ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА СИСТЕМ ТА ТЕХНОЛОГІЙ КІБЕРБЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Рекомендації щодо застосування криптографічних засобів в локальній мережі організації»

зі спеціальності *125 Кібербезпека*
(код, найменування спеціальності)

освітньо-професійної програми *Інформаційна та кібернетична безпека*
(назва програми)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Михайло ЮРКОВСЬКИЙ
(підпис)

Виконав: здобувач вищої освіти групи БСД-44
ЮРКОВСЬКИЙ Михайло
(прізвище, ім'я)

Керівник д.т.н, проф. КОЖУХІВСЬКИЙ Андрій
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент _____
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

Кафедра Систем та технологій кібербезпеки
Ступінь вищої освіти Бакалавр
Спеціальність 125 Кібербезпека
Освітньо-професійна програма Інформаційна та кібернетична безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри СТКБ
_____ Гайдур Г.І.
“___”_____2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Юрковському Михайлу Сергійовичу
(прізвище, ім’я, по батькові)

1. Тема кваліфікаційної роботи: «Рекомендації щодо застосування криптографічних засобів в локальній мережі організації»

керівник кваліфікаційної роботи Кожухівський Андрій Дмитрович, д.т.н., проф.
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «___» _____2025 року № ____.

2. Строк подання студентом кваліфікаційної роботи 29.05.2025 р.

3. Вихідні дані до кваліфікаційної роботи
локальна обчислювальна мережа організації;
програмне забезпечення для організації криптографічного захисту;
наукова та технічна література, експлуатаційна документація, нормативні документи, міжнародні стандарти.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Актуальність проблеми криптографічного захисту в локальних мережах.
2. Огляд та класифікація сучасних криптографічних засобів і протоколів.

3. Практичні методи застосування криптогрфічного захисту в мережі.
4. Розробка рекомендацій щодо впровадження обраних засобів захисту.

5. Перелік графічного матеріалу
1. Тема кваліфікаційної роботи.
2. Об’єкт, предмет, мета та наукові завдання дослідження.
3. Аналіз сучасних засобів криптографічного захисту в локальних мережах.
4. Розробка рекомендацій щодо впровадження обраних засобів захисту.
5. Висновки за результатами роботи.

6. Дата видачі завдання 24.02.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1.	Визначення актуальності застосування криптографічного захисту в локальних мережах.	24.02.2025 р.	
2.	Аналіз наукової та технічної літератури з питань теми кваліфікаційної роботи.		
3.	Дослідження сучасних методів та засобів криптографічного захисту інформації.		
4.	Реалізація прикладу впровадження захисту в тестовому середовищі.		
5.	Розробка практичних рекомендацій щодо застосування обраних засобів криптографічного захисту.		
6.	Оформлення результатів дослідження.		
7.	Підготовка доповіді до захисту.	29.05.2025 р.	

Студент Юрковський М.С.
(підпис) прізвище та ініціали

Керівник кваліфікаційної роботи Кожухівський А.Д.
(підпис) прізвище та ініціал

РЕФЕРАТ

Текстова частина кваліфікаційної роботи: 47 сторінки, 14 рисунків, 32 джерел.

Об'єкт дослідження – локальна комп'ютерна мережа організації.

Предмет дослідження – методи та засоби криптографічного захисту інформації в локальній мережі.

Мета роботи – розробити рекомендації щодо впровадження ефективних криптографічних засобів захисту інформації в локальній мережі організації.

Методи дослідження – аналіз науково-технічної літератури, нормативних документів, міжнародних стандартів (ISO/IEC, RFC), порівняльний аналіз криптографічних протоколів, моделювання захищеної мережевої інфраструктури у віртуальному середовищі. Забезпечення криптографічного захисту інформації є критично важливим для організацій будь-якого масштабу, особливо в умовах зростання кіберзагроз і перехоплення трафіку. Локальні мережі потребують надійного захисту даних під час передавання, автентифікації користувачів та доступу до ресурсів. У роботі проведено огляд криптографічних алгоритмів та протоколів, зокрема VPN (IPSec, OpenVPN), TLS/SSL, а також систем автентифікації (Kerberos, RADIUS). Досліджено принципи їх побудови, переваги, вразливості та відповідність до потреб малих і середніх організацій. На основі дослідження запропоновано комплекс практичних рекомендацій щодо впровадження криптографічного захисту в локальній мережі: вибір протоколів, налаштування VPN-з'єднання, захист веб-ресурсів через TLS, централізована автентифікація користувачів та моніторинг безпеки.

Галузь використання – кібербезпека локальних інформаційних мереж та ресурсів організацій малого та середнього бізнесу.

VPN, TLS, IPSEC, KERBEROS, RADIUS, ШИФРУВАННЯ, ЛОКАЛЬНА

МЕРЕЖА, КРИПТОГРАФІЯ, АВТЕНТИФІКАЦІЯ, ІНФОРМАЦІЙНА БЕЗПЕКА.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення

ПКК – постквантова криптографія

TLS – Transport Layer Security

SSL – Secure Sockets Layer

RFC – Request for Comments

RBAC – Role-Based Access Control

ABAC – Attribute-Based Access Control

БД – База Даних

ACL – Access Control List

SIEM – Security Information and Event Management

WAF – Web Application Firewall

EDR – Endpoint Detection and Response

ARP – Address Resolution Protocol

AD – Active Directory

IDS – Intrusion Detection System

IPS – Intrusion Prevention System

EAP – Extensible Authentication Protocol

HMAC – Hashed Message Authentication Code

MAC – Message Authentication Code

DH – Diffie-Hellman

CBC – Cipher Block Chaining

CNAME – Canonical Name Record

ВСТУП

Актуальність дослідження. У сучасних умовах стрімкого розвитку цифрових технологій питання захисту інформації в локальних мережах організацій набуває особливої ваги. Щодня зростає кількість кіберзагроз, спрямованих на перехоплення, модифікацію або знищення даних під час їх передавання по мережі. Одним із найбільш ефективних способів забезпечення конфіденційності, цілісності та автентичності інформації є застосування криптографічного захисту. Криптографічні протоколи, такі як IPSec, TLS, OpenVPN, забезпечують захищене середовище для передавання даних між користувачами, серверами та іншими елементами мережевої інфраструктури. Успішна реалізація таких рішень дозволяє значно зменшити ризики несанкціонованого доступу, витоку даних та компрометації інформації. Особливу актуальність тема набуває для малих та середніх організацій, де відсутність централізованої безпекової політики і недостатнє фінансування на інформаційну безпеку створюють потенційні вектори атак. Правильний вибір і впровадження криптографічних засобів дозволяє досягти високого рівня захисту навіть за обмежених ресурсів.

Об'єкт дослідження – локальна комп'ютерна мережа організації.

Предмет дослідження – методи та засоби криптографічного захисту інформації в локальній мережі.

Мета роботи – розробити рекомендації щодо впровадження ефективних криптографічних засобів захисту інформації в локальній мережі організації.

Завдання кваліфікаційної роботи:

проаналізувати актуальні загрози для локальних мереж організацій;

дослідити сучасні криптографічні алгоритми, протоколи та засоби захисту;

здійснити порівняльний аналіз VPN-рішень (IPSec, OpenVPN, WireGuard) та засобів TLS/SSL;

розробити приклад практичного впровадження криптографічного захисту у віртуальному середовищі;

сформулювати рекомендації щодо вибору, налаштування та підтримки криптографічного захисту в організаціях малого та середнього бізнесу.

Методи дослідження – аналіз науково-технічної літератури, нормативних документів, міжнародних стандартів (ISO/IEC, RFC), порівняльний аналіз криптографічних засобів, моделювання інфраструктури захисту у віртуальному середовищі.

Практичне значення одержаних результатів: створення комплексу практичних рекомендацій для впровадження криптографічного захисту в локальній мережі організації. Запропоновані рішення можуть бути реалізовані адміністраторами безпеки з урахуванням реальних потреб та обмежень організаційної інфраструктури.

1 ТЕОРЕТИЧНІ АСПЕКТИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ

1.1. Основні поняття криптографії

Криптологія – це дисципліна, що займається розробкою методів захисту інформації шляхом її кодування таким чином, щоб тільки призначений одержувач міг з нею ознайомитися. Основна мета криптографії полягає в захисті даних від несанкціонованого доступу, забезпечуючи конфіденційність, цілісність, автентичність та невідмовність, поки дані знаходяться в стані спокою або передаються по незахищених каналах.

Слово “криптографія” походить із давньогрецьких слів “криптос", що означає “прихований", і “графо”, що тлумачиться як “писати". Сучасна криптографія використовується для захисту вмісту повідомлень і також у процесі автентифікації користувачів та цифрового підписання документів. Вона важлива для захисту каналів зв'язку та перевірки інформації на цілісність [1]. У сучасних умовах кібербезпеки використання криптографії є критичним. Зменшення ризику кібератак та несанкціонованого доступу можливе завдяки застосуванню методів криптології. Використання інформаційних ресурсів призвело до того, що криптографічні засоби стали дуже важливими для корпоративних мереж, банківською сфери, громадських інформаційних просторів та безпечного спілкування. Якщо використовувати правильно криптографічні методи, то можна значно зменшити ризик розголошення конфіденційної інформації або підробки даних чи отримання несанкціонованого доступу до цих цінних ресурсів. Основні властивості інформаційної безпеки:

- Конфіденційність – забезпечення недоступності інформації для сторонніх осіб.
- Цілісність – гарантування того, що дані не були змінені або пошкоджені під час передавання або зберігання.

- Автентичність – підтвердження справжності джерела даних.
- Невідмовність – забезпечення доказів здійснення певної дії, які не можуть бути заперечені її учасниками.

Отже, шифрування грає рішучу роль у створенні надійних систем захисту інформацію відповідно до поточних вимог безпеки. Для кращого розуміння основ криптографії важливо знати ключові терміни, які часто зустрічаються в цій сфері. Вивчення цієї фундаментальної термінології є передумовою для подальшого дослідження методик і засобів захисту інформації із застосуванням криптографічних прийомів.

- Шифрування – процес перетворення відкритого тексту у зашифровану форму (шифротекст) за допомогою певного алгоритму і ключа, з метою приховування змісту даних від сторонніх осіб [1].
- Дешифрування – процес зворотного перетворення зашифрованого тексту у початковий відкритий текст за допомогою відповідного ключа та алгоритму [1].
- Ключ шифрування – параметр, що визначає конкретне перетворення при шифруванні і дешифруванні даних. Без знання правильного ключа відновлення вихідної інформації з шифротексту практично неможливе [1].
- Криптоаналіз – наука та практика виявлення слабких місць у криптографічних системах з метою розкриття зашифрованої інформації без знання ключа шифрування [2].
- Криптографічний протокол – набір правил, що визначають порядок обміну зашифрованою інформацією між сторонами з використанням криптографічних алгоритмів і ключів для досягнення цілей захисту даних [2].
- Криптографічна стійкість – характеристика криптографічного алгоритму, що визначає його здатність протистояти криптоаналітичним атакам протягом заданого періоду часу за заданих ресурсів атакуючого [1].

- Постквантова криптографія (ПКК) — сукупність криптографічних методів, стійких до атак із використанням квантових обчислень.

Правильне розуміння цих основних термінів є необхідною передумовою для подальшого дослідження методів і засобів криптографічного захисту інформації.

Практика криптографії почалася в давні часи, коли люди потребували захисту своїх повідомлень від небажаних читачів. Шифр давньоримського Цезаря є одним з найперших зафіксованих методів шифрування, оскільки він вимагав зсуву літер на певну кількість позицій [3]. У Середньовіччі криптографія виникла завдяки потребам дипломатичного та військового листування. Поліалфавітні шифри представляють вдосконалені криптографічні системи, серед яких шифр Віженера виділяється як яскравий приклад. Криптографічні методи еволюціонували в більш складні системи завдяки розвитку криптоаналізу протягом століть. Двадцяте століття принесло значні зміни під час Першої та Другої світових воєн. Поява електромеханічних шифрувальних машин, таких як німецька «Енігма», ознаменувала значний прогрес у розвитку криптографії. Постійна боротьба між криптографами і криптоаналітиками призвела до швидкого розвитку обчислювальних засобів і математичних методів шифрування. Друга половина XX століття принесла криптографії математичні основи. Розвиток RSA та інших фундаментальних алгоритмів відбувався завдяки використанню складності факторизації великих чисел та асиметричних систем ключів [4].

