Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: "Використання шифрування для захисту конфіденційної інформації в операційній системі MacOS"

Виконав: студент (ка)

Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Павлюк Володимир Богданович
підпис (прізвище та ініціали)

Керівник Стадник М.А.
підпис (прізвище та ініціали)

Нормоконтроль Тимощук Д. І.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Павлюку Володимиру Богдановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Використання шифрування для захисту конфіденційної інформації в операційній системі MacOS

Керівник роботи Стадник Марія Андріївна, к.т.н., доцент кафедри КБ.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «15» 04 2024 року № 4/7-350

2. Термін подання студентом завершеної роботи 12.06.2024

3. Вихідні дані до роботи Вимоги до безпеки даних в операційній системі MacOS. Операційна система MacOS.

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ
1. Огляд безпеки платформи Apple
2. Компоненти безпеки комп'ютерів Mac
3. Практична реалізація шифрування в операційній системі MacOS
4. Безпека життєдіяльності, основи охорони праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Тема, мета, задачі. Апаратна безпека в процесорі Apple. Завантаження Mac із процесором Apple. Завантаження Mac із процесором Intel. Огляд шифрування та захисту даних. Захищена підсистема Secure Enclave. Принцип взаємодії Secure Enclave з AES Engine. Механізм Data Protection. Файлова система APFS. Механізм шифрування за допомогою FileVault. Шифрування системного диску за допомогою FileVault. Шифрування даних на окремому диску. Створення зашифрованого дискового образу.
Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Мариненко С. Ю., к.т.н. доцент кафедри МТ		

7. Дата видачі завдання 29.01.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	29.01.2024	
2.	Опрацювання джерел в галузі дослідження	02.02 – 30.01	
3.	Оформлення розділу «Огляд безпеки платформи Apple»	21.02 – 10.03	
4.	Оформлення розділу «Компоненти безпеки комп’ютерів Mac»	11.03 – 25.03	
5.	Оформлення розділу «Практична реалізація шифрування в операційній системі MacOS»	10.04 – 05.05	
6.	Оформлення розділу «Безпека життєдіяльності, основи охорони праці»	10.05 – 21.05	
7.	Оформлення кваліфікаційної роботи	23.05 – 06.06	
8.	Нормоконтроль	06.06 – 10.06	
9.	Перевірка на плагіат	11.06 – 12.06	
10.	Попередній захист кваліфікаційної роботи	14.06 – 15.06	
11.	Захист кваліфікаційної роботи	27.06.2024	

Студент _____ Павлюк В. Б.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Стадник М.А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Використання шифрування для захисту конфіденційної інформації в операційній системі MacOS. // Кваліфікаційна робота ОР «Бакалавр» // Павлюк Володимир Богданович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБс-42 // Тернопіль, 2024 // С. 61, рис. – 22, табл. – 0, кресл. – 13, додат. – 0.

Ключові слова: Apple, MacOS, APFS, AES, FileVault, XTS, fdesetup, diskutil, hdiutil.

У кваліфікаційній роботі бакалавра проведено аналіз та практичне використання механізмів шифрування в операційній системі MacOS.

В роботі розглянуто безпеку платформи Apple, включаючи безпечне завантаження та захист даних користувача, а також проведено огляд основних компонентів безпеки комп'ютерів Mac, таких як Secure Enclave та Data Protection. Описано криптографічний захист файлової системи APFS та механізм шифрування даних за допомогою FileVault. Показано процедури шифрування системного диску та окремих пристроїв зберігання, використовуючи утиліти fdesetup та diskutil. Детально розглянуто процес створення зашифрованих файлових систем APFS та шифрованих дискових образів з використанням утиліт diskutil та hdiutil відповідно.

В ході роботи було показано, що методи шифрування в MacOS ефективні та надійні, а використання вбудованих інструментів дозволяє забезпечити високий рівень захисту даних у стані спокою.

ANNOTATION

Use of encryption to protect confidential information in the MacOS operating system. // Thesis of educational level "Bachelor"// Volodymyr Pavliuk // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group СБс-42 // Ternopil, 2024 // P. 61, fig. - 22, tab. - 0, chair. - 13 , added. - 0.

Keywords: Apple, MacOS, APFS, AES, FileVault, XTS, fdesetup, diskutil, hdiutil.

In the bachelor's thesis, the analysis and practical use of encryption mechanisms in the MacOS operating system was carried out.

Apple's platform security is covered, including secure boot and user data protection, and key Mac security components such as Secure Enclave and Data Protection are reviewed. The cryptographic protection of the APFS file system and the data encryption mechanism using FileVault were described. Procedures for encrypting the system disk and individual storage devices using the fdesetup and diskutil utilities were shown. The process of creating encrypted APFS file systems and encrypted disk images using the diskutil and hdiutil utilities, respectively, was discussed in detail.

In the course of the work, it was shown that encryption methods in MacOS are effective and reliable, and the use of built-in tools allows for a high level of data protection at rest.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

TRNG — True Random Number Generator

SepOS — Secure Enclave Processor Operating System

DES — Data Encryption Standard

XTS — XEX-Based Tweaked Codebook Mode With Ciphertext Stealing

MAC — Message Authentication Code

SCIP — System Coprocessor Integrity Protection

SSV — Signed System Volume

eSPI — Serial Peripheral Interface

DFU — Device Firmware Upgrade

CMAC — Cipher-based Message Authentication Code

AuxKC — Auxiliary Kernel Collection

XEX — XOR - encrypt - XOR

AES — Advanced Encryption Standard

SoC — System on Chip

RSA — Rivest-Shamir-Adleman

ECC — Elliptic Curve Cryptography

LLB — Low Level Bootloader

JTAG — Joint Test Action Group

SHA — Secure Hash Algorithm

SPA — Static Power Analysis

DPA — Dynamic Power Analysis

DMA — Direct Memory Access

PKA — Public Key Accelerator

SKP — Sealed Key Protection

APFS — Apple File System

KEK — Key Encryption Key

ВСТУП

В умовах постійного розвитку технологій та загроз кібербезпеці, забезпечення конфіденційності та цілісності даних в операційних системах стає надзвичайно важливим завданням. Актуальною є проблема забезпечення безпеки в MacOS - операційній системі, що широко використовується в корпоративному секторі та в особистих цілях. З урахуванням цього контексту, дослідження методів шифрування даних в MacOS є важливою темою для розуміння можливостей системи.

Метою даної роботи є аналіз та дослідження механізмів шифрування даних в операційній системі MacOS, а також практичне використання цих методів для забезпечення безпеки і конфіденційності інформації.

Основними завданнями дослідження є:

- огляд безпеки платформи Apple та компонентів безпеки комп'ютерів Mac;
- аналіз файлової системи APFS та механізмів шифрування, таких як FileVault.
- вивчення процедур шифрування даних на системному диску, окремих пристроях зберігання та образах дисків;
- практичне виконання процедур шифрування даних в операційній системі MacOS.

Об’єктом дослідження є методи шифрування даних в операційній системі MacOS, включаючи їхні технічні аспекти та практичне застосування.

Предметом дослідження є практичне використання механізмів шифрування в операційній системі MacOS для захисту конфіденційності та цілісності даних.

Отримані результати можуть мати важливе практичне значення для організацій та користувачів, що використовують операційну систему MacOS. Застосування рекомендацій та практик, розроблених у даній роботі, дозволить підвищити рівень безпеки та захисту даних, що зберігаються на пристроях з цією операційною системою.

РОЗДІЛ 1 ОГЛЯД БЕЗПЕКИ ПЛАТФОРМИ APPLE

Apple вбудовує безпеку в основу своїх платформ. Кожен пристрій Apple поєднує в собі апаратне забезпечення, програмне забезпечення та служби, розроблені для спільної роботи для забезпечення максимальної безпеки та прозорої взаємодії з користувачем для досягнення кінцевої мети - збереження особистої інформації. Наприклад, розроблені Apple чіпи та захисне обладнання забезпечує критичні функції безпеки. Засоби захисту програмного забезпечення забезпечують захист операційної системи та програм сторонніх розробників. Сервіси забезпечують механізм безпечного та своєчасного оновлення програмного забезпечення і сприяють безпечному зв’язку. У результаті пристрої Apple захищають не лише пристрій і його дані, але й всю екосистему, включаючи все, що користувачі роблять локально, у мережах і за допомогою ключових інтернет-служб [1].

Ключові функції безпеки, такі як апаратне шифрування пристрою, не можна вимкнути помилково.

1.1 Апаратна безпека в процесорі Apple

Щоб програмне забезпечення було безпечним, воно має базуватися на апаратному забезпеченні, яке має вбудований захист. Тому комп’ютера Apple MacOS мають засоби безпеки, які вбудовані в процесорі. Ці можливості включають центральний процесор, який забезпечує функції безпеки системи, а також додаткову мікросхему, призначену для функцій безпеки [1]. Обладнання, орієнтоване на безпеку, дотримується принципу підтримки обмежених і дискретно визначених функцій для мінімізації поверхні атаки. Такі компоненти включають Boot ROM, який формує апаратний корінь довіри для безпечного завантаження, спеціальні механізми AES для ефективного та безпечного шифрування та дешифрування, а також Secure Enclave.

Boot ROM - це найперший код, який виконується процесором пристрою під час завантаження. Будучи невід'ємною частиною процесора, він не може бути змінений ні Apple, ні зловмисником.

Secure Enclave - це компонент системи на чіпі Apple (SoC), який включено до комп'ютерів Mac із Apple Silicon, а також пристроїв із Apple T2 Security Chip. SoC – це інтегральна схема, яка містить кілька компонентів в одній мікросхемі. Процесор додатків, Secure Enclave та інші співпроцесори є компонентами SoC. Сам Secure Enclave дотримується того самого принципу дизайну, що й SoC, і містить власний дискретний Boot ROM і механізм AES. Secure Enclave також забезпечує основу для безпечного створення та зберігання ключів, необхідних для шифрування даних у стані спокою, а також захищає та оцінює біометричні дані для Face ID та Touch ID .

Шифрування даних має бути швидким і ефективним. Апаратний механізм AES виконує швидке вбудоване шифрування та дешифрування під час запису чи читання файлів. Спеціальний канал із Secure Enclave надає необхідні криптографічні ключі механізму AES, не розкриваючи цю інформацію процесору додатків (центральному процесору) або загальній операційній системі. Це допомагає гарантувати, що технології Apple Data Protection і FileVault захищають файли користувачів без розкриття довгострокових ключів шифрування.

Apple розробила безпечне завантаження, щоб захистити найнижчі рівні програмного забезпечення від втручання та дозволити завантажувати під час запуску лише надійне програмне забезпечення операційної системи від Apple. Безпечне завантаження починається з незмінного коду під назвою Boot ROM, який закладається під час виробництва Apple SoC і відомий як апаратний корінь довіри. На комп'ютерах Mac із чіпом T2 надійність безпечного завантаження MacOS починається з T2. Чіп T2 і Secure Enclave також виконують власні процеси безпечного завантаження, використовуючи власний окремий Boot ROM - це точний аналог того, як процесори серії A, M1 і M2 безпечно завантажуються.

Secure Enclave також обробляє дані обличчя та відбитків пальців із датчиків Face ID і Touch ID на пристроях Apple. Це забезпечує безпечну автентифікацію,

водночас зберігаючи конфіденційність і безпеку біометричних даних користувача. Це також дозволяє користувачам скористатися перевагами безпеки довших і складніших кодів доступу та паролів із зручністю швидкої автентифікації для доступу чи покупок у багатьох ситуаціях.

1.2 Безпека системи Apple

Використовуючи можливості апаратного забезпечення Apple, система безпеки відповідає за контроль доступу до системних ресурсів у пристроях Apple без шкоди для зручності використання. Безпека системи охоплює процес завантаження, оновлення програмного забезпечення та захист системних ресурсів комп'ютера, таких як процесор, пам'ять, диск, програмне забезпечення та збережені дані [2].

Важливою частиною безпеки Apple є безпечне завантаження, яке захищає систему від зараження зловмисним програмним забезпеченням під час завантаження. Безпечне завантаження починається з процесора та створює ланцюжок довіри за допомогою програмного забезпечення, де кожен крок розроблено для забезпечення належного функціонування наступного перед передачею керування. Ця модель безпеки підтримує не лише стандартне завантаження пристроїв Apple, але й різні режими для відновлення та своєчасного оновлення на пристроях Apple. Такі підкомпоненти, як Secure Enclave, також виконують власне безпечне завантаження, щоб забезпечити завантаження лише завідомо справного коду від Apple. Система оновлення розроблена для запобігання атакам на пониження версії, щоб пристрої не можна було повернути до старішої версії операційної системи, як метод викрадення даних користувача.

Пристрої Apple також включають захист під час завантаження та виконання, щоб вони зберігали свою цілісність під час поточної роботи. Розроблений процесор Apple забезпечує загальну архітектуру для захисту цілісності операційної системи. MacOS також має розширений і настроюваний набір

можливостей захисту для підтримки різних обчислювальних моделей, а також можливості, які підтримуються на всіх апаратних платформах Mac.

1.1.1 Завантаження Mac із процесором Apple

На рисунку 1.1 показано процес завантаження Mac з процесором Apple.

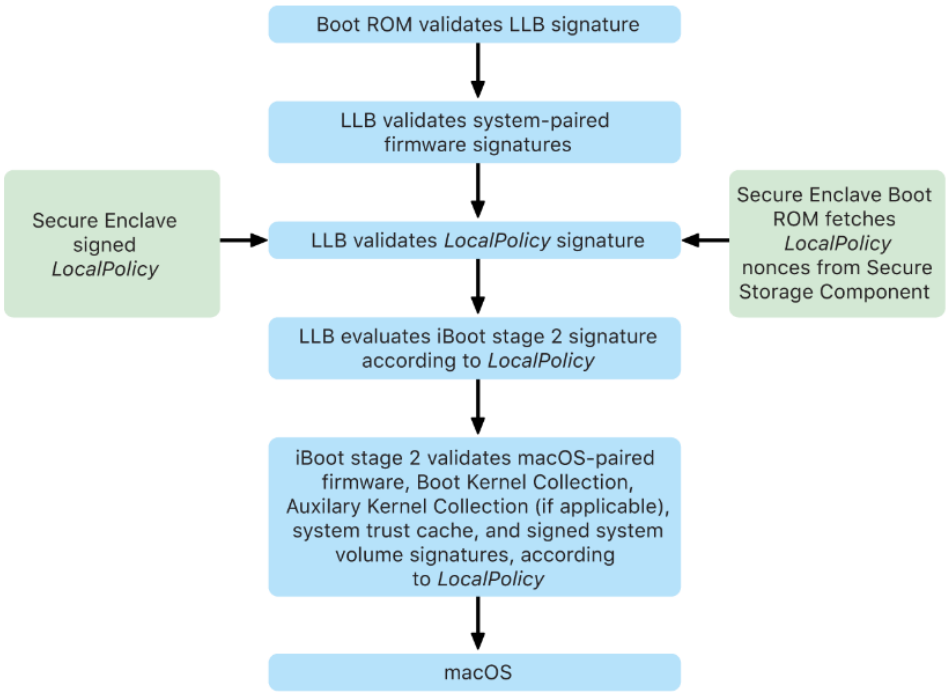


Рисунок 1.1 – Процес завантаження Mac з процесором Apple

Чіп виконує код із Boot ROM на першому етапі в ланцюжку довіри. Захищене завантаження MacOS на комп'ютері Mac із процесором Apple Silicon перевіряє не лише сам код операційної системи, але й політики безпеки налаштовані авторизованими користувачами [3].

