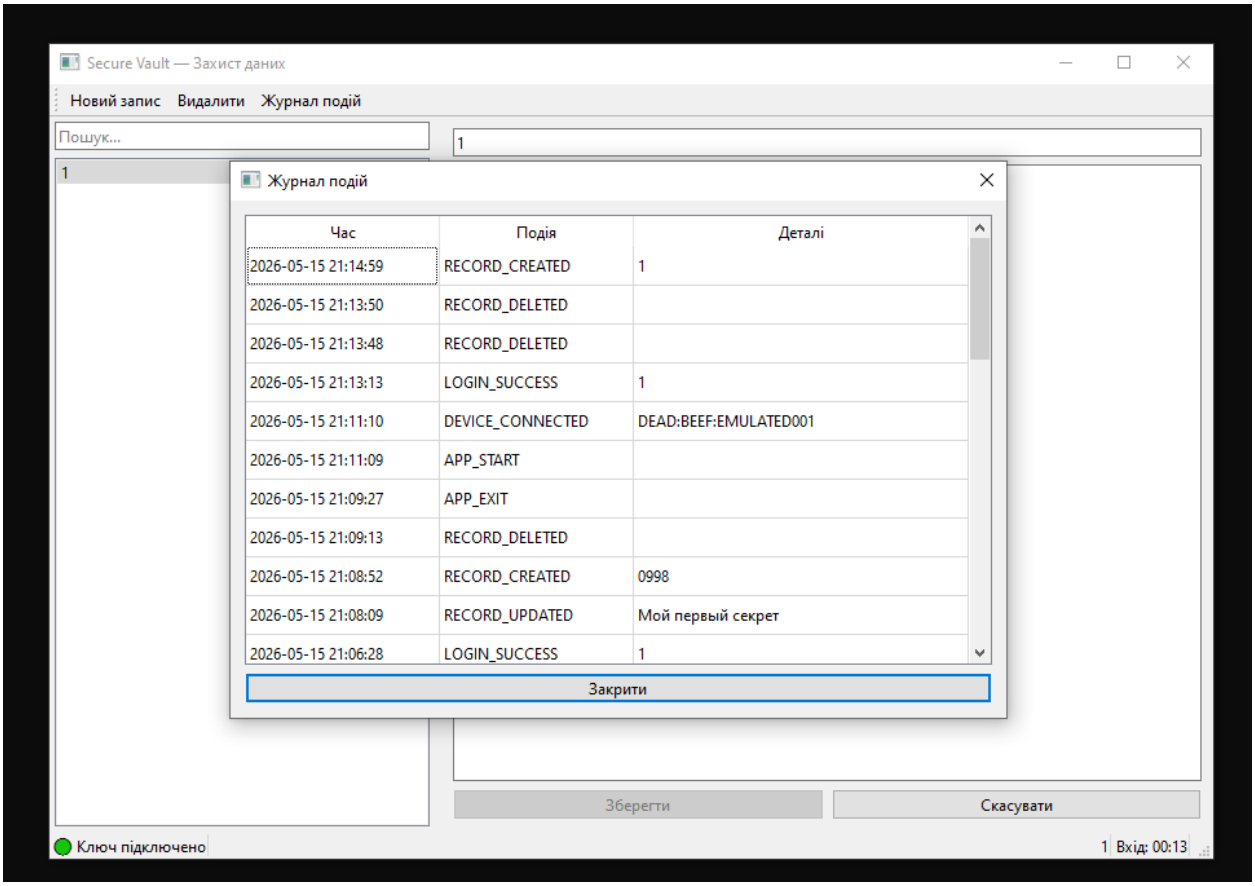


Рисунок 3.6 – Вікно журналу подій

Журнал подій містить інформацію про час виконання операції, тип події та додаткові відомості щодо виконаної дії. У журналі фіксуються успішні входи до системи, підключення та відключення USB-пристроїв, створення, редагування та видалення записів, а також інші важливі події. Наявність такого механізму дозволяє здійснювати аудит роботи системи та аналізувати можливі спроби несанкціонованого доступу [20].