Інформаційна безпека спирається на криптографію як на важливий інтегрований компонент. Сучасний криптографічний ландшафт включає широке впровадження симетричних алгоритмів шифрування (AES) та хеш-функцій (SHA-2, SHA-3), а також безпечних протоколів обміну даними (TLS/SSL, IPSec). Розвиток електронної комерції, електронного документообігу та безпеки мережевого обміну даними спирається на криптографію як на фундаментальну основу. Розвиток квантових обчислень сьогодні ставить перед криптографією нові виклики, що стимулює дослідження в галузі ПКК.

1.2. Види криптографічних алгоритмів (симетричні, асиметричні, хеш-функції)

Існує три основні категорії криптографічних алгоритмів, які слугують різним цілям завдяки принципам використання ключів, що включають симетричні, асиметричні та хеш-функції. Різні системи криптографічного захисту покладаються на кожен тип для виконання своєї специфічної задачі. Процеси шифрування і дешифрування в симетричних системах вимагають для роботи один і той самий ключ. Основною перевагою такого підходу є висока швидкість обробки даних. Розподіл ключів між учасниками комунікації стикається з проблемою безпеки. Найпоширенішими алгоритмами симетричного шифрування є:

- AES (Advanced Encryption Standard) – сучасний стандарт симетричного шифрування з блоковою структурою та довжиною ключа 128, 192 або 256 біт;
- 3DES (Triple DES) – розширення алгоритму DES шляхом потрійного шифрування, вважається застарілим;
- ChaCha20 – потоковий алгоритм, оптимізований для програмної реалізації, часто використовується в мобільних пристроях.

Для шифрування великих обсягів даних в системах з можливістю попереднього узгодження ключів використовуються симетричні алгоритми. Найбільш поширені сфери застосування включають:

- захист трафіку в VPN;
- шифрування файлів і дисків;
- мережеві протоколи передачі даних (IPSec, WPA2/3);
- корпоративні резервні копії та хмарні сховища.

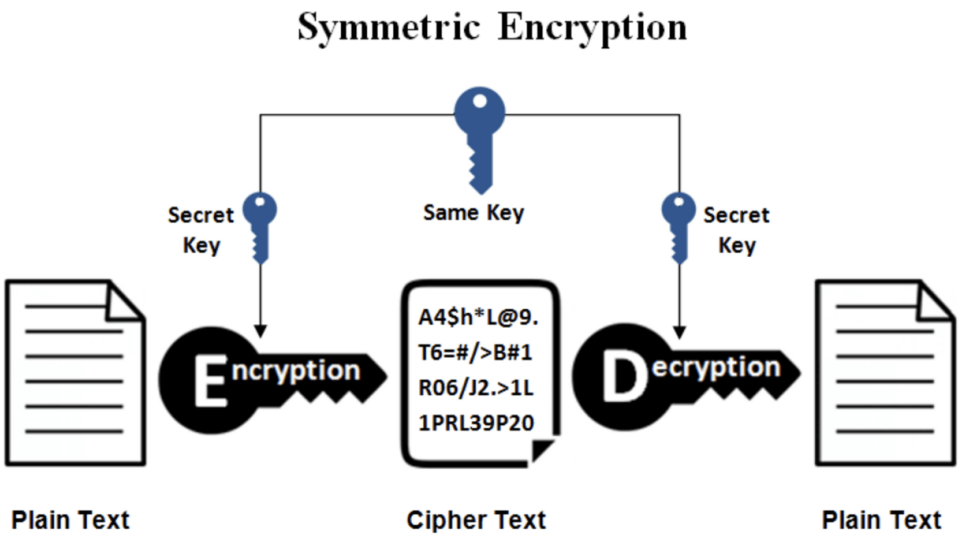

Рис. 1.1. Схема роботи симетричних шифрів [5]

Процес шифрування в асиметричних алгоритмах (або алгоритмах з відкритим ключем) вимагає відкритого ключа, але для розшифрування потрібен закритий ключ. Цей метод забезпечує безпечний обмін ключами без необхідності, але вимагає більше обчислювальних ресурсів.

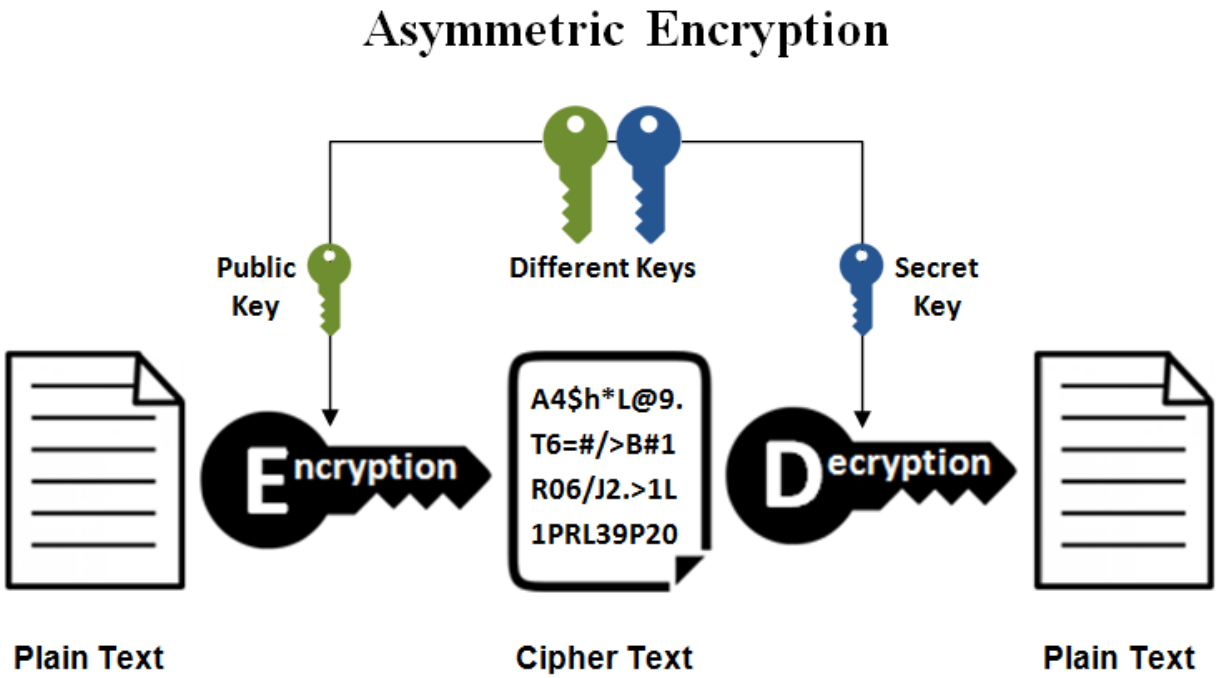

Рис. 1.2. Схема роботи асиметричних шифрів [6]

Типові приклади асиметричних алгоритмів:

- RSA – базується на складності факторизації великих простих чисел;

- ElGamal – ґрунтується на складності обчислення дискретного логарифма;
- ECC (Elliptic Curve Cryptography) – забезпечує високу криптостійкість при коротких ключах, що робить її привабливою для мобільних і вбудованих систем.

Асиметричні алгоритми застосовуються для захисту цифрових підписів, автентифікації та обміну ключами, а також для захисту електронних транзакцій.

Хеш-функції в криптографії працюють як алгоритми, які перетворюють будь-який масив вхідних даних у значення фіксованої довжини, відоме як хеш-код або дайджест. Основна вимога полягає в тому, щоб система виробляла незворотні результати, не допускаючи при цьому, щоб різні повідомлення генерували ідентичні хеш-значення. Найбільш відомі алгоритми гешування:

- SHA-2 (Secure Hash Algorithm 2) – сімейство геш-функцій із довжиною виходу 224, 256, 384 або 512 біт;
- SHA-3 – новий стандарт, заснований на алгоритмі Keccak;
- BLAKE2, BLAKE3 – сучасні, швидкі та надійні геш-функції з широким використанням у практичних системах.

Геш-функції застосовуються для перевірки цілісності даних, збереження паролів, створення цифрових підписів та у блокчейн-технологіях.

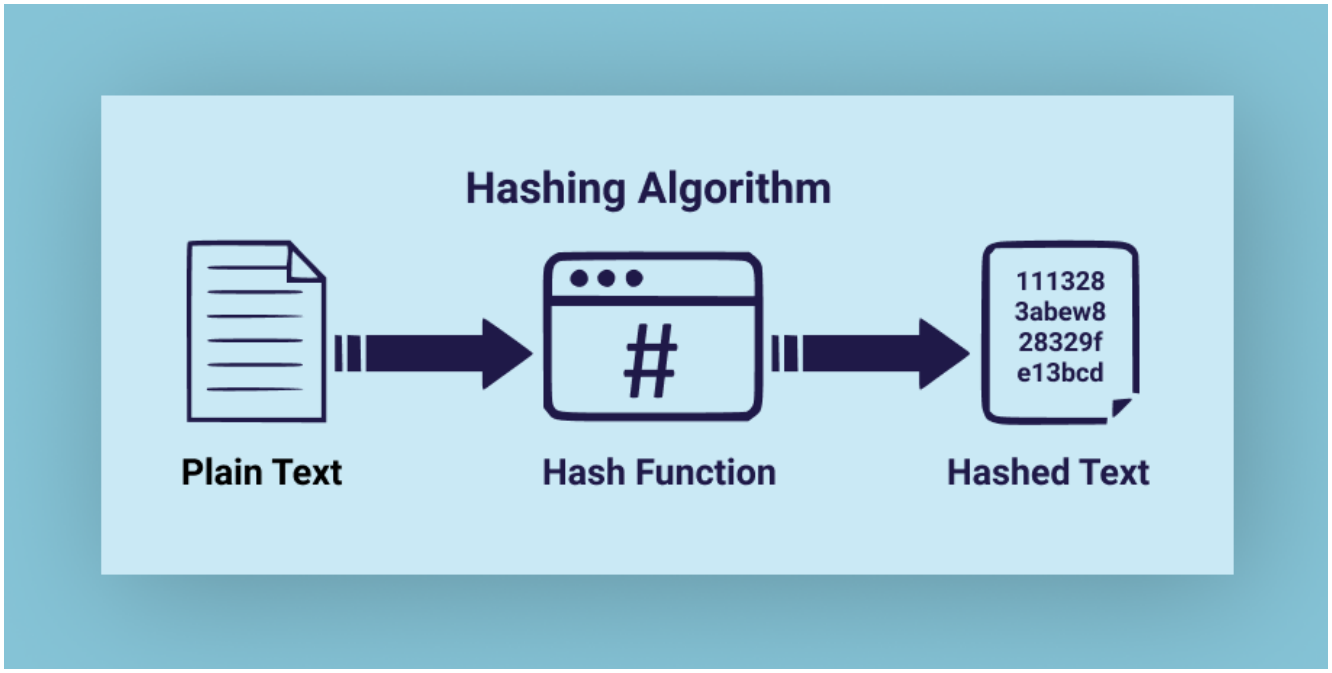

Рис. 1.3 Схема роботи геш-функцій [7]

Алгоритми криптографії з часом зазнали значного розвитку. Протокол шифрування DES еволюціонував у більш безпечні стандарти шифрування, такі як AES. Протоколи шифрування RSA, які раніше були громіздкими, були замінені швидшими алгоритмами на основі ECC в асиметричній криптографії. Хеш-функції еволюціонували завдяки заміні незахищених MD5 і SHA-1 на сучасні SHA-2 і SHA-3 та додаткові алгоритми, включаючи BLAKE2. Дослідники в галузі криптографії працюють над розробкою квантово-стійких алгоритмів для створення основи ПКК в сучасному світі.

Фундаментальною складовою повного захисту даних в сучасних інформаційних системах є криптографія. Розвиток інформаційних технологій поряд з хмарними обчисленнями, мобільними пристроями та розподіленими сервісами вимагає постійної конфіденційності, цілісності та доступності інформації на всіх етапах обробки. Криптографічні алгоритми використовуються в багатьох сферах, таких як протоколи безпечної передачі даних TLS та IPSec, створення VPN, управління електронними документами, технології блокчейн та цифрові підписи. Веб-з'єднання використовують протокол TLS для захисту трафіку, а IPSec і WireGuard VPN встановлюють безпечні з'єднання між віддаленими вузлами мережі. Асиметричні криптографічні системи дозволяють організаціям розгортати системи автентифікації, здійснювати безпечний обмін ключами та впроваджувати механізми електронного підпису. Симетричні алгоритми забезпечують ефективне шифрування великих обсягів даних за допомогою процесів, що призводять до мінімальних затримок. Хеш-функції дозволяють перевіряти дані, генерувати цифрові відбитки, зберігати паролі та контролювати їх зміну. Політика корпоративної безпеки залежить від криптографії, і ця технологія є ключовим елементом відповідності міжнародним стандартам ISO/IEC 27001, NIST SP 800-53 та PCI DSS. Сучасні криптографічні рішення повинні відповідати новим вимогам через зростаючу складність атак та розвиток APT-загроз і шкідливого програмного забезпечення. У цьому контексті зростає важливість гібридних моделей захисту, ПКК та автоматизованих систем управління ключами.