Коли запускається LLB, він перевіряє підписи та завантажує системне програмне забезпечення для внутрішніх ядер SoC, таких як сховище, дисплей, керування системою та контролери Thunderbolt. LLB також відповідає за завантаження LocalPolicy, який є файлом, підписаним Secure Enclave Processor. Файл LocalPolicy описує конфігурацію, яку користувач вибрав для політики безпеки завантаження системи та виконання. LocalPolicy має такий самий формат структури даних, як і всі інші об'єкти завантаження, але він підписується

локально приватним ключем, який доступний лише в Secure Enclave конкретного комп'ютера, а не підписується центральним сервером Apple, як оновлення програмного забезпечення.

Щоб запобігти повторному відтворенню будь-якої попередньої локальної політики, LLB має шукати значення захисту від повторного відтворення (anti-replay value) в компоненті безпечного зберігання (Secure Storage Component), підключеному до Secure Enclave. Для цього він використовує Secure Enclave Boot ROM і перевіряє чи значення захисту від повторного відтворення в LocalPolicy відповідає значенню захисту від повторного відтворення в компоненті безпечного зберігання. Це допомагає запобігти повторному застосуванню до системи старої локальної політики, яка могла бути налаштована для нижчого рівня безпеки, після оновлення безпеки. У результаті безпечне завантаження на комп'ютері Mac із процесором Apple допомагає захистити не лише від пониження версії операційної системи, але й від зниження рівня політики безпеки.

Файл LocalPolicy фіксує, чи налаштовано операційну систему на повну (Full Security), знижену (Reduced Security) або дозвільну (Permissive Security) безпеку.

При повній безпеці система дозволяє завантажувати лише найновіше програмне забезпечення, доступне під час встановлення. При зниженій безпеці LLB використовує метод довіри до глобальних підписів, які входять до складу операційної системи. Це дозволяє системі запускати старіші версії MacOS. Оскільки старіші версії MacOS неминуче мають невиправлені вразливості, цей режим безпеки описується як знижений. Це також рівень політики, необхідний для підтримки завантажувальних розширень ядра (kexts). Дозвільна система безпека поводиться як знижена безпека, оскільки вона використовує глобальну перевірку підпису для iBoot і не тільки, але вона також повідомляє iBoot, що вона повинна приймати деякі завантажувальні об'єкти, підписані Secure Enclave тим самим ключем, який використовується для підпису LocalPolicy. Цей рівень політики підтримує користувачів, які створюють, підписують і завантажують власні ядра XNU.

iBoot - це завантажувач другого рівня для всіх пристроїв Apple. Код, який завантажує XNU як частину безпечного ланцюжка завантаження. Залежно від покоління SoC, iBoot може завантажуватися за допомогою LLB або безпосередньо Boot ROM.

Якщо LocalPolicy вказує LLB, що вибрана операційна система працює в режимі повної безпеки, LLB оцінює персоналізований підпис для iBoot. Якщо він працює в режимі обмеженого захисту або дозволеного захисту, він оцінює глобальний підпис. Будь-які помилки перевірки підпису призводять до завантаження системи recoveryOS для надання варіантів відновлення.

Після того, як LLB передає процес завантаження до iBoot, він завантажує вбудоване програмне забезпечення для MacOS та переглядає інформацію про LocalPolicy, передану йому від LLB. Якщо LocalPolicy вказує на наявність допоміжної колекції ядра (AuxKC), iBoot шукає її у файловій системі, перевіряє, чи її було підписано Secure Enclave тим самим ключем, що й LocalPolicy, і перевіряє, чи її хеш відповідає хешу зберігається в LocalPolicy. Якщо AuxKC перевірено, iBoot поміщає його в пам’ять із колекцією завантажувального ядра (Boot Kernel Collection) перед блокуванням повної області пам’яті, що охоплює колекцію завантажувального ядра та AuxKC із захистом цілісності системного співпроцесора (SCIP). SCIP - це механізм призначений для запобігання модифікації мікропрограми співпроцесора. Якщо політика вказує, що AuxKC має бути присутнім, але його не знайдено, система продовжує завантажуватися MacOS без нього. iBoot також відповідає за перевірку кореневого хешу для підписаного системного тому (SSV), щоб переконатися, що файлова система, яку монтуватиме ядро, повністю перевірена на цілісність.

1.1.2 Завантаження Mac із процесором Intel

Коли комп’ютер Mac на базі Intel із мікросхемою безпеки Apple T2 увімкнено, мікросхема виконує безпечне завантаження зі свого Boot ROM так само, як із Apple Silicon. Це перевіряє завантажувач iBoot і є першим кроком у ланцюжку довіри. iBoot перевіряє ядро та код розширення ядра на чіпі T2, який

потім перевіряє мікропрограму Intel UEFI. Прошивка UEFI та відповідний підпис спочатку доступні лише для мікросхеми T2.

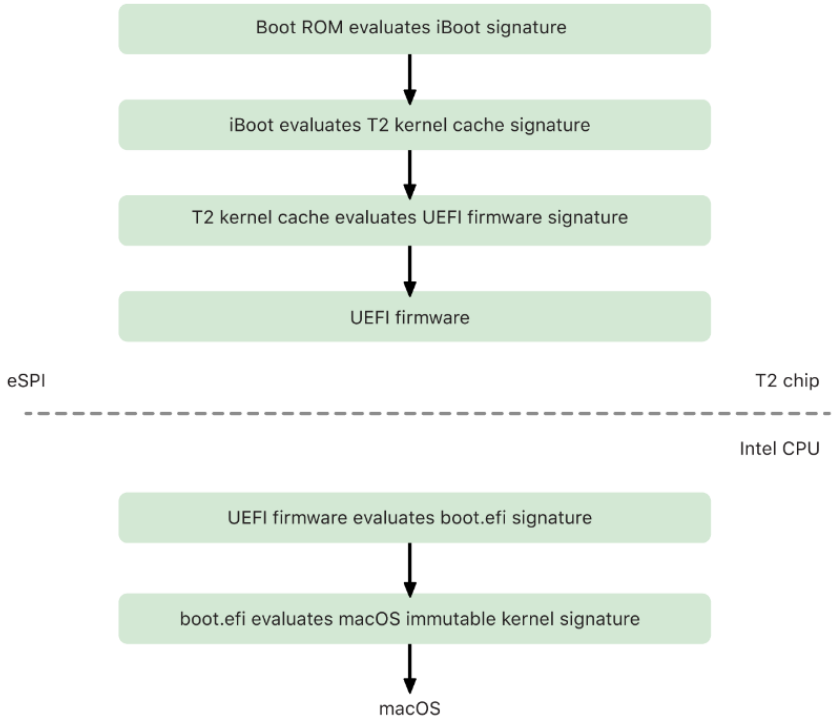


Рисунок 1.2 – Процес завантаження Mac з процесором Intel

Після перевірки, образ мікропрограми UEFI відображається в частині пам’яті чіпа T2. Ця пам’ять стає доступною для ЦП Intel через розширений послідовний периферійний інтерфейс (eSPI). Коли процесор Intel завантажується вперше, він отримує мікропрограму UEFI через eSPI з відображеної в пам’яті копії мікропрограми розташованої на мікросхемі T2. Оцінка ланцюжка довіри продовжується на процесорі Intel, при цьому мікропрограма UEFI оцінює підпис для boot.efi, який є завантажувачем MacOS. Сигнатури безпечного завантаження MacOS, зареєстровані в базі Intel, зберігаються в тому самому форматі Image4, який використовується для безпечного завантаження мікросхеми T2. Boot.efi, у свою чергу, перевіряє підпис нового файлу під назвою immutablekernel. Коли безпечне завантаження ввімкнено, файл immutablekernel представляє повний набір розширень ядра Apple, необхідних для завантаження MacOS. Дія політики безпечного завантаження припиняється після передачі до незмінного ядра, і

після цього набувають чинності політики безпеки MacOS такі як захист цілісності системи та підписані розширення ядра [4].

Якщо під час цього процесу виникнуть будь-які помилки, комп'ютер Mac перейде в режим відновлення, Apple T2 Security Chip Recovery або Apple T2 Security Chip DFU.

1.3 Огляд шифрування та захисту даних

Захищений ланцюжок завантаження, безпека системи та можливості безпеки додатків допомагають переконатися, що на пристрої працюють лише надійні коди і програми. Пристрої Apple мають додаткові функції шифрування для захисту даних користувача, навіть якщо інші частини інфраструктури безпеки зламано. Усі ці функції приносять користь як користувачам, так і IT-адміністраторам, захищаючи особисту та корпоративну інформацію та забезпечуючи методи миттєвого й повного віддаленого видалення в разі крадіжки чи втрати пристрою [5].

Дані на комп'ютерах Mac із процесором Intel захищені технологією томового шифрування під назвою FileVault [6]. Mac з Apple Silicon використовує гібридну модель із підтримкою Data Protection із двома застереженнями. Клас D, як найнижчий рівень захисту, не підтримується. Рівень за замовчуванням встановлено на клас C, який діє так само, як FileVault в Mac на базі Intel. У всіх випадках ієрархії керування ключами базуються на спеціальному процесорі Secure Enclave, а спеціальний механізм AES підтримує шифрування на швидкості лінії зв'язку та допомагає гарантувати, що довгострокові ключі шифрування не піддаються впливу операційної системи ядра чи ЦП де вони можуть бути скомпрометовані. Комп'ютер Mac на базі процесора Intel із T1 або без Secure Enclave не використовує спеціальний процесор для захисту своїх ключів шифрування FileVault.

Окрім захисту даних і FileVault для запобігання несанкціонованому доступу до даних, Apple використовує ядра операційної системи для забезпечення захисту та безпеки. Ядро використовує елементи керування доступом до програм

ізольованого програмного середовища, що обмежує доступ до даних, які програма може отримати. Механізм під назвою Data Vault, який замість того, щоб обмежувати виклики, які може здійснювати програма, обмежує доступ до даних програми з усіх інших програм, що запитують. Це механізм запроваджений ядром для захисту від неавторизованого доступу до даних незалежно від того, чи сама програма, яка запитує, знаходиться в ізольованому програмному середовищі.

1.4 Висновки до розділу

В першому розділі було проведено огляд безпеки платформи Apple. Показано, що Boot ROM формує апаратний корінь довіри для безпечного завантаження, спеціальні механізми AES використовуються для ефективного та безпечного шифрування та дешифрування на апаратному рівні. Проведено огляд безпечного завантаження Mac з процесором Apple та на базі Intel із мікросхемою безпеки Apple T2.

Показано, що пристрої Apple мають додаткові функції шифрування для захисту даних користувача, навіть якщо інші частини інфраструктури безпеки зламано.

РОЗДІЛ 2 КОМПОНЕНТИ БЕЗПЕКИ КОМП'ЮТЕРІВ MAC

2.1 Захищена підсистема Secure Enclave

Secure Enclave - це спеціальна захищена підсистема, інтегрована в системи SoC. Secure Enclave ізольовано від головного процесора (Application Processor), щоб забезпечити додатковий рівень безпеки та розроблено для захисту конфіденційних даних користувача, навіть якщо ядро Application Processor стає скомпрометованим [7] (див.рисунок 2.1).

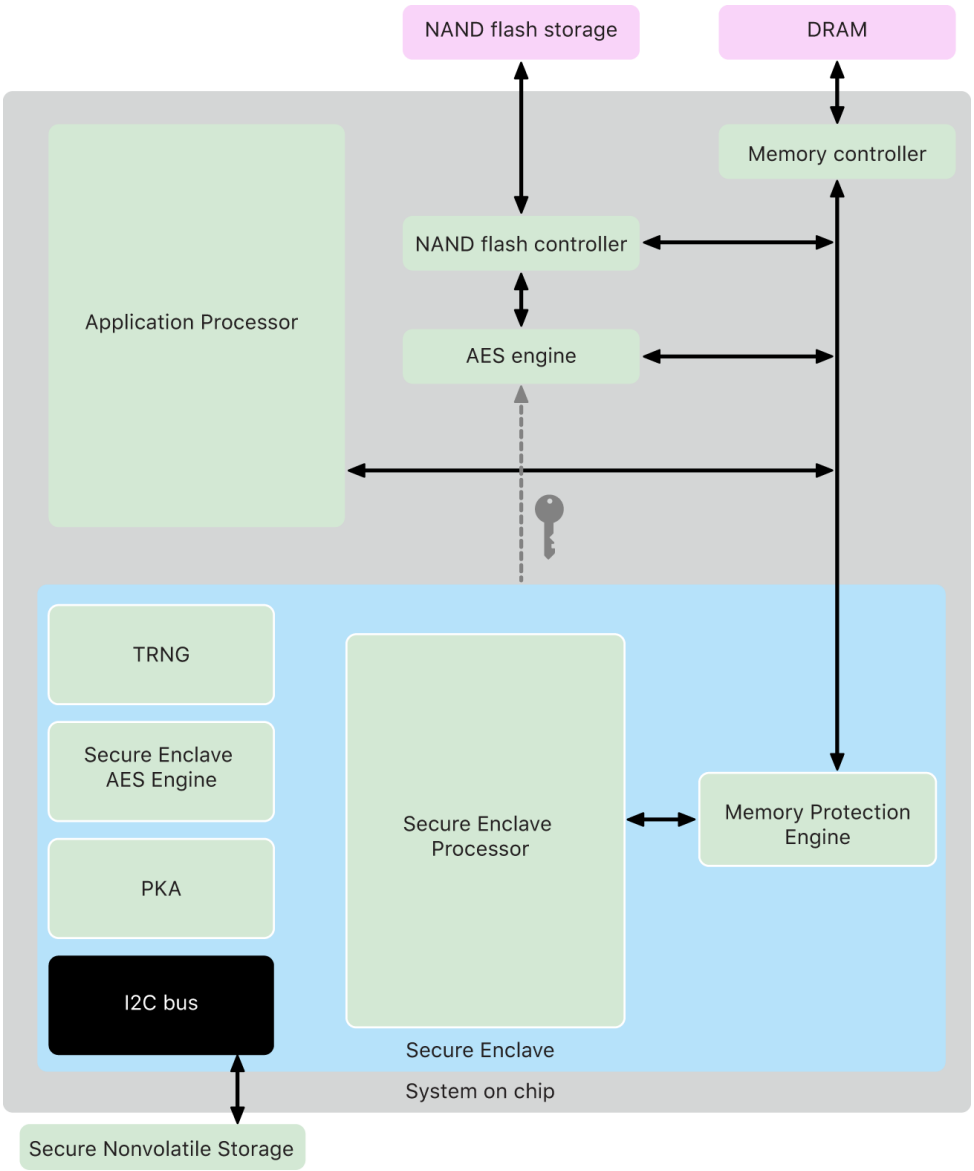


Рисунок 2.1 – Принцип взаємодії Secure Enclave з решту компонентами

Secure Enclave дотримується тих самих принципів проектування, що й SoC:

- boot ROM для встановлення апаратного кореня надійності;
- механізм AES для ефективних і безпечних криптографічних операцій і захищеної пам’яті.