Під час розробки інтерфейсу також було реалізовано механізм автоматичного блокування системи. У разі відключення USB-пристрою програмний застосунок негайно закриває доступ до захищених даних та повертає користувача до вікна автентифікації. Це забезпечує додатковий рівень безпеки та запобігає використанню системи сторонніми особами.

Особливістю реалізованого інтерфейсу є використання української мови для всіх елементів взаємодії з користувачем. Це підвищує зручність використання програмного забезпечення та спрощує роботу із системою для кінцевих користувачів.

Таким чином, реалізований графічний інтерфейс забезпечує зручну взаємодію користувача із програмним застосунком Secure Vault, підтримує виконання всіх основних функцій системи та відповідає вимогам технічного завдання щодо простоти використання, надійності та безпеки роботи із захищеними даними.

3.3 Реалізація механізмів захисту даних та апаратної автентифікації

Ключовою особливістю розробленого програмного застосунку Secure Vault є використання комплексного підходу до забезпечення інформаційної безпеки. На відміну від традиційних програмних рішень, що використовують лише парольний захист, у розробленій системі реалізовано поєднання апаратної автентифікації, криптографічного захисту даних та журналювання подій. Такий підхід дозволяє суттєво підвищити рівень захисту

конфіденційної інформації та зменшити ризик її несанкціонованого використання [21].

Основним механізмом контролю доступу до системи є апаратна автентифікація користувача. У ролі апаратного ключа використовується USB-пристрій, який має унікальний ідентифікатор. Під час запуску застосунку система автоматично виконує перевірку підключених USB-пристроїв та визначає наявність зареєстрованого носія.

Реалізація механізму роботи з апаратним ключем здійснюється за допомогою окремого модуля hardware, який відповідає за моніторинг USB-пристроїв та отримання їхніх ідентифікаційних параметрів. У разі відсутності зареєстрованого пристрою система не дозволяє користувачеві перейти до процедури введення PIN-коду та блокує доступ до інформації.

Реалізація модуля роботи з USB-пристроями наведена на рисунку 3.7.