Цифровий світ залежить від криптографії як від основного елемента для побудови систем інформаційної безпеки, що захищають дані. Проектування та впровадження сучасних безпечних інформаційних систем вимагає глибокого розуміння основних понять, типів алгоритмів та їх призначення. Майбутнє впровадження криптографічного захисту в локальних мережевих системах буде базуватися на цих знаннях для практичного аналізу та вибору.

1.3. Загальні принципи побудови захищених локальних мереж

Локальні мережі (LAN) з'єднують пристрої через мережеву систему, яка працює в межах певних територій (офісних приміщень, будівель або навчальних закладів) для забезпечення обміну даними та спільного використання ресурсів [8].

LAN існує для полегшення взаємодії між користувачем і сервером, базою даних і периферійними пристроями, а також системою контролю доступу. Типова локальна мережа має високу пропускну здатність, низьку затримку, централізовані системи контролю та функції управління. Сьогодні організації покладаються на LAN як на фундаментальну основу своєї діяльності, оскільки вони підтримують усі бізнес-функції, починаючи від бухгалтерського обліку і закінчуючи автоматизованими системами виробництва.

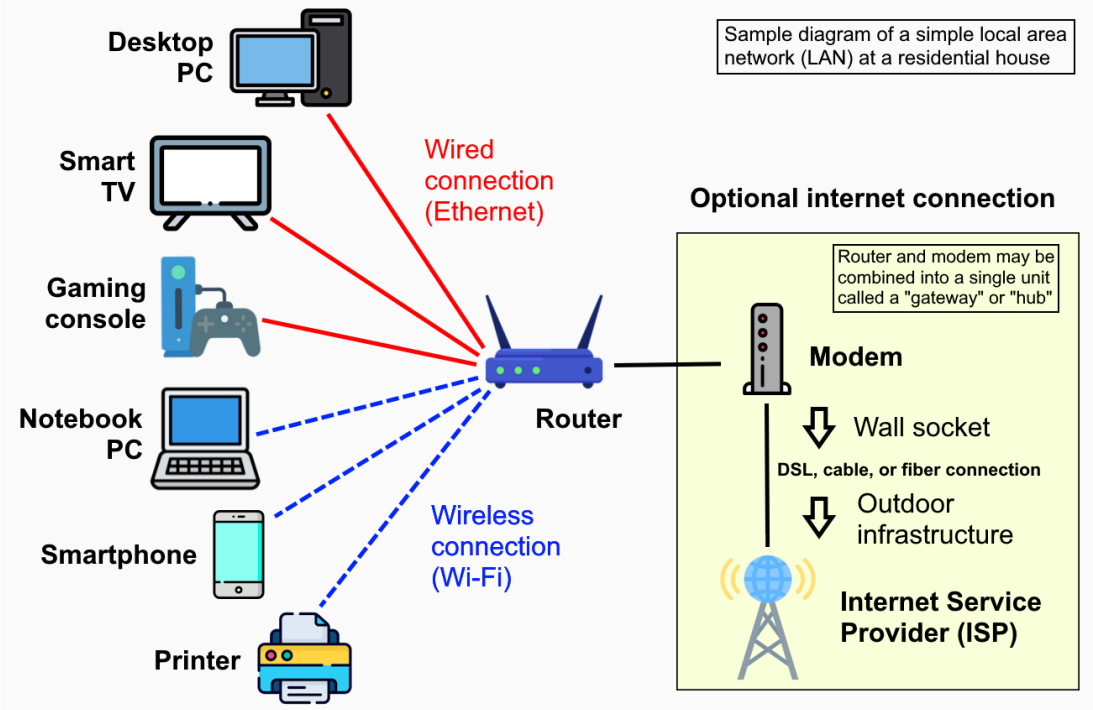

Рис. 1.4 Схема ЛОМ [12]

Локальні мережі слугують простором для розповсюдження важливої інформації, конфіденційних даних та інформації, пов'язаної з послугами. Порушення цілісності даних, що виникає внаслідок несанкціонованого доступу, модифікації, перехоплення або блокування даних, призводить до порушення функцій організації, витоку персональних даних і шкоди діловій репутації, а також до фінансових втрат і юридичної відповідальності. Сучасні локальні мережі функціонують інакше, ніж ізольовані системи минулого, оскільки вони підтримують постійну взаємодію із зовнішніми системами, включаючи мережі Інтернету, хмарні ресурси, віддалені офіси та підрядників. Розширені системні інтерфейси створюють додаткові виклики для безпеки локальних мереж. Основні загрози для локальних мереж включають:

- Несанкціонований доступ. Сторонні пристрої стають вразливими, коли підключаються до відкритих портів комутаторів або гостьових мереж Wi-Fi без фільтрації MAC-адрес [9].
- Перехоплення трафіку (сніфінг, ARP-спуфінг). Відсутність шифрування на певних сегментах мережі дозволяє здійснювати пасивне прослуховування та MITM-атаки, що використовують ARP-спуфінг.

- Спуфінг. Зловмисник видає себе за надійний пристрій, змінюючи його MAC- або IP-адресу, щоб отримати доступ до внутрішніх ресурсів.
- Розповсюдження шкідливого програмного забезпечення. Мережеві інфекції можуть поширюватися по всій системі від вразливих вузлів, які не мають належного захисту своїх служб і файлів. Атаки програм-вимагачів WannaCry і NotPetya є відомими прикладами таких загроз.
- Бічний рух. Скомпрометований вузол функціонує як точка входу для зловмисників, які намагаються розширити свій вплив на внутрішню інфраструктуру, сервери, бази даних і контролери домену [9].
- Атаки DoS/DDoS. Успішна локальна атака на сервер DHCP або DNS повністю виведе з ладу роботу мережі. Атаки типу «флуд» можуть перевантажити комутатори.
- Використання незахищених служб. Три протоколи (Telnet, SMBv1 і FTP) працюють без функцій шифрування або автентифікації, оскільки вони за замовчуванням залишаються увімкненими в більшості систем.

Більшість загроз безпеці походять від внутрішніх дій користувачів або випадкових дій персоналу, що вимагає як технічних заходів безпеки, так і організаційних протоколів. Захист локальних мереж виходить за межі використання антивірусного програмного забезпечення та встановлення брандмауерів як самостійних рішень. Безпека повинна поширюватися на всі етапи інформаційної системи, включаючи як фізичну інфраструктуру, так і людський фактор.

Таблиця 1.1.

Рівні інформаційної системи

Складова	Опис
Архітектурна безпека	Сегментація мережі, DMZ-зони, логічна ізоляція підсистем і критичних вузлів;
Ідентифікація, автентифікація, авторизація	Керування доступом на основі ролей (RBAC), політик безпеки, списків доступу (ACL);
Криптографічний захист	Шифрування трафіку (IPsec, TLS), файлів (AES), баз даних, використання сертифікатів;
Журналювання та моніторинг	Збір логів подій, аналіз за допомогою SIEM-систем, виявлення аномалій, ретроспективний

	аналіз;
Управління вразливостями	Регулярне сканування систем, оцінка ризиків, оновлення ПЗ, патч-менеджмент;
Реагування на інциденти	План реагування, процедури ескалації, аналіз наслідків, журналювання інцидентів;
Фізичний захист	Контроль доступу до серверних кімнат, блокування USB-портів, відеоспостереження;
Навчання персоналу	Протидія фішингу, курси з кібербезпеки, симуляції атак.

Принцип мінімальних привілеїв вимагає, щоб користувачі, процеси та пристрої отримували лише необхідні права доступу для виконання призначених їм завдань. Кожен дозвіл, який не має обмежень і не є необхідним, становить загрозу для контролю авторизації та збільшує площу атаки системи [10]. Впровадження принципу мінімальних привілеїв у локально обчислювальну мережу (ЛОМ) вимагає конкретних кроків:

- права доступу користувачів до спільних ресурсів повинні бути обмежені.;
- система брандмауера повинна контролювати доступність служб і портів.;
- організація повинна використовувати політику групового доступу.;
- всі дії привілейованих облікових записів повинні контролюватися..

Цей принцип безпеки є основною вимогою стандартів ISO/IEC 27001 та NIST SP 800-53 для встановлення ефективної політики доступу в корпоративних середовищах.

Принцип безпеки розділення привілеїв стверджує, що для ефективного виконання критичних операцій необхідні кілька незалежних умов та дій, а не одна умова. Впровадження цього принципу мінімізує ризики, пов'язані із зловживанням системою, помилками та порушенням її цілісності. Принцип передбачає, що одна особа або компонент не повинні мати повного контролю над критичними операціями в будь-якій ситуації [10]. Джером Солтцер і Майкл Д. Шредер розробили цей принцип у 1975 році, коли створили основні принципи проектування безпечних комп'ютерних систем. Концепція походить від політичного принципу поділу влади, який розподіляє повноваження між різними

суб'єктами, замість того щоб надавати всю владу одній особі. Приклади реалізації:

- фінансова система вимагає двох підписів для транзакцій, що перевищують 10 000 доларів.
- система адміністрування вимагає, щоб один адміністратор створював користувачів, а інший адміністратор керував правами доступу користувачів.
- система управління сертифікатами використовує двох різних користувачів для генерації ключів та підписання сертифікатів.
- секретний ключ розподіляється між кількома особами за допомогою схеми розподілу секрету Шаміра для цілей шифрування.
- система обов'язкового контролю доступу вимагає декількох атрибутів класифікації, включаючи ролі для авторизації доступу.

Підхід глибокого захисту передбачає створення декількох захисних шарів, які блокують, ідентифікують або зменшують загрози, що проходять через попередні етапи захисту [11]. Цей підхід використовує декілька рівнів захисту, які доповнюють один одного для досягнення захисту, замість того, щоб покладатися на один механізм захисту. За допомогою цього підходу система залишається стабільною, навіть якщо будь-який елемент захисту буде порушений.

Таблиця 1.2.

Можливий сценарій етапів атаки та захист на кожному рівні

Етап атаки	Рівень захисту	Засіб	Реакція системи
Сканування IP-адрес, портів	Мережевий периметр	IDS/IPS, Firewall	IPS фіксує активність з незнайомого IP, блокує трафік. SIEM реєструє аномалію;
Спроба обходу Firewall	VPN з MFA	VPN + 2FA	Атакуючий не має облікових даних, вхід не допускається;
Використання вкрадених облікових даних	Керування доступом	RBAC і SIEM	Доступ дає тільки обмежені права. SIEM фіксує вхід з незвичного гео-IP;

Переміщення вглибину	Сегментація мережі (VLAN, ACL)	ACL, ізоляція БД в окремій зоні	Перехід із сегмента користувачів у зону БД неможливий без додаткової автентифікації;
Спроба SQL-ін’єкції до веб-додатку	Захист додатків	WAF, валідація інпуту	WAF блокує запит, генерує лог; SIEM фіксує спробу експлуатації;
Спроба експорту БД, якщо ін'єкція вдалася	Шифрування та аудит	Прозоре шифрування даних, журналювання	БД зашифрована, доступ до даних недоступний без ключа. Спрацює алерт у SIEM;
Видалення логів для приховування слідів	Контроль змін	Центральний лог-сервер, MAC	Зміна логів неможлива без прав root і не залишить хеш-суму незмінною;
Захоплення системи або шифрування файлів	Антивірус, EDR	EDR	EDR фіксує нетипову активність, запускає ізоляцію вузла;
Фізичне проникнення	Фізичний захист	Біометрія, фіксація камерами, RFID-доступ	Доступ обмежений, всі події записуються, охорона отримає тривогу.

Основний принцип безпеки комп'ютерних систем, відомий як повне посередництво (повна медіація), був створений Салтцером і Шредером (1975) для встановлення надійної системи контролю доступу. Політика доступу повинна перевірятися кожного разу, коли будь-який об'єкт в інформаційній системі отримує запит на доступ. Кожен доступ до ресурсу повинен бути авторизований перед наданням дозволу. Відсутність постійної перевірки доступу дозволяє користувачам або процесам отримувати доступ до ресурсів, які колись були дозволені, хоча їхні дозволи були змінені або скасовані. Доступ стає незахищеним у період від надання дозволу до скасування дозволу.

Таблиця 1.3.