Незважаючи на те, що Secure Enclave не включає сховище, він має механізм для безпечного зберігання інформації у підключеному сховищі окремо від флеш-пам’яті NAND, яка використовується процесором додатків і операційною системою.

Secure Enclave доступний в комп’ютерах Mac із процесором Apple, комп’ютерах MacBook Pro, які містять чіп Apple T1 та комп’ютерах Mac на базі Intel, які містять Apple T2 Security Chip.

Процесор Secure Enclave забезпечує основну обчислювальну потужність для Secure Enclave. Щоб забезпечити найсильнішу ізоляцію, процесор Secure Enclave призначений виключно для використання в Secure Enclave. Це допомагає запобігти атакам побічного каналу, які залежать від шкідливого програмного забезпечення, яке використовує те саме ядро виконання, що й цільове програмне забезпечення, яке атакується. Процесор Secure Enclave працює на налаштованій Apple версії мікроядра L4. Він розроблений для ефективної роботи на нижчій тактовій частоті, що допомагає захистити його від атак тактової частоти та живлення. Процесор Secure Enclave, починаючи з A11 і S4, включає механізм із захистом пам’яті та зашифровану пам’ять із можливостями захисту від повторного відтворення, безпечне завантаження, спеціальний генератор випадкових чисел і власний механізм AES.

AES - це симетричний алгоритм шифрування. Він був прийнятий Національним інститутом стандартів і технологій (NIST) у 2001 році, замінивши застарілий DES. Він є одним з найпоширеніших алгоритмів симетричного шифрування і використовується у різних сферах, включаючи комп'ютерну безпеку. AES працює на блоках даних фіксованої довжини 128 біт і може використовуватися з ключами довжиною 128, 192 або 256 біт [8].

Secure Enclave працює з виділеної області пам’яті DRAM пристрою. Кілька рівнів захисту ізолюють пам'ять, захищену Secure Enclave, від процесора додатків. Коли пристрій запускається, Secure Enclave Boot ROM генерує

випадковий тимчасовий ключ захисту пам’яті для механізму захисту пам’яті. Щоразу, коли Secure Enclave записує у виділену область пам’яті, Memory Protection Engine шифрує блок пам’яті за допомогою AES у режимі XEX і обчислює тег автентифікації на основі CMAC для пам'яті.

CMAC є алгоритмом створення коду автентифікації повідомлення на основі блочного шифру. Він використовується для забезпечення цілісності та автентичності даних. CMAC використовує симетричний блочний шифр для генерації фіксованого розміру тегу (MAC) з вхідного повідомлення змінної довжини та секретного ключа. Основними компонентами CMAC є блочний шифр та секретний ключ, що є спільним між відправником і одержувачем [9].

Механізм захисту пам’яті зберігає тег автентифікації разом із зашифрованою пам’яттю. Коли Secure Enclave зчитує пам’ять, механізм захисту пам’яті перевіряє тег автентифікації. Якщо тег автентифікації збігається, Memory Protection Engine розшифровує блок пам’яті. Якщо тег не збігається, Memory Protection Engine повідомляє Secure Enclave про помилку. Після помилки автентифікації пам’яті Secure Enclave припиняє приймати запити до перезавантаження системи.

Memory Protection Engine додає захист від повторного відтворення для пам’яті Secure Enclave. Щоб запобігти повторному відтворенню важливих для безпеки даних, механізм захисту пам’яті зберігає унікальний одноразовий номер, який називається значенням захисту від повторного відтворення (anti-replay value), для блоку пам’яті поряд з тегом автентифікації. Значення захисту від повторного відтворення використовується як додаткове налаштування для тегу автентифікації CMAC. Значення захисту від повторного відтворення для всіх блоків пам’яті захищено за допомогою дерева цілісності, що входить до виділеної SRAM у Secure Enclave. Для запису Memory Protection Engine оновлює значення захисту від повторного відтворення та кожен рівень дерева цілісності до SRAM. Для процесу читання механізм захисту пам’яті перевіряє значення захисту від повторного відтворення та кожен рівень дерева цілісності аж до SRAM. Невідповідності значень захисту від повтору обробляються так само, як і невідповідності тегів автентифікації.

На Apple A14, M1 або пізніших SoC механізм захисту пам’яті підтримує два тимчасові ключі захисту пам’яті. Перший використовується для даних, приватних для Secure Enclave, а другий використовується для даних, наданих спільно з Secure Neural Engine.

Механізм захисту пам’яті працює вбудовано та прозоро для Secure Enclave. Secure Enclave зчитує та записує в пам’ять так, ніби це звичайна незашифрована DRAM, тоді як за межами Secure Enclave можна побачити лише зашифровану та автентифіковану версію пам’яті. Результатом є надійний захист пам’яті без компромісів у продуктивності чи складності програмного забезпечення.

На Apple A13 і пізніших SoC Secure Enclave містить Boot Monitor, призначений для забезпечення покращеної цілісності хешу завантаженої sepOS.

Під час запуску системи конфігурація SCIP процесора Secure Enclave допомагає запобігти виконанню процесором Secure Enclave будь-якого коду, крім Boot ROM Secure Enclave. Boot Monitor допомагає запобігти Secure Enclave безпосередньо змінювати конфігурацію SCIP. Щоб виконати завантажений sepOS, Secure Enclave Boot ROM надсилає Boot Monitor запит із адресою та розміром завантаженого sepOS.

SepOS - це мікропрограма, яка міститься в окремому спеціальному Boot ROM для Secure Enclave.

Отримавши запит, Boot Monitor скидає Secure Enclave Processor, хешує завантажений sepOS, оновлює параметри SCIP, щоб дозволити виконання завантаженого sepOS, і починає виконання в рамках щойно завантаженого коду. Оскільки система продовжує завантажуватися, цей самий процес використовується щоразу, коли новий код стає виконуваним. Кожного разу Boot Monitor оновлює поточний хеш процесу завантаження. Монітор завантаження також включає критичні параметри безпеки в поточний хеш.

Генератор справжніх випадкових чисел (TRNG) використовується для створення безпечних випадкових даних. Secure Enclave використовує TRNG щоразу, коли генерує випадковий криптографічний ключ, початкове число випадкового ключа або іншу ентропію. TRNG базується на блокових шифрах у режимі CTR.

Secure Enclave включає кореневий криптографічний ключ з унікальним ідентифікатором UID. UID є унікальним для кожного окремого пристрою та не пов’язаний з жодним іншим ідентифікатором на пристрої. Випадково згенерований UID вставляється в SoC під час виробництва. Починаючи з A9 SoC, UID генерується Secure Enclave TRNG під час виробництва та записується за допомогою програмного процесу, який повністю виконується в Secure Enclave. Цей процес захищає UID від видимості за межами пристрою під час виробництва, тому Apple або будь-який із її постачальників не може отримати доступ до нього чи зберегти його. SepOS використовує UID для захисту секретів пристрою. UID дозволяє криптографічно прив’язувати дані до конкретного пристрою. Наприклад, ієрархія ключів, що захищає файлову систему, включає UID, тому, якщо внутрішню пам’ять SSD фізично перемістити з одного пристрою на інший, файли будуть недоступні. Інші захищені секрети пристрою включають дані Face ID або Touch ID . На Mac лише внутрішня пам’ять, яка пов’язана з AES Engine, отримує цей рівень шифрування. Наприклад, ані зовнішні накопичувачі, підключені через USB, ані накопичувачі на основі PCIe, не шифруються таким чином.

Secure Enclave також має ідентифікатор групи пристроїв (GID), який є спільним для всіх пристроїв, які використовують даний SoC (наприклад, усі пристрої, які використовують Apple A15 SoC, мають однаковий GID).

UID та GID недоступні через JTAG або інші інтерфейси налагодження.

Secure Enclave AES Engine - це апаратний блок, який використовується для виконання симетричної криптографії на основі шифру AES. Механізм AES розроблено для захисту від витоку інформації за допомогою синхронізації та SPA. Починаючи з A9 SoC, AES Engine також включає засоби протидії DPA.

AES Engine підтримує апаратні та програмні ключі. Апаратні ключі походять від Secure Enclave UID або GID. Ці ключі залишаються в AES Engine і не стають видимими навіть для програмного забезпечення sepOS. Хоча програмне забезпечення може запитувати операції шифрування та дешифрування за допомогою апаратних ключів, воно не може отримати ключі.

На процесорах Apple A10 і новіших SoC AES Engine включає початкові біти, які можна заблокувати, які урізноманітнюють ключі, отримані з UID або GID. Це дозволяє обумовлювати доступ до даних залежно від режиму роботи пристрою. Наприклад, початкові біти, які можна заблокувати, використовуються для заборони доступу до захищених паролем даних під час завантаження з режиму оновлення мікропрограми пристрою DFU.

Кожен пристрій Apple із Secure Enclave також має спеціальну систему шифрування AES256 - AES Engine, вбудовану в шлях DMA між NAND енергонезалежною флеш-пам'яттю та основною системною пам'яттю, що робить шифрування файлів високоефективним. На A9 або пізніших процесорах серії A підсистема флеш-пам'яті знаходиться на ізольованій шині, якій надається доступ лише до пам'яті, що містить дані користувача, через механізм шифрування DMA.

Під час завантаження sepOS генерує тимчасовий ключ шифрування (EWK) за допомогою TRNG. Secure Enclave передає цей ключ до AES Engine за допомогою спеціальних доріжок, призначених для запобігання доступу будь-якого програмного забезпечення за межами Secure Enclave. Потім sepOS може використовувати тимчасовий ключ EWK для шифрування ключів файлів з подальшим використання драйвером файлової системи Application Processor. Коли драйвер файлової системи читає або записує файл, він надсилає зашифрований ключ до AES Engine, який розшифровує ключ. Механізм AES ніколи не надає програмному забезпеченню розшифрований ключ.

AES Engine є окремим компонентом як від Secure Enclave, так і від Secure Enclave AES Engine, але його робота тісно пов'язана з Secure Enclave, як показано нижче (див.рисунок 2.2).

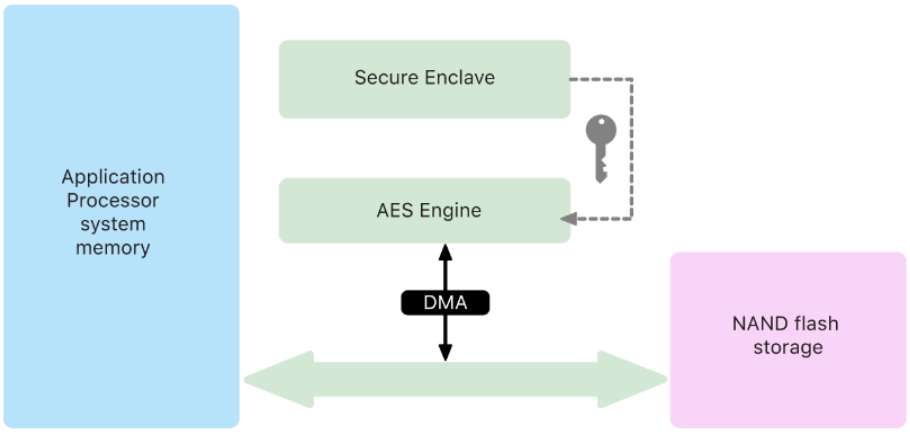


Рисунок 2.2 – Принцип взаємодії Secure Enclave з AES Engine

PKA - це апаратний блок, який використовується для виконання операцій асиметричної криптографії. PKA підтримує алгоритми підписання та шифрування RSA та ECC. PKA розроблений, щоб протистояти витоку інформації за допомогою синхронізації та атак на бокових каналах, таких як SPA та DPA. PKA підтримує програмні та апаратні ключі. Апаратні ключі походять від Secure Enclave UID або GID. Ці ключі залишаються в PKA і не стають видимими навіть для програмного забезпечення sepOS. На Apple A10 і пізніших SoC PKA підтримує пов'язані з ОС ключі, які також називають захистом запечатаних ключів SKP. Ці ключі генеруються за допомогою комбінації UID пристрою та хешу sepOS, запущеного на пристрої. Хеш забезпечує Secure Enclave Boot ROM або Secure Enclave Boot Monitor на Apple A13 і новіших SoC. Ці ключі також використовуються для перевірки версії sepOS під час надсилання запитів до певних служб Apple, а також для покращення безпеки даних, захищених паролем, допомагаючи запобігти доступу до матеріалів із ключами, якщо критичні зміни вносяться до системи без авторизації користувача.

Secure Enclave оснащено спеціальним безпечним енергонезалежним пристроєм зберігання. Захищене енергонезалежне сховище підключено до Secure Enclave за допомогою спеціальної шини I2C, тому до нього може отримати доступ лише Secure Enclave. Усі ключі шифрування даних користувача базуються на ентропії, що зберігається в енергонезалежному сховищі Secure Enclave.

У пристроях із SoC A12, S4 і пізнішими Secure Enclave поєднується з компонентом Secure Storage для ентропійного зберігання. Компонент Secure Storage сам розроблений із незмінним кодом ROM, апаратним генератором випадкових чисел, унікальним криптографічним ключем для кожного пристрою, механізмами криптографії та виявленням фізичного втручання. Secure Enclave і Secure Storage Component взаємодіють за допомогою зашифрованого та автентифікованого протоколу, який надає ексклюзивний доступ до ентропії.

2.2 Механізм Data Protection

Apple використовує технологію під назвою Data Protection для захисту даних, що зберігаються у флеш-пам’яті на пристроях Mac із Apple Silicon [10]. За допомогою Data Protection пристрій може забезпечити високий рівень шифрування даних користувача. Певні системні програми використовують механізм Data Protection за замовчуванням. Програми сторонніх розробників отримують цей захист автоматично [11].

Захист даних реалізується шляхом побудови та керування ієрархією ключів і базується на технологіях апаратного шифрування, вбудованих у пристрої Apple. Захист даних контролюється на основі кожного файлу шляхом присвоєння кожному файлу класу. Доступність визначається відповідно до того, чи були розблоковані ключі класу. APFS дозволяє файловій системі додатково розділяти ключі на основі екстентів де частини файлу можуть мати різні ключі. APFS – це файлова система за замовчуванням для комп’ютерів Mac із MacOS 10.13 або новішої версії. APFS має надійне шифрування, спільне використання простору, знімки, швидка зміна розміру каталогу та вдосконалені основи файлової системи.

Щоразу, коли створюється файл на томі даних, Data Protection створює новий 256-бітний ключ (ключ для кожного файлу) і передає його апаратному механізму AES Engine, який використовує ключ для шифрування файлу під час його запису у флеш-пам’ять. зберігання. На пристроях A14–A17 і M1–M3 шифрування використовує AES-256 у режимі XTS, де 256-бітний ключ для

кожного файлу проходить через Key Derivation Function згідно NIST 800-108, щоб отримати 256-бітний ключ налаштування (tweak) та 256-бітний ключ шифрування. На пристроях A9–A13 і S5–S9 шифрування використовує AES-128 у режимі XTS, де 256-бітний ключ на файл розділяється, щоб забезпечити 128-бітове налаштування та 128-бітний ключ шифрування [6].

На комп'ютері Mac із процесором Apple для захисту даних за замовчуванням використовується клас захисту даних C, але використовується ключ тому, а не ключ для екстенту чи файлу, фактично відтворюючи модель безпеки FileVault для даних користувача..