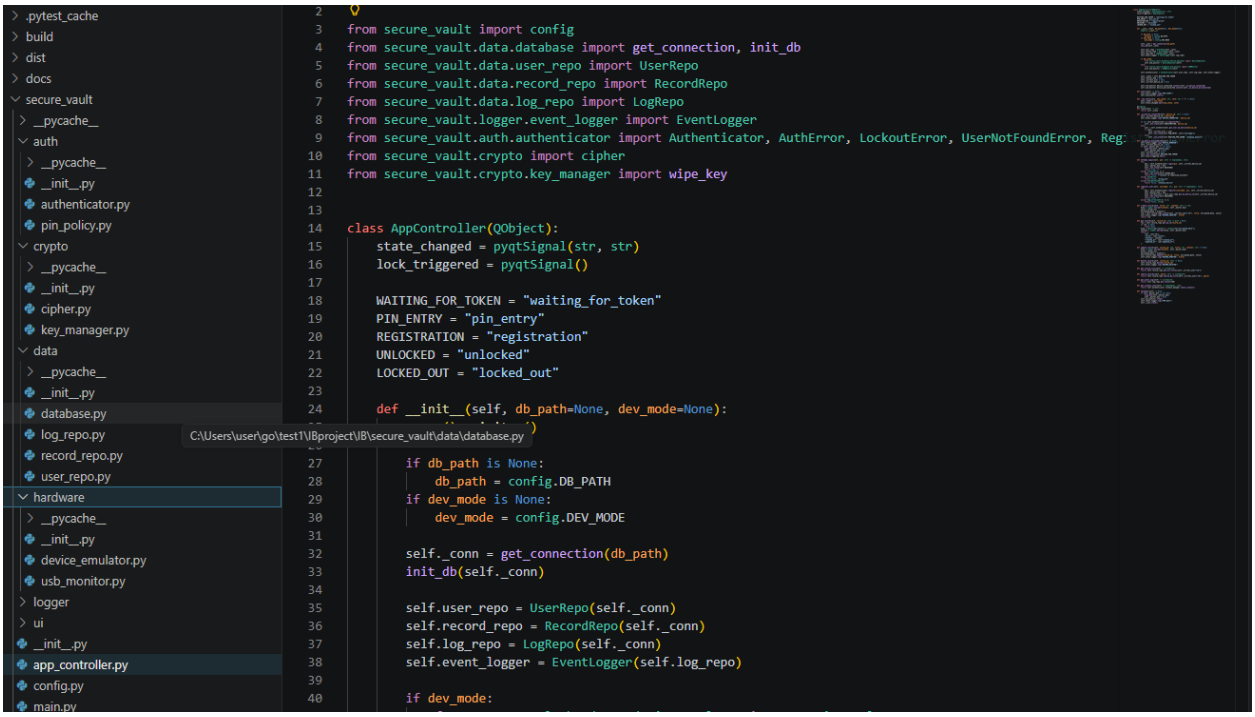


Рисунок 3.7 – Реалізація модуля роботи з USB-пристроями

Після успішного визначення USB-пристрою система переходить до етапу автентифікації користувача. Для підтвердження особи

використовується персональний PIN-код, який вводиться у відповідному діалоговому вікні. Таким чином реалізується двофакторна автентифікація, що базується на принципі поєднання двох незалежних факторів:

– фізичного володіння USB-пристроєм;

– знання секретного PIN-коду.

Такий механізм значно підвищує рівень захисту системи, оскільки отримання доступу до інформації можливе лише за наявності обох факторів одночасно [22].

Для захисту конфіденційних даних у програмному застосунку використовується криптографічний алгоритм AES-256-GCM. Даний алгоритм забезпечує не лише шифрування інформації, але й контроль її цілісності. Усі записи, що створюються користувачем, перед збереженням автоматично шифруються, а під час відкриття проходять процедуру дешифрування.

Використання алгоритму AES-256-GCM дозволяє забезпечити високий рівень криптографічної стійкості та відповідає сучасним вимогам щодо захисту інформації [21].

Для формування криптографічного ключа використовується механізм генерації на основі PIN-коду користувача та унікального ідентифікатора USB-пристрою. Даний підхід дозволяє уникнути зберігання ключа шифрування у відкритому вигляді та забезпечує його унікальність для кожного користувача системи.

Для підвищення рівня безпеки в системі реалізовано механізм очищення криптографічного ключа після завершення роботи користувача. У випадку виходу із системи або відключення USB-пристрою ключ автоматично видаляється з оперативної пам’яті, що унеможливлює його подальше використання сторонніми особами.

Важливим елементом системи є автоматичне блокування доступу до інформації. Після проходження автентифікації застосунок продовжує постійно контролювати наявність USB-пристрою. Якщо пристрій

відключається під час роботи користувача, система негайно переходить у заблокований стан.

Алгоритм роботи механізму блокування включає:

– постійний моніторинг USB-пристрою;
– виявлення факту відключення носія;
– очищення криптографічного ключа;
– закриття доступу до зашифрованих даних;
– повернення до вікна авторизації.

Такий підхід дозволяє забезпечити захист інформації навіть у випадку фізичного доступу сторонньої особи до комп’ютера після завершення роботи користувача.

Окрім механізмів захисту доступу до інформації, у програмному застосунку реалізовано систему журналювання подій. Усі важливі дії користувача та системні події автоматично реєструються у журналі.

До журналу заносяться:

– запуск програмного застосунку;
– підключення USB-пристрою;
– успішна автентифікація;
– помилки введення PIN-коду;
– створення записів;
– редагування записів;
– видалення записів;
– завершення роботи застосунку.