Можливий сценарій

Сценарій	Без повної медіації	З повною медіацією
Співробітник отримав доступ до файлу	Залишається відкритим у кеші навіть після звільнення	При кожному відкритті перевіряється, чи ще має право

Веб-сесія з JWT токеном	Токен живе 1 годину, далі доступ не перевіряється після видачі	Сесія перевіряється при кожному запиті до API
Кешування ACL у файловій системі	Зміна ACL не впливає на відкриті дескриптори	Доступ перевіряється щоразу при операціях читання/запису

Повна медіація в системах:

- операційна система повинна перевіряти дозволи для кожного системного виклику, що включає open(), read() та write().
- кожен SQL-запит повинен перевіряти дозволи користувача на доступ до цільової таблиці бази даних.
- мережеві служби повинні реалізувати проміжне програмне забезпечення API для перевірки дозволів користувачів для кожного HTTP-запиту.
- Системи RBAC/ABAC виконують перевірку політики доступу при кожному запиті, запобігаючи кешуванню політики без терміну дії.

Мінімальний механізм спільного використання встановлює принцип безпеки, який вимагає, щоб спільні ресурси, механізми та компоненти залишалися ексклюзивними для одного суб'єкта, якщо це не є необхідним, оскільки це мінімізує можливості атак та ризики побічних ефектів [10]. Основна концепція зосереджена на запобіганні спільного використання ресурсів або механізмів суб'єктами з різними рівнями довіри. Наступні приклади показують потенційні загрози, які можуть виникнути:

- черга друку, доступна для декількох користувачів, дозволяє зловмисному користувачеві отримати доступ до документів інших користувачів.
- використання спільного тимчасового каталогу робить можливими як атаки заміни файлів, так і атаки на символічні посилання.
- атака на службу журналу через єдине ядро дозволяє отримати повний доступ до історичних даних.
- бічні канали можуть використовуватися для шпигунства, коли використовуються спільні бібліотеки або кеші.

- один адміністративний обліковий запис, який контролює кілька служб, дозволяє зловмиснику отримати повний доступ до всіх керованих служб.

За моделлю безпеки нульової довіри (Zero Trust) всі пристрої та програми користувачів залишаються ненадійними, навіть якщо вони знаходяться в межах корпоративної мережі. Кожен запит на доступ потребує ретельної перевірки перед затвердженням на основі контекстної оцінки [13]. Традиційна модель безпеки залежить від безпеки мережі, тоді як зовнішні загрози становлять основну небезпеку. Сучасні загрози безпеці здебільшого виникають з внутрішніх джерел через VPN, MFA та AD, а також через техніки латерального переміщення, які дозволяють зловмисникам переходити між вузлами мережі після початкового вторгнення. Модель Zero Trust працює на основі припущення, що мережі залишаються в зоні ризику, тому кожен запит повинен пройти сувору автентифікацію та авторизацію, а також оцінку політики доступу. Система приймає рішення про доступ на основі динамічної оцінки ідентичності користувача, типу та стану пристрою, географічного розташування та моделей поведінки, що залежать від часу. Система обмежує доступ після автентифікації за принципом мінімальних привілеїв і постійно контролює та реєструє всі дії для виявлення ненормальної поведінки.

Сучасне середовище загроз вимагає від організацій використання декількох механізмів захисту, оскільки загрози походять як з внутрішніх, так і з зовнішніх джерел. Чотири принципи Zero Trust – мінімальні привілеї, повне посередництво та глибока оборона – мають різні цілі, але призводять до однакових технічних та організаційних наслідків. Найкращі результати досягаються, коли ці підходи працюють разом: Zero Trust створює стратегічну базу, мінімальні привілеї контролюють доступ, повне посередництво забезпечує постійну верифікацію, а глибока оборона будує гнучку багаторівневу систему захисту. Стратегічне впровадження цих принципів дозволяє розробляти системи, які адаптуються до нових викликів та протидіють складним загрозам.

2 АНАЛІЗ ЗАСОБІВ КРИПТОГРАФІЧНОГО ЗАХИСТУ В ЛОКАЛЬНІЙ МЕРЕЖІ ОРГАНІЗАЦІЇ

2.1. Використання VPN-протоколів (IPSec, OpenVPN, WireGuard)

Елементи локальної мережі потребують встановлення безпечних каналів зв'язку через зростання кіберзагроз, тоді як хмарні технології продовжують активно використовуватися. Основне рішення цієї проблеми полягає у використанні технології віртуальної приватної мережі (VPN) для встановлення зашифрованих логічних з'єднань між мережами, які можуть бути інтернет-мережами або іншими незахищеними мережами. VPN захищають передану інформацію за допомогою методів шифрування, а також перевірки користувачів і контрольних сум для забезпечення цілісності та автентичності даних. VPN виконують три основні функції для безпеки локальної мережі:

- віддалені співробітники та офіси можуть отримати доступ до корпоративної мережі через це з'єднання;
- мережа відокремлює ізольовані сегменти від потенційно небезпечних зон;
- система створює безпечні мережеві з'єднання для доступу до хмарних ресурсів та підключення до серверів центрів обробки даних.

Впровадження VPN є важливою вимогою безпеки для захисту інформації, що відповідає стандартам ISO/IEC 27001, NIST SP 800-77 та архітектурі Zero Trust.

Концепція віртуальних приватних мереж створює захищені шляхи передачі інформації через ненадійні мережі, включаючи Інтернет. Функціонування VPN поєднує дві основні функції: тунелювання та криптографічний захист. Завдяки своїй функціональності VPN забезпечує передачу даних між декількома вузлами з гарантованою конфіденційністю, цілісністю та автентичністю, так само як, якщо б вузли існували в одній фізичній мережі. Метод інкапсуляції одного мережевого протоколу в інший називається тунелюванням. Передача пакетів

клієнта VPN в мережу передбачає інкапсуляцію даних в додатковий IP-пакет перед доставкою в Інтернет на сервер VPN. Сервер VPN отримує пакет, а потім видаляє його для передачі в внутрішню мережу призначення. Створення тунелів відбувається за допомогою трьох різних протоколів:

- Протокол OpenVPN працює через TCP/UDP на транспортному або сесійному рівні;
- IPsec працює на рівні мережі (IPsec);
- VPN на рівні інтерфейсу WireGuard створює власний віртуальний мережевий інтерфейс для роботи.

VPN покладається на шифрування як основну криптографічну функцію для захисту даних. Реалізація протоколу дозволяє використовувати симетричні (наприклад, AES) та асиметричні алгоритми (RSA, ECC) разом з хеш-функціями (SHA, BLAKE2) для забезпечення цілісності. Основні криптографічні функції у VPN:

- AES-256 і ChaCha20 служать для захисту конфіденційності даних за допомогою шифрування;
- Для перевірки цілісності даних використовуються HMAC (SHA-256, BLAKE2s) і хеш-функції;
- Аутентифікація відбувається за допомогою цифрових сертифікатів і ключів, а також комбінацій логіна/пароля;
- Функція обміну ключами використовує протоколи DH або Curve25519 (у WireGuard) для реалізації.

Сесія VPN встановлюється за допомогою такого загального процесу:

1. Користувач встановлює зв'язок із сервером VPN під час процесу ініціювання.
2. Користувач підтверджує свою особу за допомогою методів верифікації, що включають сертифікати, а також паролі та ключі.
3. Система створює криптографічні параметри, що включають алгоритм, ключі та вибір протоколу під час встановлення тунелю.
4. Ця точка позначає початок шифрування для передачі даних через тунель.

5. Завершення сеансу – сеанс може бути завершено вручну або за допомогою таймера.

Набір IPsec (Internet Protocol Security) включає протоколи, які шифрують і автентифікують IP-пакети, перевіряючи їх цілісність на рівні мережі. Рішення безпеки IPSec працює в корпоративних мережах і міжофісних тунелях, а також на шлюзах безпеки і платформах мережевого обладнання всіх сучасних операційних систем [14]. Система IPSec складається з декількох функціональних компонентів:

- Протокол AH (Authentication Header) перевіряє цілісність і автентичність IP-пакетів без шифрування вмісту даних. Основна роль AH полягає в захисті даних від змін під час передачі (цілісність) і перевірці джерела передачі (автентичність), а також у захисті лише IP-заголовків, а не корисного навантаження. Протокол AH вставляє додатковий заголовок, який розміщується між заголовком TCP/UDP і заголовком IP. Обчислення хешу (HMAC) відбувається в певних полях пакета, які потім передаються в заголовку AH під час передачі даних. Коли одержувач перекомпілює хеш, він виконує ще одне обчислення перед проведенням процесу порівняння. Функція автентифікації AH не захищає конфіденційність, оскільки весь вміст пакета залишається відкритим під час передачі. Перевірка хешу стає недійсною при будь-якій модифікації заголовка IP, оскільки AH поширює свій захист на заголовок IP. Протокол працює в режимах Tunnel і Transport відповідно до стандартів AH [15].
- IKEv2 працює як протокол, який дозволяє сторонам обмінюватися ключами під час створення параметрів безпечного з'єднання та автентифікації учасників за допомогою паролів або сертифікатів та EAP. IKEv2 виконує автоматичну конфігурацію асоціації безпеки (SA) для використання протоколами AH або ESP.
- Основний криптографічний протокол IPSec використовує ESP (Encapsulating Security Payload) для забезпечення шифрування даних (конфіденційність), контролю цілісності (HMAC або хеш-функція) та

автентифікації джерела. Протокол включає заголовок ESP і зашифрований корисний вантаж разом з опціональним трейлером автентифікації, коли необхідне шифрування. Основні компоненти ESP включають SPI (Security Parameter Index) як ідентифікатор з'єднання, поряд з послідовним номером для захисту від повторних передач, і корисні дані як зашифрований сегмент, а також дані автентифікації для перевірки HMAC. Алгоритми шифрування, що підтримуються цим протоколом, включають AES-CBC і AES-GCM, а також ChaCha20, а алгоритми цілісності підтримують HMAC-SHA1, SHA-256 і BLAKE2s. Процес шифрування ESP відбувається в двох режимах роботи, де режим транспорту шифрує тільки дані, залишаючи заголовок IP відкритим, а режим тунелю повністю шифрує весь IP-пакет. Протокол підтримує три типи VPN-додатків: VPN «сайт-сайт», VPN «хост-сайт» та шлюзи безпеки. Стандартний протокол для корпоративних IPSec-тунелів поєднує ESP у тунельному режимі з AES-GCM та IKEv2. IKEv2 (Internet Key Exchange v2) встановлює параметри безпечного тунелю за допомогою протоколу обміну ключами [16].

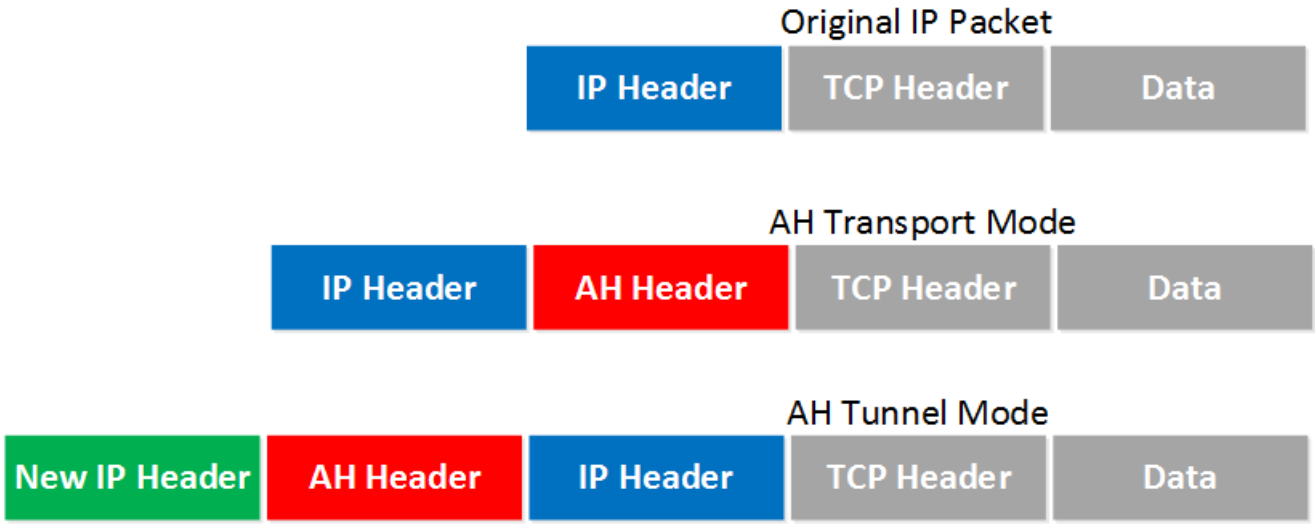

Рис. 2.1. Порівняння вмісту пакетів в залежності від режиму передачі даних [17]

Механізм IPSec реалізує безпечний тунель між сторонами за допомогою послідовного протоколу IKEv2 та ESP або AH з операціями протоколу Tunnel Mode. Під час стандартних операцій VPN між сайтами тунель IPSec служить каналом, через який відбувається шифрування всіх даних [17]. Під час

IKE_SA_INIT ініціюючий хост або шлюз починає з'єднання, надсилаючи іншим сторонам криптографічні набори та випадкові числа (nonces), а також параметри алгоритму DH. Спочатку необхідно згенерувати спільний секрет для створення ключів для шифрування. Процес шифрування починається після встановлення IKE SA, що створює базовий канал довіри.