На пристроях Apple із захистом даних кожен файл захищено унікальним ключем. Ключ, зашифрований за допомогою алгоритму шифрування ключів NIST AED, додатково шифрується одним із кількох ключів класу, залежно від того, як передбачається отримати доступ до файлу. Зашифрований ключ файлу потім зберігається в метаданих файлу.

Пристрої з форматом APFS можуть підтримувати клонування файлів. Якщо файл клоновано, кожна половина клону отримує новий ключ для прийняття вхідних записів, щоб нові дані записувалися на носій із новим ключем. З часом файл може складатися з різних фрагментів, кожен з яких відповідає різним ключам. Однак усі фрагменти, які складають файл, захищаються одним і тим же ключем класу.

Коли файл відкривається, його метадані розшифровуються за допомогою ключа файлової системи, відкриваючи зашифрований ключ для кожного файлу та запис про те, який клас його захищає. Ключ для кожного файлу (або для кожного фрагменту) розшифровується разом із ключем класу, а потім передається апаратному механізму AES Engine, який розшифровує файл під час його читання з флеш-пам'яті. Уся обробка ключів розшифрованого файлу відбувається в Secure Enclave; ключ файлу ніколи не піддається безпосередньому впливу Application Processor. Під час запуску Secure Enclave узгоджує тимчасовий ключ із AES Engine.

Вміст файлу може бути зашифровано за допомогою одного або кількох ключів для кожного файлу (фрагменту), які зашифровані ключем класу та

зберігаються в метаданих файлу, які, у свою чергу, зашифровані за допомогою ключа файлової системи. Ключ класу захищено апаратним UID і для деяких класів також паролем користувача. Ця ієрархія забезпечує як гнучкість і продуктивність. Наприклад, зміна класу файлу вимагає лише перешифрування ключа для кожного файлу, а зміна коду доступу лише переоформлює ключ класу.

2.3 Файлова система APFS

Файлова система APFS - це власна файлова система Apple, розроблена з урахуванням шифрування. APFS працює на всіх платформах [12]. Оптимізована для зберігання на флеш-пам’яті або SSD, вона має надійне шифрування, копіювання під час запису метаданих, спільне використання простору, клонування файлів і каталогів, знімки, швидку зміну розміру каталогу, атомарні примітиви безпечного збереження та вдосконалені основи файлової системи, а також унікальну конструкція копіювання при записі, яка використовує об’єднання вводу-виводу для забезпечення максимальної продуктивності, забезпечуючи при цьому надійність даних.

APFS виділяє місце для зберігання на вимогу. Якщо один контейнер APFS має кілька томів, вільний простір контейнера є спільним і може бути виділений будь-якому з окремих томів за потреби. Кожен том використовує лише частину загального контейнера, тому доступний простір - це загальний розмір контейнера мінус простір, який використовується в усіх томах у контейнері.

У MacOS 10.15 або пізнішої версії контейнер APFS, який використовується для запуску Mac, повинен містити принаймні п’ять томів, перші три з яких приховані від користувача. Передзавантажувальний том незашифрований і містить дані, необхідні для завантаження кожного системного тому в контейнері. Том віртуальної машини незашифрований і використовується MacOS для зберігання зашифрованих файлів підкачки. Том відновлення незашифрований і має бути доступним без розблокування для запуску системного тому в recoveryOS. Системний том містить усі необхідні файли для запуску Mac та усі програми, інстальовані MacOS (/System/Applications). За замовчуванням жоден

процес не може писати на системний том, навіть системні процеси Apple. Том з даними містить дані, які можуть змінюватися. Це можуть бути будь-які дані в папці користувача, включаючи фотографії, музику, відео та документи. Також програми, які встановив користувач, спеціальні фреймворки та демони, встановлені користувачем. Інші розташування, які належать користувачу та доступні для запису.

Том даних створюється для кожного додаткового системного тому. Том передзавантаження, віртуальної машини та відновлення є спільними та не дублюються.

У MacOS 11 або новішої версії системний том фіксується на знімку. Операційна система завантажується зі знімка системного тому, а не лише з монтування змінного системного тому, доступного лише для читання.

У MacOS 10.15 Apple представила системний том лише для читання, виділений ізольований том для системного вмісту. MacOS 11 або пізнішої версії додає потужний криптографічний захист системного вмісту за допомогою підписаного системного тому SSV. SSV містить механізм ядра, який перевіряє цілісність системного вмісту під час виконання та відхиляє будь-які дані (кодові та некодовані) без дійсного криптографічного підпису від Apple.

SSV не тільки допомагає запобігти втручанню в будь-яке програмне забезпечення Apple, яке є частиною операційної системи, але й робить оновлення програмного забезпечення MacOS надійнішим і набагато безпечнішим. А оскільки SSV використовує знімки APFS, якщо оновлення не вдається виконати, стару версію системи можна відновити без переустановлення.

З моменту появи APFS забезпечує цілісність метаданих файлової системи за допомогою некриптографічних контрольних сум на внутрішньому пристрої зберігання. SSV посилює механізм цілісності, додаючи криптографічні хеші, таким чином розширюючи його, щоб охоплювати кожен байт даних файлу. Дані з внутрішнього накопичувача (включаючи метадані файлової системи) криптографічно хешуються на шляху зчитування, а потім хеш порівнюється з очікуваним значенням у метаданих файлової системи [13]. У разі невідповідності

система припускає, що дані були підроблені, і не повертатиме їх програмному забезпеченню, яке запитує.

Кожен SHA256 хеш SSV зберігається в основному дереві метаданих файлової системи, яке саме хешується. І оскільки кожен вузол дерева рекурсивно перевіряє цілісність хешів своїх нащадків, подібно до бінарного хеш-дерева (Merkle), хеш-значення кореневого вузла, яке називається seal, охоплює кожен байт даних у SSV, що означає криптографічний підпис покриває весь обсяг системи.

Під час інсталяції та оновлення MacOS seal повторно обчислюється з файлової системи на пристрої, і це вимірювання звіряється з вимірюванням, підписаним Apple. На Mac із Apple Silicon завантажувач перевіряє seal перед передачею керування ядру. На комп'ютері Mac на базі Intel із чіпом безпеки Apple T2 завантажувач пересилає вимірювання та підпис до ядра, яке потім перевіряє seal безпосередньо перед монтуванням кореневої файлової системи. У будь-якому випадку, якщо перевірка не вдається, процес запуску зупиняється, і користувачеві пропонується перевстановити MacOS. Ця процедура повторюється під час кожного завантаження, якщо користувач не вибрав режим нижчого рівня безпеки та окремо вимкнув підписаний системний том.

2.4 Механізм шифрування томів за допомогою FileVault

Комп'ютери Mac пропонують FileVault, вбудовану можливість шифрування, щоб захистити всі дані, що знаходяться в спокої. FileVault використовує алгоритм шифрування даних AES-XTS для захисту повних томів на внутрішніх і знімних пристроях зберігання [14].

FileVault на Mac із Apple Silicon реалізовано за допомогою Data Protection Class C із ключем тому. На Mac з Apple Silicon і Mac з Apple T2 Security Chip зашифровані внутрішні накопичувачі, безпосередньо підключені до Secure Enclave, використовують апаратні можливості безпеки, а також механізм AES. Після того, як користувач увімкне FileVault на Mac, його облікові дані будуть потрібні під час процесу завантаження.

Для комп’ютерів Mac до комп’ютерів із чіпом T2 або із внутрішньою пам’яттю, яка спочатку не постачалася з Mac, або із підключеним зовнішнім накопичувачем після ввімкнення FileVault усі наявні файли та будь-які інші записані дані будуть зашифровані. Дані, які було додано, а потім видалено перед увімкненням FileVault, не шифруються, і їх можна відновити за допомогою інструментів відновлення даних.

Без дійсних облікових даних для входу або криптографічного ключа відновлення, внутрішні томи APFS залишаються зашифрованими та захищені від несанкціонованого доступу, навіть якщо фізичний пристрій зберігання даних підключено до іншого комп’ютера. У MacOS 10.15 це включає як системний том, так решту даних на інших томах. Починаючи з MacOS 11, системний том захищено функцією підписаного системного тому SSV, але том даних залишається захищеним шифруванням. SSV забезпечує еквівалентний захист у стані спокою для системного вмісту, тому системний том більше не потрібно шифрувати. Будь-які зміни, внесені у файлову систему, коли вона перебуває в стані спокою, виявляються файловою системою під час їх читання. Якщо користувач увімкнув FileVault, вміст користувача в томі даних усе ще зашифровано за допомогою наданого користувачем секрету.

Якщо користувач вирішує вимкнути SSV, система в стані спокою стає вразливою до втручання, і таке втручання може дозволити зловмиснику отримати зашифровані дані користувача під час наступного запуску системи. Тому система не дозволить користувачеві вимкнути SSV, якщо FileVault увімкнено. Захист у стані спокою потрібно ввімкнути або вимкнути для обох томів узгоджено.

У MacOS 10.15 або раніших версіях FileVault захищає програмне забезпечення операційної системи в стані спокою шляхом шифрування вмісту користувача та системи за допомогою ключа, захищеного наданим користувачем секретом. Це захищає від зловмисника, який має фізичний доступ до пристрою, від доступу або ефективної зміни файлової системи, що містить системне програмне забезпечення.

Внутрішнє шифрування томів на комп'ютерах Mac із Apple Silicon, а також з чіпом T2 реалізується шляхом побудови та керування ієрархією ключів і базується на технологіях апаратного шифрування, вбудованих у чіп (див. рисунок 2.3).

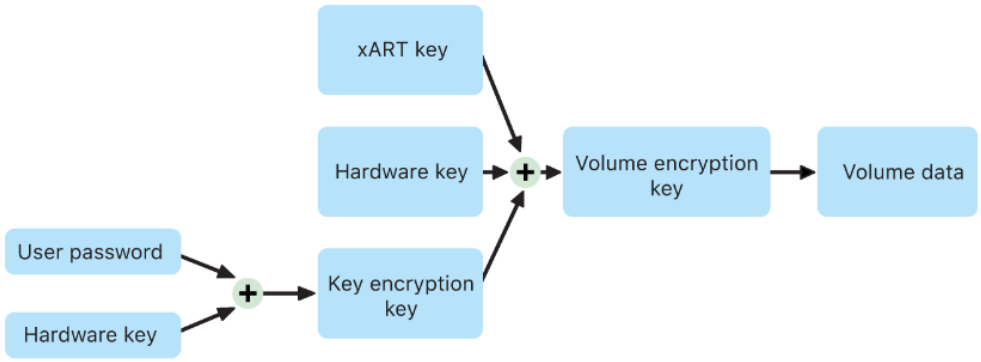


Рисунок 2.3 – Взаємодія та керування ієрархією ключів при ввімкненому FileVault

Ця ієрархія ключів призначена для одночасного досягнення чотирьох цілей:

- вимагати пароль користувача для розшифрування;
- захистити систему від brute-force атак безпосередньо на носій даних, який видалено з Mac;
- забезпечити швидкий і безпечний метод видалення вмісту шляхом видалення необхідного криптографічного матеріалу;
- дозволити користувачам змінювати свій пароль та криптографічні ключі, які використовуються для захисту файлів, не вимагаючи повторного шифрування всього тому.

На Mac з Apple Silicon та Mac з чіпом T2 уся обробка ключів FileVault відбувається в Secure Enclave; ключі шифрування ніколи не піддаються безпосередньому впливу ЦП Intel. За замовчуванням усі томи APFS створюються з ключем шифрування томів. Вміст томів і метаданих зашифровано за допомогою ключа шифрування тому, який містить ключ шифрування ключа KEK. Коли FileVault увімкнено, KEK захищено комбінацією пароля користувача та UID апаратного забезпечення.

Якщо FileVault не ввімкнуто на комп'ютері Mac із процесором Apple Silicon або комп'ютері Mac із мікросхемою T2 під час початкового процесу Setup Assistant, том усе ще зашифровано, але ключ шифрування тому захищено лише UID апаратного забезпечення в Secure Enclave.

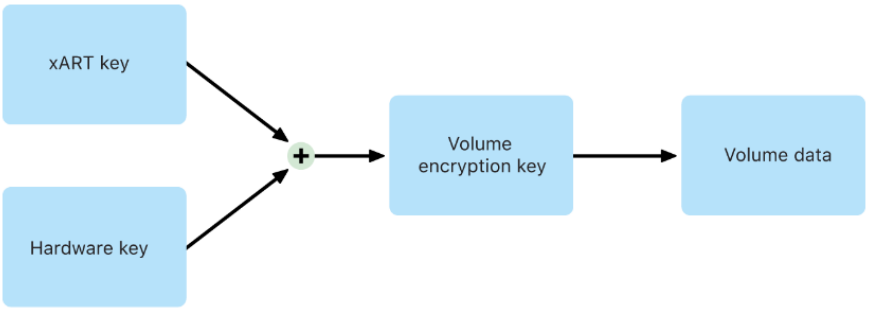


Рисунок 2.4 – Взаємодія та керування ієрархією ключів при вимкненому FileVault

Якщо FileVault увімкнути пізніше процес відбудеться негайно, оскільки дані вже зашифровано. Механізм запобігання повторному відтворенню допомагає запобігти використанню старого ключа (лише на основі UID обладнання) для розшифрування тому. Потім том захищається комбінацією пароля користувача з UID обладнання, як описано раніше.

Під час видалення тому його ключ шифрування безпечно видаляється за допомогою Secure Enclave. Це допомагає запобігти майбутньому доступу за допомогою цього ключа навіть з боку Secure Enclave. Крім того, усі ключі шифрування томів зашифровані за допомогою медіа-ключа. Медіа-ключ не забезпечує додаткової конфіденційності даних. Він розроблений для забезпечення швидкого та безпечного видалення даних, оскільки без нього розшифрування неможливе.

На комп'ютерах Mac із Apple Silicon і комп'ютерах Mac із мікросхемою T2 медіа-ключ гарантовано видаляється за допомогою технології, що підтримується в Secure Enclave, наприклад за допомогою віддалених команд MDM. Таким чином стирання медіа-ключа робить том криптографічно недоступним.

Шифрування знімних пристроїв зберігання даних не використовує можливості безпеки Secure Enclave, і його шифрування виконується так само, як і Mac на базі Intel без чіпа T2.

На томах APFS в macOS 10.13 і новіших відбулися суттєві зміни в управлінні ключами шифрування FileVault. У попередніх версіях, коли використовувалися томи CoreStorage, ключі шифрування створювалися під час увімкнення FileVault користувачем або організацією. Однак, з APFS, ключі шифрування генеруються під час створення користувача, задання пароля або першої авторизації користувача на Mac. Це дозволяє забезпечити безпеку з самого початку використання системи [15].

Важливим нововведенням є механізм, відомий як токен безпеки (Secure Token). Токен безпеки є ізольованою версією Key Encryption Key (KEK), захищеною паролем користувача. Токен створюється під час першого входу користувача або під час створення облікового запису, що гарантує захищений доступ до зашифрованих даних.

Нове управління шифруванням в APFS робить шифрування даних більш безпечним і інтегрованим у саму файлову систему, забезпечуючи додатковий рівень захисту для даних користувачів з моменту створення облікового запису або першого входу в систему.