Використання журналу подій дозволяє контролювати роботу системи та здійснювати аудит інформаційної безпеки [20].

Для перевірки працездатності реалізованих механізмів захисту під час розробки було створено набір автоматизованих тестів. Тестування охоплює процедури реєстрації користувачів, автентифікації, перевірки USB-пристрою, генерації криптографічних ключів та роботи із зашифрованими даними.

Наявність окремого набору тестів дозволяє перевіряти коректність роботи основних функцій системи після внесення змін до програмного коду та підвищує надійність програмного забезпечення.

Таким чином, у результаті реалізації механізмів захисту даних було створено багаторівневу систему безпеки, яка поєднує апаратну автентифікацію користувача, криптографічний захист інформації, автоматичне блокування доступу та журналювання подій. Використання такого підходу забезпечує високий рівень захисту конфіденційних даних та відповідає поставленим вимогам технічного завдання кваліфікаційної роботи.

3.4 Тестування та оцінка ефективності програмного за стосунку

Після завершення розробки програмного застосунку Secure Vault було проведено тестування його функціональних можливостей. Основною метою тестування стала перевірка коректності роботи механізмів автентифікації, захисту даних, взаємодії з USB-пристроєм, шифрування інформації та журналювання подій. Проведення тестування дозволяє підтвердити відповідність програмного забезпечення вимогам технічного завдання та оцінити його готовність до практичного використання [15].

У процесі тестування використовувався метод функціонального тестування, який передбачає перевірку роботи кожної функції програмного застосунку окремо. Такий підхід дозволяє виявити помилки реалізації, перевірити правильність взаємодії між окремими модулями та оцінити стабільність роботи системи в різних умовах експлуатації [16].

Для перевірки працездатності системи було сформовано набір тестових сценаріїв, які охоплюють усі основні функції програмного застосунку.

Таблиця 3.1 – Результати функціонального тестування системи

Тестовий сценарій	**Очікуваний результат**	**Результат**
Запуск застосунку без USB-пристрою	Доступ заблоковано	Успішно
Запуск застосунку з коректним USB-пристроєм	Відображення форми входу	Успішно
Введення правильного PIN-коду	Успішна авторизація	Успішно
Введення неправильного PIN-коду	Відмова у доступі	Успішно
Створення нового запису	Запис успішно збережено	Успішно
Редагування запису	Дані оновлено	Успішно
Видалення запису	Запис видалено	Успішно
Відключення USB-пристрою під час роботи	Автоматичне блокування	Успішно
Перегляд журналу подій	Відображення журналу	Успішно
Повторна авторизація після блокування	Доступ відновлено	Успішно

Результати тестування свідчать про коректне функціонування всіх основних компонентів системи. Під час перевірки не було виявлено критичних помилок, які могли б призвести до втрати інформації або порушення роботи механізмів захисту.

Особливу увагу під час тестування було приділено перевірці механізму апаратної автентифікації. Було встановлено, що система коректно визначає підключення зареєстрованого USB-пристрою та блокує доступ у разі його відсутності. Крім того, підтверджено правильність роботи механізму автоматичного блокування після відключення пристрою під час роботи користувача.

Окремо було проведено перевірку криптографічного захисту інформації. Під час тестування створювалися та зберігалися тестові записи,

після чого здійснювався аналіз вмісту бази даних. Було встановлено, що інформація зберігається виключно у зашифрованому вигляді та не може бути прочитана без проходження процедури автентифікації та отримання криптографічного ключа [22].

Для перевірки стійкості механізму автентифікації було виконано серію тестів із введенням неправильних PIN-кодів. У результаті підтверджено коректну роботу механізму обмеження доступу та захисту від перебору облікових даних.