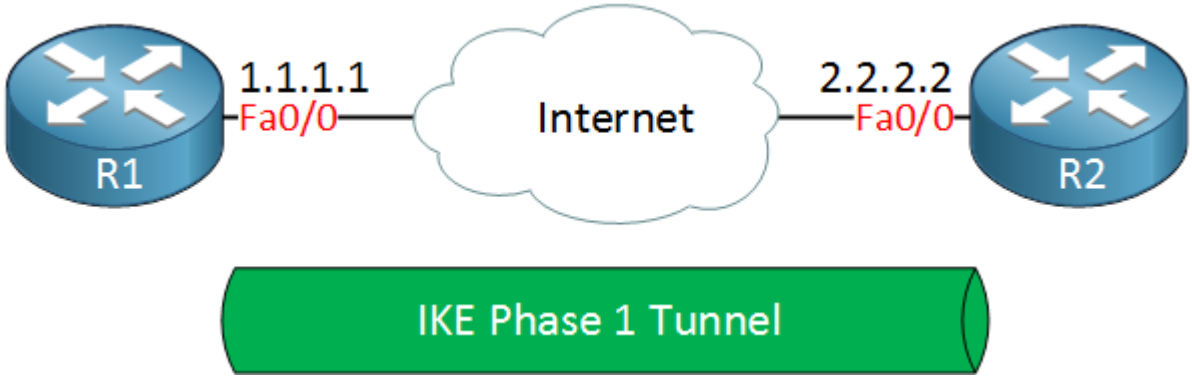

Рис. 2.2. Перший етап встановлення захищеного тунелю [17]

Під час фази IKE_AUTH сторони встановлюють взаємну автентифікацію за допомогою сертифікатів X.509 та комбінацій імені користувача/пароля або зовнішніх методів, таких як EAP та RADIUS. В процесі комунікації з'являються як IP-адреси, так і DNS-імена. Після успішної автентифікації створюється другий безпечний канал IKE_AUTH SA, що дозволяє встановити Child SA для майбутньої роботи IPSec.

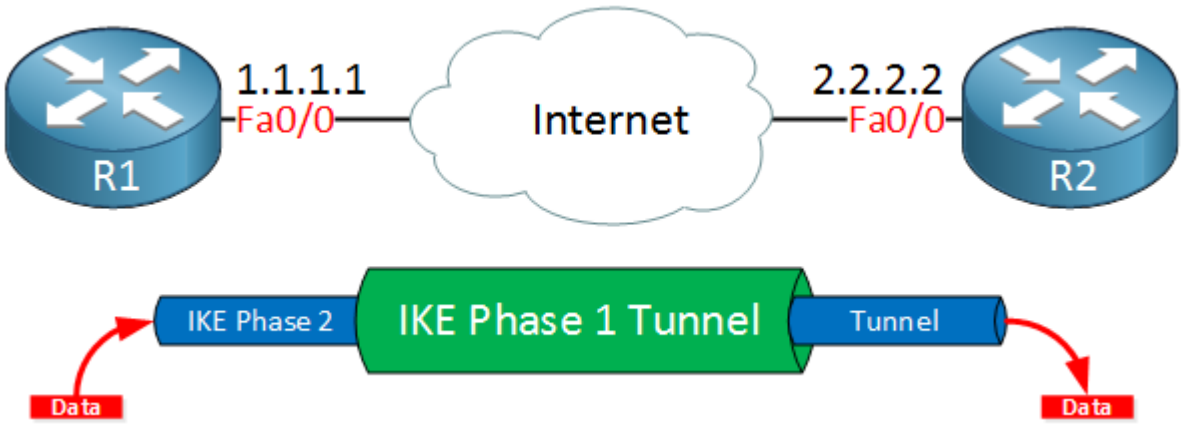

Рис. 2.3. Другий етап встановлення захищеного тунелю [17]

Сторони використовують конфігурацію IPSec SA для визначення протоколів ESP та AH, а також для вибору методів шифрування, таких як AES або ChaCha20, та методів цілісності, таких як HMAC-SHA256 або BLAKE2, та для встановлення додаткових параметрів, включаючи тривалість сеансу та обсяг

трафіку. Кожне з'єднання використовує окремий SPI (індекс параметрів безпеки), який функціонує як ідентифікатор сеансу.

Після цього, на четвертому етапі, починається передача даних. Передача даних відбувається через процес, під час якого IP-пакети отримують додаткові IP-пакети, які слугують зовнішніми конвертами, перш ніж пройти шифрування за вибраним алгоритмом і перевірку цілісності з додаванням хеш-суми разом із полями обслуговування (наприклад, послідовний номер для запобігання атакам повторного відтворення). Весь зашифрований пакет передається через мережу. Модуль IPSec розшифровує отримані дані на приймальному кінці перед перевіркою хеш-сум, одночасно видаляючи зовнішню оболонку перед доставкою внутрішнього IP-пакета до системи призначення. На останньому етапі з'єднання підтримується: дочірній SA має обмежений час існування, після якого він може бути автоматично поновлений без повного розриву з'єднання за допомогою існуючого IKE SA. Обидва SA – IKE і дочірній – знищуються, коли одна зі сторін ініціює завершення, тим самим закриваючи тунель.

OpenVPN встановлює безпечні тунелі на рівні додатків між пристроями через незахищені мережі, включаючи Інтернет-з'єднання. OpenVPN відрізняється від IPSec тим, що працює в просторі користувача, а не в просторі ядра, використовуючи TLS/SSL для шифрування трафіку та автентифікації учасників. Протокол досягає надзвичайної сумісності, оскільки працює на цьому рівні. OpenVPN завоював популярність завдяки сумісності з різними платформами, гнучкій конфігурації, простоті розгортання, підтримці NAT та придатності для організацій будь-якого розміру. Архітектура OpenVPN працює за принципом клієнт-сервер. Сервер працює в захищеному сегменті локальної мережі, а клієнт – як віддалений пристрій, наприклад, комп'ютер співробітника, філія або хмарний сервер. Для інкапсуляції IP або Ethernet-пакетів OpenVPN використовує віртуальні мережеві інтерфейси, такі як TUN або TAP [18]. Протокол надає два варіанти використання транспортного протоколу: TCP і UDP, причому останній є кращим вибором для кращої продуктивності.

Основним протоколом, який OpenVPN використовує для створення

сеансів, є TLS або SSL. Сервер перевіряє свою ідентичність за допомогою сертифіката X.509, але клієнт може встановити автентифікацію за допомогою імені користувача та пароля. Після встановлення з обох сторін TLS-з'єднання вони визначають криптографічні параметри, які будуть використовувати. Протоколи шифрування за замовчуванням в OpenVPN використовують TLS 1.2 і 1.3, які забезпечують підтримку безпечних і сучасних криптографічних алгоритмів. Протоколи безпеки, які OpenVPN використовує для шифрування, включають алгоритми AES-128 і AES-256, а також ChaCha20 і використовують функції HMAC, такі як SHA1 і SHA256, для перевірки цілісності. Протокол TLS ініціює ініціалізацію, а також методи обміну ключами DH і ECDH [18]. Конфігураційні файли сервера і клієнта дозволяють гнучко змінювати всі ці параметри.

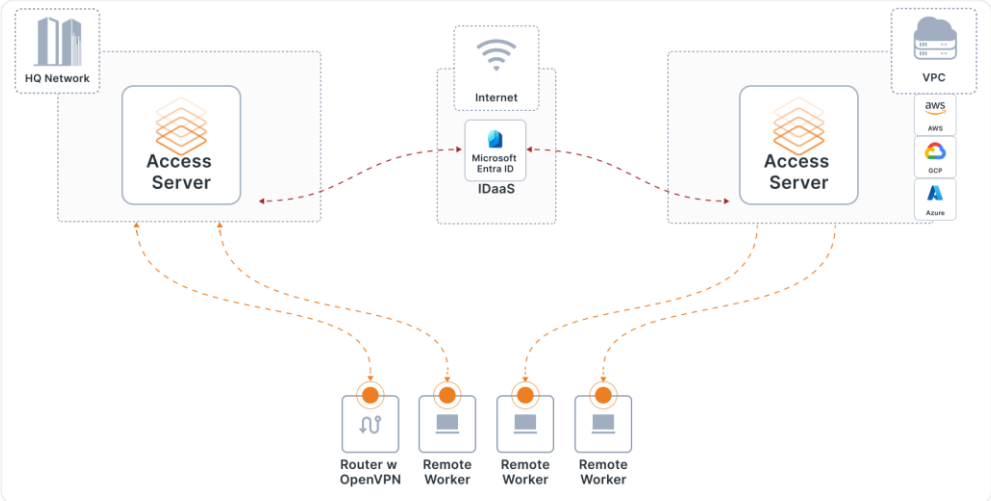

Рис. 2.4. Схема роботи OpenVPN [18]

OpenVPN забезпечує аутентифікацію за допомогою сертифікатів як основного методу, але також підтримує зовнішню аутентифікацію через RADIUS, LDAP, PAM і MFA. Система також підтримує двофакторну аутентифікацію через OTP і Google Authenticator. OpenVPN виділяється тим, що може пройти через брандмауери NAT через стандартний порт 1194 разом з NAT-T, переадресацією портів і додатковими плагінами для маскування трафіку. Протокол забезпечує надійну роботу в мережах, що мають обмеження.

Користувачі OpenVPN можуть створити два різних інтерфейси, які

називаються TUN або TAP. TUN працює на рівні мережі, інкапсулюючи IP-пакети, а TAP працює на рівні каналу, інкапсулюючи Ethernet-фрейми [18]. TUN є кращим вибором для більшості користувачів, оскільки забезпечує просту роботу та ефективну продуктивність. Єдиний випадок, коли TAP стає необхідним, – це коли потрібно створити мережеву симуляцію для передачі трафіку широкого або багатоадресного мовлення. OpenVPN пропонує користувачам кілька переваг, оскільки працює в просторі користувача, підтримує підключення NAT, шифрування TLS і сертифікати, має гнучку конструкцію і розширювану архітектуру. Протокол має три основні недоліки: він працює гірше, ніж IPSec, оскільки працює поза ядром і вимагає більше налаштувань, ніж WireGuard, а також потребує зовнішнього сервера і може мати проблеми з мобільністю без додаткової конфігурації. Основні сфери застосування OpenVPN включають безпечний віддалений доступ до корпоративної мережі, а також доступ до внутрішніх ресурсів підприємства, обмеження мережі та обхід цензури, а також об'єднання офісних мереж.

WireGuard – це сучасний протокол VPN, який забезпечує вищу продуктивність, підвищену безпеку та простішу експлуатацію, ніж IPSec та OpenVPN. Розробка програмного забезпечення WireGuard відбулася у 2015 році Джейсоном Доненфельдом, який зосередився на створенні легкого рішення з високою швидкістю та сучасними алгоритмами шифрування. WireGuard був інтегрований в ядро Linux у версії 5.6 у 2020 році. Віртуальний інтерфейс, через який передається весь зашифрований трафік, працює на мережевому рівні моделі OSI. Дизайн протоколу досяг такого рівня спрощення, що TLS, SSL, IKEv2 та HMAC були усунені, оскільки криптографічні елементи існують в рамках протоколу. Вузли зберігають власні пари відкритих і закритих криптографічних ключів. Після початку роботи тунелю пакети шифруються перед передачею через UDP, причому WireGuard не потребує інформації про стан для своєї роботи, оскільки встановлює з'єднання виключно під час передачі даних.

WireGuard реалізує свою криптографічну структуру за допомогою сучасних безпечних криптографічних примітивів. Протокол використовує

ChaCha20 для шифрування, а також Poly1305 для автентифікації з Curve25519 для обміну ключами та BLAKE2s для хешування. AEAD (Authenticated Encryption with Associated Data) забезпечує вбудовану функцію контролю цілісності в рамках протоколу. Протокол відхиляє застарілі методи шифрування, такі як RSA, AES та SHA1, для захисту від сучасних загроз та підтримки постквантових криптографічних операцій [19].