2.5 Висновки до розділу

У другому розділі було проведено огляд основних компонентів безпеки комп’ютерів Mac. Описано захищену підсистему Secure Enclave, яка ізольована від головного процесора для забезпечення додаткового рівня безпеки. Проказано що в своїй роботі Secure Enclave використовує спеціальний генератор випадкових чисел TRNG, який базується на блокових шифрах у режимі CTR і власний механізм AES Engine з алгоритмом AES256, який підтримує апаратні та програмні ключі. Показано принцип роботи PKA, який є апаратним блоком, що використовується для виконання операцій асиметричної криптографії. PKA підтримує алгоритми підписання та шифрування RSA та ECC.

Проведено опис механізму Data Protection для захисту даних, що зберігаються у флеш-пам’яті на пристроях Mac із Apple Silicon. Показано, що Data Protection створює новий 256-бітний ключ (ключ для кожного файлу) і передає його апаратному механізму AES Engine, який використовує ключ для шифрування файлу під час його запису у флеш-пам’ять з використанням алгоритму шифрування AES-256 у режимі XTS.

Проведено огляд файлової системи APFS, яка розроблена з урахуванням шифрування та є власною файловою системою Apple. Описано криптографічний захист системного вмісту за допомогою підписаного системного тому SSV з використанням SHA256 хешу.

Описано механізм шифрування внутрішніх і знімних пристроїв зберігання за допомогою FileVault, який використовує алгоритм шифрування даних AES-XTS. Показано взаємодію FileVault та Secure Enclave.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ШИФРУВАННЯ В ОПЕРАЦІЙНІЙ СИСТЕМІ MACOS

3.1 Процедура шифрування системного диску за допомогою FileVault

Щоб зашифрувати диск за допомогою FileVault на MacOS потрібно скористатися утилітою `fdesetup` [16]. Під час увімкнення FileVault потрібно зберегти ключ відновлення. Команда `fdesetup status` використовується для перевірки поточного статусу FileVault на Mac. Вона показує, чи ввімкнено FileVault і який стан шифрування диска (див. рисунок 3.1).

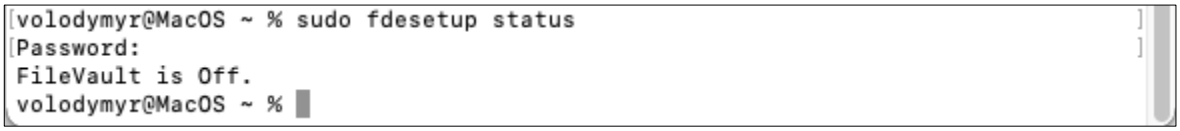


Рисунок 3.1 – Вивід статусу шифрування диска за допомогою FileVault

Вивід команди показує, що шифрування диска FileVault в даний момент вимкнене. Це означає, що дані на диску не зашифровані, коли система вимкнена, що може потенційно зробити їх більш доступними для несанкціонованих користувачів, особливо у випадку крадіжки або втрати обладнання. Для підвищення безпеки даних, особливо для ноутбуків або інших мобільних пристроїв, які можуть зберігати конфіденційну особисту або службову інформацію, потрібно увімкнути FileVault.

На рисунку 3.2 показано вивід команди для активації FileVault із збереженням ключа відновлення у файл.

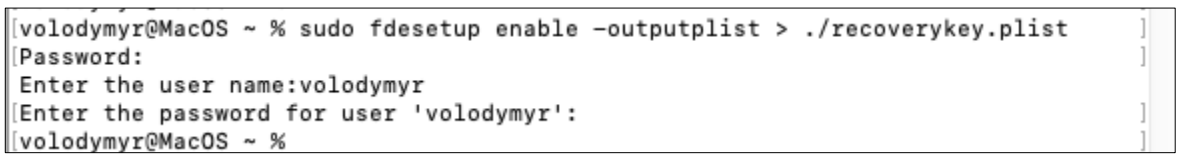


Рисунок 3.2 – Вивід команди активації FileVault

Команда увімкне FileVault та збереже ключ відновлення у вигляді plist-файлу. Згенерований ключ відновлення може бути використаний для розблокування диска у випадку забутого пароля користувача.

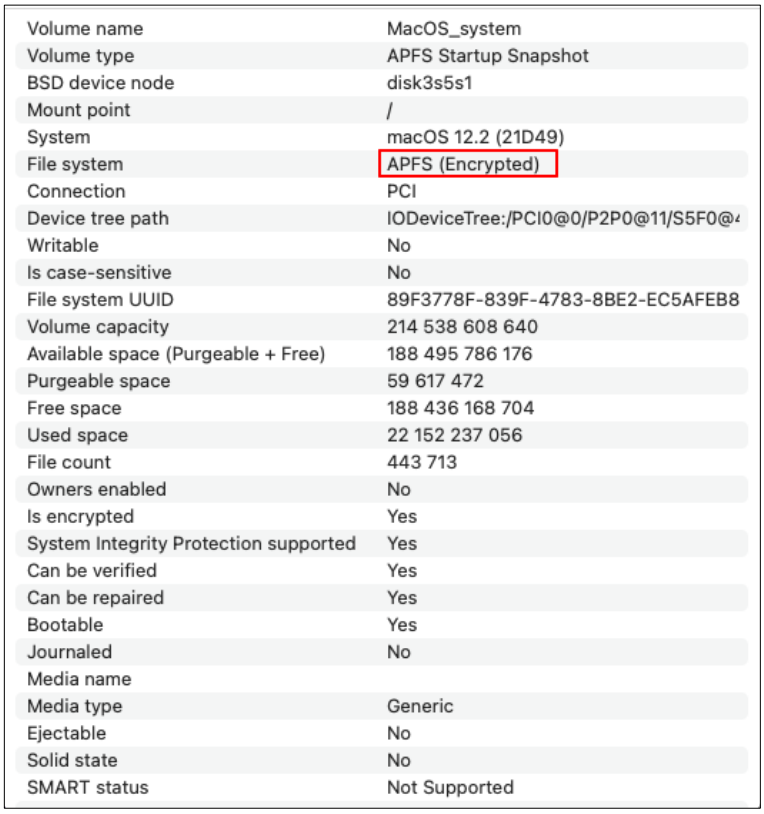
На рисунку 3.3 показано вивід команди для перевіряється статус шифрування FileVault.

```
volodymyr@MacOS ~ % sudo fdesetup status
FileVault is On.
Encryption in progress: Percent completed = 16
volodymyr@MacOS ~ %
```

Рисунок 3.3 – Вивід команди перевірки статусу шифрування FileVault

Вивід FileVault is On означає, що шифрування FileVault активоване на вашому пристрої. Вивід Encryption in progress: Percent completed = 16 показує, що процес шифрування ще не завершений і на даний момент зашифровано 16% даних на диску. Шифрування FileVault зазвичай запускається автоматично після його активації і може займати деякий час залежно від обсягу даних на диску і швидкості самого диску.

На рисунку 3.4 показано детальну інформацію про системний том на комп'ютері Mac, яка виведена за допомогою Disk Utility.



Volume name	MacOS_system
Volume type	APFS Startup Snapshot
BSD device node	disk3s5s1
Mount point	/
System	macOS 12.2 (21D49)
File system	APFS (Encrypted)
Connection	PCI
Device tree path	IODeviceTree:/PCI0@0/P2P0@11/S5F0@4
Writable	No
Is case-sensitive	No
File system UUID	89F3778F-839F-4783-8BE2-EC5AFEB8
Volume capacity	214 538 608 640
Available space (Purgeable + Free)	188 495 786 176
Purgeable space	59 617 472
Free space	188 436 168 704
Used space	22 152 237 056
File count	443 713
Owners enabled	No
Is encrypted	Yes
System Integrity Protection supported	Yes
Can be verified	Yes
Can be repaired	Yes
Bootable	Yes
Journaled	No
Media name	
Media type	Generic
Ejectable	No
Solid state	No
SMART status	Not Supported

Рисунок 3.4 – Вивід інформації про системний том

Статус `File System: APFS (Encrypted)` вказує на не, що APFS налаштована як зашифрована. Це забезпечує захист від несанкціонованого доступу до даних. Шифрування виконується на рівні файлової системи, що дозволяє користувачам безпечно зберігати чутливу інформацію та управляти доступом до даних через власні ключі шифрування та паролі. APFS також дозволяє здійснювати шифрування і дешифрування "на льоту", що мінімізує час, необхідний для доступу до зашифрованих даних, і робить процес прозорим для користувача.

На рисунку 3.5 показано вміст файла recoverykey.plist, який використовується в системі MacOS для зберігання ключ відновлення у форматі XML.

```
volodymyr@MacOS ~ % cat ./recoverykey.plist
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>EnabledDate</key>
        <string>2024-05-28 12:52:30 +0300</string>
        <key>EnabledUser</key>
        <string>volodymyr</string>
        <key>HardwareUUID</key>
        <string>564DAA5E-E206-BE2D-6101-3BF6E2E616A9</string>
        <key>RecoveryKey</key>
        <string>4JYZ-9LOR-3V3Y-63RX-5ENX-4V95</string>
        <key>SerialNumber</key>
        <string>VMIsypxTZcwR</string>
</dict>
</plist>
volodymyr@MacOS ~ %
```

Рисунок 3.5 – Вивід інформації про ключ відновлення FileVault

Цей файл містить ключі та значення, які пов'язані з файлом відновлення для FileVault. Значення параметрів наступне:

- `EnabledDate` - дата та час, коли було активовано відповідний ключ або параметр;
- `EnabledUsers` - містить інформацію про користувачів, для яких активовано цей ключ;
- `HardwareUUID` - універсальний унікальний ідентифікатор апаратного забезпечення;
- `RecoveryKey` - ключ, що використовується для відновлення або розшифрування даних, якщо основні ключі доступу втрачено або вони недоступні;
- `SerialNumber` - серійний номер, що використовується для додаткової ідентифікації пристрою.

При наступному завантаженні операційної системи MacOS буде виведено запит паролю користувача для розблокування системного диска та продовження процесу завантаження (див. рисунок 3.6).

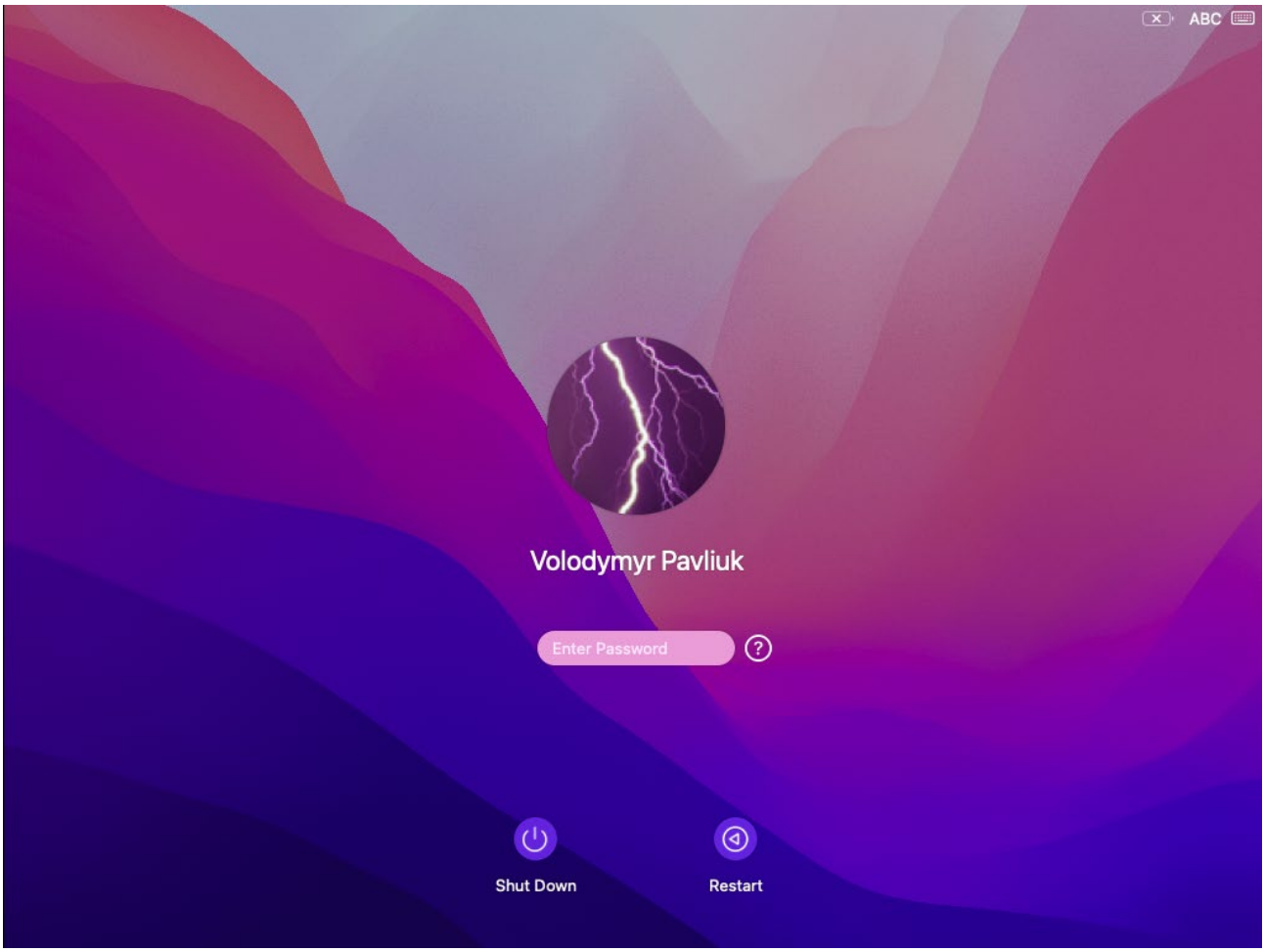


Рисунок 3.6 – Запит паролю користувача для розблокування системного диска та продовження процесу завантаження

Процес завантаження операційної системи MacOS з ввімкненим FileVault описано в пункті 2.4.

3.2 Процедура шифрування даних на окремому диску

Утиліта командного рядка `diskutil` в macOS використовується для управління дисками і томами [17]. За допомогою `diskutil` можна виконувати різні завдання, такі як перегляд списку дисків, форматування, створення, видалення розділів, управління шифруванням, і багато іншого. Утиліта надає потужні інструменти для адміністрації дисків як на фізичному, так і на логічному рівнях.

Команда `diskutil list` використовується для виводу списку всіх підключених дисків і їх розділів на MacOS. На рисунку 3.7 показано вивід даної команди.