Під час оцінки ефективності роботи системи також було проаналізовано швидкість виконання основних операцій. Практичні випробування показали, що час автентифікації користувача, відкриття записів та збереження інформації є достатньо малим і не створює незручностей під час роботи із застосунком.

Основні переваги розробленого програмного забезпечення полягають у використанні двофакторної автентифікації, автоматичному блокуванні доступу при відключенні USB-пристрою, шифруванні інформації та веденні журналу подій. Поєднання цих механізмів забезпечує комплексний підхід до захисту конфіденційної інформації та відповідає сучасним вимогам інформаційної безпеки.

Таким чином, проведене тестування підтвердило працездатність програмного застосунку Secure Vault та його відповідність вимогам технічного завдання. Реалізовані механізми захисту забезпечують належний рівень інформаційної безпеки, а результати перевірки свідчать про готовність програмного забезпечення до практичного використання.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було вирішено актуальне завдання розробки програмного застосунку захисту даних від несанкціонованого використання із застосуванням апаратних закладних засобів. Проведене дослідження підтвердило, що в умовах стрімкого розвитку інформаційних технологій та постійного зростання кількості кіберзагроз проблема забезпечення конфіденційності та цілісності інформації залишається однією з найважливіших у сфері інформаційної безпеки.

У ході роботи було проаналізовано сучасний стан проблеми захисту даних у комп'ютерних системах, розглянуто основні методи та засоби забезпечення інформаційної безпеки, а також досліджено можливості використання апаратних засобів автентифікації як додаткового рівня захисту інформації. Аналіз існуючих програмних рішень показав, що найбільш ефективними є системи, які поєднують декілька механізмів безпеки та реалізують багаторівневий захист даних.

У результаті виконаного кваліфікаційної роботи було обґрунтовано доцільність використання USB-пристрою як апаратного ключа автентифікації користувача. На відміну від традиційних систем, що використовують лише парольний захист, запропонований підхід передбачає використання двох незалежних факторів автентифікації: фізичного носія та персонального PIN-коду. Це дозволяє суттєво підвищити рівень захисту інформації та зменшити ризик несанкціонованого доступу до даних.

У межах кваліфікаційної роботи було спроєктовано архітектуру програмного застосунку Secure Vault та визначено основні принципи його функціонування. Розроблена архітектура побудована за модульним принципом та включає модулі автентифікації користувачів, роботи з USB-пристроями, криптографічного захисту інформації, журналювання подій, взаємодії з базою даних та графічного інтерфейсу користувача. Такий підхід

забезпечує гнучкість програмної структури, спрощує супровід програмного забезпечення та створює можливість його подальшого розвитку.

У процесі роботи було виконано проєктування структури бази даних програмного застосунку. Для зберігання інформації використано локальну систему керування базами даних SQLite. Розроблена структура забезпечує зберігання інформації про користувачів, захищені записи, журнал подій та параметри блокування доступу. Особливістю реалізації є відсутність зберігання паролів у відкритому вигляді та використання криптографічних механізмів для захисту службових даних.

Практичним результатом кваліфікаційної роботи стало створення програмного застосунку Secure Vault мовою програмування PYTHON із використанням бібліотеки PyQt6 для реалізації графічного інтерфейсу користувача. У програмному забезпеченні реалізовано механізми реєстрації користувачів, автентифікації за допомогою USB-пристрою та PIN-коду, створення і зберігання захищених записів, перегляду журналу подій та автоматичного блокування системи у разі відключення апаратного ключа.

Для забезпечення конфіденційності інформації в застосунку реалізовано криптографічний захист даних із використанням алгоритму AES-256-GCM. Усі користувацькі записи зберігаються у зашифрованому вигляді, а доступ до них можливий лише після успішного проходження процедури автентифікації. Використання сучасних криптографічних алгоритмів дозволило забезпечити високий рівень захисту інформації та відповідність сучасним вимогам інформаційної безпеки.