Процес встановлення з'єднання для WireGuard залишається простим. Кожен учасник має власний приватний ключ, а також список дозволених публічних ключів інших учасників. Передача пакетів вузлів передбачає шифрування ChaCha20, яке додає одноразовий код разом із позначкою часу та MAC перед передачею. Приймаюча сторона розшифровує вхідні дані, а потім виконує процедури автентифікації. Процес з'єднання в WireGuard працює інакше, ніж в IPSec і TLS, оскільки немає окремої фази встановлення сеансу. Конфігурація WireGuard здійснюється за допомогою простих текстових файлів. Користувачі повинні ввести свій приватний ключ разом з IP-адресою інтерфейсу та публічним ключем сервера на стороні клієнта, а потім вказати порт та дозволені IP-адреси. Конфігурація сервера відрізняється від конфігурації клієнта, оскільки використовує власний ключ разом з адресою та портом і публічним ключем клієнта.

Конфіг-файл клієнта:

```
[Interface]
PrivateKey = <client_private_key>
Address = 10.0.0.2/24

[Peer]
PublicKey = <server_public_key>
Endpoint = vpn.example.com:51820
AllowedIPs = 0.0.0.0/0
```

Конфіг-файл сервера:

```
[Interface]
PrivateKey = <server_private_key>
Address = 10.0.0.1/24
ListenPort = 51820

[Peer]
```

```
PublicKey = <client_public_key>
AllowedIPs = 10.0.0.2/32
```

Система WireGuard має численні переваги. Компактний розмір коду (менше 5000 рядків) спрощує проведення аудитів безпеки. Продуктивність WireGuard у 3-4 рази перевищує продуктивність OpenVPN та IPSec завдяки вбудованій підтримці ядра Linux. Протокол використовує виключно сучасні методи шифрування, забезпечуючи просту конфігурацію та підтримку NAT-T без додаткових вимог, а також працює без сторонніх бібліотек або протоколів. Однак WireGuard також має обмеження. Усі операції з WireGuard залежать від ключів, оскільки система не має вбудованих функцій автентифікації користувачів. Система не має автоматичних можливостей динамічної маршрутизації, тому користувачі повинні вручну налаштовувати всі маршрути. Обмеження стандартизації та підтримки існують для WireGuard у певних корпоративних мережах. Протокол не має власної системи контролю трафіку, тому користувачі повинні використовувати iptables або nftables для таких функцій. VPN-протокол WireGuard служить альтернативою IPSec для створення внутрішніх VPN-тунелів між офісами та захисту мобільного трафіку на пристроях Android і iOS, а також для середовищ DevOps і налаштувань Docker або Kubernetes і хмарних платформ, включаючи AWS, GCP і Azure.

Три найпопулярніші протоколи VPN мають різні характеристики, які впливають на їх придатність для конкретних випадків використання завдяки своїм унікальним перевагам і недолікам. Захист на рівні мережі вимагає використання IPSec як традиційного рішення. Це рішення забезпечує потужні функції шифрування, а також відповідність стандартам і функціональність тунельного та транспортного режиму. Основні переваги IPSec включають апаратне прискорення, гнучку конфігурацію та широку сумісність з корпоративним обладнанням. Складна конфігурація IPSec і його чутливість до NAT, а також численні параметри роблять це рішення менш зручним для динамічних і невеликих мереж. OpenVPN функціонує як протокол рівня

додатків, який використовує TLS/SSL для своєї роботи. Відкритий вихідний код і підтримка різних платформ, а також підтримка NAT роблять це рішення придатним для малих підприємств і індивідуальних користувачів. OpenVPN забезпечує масштабованість, а також підтримку аутентифікації на основі сертифікатів і гнучкі можливості адаптації до умов мережі. OpenVPN працює в просторі користувача, а не в ядрі ОС, що призводить до зниження продуктивності, що є його основним недоліком. Сучасний протокол VPN WireGuard пропонує просту архітектуру та компактний код з високою швидкістю і використовує новітні методи шифрування. Система працює на рівні мережі за допомогою пар ключів між вузлами, що забезпечує підвищену безпеку та зменшує навантаження на систему без необхідності складних обмінів. Ця технологія чудово підходить для мобільних і хмарних обчислювальних середовищ, а також для контейнерних систем. WireGuard не має вбудованої аутентифікації користувачів і надає обмежену підтримку в традиційних корпоративних мережах. Вибір протоколу VPN залежить від конкретних вимог кожного впровадження. Протокол IPSec оптимально працює у великих мережах, які мають стабільну роботу і вимагають надійних заходів безпеки. Поєднання гнучких і складних топологій з підтримкою TLS і NAT робить OpenVPN найкращим вибором. WireGuard є найкращим рішенням для сучасних систем, які вимагають високої швидкості, простоти експлуатації і надійної криптографічної безпеки.

2.2. Аналіз протоколів TLS/SSL для захисту трафіку

TLS (Transport Layer Security) та SSL (Secure Sockets Layer) були спочатку розроблені в середині 1990-х років компанією Netscape для забезпечення безпечного обміну даними між клієнтом і сервером в комп'ютерних мережах. SSL був спочатку створений в середині 1990-х років як спосіб шифрування веб-трафіку в браузерах. Перша версія SSL 2.0, яка була доступна для широкого

загалу, була випущена в 1995 році, але мала критичні проблеми з безпекою і зрештою була замінена SSL 3.0. Однак ця версія була визнана застарілою та вразливою до атаки POODLE (Padding Oracle On Downgraded Legacy Encryption), і її використання зараз настійно не рекомендується [20]. У 1999 році Інженерна група Інтернету (IETF) офіційно прийняла TLS 1.0 як наступника SSL 3.0. З того часу TLS став основним стандартом захисту мережевого трафіку. TLS 1.1 з'явився в 2006 році, а TLS 1.2 в 2008 році і протягом багатьох років залишався галузевим стандартом. Однак із розвитком криптоаналітичних методів та зростанням обчислювальної потужності виникла потреба в оновленому, більш безпечному та оптимізованому протоколі. У 2018 році було опубліковано TLS 1.3 – повністю перероблену версію, яка значно скорочує кількість кроків у рукостисканні, усуває застарілі криптографічні алгоритми (такі як криптографія RSA в обміні ключами, режими шифрування CBC) та покращує загальну безпеку та продуктивність [21].

Архітектура TLS базується на принципі розділення шарів. Основою є протокол записів, який відповідає за шифрування, автентифікацію та цілісність даних. Над ним працюють інші протоколи: протокол рукостискання для встановлення ключів і параметрів, протокол сповіщення для обробки помилок, протокол зміни шифрування для переходу до узгоджених криптографічних параметрів, та протокол даних додатків для передачі захищеного вмісту. У TLS 1.2 під час рукостискання клієнт і сервер узгоджують криптографічні параметри, обмінюються сертифікатами X.509, встановлюють спільний ключ (через RSA або DH) і тільки потім передають зашифровані дані. Рукостискання складається з декількох раундів, включає обчислення MAC, підтримує взаємну автентифікацію і дозволяє використовувати велику кількість криптографічних комбінацій, що ускладнює аудит і сприяє виникненню помилок. TLS 1.3 значно спрощує архітектуру. Установлення з'єднання зведено до одного раунду обміну. Підтримка застарілих алгоритмів (MD5, RC4, SHA1, RSA для обміну ключами, режими CBC) була повністю видалена, дозволено лише шифрування AEAD (наприклад, AES-GCM або ChaCha20-Poly1305) та обмін ключами через

(EC)DHE. Це підвищує безпеку, зменшує затримки з'єднання та покращує стійкість до атак з пониженням версії та протоколів Forward Secrecy. TLS 1.3 також надає механізм 0-RTT (Zero Round-Trip Time), який дозволяє клієнту надсилати дані одразу після першого повідомлення, але з обмеженим рівнем безпеки для повторних сесій [22].

TLS є основою безпечного Інтернету: він лежить в основі HTTPS, VPN-з'єднань, електронної пошти, VoIP, захисту баз даних та внутрішнього корпоративного трафіку. Ось чому розуміння його архітектури та криптографічних основ є критично важливим для побудови сучасних безпечних систем. Після встановлення TCP-з'єднання між клієнтом і сервером протокол TLS переходить до фази встановлення безпечного каналу, яка називається TLS-рукостисканням. Під час рукостискання сторони узгоджують криптографічні параметри, автентифікують одна одну (якщо потрібно), генерують спільний сеансовий ключ і переходять до обміну зашифрованими даними. Цей етап є критично важливим, оскільки він визначає не тільки безпеку каналу, але й його стійкість до потенційних атак. У TLS 1.2 рукостискання включає кілька обов'язкових кроків. Спочатку клієнт надсилає повідомлення ClientHello, яке містить список підтримуваних криптографічних наборів (наборів шифрів), версію протоколу, випадкове значення, а також може містити SNI, ALPN та інші розширення. Сервер відповідає повідомленням ServerHello, в якому вибирає один із запропонованих наборів, надсилає власне випадкове значення та свій сертифікат з відкритим ключем. Далі, залежно від типу обраного обміну ключами (наприклад, RSA або DH), клієнт генерує попередній секретний ключ, шифрує його відкритим ключем сервера і відправляє назад. Обидві сторони незалежно обчислюють спільний сесійний ключ на основі попереднього секретного ключа. Після цього надсилається повідомлення ChangeCipherSpec і передаються зашифровані повідомлення Finished, що сигналізують про закінчення фази рукостискання.

У TLS 1.3 кількість повідомлень було мінімізовано, а підтримка застарілих протоколів обміну ключами було скасовано. Алгоритми обміну ECDHE та DH

слугують основою для рукостискання, забезпечуючи пряму секретність. Клієнт передає повідомлення ClientHello, що містить набори криптографічних алгоритмів та параметри обміну ключами, що спонукає сервер надіслати ServerHello з вибраними параметрами та частиною відкритого ключа. Відразу після рукостискання сторони генерують спільний сесійний ключ за допомогою протоколу DH. Сервер надає сертифікат і підписує дані для завершення процедури автентифікації. Усі подальші комунікації відбуваються за допомогою шифрування. Зменшена кількість раундів прискорює встановлення з'єднання, що є перевагою як для мобільних пристроїв, так і для серверів з високим навантаженням [22].

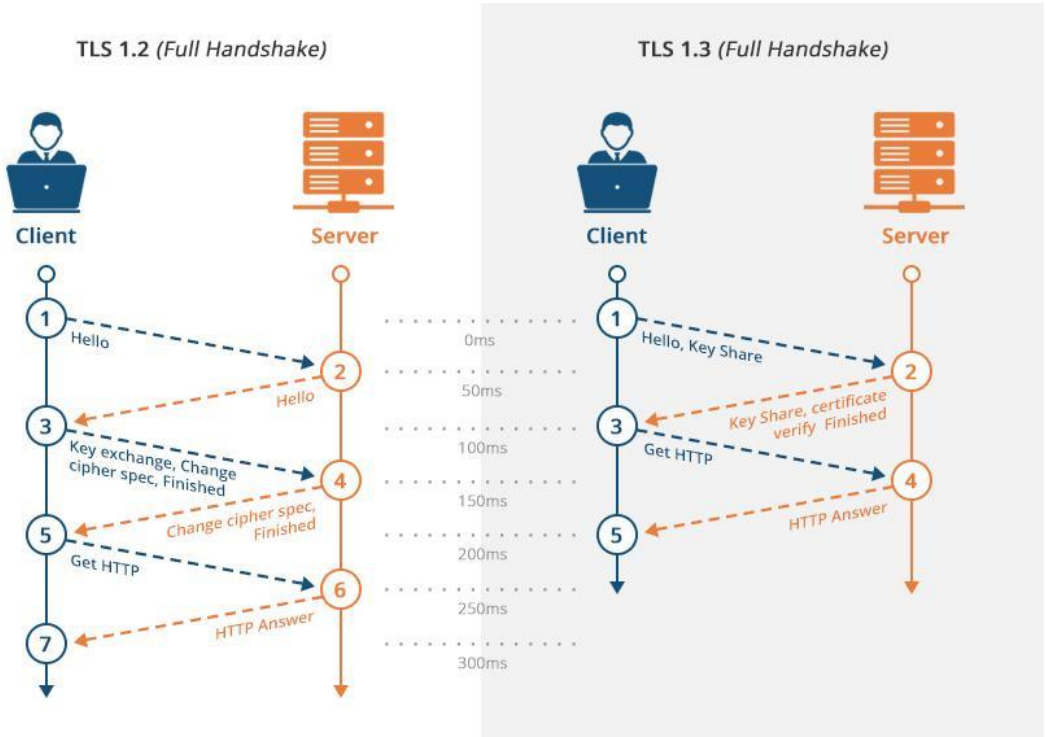

Рис. 2.5. Схематичне порівняння TLS версій 1.2 та 1.3 [23]

TLS виконує шифрування даних за допомогою алгоритмів AEAD, таких як AES-GCM і ChaCha20-Poly1305. Поєднання конфіденційності даних з цілісністю та автентичністю в цих алгоритмах усуває необхідність окремого впровадження MAC, мінімізуючи ризики, пов'язані з помилками впровадження. TLS 1.2 дозволяє використовувати окремі функції MAC (HMAC-SHA256, HMAC-SHA384) разом з режимами CBC, проте ці режими залишаються вразливими до атак типу padding oracle, тому TLS 1.3 видаляє їх зі своєї специфікації. Механізм

обміну ключами ECDHE дозволяє обчислювати спільні секрети, не піддаючи їх передачі по мережі. Короткочасний характер сесійних ключів гарантує, що перехоплені дані не можуть бути розшифровані, навіть якщо хакерам вдасться викрасти постійні ключі шифрування. Концепція пересилання секретності є фундаментальним принципом, який TLS 1.3 вимагає як обов'язкову вимогу [24].