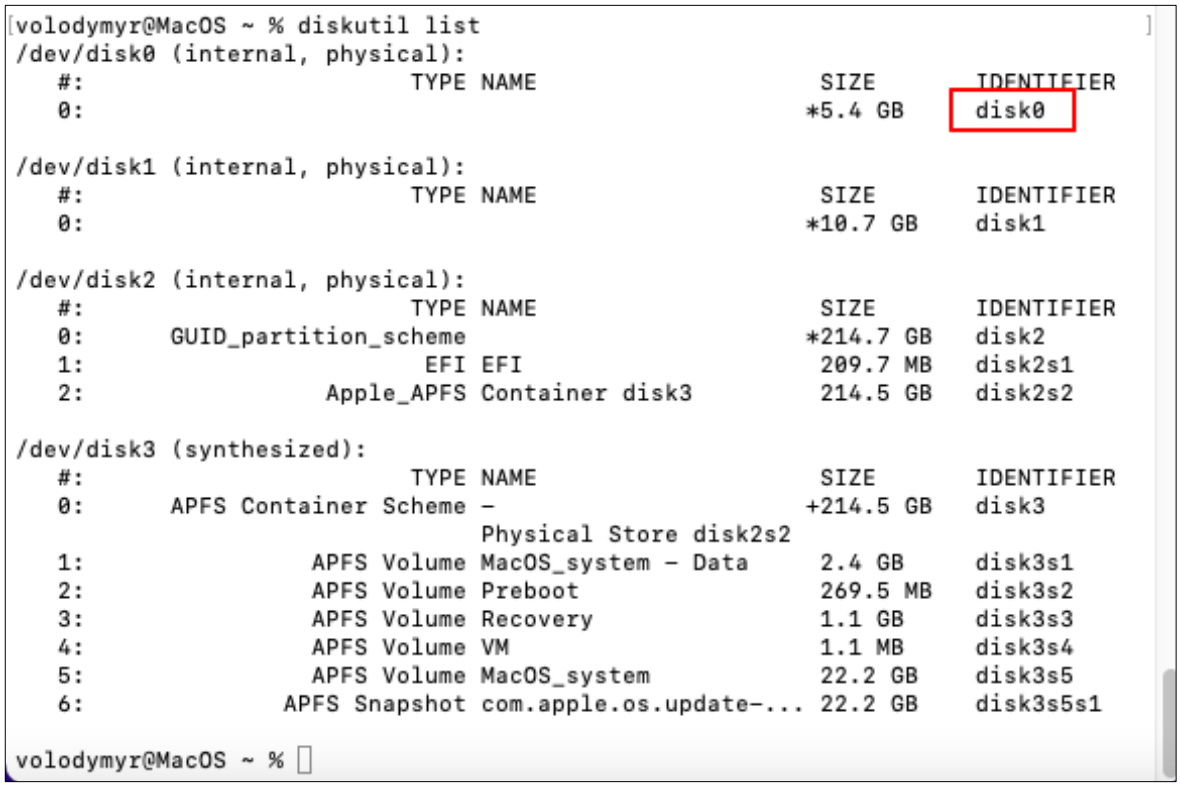


Рисунок 3.7 – Запит паролю користувача для розблокування системного диска та продовження процесу завантаження

Проведемо шифрування диску disc0 за допомогою `diskutil`. На рисунку 3.8 показано вивід терміналу MacOS, де виконується процес стирання та шифрування диска за допомогою команд `diskutil`.

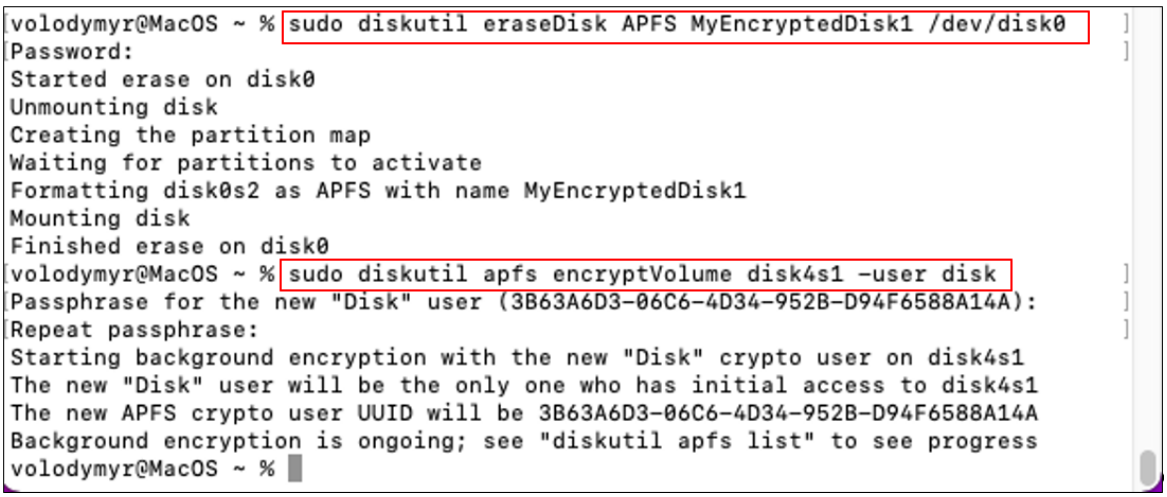


Рисунок 3.8 – Виконнання процесу стирання та шифрування диска за допомогою команд `diskutil`

Процес починається зі стирання диска `disk0`, форматування його у файлову систему APFS під назвою MyEncryptedDisk1. Диск демонтується, створюється таблиця розділів, і диск знову монтується після форматування.

Вказано, що диск `disk4s1` має бути зашифрований, і встановлюється пароль для відкриття диску. Вказується, що шифрування виконується у фоновому режимі і можна використати команду `diskutil apfs list` для перегляду прогресу шифрування.

Команда `diskutil list` використовується для виводу списку всіх підключених дисків і їх розділів на MacOS. На рисунку 3.7 показано вивід даної команди.

На рисунку 3.9 показано вивід команди `diskutil list` для перегляду всіх доступних дисків після виконнання процесу стирання та шифрування диска `disk0`.

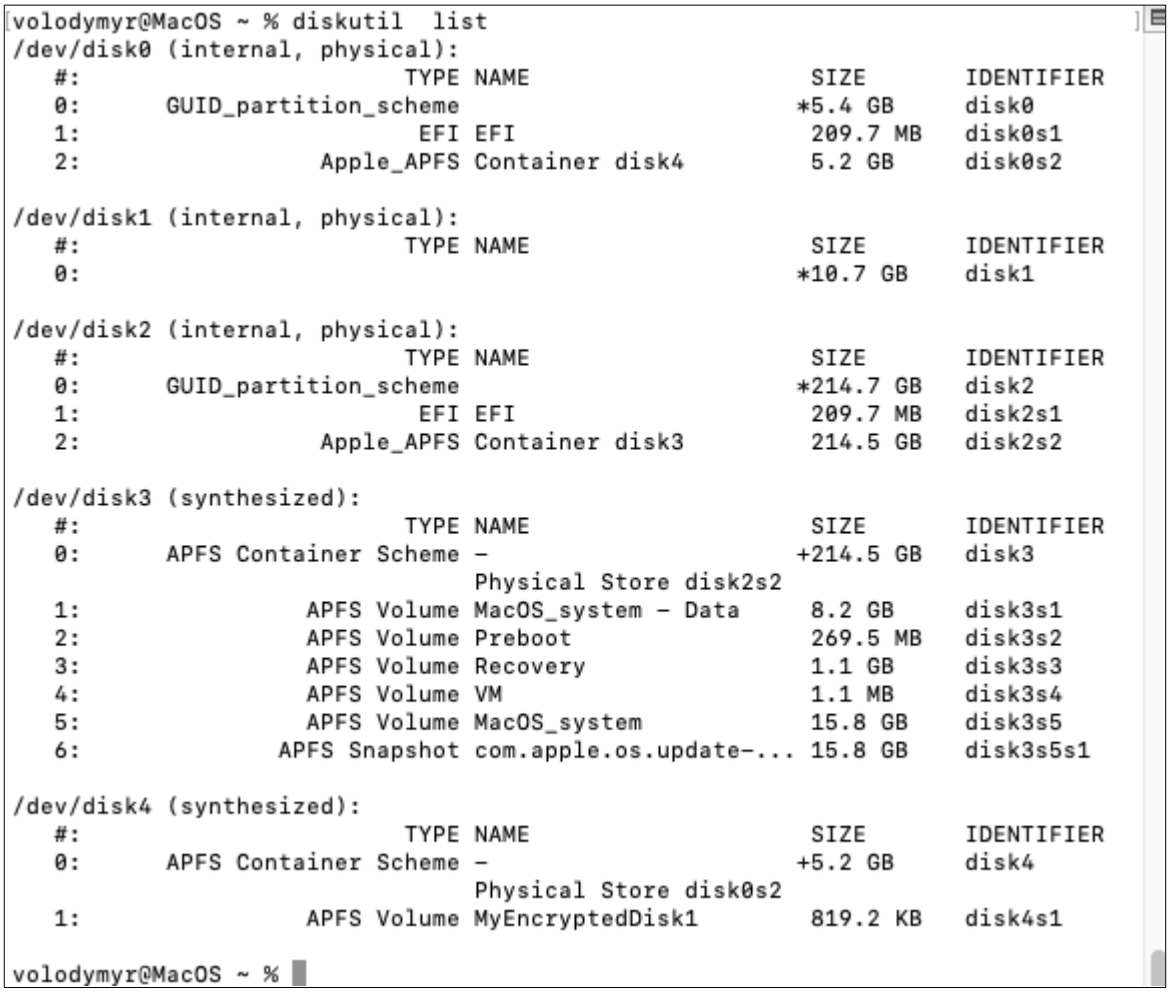


Рисунок 3.9 – Вивід команди `diskutil list` для перегляду всіх доступних дисків

В системі присутній диск `disk0` - це внутрішній фізичний диск об'ємом 5.4 GB. Він містить контейнер APFS, який має ідентифікатор `disk4`. Синтезований диск `disk4` - це контейнер APFS із фізичним сховищем `disk0s2`. У цьому контейнері міститься лише один том APFS `disk4s1` з назвою `MyEncryptedDisk1`.

На рисунку 3.10 показано вивід команди детального перегляду інформації про контейнер APFS на MacOS.

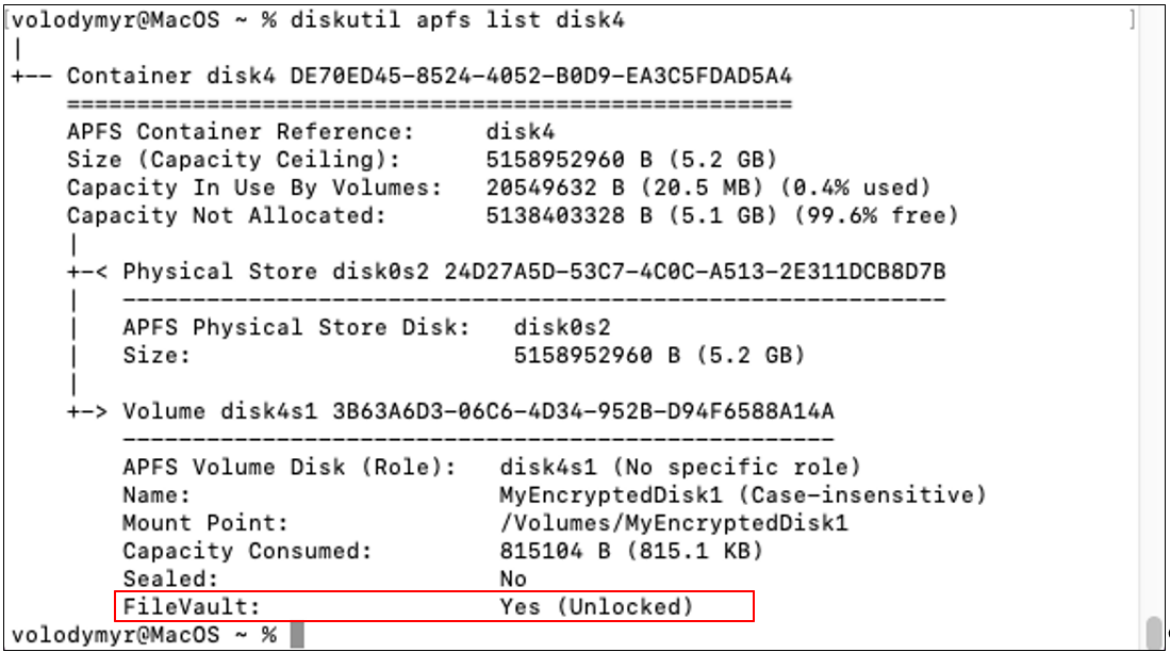


Рисунок 3.10 – Вивід команди детального перегляду інформації про контейнер `disk4`

Ця інформація вказує на те, що том `MyEncryptedDisk1` є зашифрованим і наразі відкритим, що дозволяє доступ до даних.

Команда `diskutil apfs list` надає детальну інформацію про APFS контейнери і включене шифрування, але не вказує конкретний алгоритм шифрування. В MacOS за замовчуванням використовує AES-XTS для шифрування FileVault (див. пункт 2.4).

3.3 Процедура створення зашифрованого дискового образу

У MacOS дисковий образ (disk image) є файлом, що може містити довільний вміст. Дискові образи дозволяють зберігати, передавати та відновлювати цілі файлові системи чи набори файлів у зручний спосіб. У MacOS дискові образи часто використовуються для створення резервних копій, дистрибуції програмного забезпечення, захисту даних за допомогою шифрування та інших завдань.

Формат DMG image - це найпоширеніший формат дискових образів у MacOS. Використовується для дистрибуції програмного забезпечення та даних. Підтримує шифрування і стиснення. ISO image – це стандартний формат для

створення копій оптичних дисків, таких як CD і DVD. Формат дискового образу sparsebundle збільшується в розмірі в міру додавання даних, зберігаючи окремі частини в окремих файлах (bundle). Формат sparseimage схожий на sparsebundle, але зберігається як єдиний файл.

Багато програм у MacOS розповсюджуються у вигляді DMG файлів. Користувачі завантажують DMG файл, монтують його та встановлюють програму. Дискові образи використовуються для створення резервних копій системи або даних. Наприклад, за допомогою Time Machine можна створювати резервні копії у вигляді sparsebundle образів. Дискові образи можуть бути зашифровані для захисту конфіденційної інформації. Шифровані дискові образи вимагають введення пароля для доступу до їх вмісту. Дискові образи можна монтувати як віртуальні диски, що дозволяє отримати доступ до їх вмісту так, як до звичайних фізичних дисків.

Щоб створити зашифрований дисковий образ у MacOS потрібно використати утиліту `hdiutil`.

Утиліта командного `hdiutil` використовується для створення та монтування дискових образів [18]. Ця утиліта забезпечує різноманітні функції для роботи з дисковими образами, включаючи створення зашифрованих образів, перетворення форматів образів, монтування образів як віртуальних дисків та багато іншого.

На рисунку 3.11 показано вивід команди створення зашифрованого дискового образу за допомогою утиліти `hdiutil`.

```
volodymyr@MacOS ~ % hdiutil create -size 1g -type SPARSEBUNDLE -fs 'APFS' -volna
me 'EncryptedImage' -encryption AES-256 -stdinpass ~/Desktop/EncryptedImage.spar
sebundle
Enter disk image passphrase:
created: /Users/volodymyr/Desktop/EncryptedImage.sparsebundle
volodymyr@MacOS ~ %
```

Рисунок 3.11 – Вивід команди створення зашифрованого дискового образу за допомогою утиліти `hdiutil`

Параметри команди наступні:

1) `-size 1g` - вказує розмір образу, який створюється, у цьому випадку 1 гігабайт;

2) `-type SPARSEBUNDLE` - вказує тип дискового образу як SPARSEBUNDLE, що дозволяє динамічно змінювати розмір образу залежно від використання;

3) `-fs 'APFS'` - встановлює файлову систему образу як APFS;

4) `-volname 'EncryptedImage'` - вказує назву тому, яка з'явиться при монтуванні образу;

5) `-encryption AES-256` - встановлює шифрування образу з алгоритмом AES-256;

6) `-stdinpass` - вказує, що пароль для шифрування буде введений з консолі.