Під час розробки також було реалізовано механізм автоматичного контролю підключення USB-пристрою. У разі його відключення система автоматично блокує доступ до захищених даних та очищає криптографічні ключі з оперативної пам’яті. Такий підхід забезпечує додатковий рівень захисту інформації та підвищує загальну стійкість системи до спроб несанкціонованого використання.

Проведене функціональне тестування підтвердило працездатність усіх основних компонентів програмного застосунку. Результати тестування показали коректну роботу механізмів автентифікації, шифрування інформації, контролю доступу, журналювання подій та взаємодії з апаратним ключем. Усі функції, передбачені технічним завданням, були реалізовані та успішно пройшли перевірку.

Практичне значення отриманих результатів полягає у можливості використання розробленого програмного забезпечення для захисту конфіденційної інформації на персональних комп’ютерах, у навчальних закладах, невеликих організаціях та інших локальних інформаційних системах. Запропоноване рішення може бути використане як основа для подальшого розвитку програмних засобів захисту даних із використанням апаратних засобів автентифікації.

Таким чином, поставлена мета кваліфікаційної роботи була досягнута, а всі визначені завдання виконані в повному обсязі. У результаті роботи розроблено програмний застосунок захисту даних із використанням апаратних закладних засобів, який забезпечує надійний контроль доступу до інформації, підтримує сучасні механізми криптографічного захисту та відповідає вимогам безпеки локальних інформаційних систем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Allen, G., & Owens, M. (2010). SQLite database system design and implementation. Packt Publishing.

2. Andress, J. (2019). The basics of information security: Understanding the fundamentals of InfoSec in theory and practice (3rd ed.). Elsevier.

3. Bishop, M. (2018). Computer security: Art and science (2nd ed.). Addison-Wesley.

4. Ciampa, M. (2022). Security+ guide to network security fundamentals (8th ed.). Cengage Learning.

5. Conklin, W., White, G., & Williams, D. (2015). Principles of computer security: CompTIA Security+ and beyond (5th ed.). McGraw-Hill Education.

6. Elmasri, R., & Navathe, S. (2016). Fundamentals of database systems (7th ed.). Pearson.

7. Gollmann, D. (2011). Computer security (3rd ed.). John Wiley & Sons.

8. Goodrich, M., & Tamassia, R. (2014). Introduction to computer security. Pearson.

9. Google. Titan Security Key. Google Store. URL: https://store.google.com/product/titan_security_key (дата звернення: 01.07.2026).

10. IDRIX. VeraCrypt Documentation. VeraCrypt. URL: https://www.veracrypt.fr/en/Documentation.html (дата звернення: 01.07.2026).

11. Kim, D., & Solomon, M. (2016). Fundamentals of information systems security (3rd ed.). Jones & Bartlett Learning.

12. Menezes, A., van Oorschot, P., & Vanstone, S. (2018). Handbook of applied cryptography. CRC Press.

13. Microsoft. BitLocker Drive Encryption Overview. Microsoft Learn. URL: https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/ (дата звернення: 01.07.2026).

14. Pfleeger, C., & Pfleeger, S. (2015). Security in computing (5th ed.). Prentice Hall.

15. Pressman, R., & Maxim, B. (2020). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education.

16. Sommerville, I. (2016). Software engineering (10th ed.). Pearson.

17. Stallings, W. (2017). Effective cybersecurity: A guide to using best practices and standards. Addison-Wesley.

18. Sweigart, A. (2020). Automate the boring stuff with Python (2nd ed.). No Starch Press.

19. USB Raptor. Official Project Page. SourceForge. URL: https://sourceforge.net/projects/usbraptor/ (дата звернення: 01.07.2026).

20. Whitman, M., & Mattord, H. (2021). Principles of information security (7th ed.). Cengage Learning.

21. Yubico. YubiKey Hardware Authentication Device. Yubico. URL: https://www.yubico.com/products/yubikey-5-overview/ (дата звернення: 01.07.2026).