TLS захищає канал зв'язку за допомогою декількох рівнів шифрування, що поєднують шифрування та автентифікацію, перевірку цілісності та динамічне узгодження параметрів. Функції безпеки TLS 1.3 були вдосконалені, а також досягнуто більшої ефективності, що робить протокол корисним для різних сучасних мережевих додатків, включаючи веб-трафік та середовища IoT. Інфраструктура відкритих ключів (PKI) служить основою для операцій TLS для автентифікації мережевих об'єктів. Цифрові сертифікати, видані X.509, є важливим компонентом цієї системи, оскільки вони містять відкриті ключі разом з даними про власника, терміни дії та цифрові підписи від центрів сертифікації (CA). Центри сертифікації виступають довіреними суб'єктами, які перевіряють справжність відкритих ключів. Під час операцій узгодження TLS сервер надсилає свій сертифікат клієнтам, які потім перевіряють його дійсність за допомогою довірених підписів CA та статусу закінчення терміну дії, а також скасування сертифіката CRL або OCSP (за допомогою CRL або OCSP) [25]. PKI забезпечує безпечні з'єднання між незнайомими сторонами за допомогою механізму автентифікації. TLS підтримує взаємну автентифікацію за допомогою сертифікатів, які сервер і клієнт обмінюються під час з'єднання, особливо в корпоративних VPN та фінансових службах.

Протягом усього періоду свого існування SSL/TLS зазнав впливу численних відомих вразливостей. BEAST (Browser Exploit Against SSL/TLS) є критичною вразливістю, оскільки вона дозволяла розшифрувати дані в TLS 1.0 через слабкі режими CBC. Порушення стиснення даних дозволяють здійснювати атаки CRIME та BREACH. Атака POODLE використовувала неправильну обробку заповнення для атак на системи SSL 3.0. Багато реалізацій TLS містили помилки управління сертифікатами, що дозволяли атаки типу «людина

посередині». Старі версії протоколів стикаються з ризиками безпеки через відомі недоліки, тому вони отримали офіційний статус застарілих (SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1) [26]. Версія TLS 1.3 забезпечує підвищену безпеку завдяки усуненню застарілих компонентів та обмежених криптографічних опцій, що мінімізує ймовірність помилок реалізації. Поточна версія більше не підтримує статичні ключі та RSA для режимів обміну або CBC, а також RC4 та стиснення. Протокол забезпечує пересилання секретності шляхом застосування ECDHE разом із шифруванням AEAD та перевіреними хеш-функціями. TLS 1.3 забезпечує повний захист від різних типів атак завдяки своїм розширеним функціям безпеки [27]. TLS має широкий спектр застосування. Основним застосуванням TLS є HTTPS, який забезпечує шифрування трафіку веб-сайтів за допомогою безпечної передачі гіпертексту і функціонує як стандартний протокол. TLS працює в протоколах електронної пошти, таких як SMTP, IMAP і POP3, використовуючи методи STARTTLS і пряме шифрування портів. TLS виконує функції тунелювання та автентифікації в OpenVPN, а також у системах віддаленого доступу AnyConnect і SoftEther. TLS служить для шифрування каналів у різних додатках, включаючи бази даних PostgreSQL, MySQL і MongoDB, мобільні додатки та VoIP через SIP через TLS і внутрішні системи обміну повідомленнями для бізнесу. TLS версії 1.3 є необхідною вимогою для організацій будь-якого розміру для забезпечення інформаційної безпеки.

2.3. Порівняльний аналіз алгоритмів шифрування (AES, RSA, ECC, SHA)

AES служить симетричним блоковим шифром, який уряд США встановив як стандарт шифрування в 2001 році через FIPS PUB 197. Основна мета AES полягає в захисті цифрової інформації від несанкціонованого доступу під час обміну даними в Інтернеті. Основні особливості:

- Тип: симетричний (один ключ для шифрування та дешифрування);

- Розмір блоку: 128 біт;
- Довжина ключа: 128, 192 або 256 біт;
- Кількість раундів:
    - AES-128 – 10 раундів;
    - AES-192 – 12 раундів;
    - AES-256 – 14 раундів.

AES базується на мережі заміщення-перестановки (Substitution-Permutation Network). Основні етапи шифрування:

1. SubBytes – нелінійна заміна байтів за S-блоком;
2. ShiftRows – циклічний зсув рядків блоку;
3. MixColumns – лінійне змішування колонок (крім останнього раунду);
4. AddRoundKey – побітове додавання з раундовим ключем.

Ключі для кожного раунду генеруються з головного ключа через Key Expansion [28].

Таблиця 2.1.

Переваги та недоліки AES

Переваги	Недоліки
Висока швидкість програмної й апаратної реалізації;	Чутливість до атак при неправильному використанні режимів шифрування (наприклад, ECB);
Широка підтримка на всіх платформах;	Не забезпечує цілісність або автентичність – лише конфіденційність;
Стандартизований;	Як і всі симетричні алгоритми, вимагає безпечного розповсюдження ключів;
Стійкий до відомих атак (диференційний криптоаналіз, лінійний аналіз).	AES не є квантово-стійким – квантові комп’ютери можуть скоротити ефективну криптостійкість AES удвічі завдяки алгоритму Ґровера.

Перша система асиметричного шифрування та цифрового підпису, яка отримала широке поширення, з'явилася в 1977 році завдяки Рональду Рівесту, Аді Шаміру та Леонарду Адлеману. Криптографічна міцність RSA базується на складному процесі розкладання великих чисел на прості множники [29]. Основні особливості:

- Тип: асиметричний (пара ключів: відкритий + закритий);

- Криптографічна основа: складність факторизації добутку двох великих простих чисел;
- Розмір ключа: 1024-4096 біт (рекомендовано ≥ 2048 біт згідно NIST);
- Призначення: шифрування, підпис, обмін ключами.

Математична основа RSA:

1. Алгоритм RSA базується на складності розкладання добутку великих простих чисел. На першому етапі генеруються два великі прості числа – p та q. На їх основі обчислюється модуль:
2. $$n = p \times q \tag{2.1}$$
3. Далі обчислюється значення функції Ейлера:

$$\varphi(n) = (p-1)(q-1) \tag{2.2}$$

4. Вибирається відкритий експонент e, який є взаємно простим з $\varphi(n)$, тобто:

$$\gcd(e, \varphi(n)) = 1 \tag{2.3}$$

5. Після цього обчислюється закритий експонент d, який є мультиплікативно оберненим до е за модулем $\varphi(n)$:

$$d = e^{-1} \, mod \, \varphi(n) \tag{2.4}$$

Принцип роботи алгоритму:

1. Шифрування повідомлення M виконується за допомогою відкритого ключа:

$$C = M^e \, mod \, n \tag{2.5}$$

2. Розшифрування здійснюється за допомогою закритого ключа:

$$\text{M}= C^d \ mod \ n \tag{2.6}$$

3. Створення цифрового підпису базується на хеші повідомлення $H(M)$:

$$S = H(M)^d \ mod \ n \tag{2.7}$$

4. Перевірка підпису виконується через піднесення підпису до степеня e:

$$H(M) = S^e \ mod \ n \tag{2.8}$$

Таблиця 2.2.

Переваги та недоліки RSA

Переваги	Недоліки
Підтримує цифровий підпис;	Повільний у порівнянні з симетричними алгоритмами (зокрема AES);
Не потребує попередньої передачі ключів;	Не придатний для шифрування великих обсягів даних;
Може використовуватись для обміну ключами симетричних алгоритмів;	Чутливий до атак при неправильному заповненні повідомлень;
Поширений у TLS, S/MIME, PGP.	Уразливий до квантових атак.

ECC це сучасна система асиметричної криптографії, яка використовує дискретні логарифми в групах точок еліптичних кривих над скінченними полями замість розкладу на прості множники або дискретних логарифмів у простих групах. Ніл Кобліц і Віктор Міллер незалежно один від одного запропонували ECC у 1985 році. ECC забезпечує порівнянний рівень безпеки за допомогою

набагато менших розмірів ключів, ніж системи RSA або DH. Основні особливості:

- Тип: асиметричний (пара ключів: відкритий + закритий);
- Математична основа: складність задачі дискретного логарифма на еліптичних кривих (ECDLP);
- Розміри ключів:
    - ECC-256 ≈ безпека RSA-3072;
    - ECC-384 ≈ RSA-7680;
    - ECC-521 ≈ RSA-15360;
- Алгоритми: ECDSA, ECDH, EdDSA.

Еліптична крива над скінченним полем описується рівнянням виду:

$$y^2 = x^3 + ax + b \, mod(p) \tag{2.9}$$

де:

- a і b – коефіцієнти, що визначають форму кривої;
- p – просте число, яке задає характеристику поля;
- рівняння визначає множину точок (x,y), що задовольняють умову для певного a, b, та p.

Для того щоб крива була коректною (без особливих точок), має виконуватись умова:

$$4a^3 + 27b^3 \not\equiv 0 \, mod(p) \tag{2.10}$$

Точки, які лежать на еліптичній кривій (разом із спеціальною точкою «на нескінченності»), утворюють абелеву групу відносно операції додавання точок.

Операція "додавання" точок означає:

- геометричне поєднання двох точок на кривій;

- або повторне додавання однієї й тієї ж точки (як у множенні).

Генерація ключів:

1. Фіксується точка G – генератор на кривій (публічно відома).
2. Закритий ключ d – випадково вибране ціле число в діапазоні:

$$1 \leq d \leq n \tag{2.11}$$

де n – порядок точки G, тобто найменше число, для якого $nG = O$ (нейтральний елемент).

3. Відкритий ключ обчислюється як добуток:

$$Q = d \times G \tag{2.12}$$

де Q – точка на кривій, що є результатом додавання точки G до себе d разів.

Поширені алгоритми:

- ECDH (Elliptic Curve Diffie-Hellman) – безпечний обмін ключами;
- ECDSA (Elliptic Curve Digital Signature Algorithm) – цифровий підпис;
- EdDSA (Edwards-Curve Digital Signature Algorithm) – новітня, оптимізована версія ECDSA з покращеною стійкістю до атак сторонніх каналів.

Таблиця 2.3.

Переваги та недоліки ECC

Переваги	Недоліки
Малі розміри ключів та підписів – менше навантаження на пам’ять і трафік;	Вища складність реалізації – більший ризик помилок;
Швидша генерація ключів, підписів і верифікація у порівнянні з RSA;	Складний для аудиту – глибока математика, яка потребує ретельної перевірки;
Високий рівень криптостійкості при коротших параметрах;	Вразливість до атак сторонніх каналів, якщо не використовуються захищені реалізації;
Підтримка forward secrecy через ECDH;	Піддається атаці Шора, як і RSA.

SHA представляє сімейство криптографічних хеш-функцій, розроблених

Національним інститутом стандартів і технологій США (NIST) для захисту цілісності даних і автентифікації цифрових підписів. Хеш-функція працює як односторонній алгоритм, який перетворює довільні вхідні повідомлення в хеш-коди фіксованого розміру. Криптографічна хеш-функція повинна відповідати трьом основним вимогам:

- хеш-функція повинна запобігати відновленню користувачами вихідного повідомлення з вихідних даних хешу (односторонній);
- система повинна бути стійкою до знаходження двох різних повідомлень, які генерують ідентичні хеш-виходи (стійкість до колізій);
- Алгоритм повинен запобігати зловмисникам виявити будь-яке повідомлення, яке генерує заданий хеш-вихід.