Користувачу буде запропоновано ввести пароль для дискового образу. Після введення пароля буде створено дисковий образ з назвою EncryptedImage.sparsebundle на робочому столі користувача.

Цей зашифрований дисковий образ може використовуватися для безпечного зберігання конфіденційної інформації. Він буде вимагати введення пароля для доступу до збережених у ньому даних кожного разу, коли він монтується.

На рисунку 3.12 показано процес монтування дискового образу за допомогою команди `hdiutil attach` у macOS.

```
volodymyr@MacOS ~ % sudo hdiutil attach ~/Desktop/EncryptedImage.sparsebundle
Password:
Enter password to access "EncryptedImage.sparsebundle":
/dev/disk5              GUID_partition_scheme
/dev/disk5s1            Apple_APFS
/dev/disk6              EF57347C-0000-11AA-AA11-0030654
/dev/disk6s1            41504653-0000-11AA-AA11-0030654 /Volumes/EncryptedImage
volodymyr@MacOS ~ %
```

Рисунок 3.12 – Вивід команди монтування дискового образу

Користувачу буде запропоновано ввести пароль для доступу до зашифрованого образу. Після введення пароля образ успішно монтується, і система визначає новий пристрій.

Автоматична точка монтування для `/dev/disk6s1` вказана як `/Volumes/EncryptedImage`.

На рисунку 3.13 показано вивід команди `diskutil list` для перегляду доступних дисків після виконання процесу монтування дискового образу.

```
/dev/disk5 (disk image):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      GUID_partition_scheme                        +1.1 GB     disk5
   1:                 Apple_APFS Container disk6         1.1 GB     disk5s1

/dev/disk6 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +1.1 GB     disk6
                                 Physical Store disk5s1
   1:                APFS Volume EncryptedImage          24.6 KB    disk6s1

volodymyr@MacOS ~ % diskutil list
```

Рисунок 3.13 – Вивід команди `diskutil list` для перегляду змонтованого дискового образу

В системі присутній диск `disk5` - це дистокаий образ об'ємом 1.1 GB. Він містить контейнер APFS, який має ідентифікатор `disk6`. Синтезований диск `disk6` - це контейнер APFS із сховищем `disk5s1`. У цьому контейнері міститься лише один том APFS `disk6s1` з назвою `EncryptedImage`.

На рисунку 3.14 показано вивід терміналу MacOS, де виконується процес шифрування дискового образу за допомогою команд `diskutil`.

```
volodymyr@MacOS ~ % sudo diskutil apfs encryptVolume disk6s1 -user disk
Password:
Passphrase for the new "Disk" user (6C33B438-6C03-4489-9D3A-4559DA67E392):
Repeat passphrase:
Starting background encryption with the new "Disk" crypto user on disk6s1
The new "Disk" user will be the only one who has initial access to disk6s1
The new APFS crypto user UUID will be 6C33B438-6C03-4489-9D3A-4559DA67E392
Background encryption is ongoing; see "diskutil apfs list" to see progress
volodymyr@MacOS ~ %
```

Рисунок 3.14 – Виконання шифрування дискового образу за допомогою команд `diskutil`

Вказано, що диск `disk6s1` має бути зашифрований. Також встановлюється пароль для відкриття диску. Шифрування виконується у фоновому режимі та

можна використати команду `diskutil apfs list` для перегляду прогресу шифрування.

На рисунку 3.15 показано вивід команди детального перегляду інформації про контейнер APFS `disk6` на MacOS.

```
volodymyr@MacOS ~ % diskutil apfs list disk6
|
+-- Container disk6 2F92B858-0B3C-41B3-BE80-E6A8D9BF09C6
    ====================================================
    APFS Container Reference:     disk6
    Size (Capacity Ceiling):      1073700864 B (1.1 GB)
    Capacity In Use By Volumes:   3387392 B (3.4 MB) (0.3% used)
    Capacity Not Allocated:       1070313472 B (1.1 GB) (99.7% free)
    |
    +-< Physical Store disk5s1 30D9E564-7FC4-4942-B640-8E6BA71B1446
    |   -----------------------------------------------------------
    |   APFS Physical Store Disk:   disk5s1
    |   Size:                       1073700864 B (1.1 GB)
    |
    +-> Volume disk6s1 6C33B438-6C03-4489-9D3A-4559DA67E392
        ---------------------------------------------------
        APFS Volume Disk (Role):   disk6s1 (No specific role)
        Name:                      EncryptedImage (Case-insensitive)
        Mount Point:               /Volumes/EncryptedImage
        Capacity Consumed:         24576 B (24.6 KB)
        Sealed:                    No
        FileVault:                 Yes (Unlocked)
volodymyr@MacOS ~ %
```

Рисунок 3.15 – Вивід команди детального перегляду інформації про контейнер `disk6`

Ця інформація вказує на те, що том `EncryptedImage` є зашифрованим і відкритим, що дозволяє доступ до даних.

На рисунку 3.16 показано вікно Finder у MacOS, в якому відображено вміст зашифрованого дискового образу після його монтування.

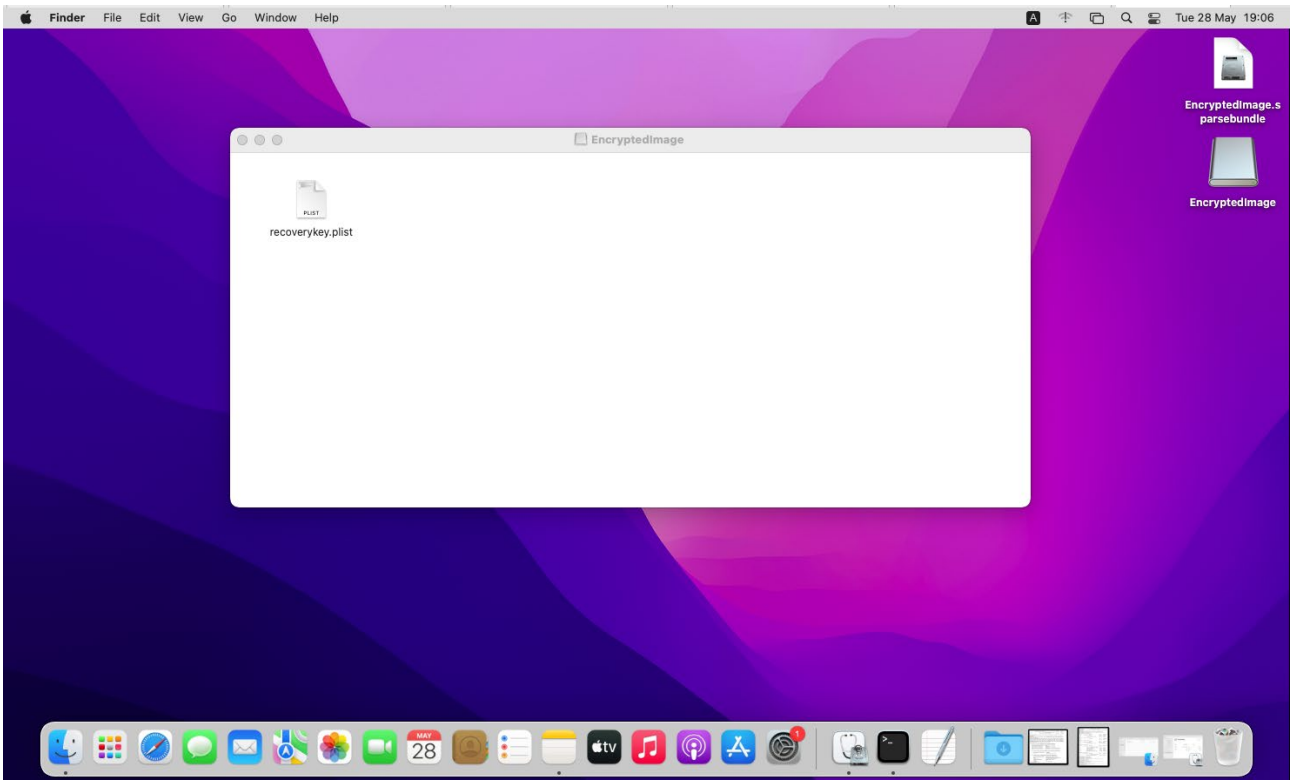


Рисунок 3.16 – Вміст зашифрованого дискового образу після його монтування

Вміст образу включає файл recoverykey.plist: Цей файл містить ключ відновлення FileVault, який був створений в пункті 3.1.

В подальшому зашифрований дисковий образ можна скопіювати на знімний носій та використовувати для безпечного зберігання та переміщення файлів між комп’ютерами Mac.

3.3 Висновки до розділу

В третьому розділі було проведено процедуру шифрування системного диску в операційній системі MacOS за допомогою FileVault. Показано можливості утиліти fdesetup. Показано, що при включенні FileVault відбувається конвертування файлової системи APFS в шифровану APFS. Також описано вміст ключа відновлення FileVault.

Показано процедуру шифрування даних на окремому диску за допомогою утиліта командного рядка diskutil. Описано параметри та можливості утиліти, що до перегляд списку дисків, форматування, створення, видалення розділів,

управління шифруванням. Показано етапи створення зашифрованої файлової системи APFS.

Показано процедуру створення зашифрованого дискового образу за допомогою утиліти hdiutil. Описано параметри та можливості утиліти при роботи з дисковими образами, включаючи створення зашифрованих образів, монтування образів як віртуальних дисків.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Загальні вимоги безпеки з охорони праці для користувачів ПК

В загальному, поняття охорона праці в комп’ютерних системах являє собою дотримання всіх вимог і нормативів, що присутні в законодавчих актах про охорону праці. Закони цієї області спрямовані на якісну і безпечну експлуатацію робочих приладів і приміщень, дотримання санітарно-гігієнічних умов праці і захист від інших небезпечних чинників на підприємстві. Ці засоби є складовими дослідження математичного і програмного забезпечення автоматизованої системи підбору команди розробників комп’ютерних систем. В основних законодавчих актах про охорону праці приділяється велика увага поліпшенню умов праці в усіх галузях господарства, впровадженню сучасних засобів техніки безпеки і забезпечення санітарно-гігієнічних умов, що запобігають виробничому травматизму і професійним захворюванням.

Охорона життя і здоров’я людини є пріоритетним напрямком соціальної політики держави. В Україні прийнято закон прямої дії «Про охорону праці», який регламентує захист конституційного права працівників на безпечні умови праці. Законодавство України про охорону праці складається із загальних законів України та спеціальних законодавчих актів. Загальними законами України, що визначають основні положення з охорони праці є Конституція України, Закон України «Про охорону праці», Кодекс законів про працю (КЗпП), Закон України «Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності».

Виконання досліджень кваліфікаційної роботи передбачали використання ПК, де площа та об'єм для одного робочого місця оператора визначається згідно вимог НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями», зокрема площа повинна становити не менше 6,0 квадратних метрів, об'єм - не менше 20,0 кубічних метрів.

Згідно вимог охорони праці та державних санітарних правил, стіни, стеля та підлога приміщень, в яких розміщені ЕОМ, повинні бути виготовлені з матеріалів, дозволених для оформлення приміщень органами державного санітарно-епідеміологічного нагляду.

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця операторів (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), повинні бути надійно захищені діелектричними щитками та сітками з метою недопущення потрапляння працівника під напругу.

Організація робочого місця оператора повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам. У приміщенні, де одночасно експлуатуються понад п'ять електронно-обчислювальних машин (ЕОМ), на помітному та доступному місці мають бути встановленні аварійні резервні вимикачі, які можуть повністю вимкнути електричне живлення приміщення, крім освітлення [19].

Дотримання правил значно знижує наслідки несприятливої дії на працівників шкідливих та небезпечних факторів, які супроводжують роботу з відео-дисплейними терміналами, зокрема можливість зорових, нервово-емоційних переживань, серцево-судинних захворювань. Виходячи з цього, роботодавець повинен забезпечити гігієнічні й ергономічні вимоги щодо організації робочих приміщень для експлуатації електронно-обчислювальних машин (ЕОМ) з ВДТ, робочого середовища, робочих місць з ЕОМ, режиму праці і відпочинку при роботі з ЕОМ тощо, які викладені у нормах НПАОП 0.00-7.15-18. Відповідно до встановлених гігієнічно-санітарних вимог роботодавець зобов'язаний забезпечити в приміщеннях з ЕОМ оптимальні параметри виробничого середовища [19].

Для захисту від прямих сонячних променів, які створюють прямі та відбиті відблиски з поверхні екранів персонального комп’ютера і клавіатури повинні бути передбачені сонцезахисні пристрої, вікна повинні мати жалюзі або штори.

Основні задачі охорони праці при використанні комп'ютерної техніки:

- аналіз впливу факторів виробничого середовища на здоров'я і працездатність користувачів персональних комп’ютерів;

- вдосконалювання методів оцінки працездатності і стану здоров'я користувачів ПК;

- розробка і впровадження організаційно-технічних, гігієнічних і соціально-економічних заходів щодо раціоналізації виробничого середовища;

- розробка і впровадження профілактичних і оздоровчих заходів, що дозволяють зберігати здоров'я людини і підвищувати її працездатність;

- вдосконалення методик навчання користувачів ПК питанням охорони праці.

Вимогам нормативних актів з охорони праці мають відповідати:

- умови праці на кожному робочому місці;

- безпека технологічних процесів, машин, механізмів, обладнання й інших засобів виробництва;

- стан засобів колективного та індивідуального захисту;

- санітарно-побутові умови.

При дослідженні методів і засобів створення програмних систем та проектуванні системи захисту від атак на сервери важливо було проаналізовати та враховати необхідні вимоги щодо охорони праці при використанні електронно-обчислювальної техніки і забезпечити умови для зручної та ефективної роботи працівників.

4.2 Підвищення стійкості роботи об’єктів господарської діяльності у воєнний час

Для покращення стійкості роботи об’єктів вивчають фактори, які впливають на стійкість та оцінюють стійкість елементів і галузей виробництва проти уражаючих факторів ядерної, хімічної і біологічної зброї, стихійних лих і виробничих аварій. Щоб підвищити стійкість необхідно завчасно організувати і провести організаційні, інженерно-технічні й технологічні заходи [20].

Здійснення організаційних заходів передбачає завчасну підготовку всіх структур цивільного захисту, служб і формувань до надзвичайних ситуацій, в тому числі і військових дій. Вжиттям технологічних заходів підвищується

стійкість роботи об'єктів шляхом змінювання технологічних процесів, режимів, можливих в умовах різних надзвичайних ситуацій. Інженерно-технічні заходи мають забезпечити підвищену стійкість виробничих споруд, технологічних ліній, устаткування, комунікацій об'єкта до впливу уражаючих факторів під час військових дій. При проведенні цих заходів необхідно враховувати конкретні умови об'єкта народного господарства. Проте є загальні організаційні інженерно-технічні заходи, які мають проводитись на всіх об'єктах.

Одним з найбільш важливих завдань в умовах воєнного часу і надзвичайних ситуацій є забезпечення захисту людей та їх життєдіяльності.

Для підвищення стійкості об’єктів господарювання та захисту людей необхідно:

– створити на об'єкті надійну систему оповіщення про загрози нападу противника, радіоактивне забруднення, хімічне і біологічне зараження, загрозу стихійного лиха і виробничої аварії;

– організувати розвідку і спостереження за радіоактивним забрудненням, хімічним і біологічним зараженням;

– організувати гідрометеорологічне спостереження за рівнем води, напрямком і швидкістю вітру, рухом і поширенням хмари радіоактивного забруднення, сильнодіючих отруйних речовин і отруйних речовин;

– створити фонд захисних споруд ЦО, запасів засобів індивідуального захисту і забезпечення своєчасної видачі їх населенню;

– завчасно підготуватись до масової санітарної обробки населення і знезаражування одягу;

– організувати взаємодію з установами охорони здоров'я для медичного обслуговування населення в умовах воєнного часу.

Також в умовах воєнного часу необхідно провести підготовку до евакуації населення, розміщеного в зонах можливих руйнувань і катастрофічного затоплення. Це передбачає завчасну підготовку місць евакуації, організацію прийому евакуйованого населення на територію населених пунктів. Окрім цього, необхідно забезпечити постачання продуктів харчування, питної води, предметів першої необхідності та провести заходи щодо морально-психологічної

підготовки населення до виживання в умовах воєнного часу, забезпечити процес чіткого інформування про обстановку та правила дій і поведінки населення в надзвичайних ситуаціях воєнного часу [20].

Для забезпечення стійкості роботи об'єктів повинні проводитись інженерно-технічні заходи на мережах комунального господарства з метою захисту джерел тепла із заглибленням у ґрунт комунікацій. Котельні слід розміщувати в спеціальному окремо розміщеному приміщенні.

Якщо об'єкт одержує тепло з міської теплоцентралі, необхідно провести заходи для забезпечення стійкості трубопроводів і розподільних пристроїв, підведених до об'єкта. Теплова мережа має будуватися за кільцевою системою з прокладанням труб у спеціальних каналах зі з'єднанням паралельних ділянок. Для відключення пошкоджених ділянок мають бути встановлені запірно-регулюючі засувки, вентилі та ін. Ці пристосування необхідно розміщувати в оглядових колодязях, на території, що не завалюється при руйнуванні будівель.

Система каналізації має будуватись окремо: одна для дощових, друга для промислових і господарських вод. На об'єкті має бути не менше двох виводів з підключенням до міських каналізаційних колекторів, а також виводи і колодязі з аварійними засувками на об'єктових колекторах з інтервалом 50 м на території, що не завалюється, для аварійного скидання неочищеної води в найближчі штучні та природні заглиблення.

На деяких промислових об'єктах є системи для забезпечення технології виробництва: для подання кисню, аміаку, стиснутого повітря та інших рідких і газових реактивів. Для цих систем розробляють заходи для попередження виникнення вторинних факторів зброї, стихійних лих та виробничих аварій і катастроф.

Створення резерву енергетичних потужностей за рахунок автономних пересувних електростанцій, а також місцевих джерел електроенергії. Підготовка автономних електростанцій до роботи за спеціальним режимом (графіком) для забезпечення технологічних процесів виробництва, для яких неможливі тривалі перерви в електропостачанні. З метою попередження аварій на електричних мережах необхідно установити автоматичну систему відключення при

виникненні перенапруги. Повітряні лінії електропостачання замінити на підземно-кабельні. Створення необхідних запасів (резервів) паливно-мастильних матеріалів та інших видів палива й організація їх безпечного зберігання.

Щоб не допустити зупинки підприємства через дефіцит палива, необхідно підготуватись для роботи на різних видах палива: нафта, вугілля, газ. Для підвищення стійкості забезпечення водою слід провести такі заходи.

Необхідно створити основні і резервні джерела водопостачання. Як резервне джерело краще мати артезіанську свердловину, яку необхідно підключити до системи водопостачання. Крім того, воду можна брати з близько розміщеної природної водойми або спорудити штучну водойму чи резервуари з обладнанням пристроїв для збору і перекачування води. Всі ділянки водопостачання повинні бути заглиблені в ґрунт з обладнанням пожежних гідрантів і пристроїв для відключення пошкоджених ділянок. Локальні мережі водопостачання окремих великих підприємств варто з'єднати із 80 загальноміською системою водопостачання в єдине кільце.

Підвищенню стійкості забезпечення водою сприяє подавання води безпосередньо в мережу поза водонапірними баштами, спорудження обвідних ліній для подання води поза пошкодженими спорудами.

Завчасне вжиття заходів захисту джерел водопостачання, водопровідних споруд, свердловин і шахтних колодязів від забруднення радіоактивними речовинами, зараження хімічними і біологічними засобами. Підготовка меліоративних, гідротехнічних та іригаційних споруд і систем до експлуатації в надзвичайних умовах.

Для забезпечення виробництва продукції необхідні електроенергія, паливо, мастила, засоби захисту рослин, міндобрива, профілактичні й лікувальні препарати ветеринарної медицини, запасні частини, сировина та інші матеріально-технічні засоби. Забезпечення об'єктів цими ресурсами дасть можливість випускати необхідну продукцію в надзвичайних умовах мирного і воєнного часу. Тому повинні проводитись такі заходи, які б забезпечили стійкість постачання і сприяли підвищенню захисту мережі електро-, водо-,

газопостачання, транспортних комунікацій і джерел постачання всім необхідним для забезпечення функціонування галузей сільського господарства в надзвичайних умовах.

З метою попередження аварій на електричних мережах необхідно встановити автоматичну систему відключення перенапруги. Повітряні лінії електропостачання слід замінити на підземно-кабельні. Газ використовується як паливо і на хімічних підприємствах у технологічному процесі. Для безперебійного забезпечення газом, газові мережі необхідно підводити до об'єкта з двох напрямків, які мають бути з'єднані в єдине кільце з обладнанням для можливого дистанційного автоматичного управління й у разі необхідності відключення пошкоджених ділянок. На великих підприємствах необхідно мати підземні ємності із закачаним резервним газом.

На підприємствах, де використовується пара, необхідно захистити джерела його постачання, заглибити в ґрунт комунікації паропостачання і встановити запірні пристосування.

Запас резервних матеріалів необхідно розраховувати на такі строки роботи підприємства, за які можливе відновлення регулярного постачання.

Передбачити, на випадок перебоїв в постачанні підприємствами-суміжниками, створення місцевих матеріалів, сировини для виготовлення комплектуючих виробів і інструментів силами свого підприємства [20].

Для підвищення стійкості та забезпечення збереження (відновлення) будівель і споруд в умовах воєнного часу необхідно:

– провести оцінку можливих ступенів руйнування будівель і споруд господарства населеного пункту, визначити обсяг невідкладних ремонтних робіт, потреби в будівельних матеріалах;

– створити і підготувати спеціальні формування для ремонтно-відновних, будівельних та інших робіт на об'єкті;

– розробити комплекс протипожежних заходів, які виключали б можливість виникнення масових пожеж.

Для забезпечення надійності системи управління і зв'язку потрібно організувати захищений пункт управління, забезпечити його засобами зв'язку,

які б дали можливість швидко доводити сигнали ЦЗ до всіх виробничих підрозділів і населення у місцях проживання. При цьому необхідно здійснити планування збору даних про обстановку, передачу команд і розпоряджень в умовах впливу на об'єкт уражуючих факторів. Для підвищення стійкості системи управління і зв’язку в умовах воєнного часу необхідно організувати використання радіозасобів, засобів телефонного зв'язку а також забезпечити зв’язок із колонами евакуйованого населення, що перебувають у дорозі, і відповідальними особами, які супроводжують їх під час евакуації, забезпечити дублювання ліній і каналів зв'язку.

ВИСНОВКИ

У процесі написання кваліфікаційної роботи було проведено аналіз та показано практичне використання механізмів шифрування в операційній системі MacOS.

У першому розділі було здійснено огляд безпеки платформи Apple. Продемонстровано, що Boot ROM забезпечує апаратний корінь довіри для безпечного завантаження системи, а спеціальні механізми AES використовуються для ефективного та безпечного шифрування і дешифрування на апаратному рівні. Проведено аналіз безпечного завантаження Mac з процесором Apple та на базі Intel із мікросхемою безпеки Apple T2. Встановлено, що пристрої Apple мають додаткові функції шифрування для захисту даних користувача, навіть якщо інші елементи інфраструктури безпеки були зламані.

У другому розділі проведено огляд основних компонентів безпеки комп'ютерів Mac. Описано захищену підсистему Secure Enclave, яка ізольована від головного процесора для забезпечення додаткового рівня безпеки. Показано, що у своїй роботі Secure Enclave використовує спеціальний генератор випадкових чисел TRNG, заснований на блокових шифрах у режимі CTR, а також власний механізм AES Engine з алгоритмом AES256, який підтримує апаратні та програмні ключі. Пояснено принцип роботи PKA - апаратного блоку, що виконує операції асиметричної криптографії, підтримуючи алгоритми підписання та шифрування RSA і ECC. Описано механізм Data Protection для захисту даних, що зберігаються у флеш-пам'яті на пристроях Mac із Apple Silicon. Показано, що Data Protection створює новий 256-бітний ключ для кожного файлу і передає його апаратному механізму AES Engine, який використовує цей ключ для шифрування файлу під час його запису у флеш-пам'ять за допомогою алгоритму AES-256 у режимі XTS.

Проведено огляд файлової системи APFS, яка розроблена з урахуванням шифрування та є власною файловою системою Apple. Описано криптографічний захист системного вмісту за допомогою підписаного системного тому SSV (signed system volume) з використанням хешу SHA256. Також описано механізм

шифрування внутрішніх і знімних пристроїв зберігання за допомогою FileVault, який використовує алгоритм шифрування даних AES-XTS. Показано взаємодію FileVault та Secure Enclave.

У третьому розділі була виконана процедура шифрування системного диску в операційній системі macOS за допомогою FileVault. Показано можливості утиліти fdesetup, яка використовується для управління FileVault. Пояснено, що під час включення FileVault відбувається конвертування файлової системи APFS в шифровану APFS. Також описано зміст ключа відновлення FileVault, який забезпечує можливість відновлення доступу до даних у випадку втрати основного пароля. Показано етапи процедури шифрування даних на окремому диску за допомогою утиліти командного рядка diskutil. Описано параметри та можливості цієї утиліти, включаючи перегляд списку дисків, форматування, створення та видалення розділів, а також управління шифруванням. Продемонстровано етапи створення зашифрованої файлової системи APFS. Була описана процедура створення зашифрованого дискового образу за допомогою утиліти hdiutil. Пояснено параметри та можливості цієї утиліти при роботі з дисковими образами, включаючи створення зашифрованих образів і монтування їх як віртуальних дисків.

Дослідження та практична реалізація підтвердили, що методи шифрування даних у стані спокою в операційній системі MacOS є ефективними і надійними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hardware security overview. URL: https://support.apple.com/uk-ua/guide/security/secf020d1074/1/web/1 (дата звернення: 29.05.2024).

2. System security overview. URL: https://support.apple.com/uk-ua/guide/security/sec114e4db04/1/web/1 (дата звернення: 29.05.2024).

3. Boot process for a Mac with Apple silicon. URL: https://support.apple.com/uk-ua/guide/security/secac71d5623/1/web/1 (дата звернення: 29.05.2024).

4. Karnaukhov, A., Tymoshchuk, V., Orlovska, A., & Tymoshchuk, D. (2024). USE OF AUTHENTICATED AES-GCM ENCRYPTION IN VPN. Матеріали конференцій МЦНД,(14.06. 2024; Суми Україна), 191-193. https://doi.org/10.62731/mcnd-14.06.2024.004

5. Encryption and Data Protection overview. URL: https://support.apple.com/uk-ua/guide/security/sece3bee0835/1/web/1 (дата звернення: 29.05.2024).

6. Тимощук, В., & Стебельський, М. (2023). Шифрування даних в операційних системах. Матеріали VI Міжнародної студентської науково-технічної конференції „Природничі та гуманітарні науки. Актуальні питання “, 183-184.

7. Secure Enclave URL: https://support.apple.com/uk-ua/guide/security/sec59b0b31ff/1/web/1 (дата звернення: 29.05.2024).

8. Stadnyk, M., Fryz, M., Zagorodna, N., Muzh, V., Kochan, R., Nikodem, J., & Hamera, L. (2022). Steady state visual evoked potential classification by modified KNN method. Procedia Computer Science, 207, 71-79.

9. Tymoshchuk, V., Karnaukhov, A., & Tymoshchuk, D. (2024). USING VPN TECHNOLOGY TO CREATE SECURE CORPORATE NETWORKS. Collection of scientific papers «ΛΌГОΣ», (June 21, 2024; Seoul, South Korea), 166-170. https://doi.org/10.36074/logos-21.06.2024.034

10. Data Protection in Apple devices. URL: https://support.apple.com/uk-ua/guide/security/sece8608431d/web (date of access: 29.05.2024).

11. Tymoshchuk, V., Dolinskyi, A., & Tymoshchuk, D. (2024). MESSENGER BOTS IN SMART HOMES: COGNITIVE AGENTS AT THE FOREFRONT OF THE INTEGRATION OF CYBER-PHYSICAL SYSTEMS AND THE INTERNET OF THINGS. Матеріали конференцій МЦНД, (07.06. 2024; Луцьк, Україна), 266-267. https://doi.org/10.62731/mcnd-07.06.2024.004

12. Role of Apple File System. URL: https://support.apple.com/en-vn/guide/security/seca6147599e/web (date of access: 29.05.2024).

13. Тимощук, В., Долінський, А., & Тимощук, Д. (2024). ЗАСТОСУВАННЯ ГІПЕРВІЗОРІВ ПЕРШОГО ТИПУ ДЛЯ СТВОРЕННЯ ЗАХИЩЕНОЇ ІТ-ІНФРАСТРУКТУРИ. Матеріали конференцій МЦНД, (24.05. 2024; Запоріжжя, Україна), 145-146. https://doi.org/10.62731/mcnd-24.05.2024.001

14. Volume encryption with FileVault in macOS. URL: https://support.apple.com/en-vn/guide/security/sec4c6dc1b6e/web (date of access: 29.05.2024).

15. Revniuk O.A., Zagorodna N.V., Kozak R.O., Karpinski M.P., Flud L.O. “The improvement of web-application SDL process to prevent Insecure Design vulnerabilities”. Applied Aspects of Information Technology. 2024; Vol. 7, No. 2: 162–174. DOI:https://doi.org/10.15276/aait.07.2024.12.

16. Тимощук, В., & Тимощук, Д. (2022). Віртуалізація в центрах обробки даних-аспекти відмовостійкості. Матеріали X науково-технічної конференції „Інформаційні моделі, системи та технології “Тернопільського національного технічного університету імені Івана Пулюя, 95-95.

17. Macintosh Terminal Pocket Guide by Daniel J. Barrett. URL: https://www.oreilly.com/library/view/macintosh-terminal-pocket/ 9781449328962/re68.html (дата звернення: 29.05.2024).

18. Kharchenko, A., Halay, I., Zagorodna, N., & Bodnarchuk, I. (2015). Trade-off optimal decision of the problem of software system architecture choice. In Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT 2015 (pp. 198-205)

19. Жидецький В.Ц. Охорона праці користувачів комп´ютерів. Львів: Афіша, 2011. 176 с.

20. Стручок В.С. Техноекологія та цивільна безпека. Частина «Цивільна безпека». Навчальний посібник. Тернопіль: ТНТУ. 2022. 150 с.