Алгоритм SHA-1 створює 160-бітний хеш. Алгоритм застарів, оскільки дослідники з Google і CWI успішно продемонстрували колізії проти нього. Більшість криптографічних систем виключили SHA-1 зі своїх сучасних фреймворків через виявлені вразливості [30]. Група SHA-2 включає в себе кілька варіантів, таких як SHA-224, SHA-256, SHA-384 і SHA-512, які прийшли на зміну алгоритму SHA-1. SHA-256 разом з SHA-512 є найпоширенішими алгоритмами в цій галузі. Протоколи TLS разом з електронними підписами (PDF-документами) та технологіями блокчейн активно використовують ці алгоритми. Мережа Bitcoin використовує SHA-256 як основну хеш-функцію. Структура SHA-2 використовує мережі Файстеля і виконує побітові логічні операції, включаючи XOR і AND, а також побітове обертання (ROTR) і додавання за модулем 2^{32} або 2^{64}, залежно від варіанту [32]. SHA-2 залишається криптографічно безпечним станом на 2024 рік, оскільки практичних вразливостей виявлено не було. Для підвищення стійкості до потенційних атак на SHA-2 було розроблено новий криптографічний стандарт під назвою SHA-3. Математична основа SHA-3 відрізняється від попередників, оскільки використовує структуру «губки» алгоритму Keccak. Внутрішня структура SHA-3 функціонує інакше, ніж у попередників, хоча підтримує ті самі розміри хеш-функцій (224, 256, 384 та 512 бітів). Система SHA-3 включає вбудовані варіанти автентифікації, такі як SHAKE128 та SHAKE256,

які пропонують гнучкі довжини вихідних хеш-функцій та додаткові можливості [33].

Хеш-функції сімейства SHA мають кілька переваг, які зумовлюють їх широке застосування в системах інформаційної безпеки. SHA-2 забезпечує високу продуктивність на більшості сучасних апаратних і програмних платформ, що робить його невід'ємною частиною цих систем. Алгоритми SHA-2 і SHA-3 мають математичний доказ стійкості до колізій, атак на преображення та вразливостей відновлення повідомлень. Їхні потужні функції безпеки роблять їх придатними для важливих криптографічних операцій. Впровадження цих алгоритмів у практичні протоколи та технології є однією з їхніх основних переваг. Цифрові підписи та безпечні протоколи TLS/SSL і системи блокчейнів, а також функції контролю цілісності даних і зберігання паролів із використанням солоних хеш-функцій (наприклад, SHA-2 і SHA-3) використовуються в цих додатках. Структура «губка» разом з альтернативною математичною основою в SHA-3 забезпечує додаткову стійкість до нових класів атак, що підвищує його безпеку. Кожен алгоритм має певні слабкі місця. Алгоритм SHA-1 більше не вважається безпечним, оскільки були доведені фактичні колізії, тому його було вилучено з криптографічних протоколів. Незважаючи на те, що на сьогодні SHA-2 є криптографічно безпечним, він має потенційні ризики, оскільки його конструкція має схожість із SHA-1. Хоча SHA-3 забезпечує високу стійкість, він вимагає більш складного впровадження, що може створити проблеми під час його розгортання в певних середовищах.

3 ПРАКТИЧНІ РЕКОМЕНДАЦІЇ ЩОДО ВПРОВАДЖЕННЯ КРИПТОГРАФІЧНОГО ЗАХИСТУ

3.1. Рекомендації щодо вибору алгоритмів та протоколів залежно від задач організації

Організації повинні вибирати відповідні криптографічні алгоритми та протоколи, а також сучасні інструменти при побудові своєї системи захисту. Різні криптографічні механізми мають унікальні переваги та обмеження, які визначають сфери їх застосування. Ефективність криптографічних рішень у банківському секторі не означає, що вони підходять для потреб малого бізнесу, і навпаки. Кожна організація має свою унікальну комбінацію бізнес-операцій, викликів у сфері безпеки, технологічних ресурсів та організаційного потенціалу. Вибір протоколів та методів шифрування вимагає індивідуальних рекомендацій, а не загальних, оскільки вони повинні базуватися на конкретних особливостях бізнесу.

Перш за все, слід звернути увагу на важливість інформації, що обробляється. Організації, які обробляють персональні дані разом з фінансовими звітами, медичною та урядовою інформацією, повинні використовувати надійні алгоритми шифрування, такі як AES-256 і RSA-3072, ECC-SECP384R1 і SHA-2 або SHA-3, які мають національну або міжнародну сертифікацію регуляторних органів (наприклад, NIST, ISO, ДСТУ). Малі підприємства потребують базового рівня захисту за допомогою простих, але надійних засобів, що відповідають сучасним протоколам внутрішнього документообігу. Невеликі організації з обмеженою кількістю серверів, IT-персоналу та бюджету на безпеку повинні використовувати прості, але надійні рішення:

- VPN на базі WireGuard або OpenVPN з шифруванням AES-128 або AES-256;

- TLS 1.3 слід використовувати для веб-доступу як до панелей адміністратора, так і до поштових служб.
- Для ресурсів, що не є критично важливими, слід використовувати самопідписані сертифікати або сертифікати Let's Encrypt;
- Система автентифікації використовує паролі та багатофакторну автентифікацію (2FA) без впровадження складних процедур авторизації.

Організації, що мають хмарні операції з клієнтами та декілька офісів, повинні впровадити більш адаптивну структуру для своїх операцій:

- Налаштування VPN на основі IPSec або OpenVPN повинно бути інтегровано з брандмауерами, маршрутизаторами та корпоративними шлюзами;
- Для аутентифікації слід використовувати сертифікати TLS, видані зовнішніми сертифікаційними органами (GlobalSign, Sectigo), з функціями автоматичного поновлення сертифікатів.
- Для оптимізації продуктивності система повинна використовувати AES-256 разом з SHA-256 або SHA-3.
- Великі організації з централізованими системами управління ключами повинні використовуватися для обробки великих обсягів обміну даними.

Великим корпораціям та організаціям, що працюють зі складною інфраструктурою та декількома рівнями доступу, слід розглянути можливість впровадження інтегрованих криптографічних інструментів:

- Криптографічна система автентифікації повинна пов'язувати політики доступу, що розділяють ролі користувачів (RBAC, ABAC), з системами автентифікації Kerberos та RADIUS.
- Усі внутрішні вузли та сервери вимагають впровадження сертифікатів X.509, включаючи системи CRM та ERP для внутрішнього HTTPS-з'єднання.
- Для зберігання критичних ключів шифрування системи необхідні апаратні системи зберігання ключів, що включають HSM і TPM.
- Для передачі ключів шифрування слід використовувати асиметричну

криптографію (наприклад, ECC) і систему автоматичної ротації ключів.

Організації, що працюють в області IoT і мобільних технологій, а також мають обмежені обчислювальні ресурси, потребують легких, але надійних алгоритмів. Основними факторами в цій ситуації є невеликі накладні витрати, сумісність з процесорами ARM і енергоефективність. Тут доцільно:

- У цій ситуації RSA слід замінити на ECC (Curve25519).
- Збалансоване поєднання швидкості та надійності досягається завдяки вибору AES-128-GCM.
- Протокол VPN WireGuard є відповідним рішенням завдяки мінімальній кодовій базі.
- Система автентифікації повинна бути доступною на всіх пристроях, включаючи невеликі контролери.

Вибір протоколів та алгоритмів повинен ґрунтуватися на конкретних сценаріях впровадження, а не на виборі з переліку «найкращих технологій». Кожна організація потребує конкретного балансу між заходами безпеки та продуктивністю, а також простими методами впровадження.

3.2. Реалізація та налаштування VPN-з’єднання в локальній мережі (на прикладі IPSec/OpenVPN)

Для криптографічного захисту трафіку в локальній мережі організації слід розглянути можливість впровадження VPN-тунелів, що використовують протоколи IPSec або OpenVPN. У наступному розділі наведено типову архітектуру та рекомендації щодо конфігурації.

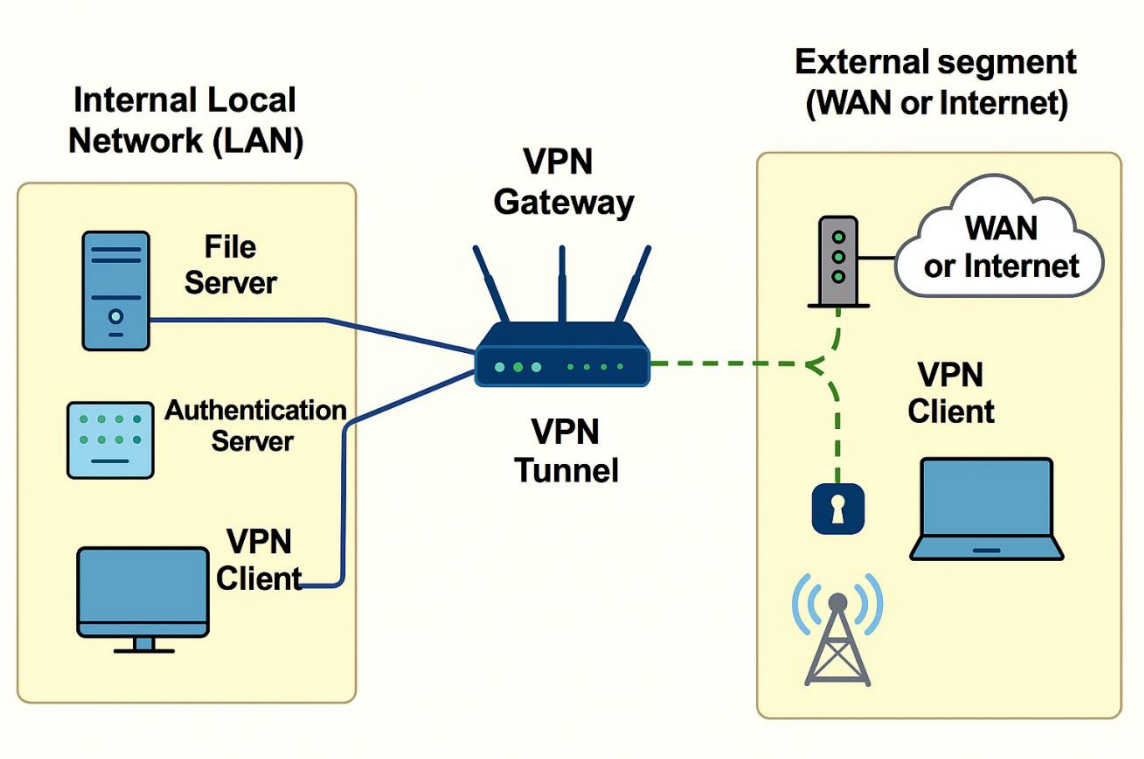

Рис. 3.1. Архітектура VPN-з’єднання в локальній мережі

1. Внутрішня локальна мережа (LAN). Всі внутрішні ресурси, такі як файлові сервери та бази даних з веб-додатками та робочі станції співробітників, підключаються через цю мережеву інфраструктуру.
2. VPN-шлюз. VPN-шлюз працює як окремий мережевий пристрій, який обробляє VPN-з'єднання та маршрутизує їх через мережу. До пристроїв VPN-шлюзів, що обробляють протоколи IPSec, належать маршрутизатори Cisco, MikroTik і pfSense, а також інші апаратні та програмні шлюзи. Сервер OpenVPN працює на системах Linux з OpenVPN і pfSense або OpenWrt, встановленими для роботи VPN.
3. Зовнішній сегмент (WAN або Інтернет). Зовнішня мережа надає віддаленим працівникам і філіям відкритий доступ для підключення до внутрішньої мережі через шифрування VPN.
4. VPN-клієнти. VPN-клієнти складаються з мобільних і стаціонарних робочих станцій, які створюють безпечні мережеві з'єднання з внутрішньою мережею через VPN-шлюз.

Детальна схема мережевого трафіку та взаємодії**:**

1. Робочі станції користувачів поза організацією підключаються до VPN-

шлюзу.

2. VPN-шлюз виконує аутентифікацію клієнтів за допомогою сертифікатів, логіна/пароля або ключів перед створенням зашифрованого VPN-з'єднання.
3. Клієнт встановлює доступ до ресурсів внутрішньої мережі після успішної аутентифікації.
4. VPN-тунель автоматично шифрує всі дані, що проходять через нього, за допомогою обраного протоколу (IPSec/OpenVPN).
5. VPN-шлюз інтегрує сервер аутентифікації, який може бути RADIUS або AD, для централізованого управління обліковими записами.

Для встановлення безпечного з'єднання в корпоративній мережі IPSec працює в тунельному режимі з протоколом ESP для шифрування IP-пакетів між VPN-шлюзами та клієнтськими пристроями. Для захисту даних використовується шифрування AES з 256-бітним ключем. Хеш-функція SHA-256 перевіряє цілісність та автентичність повідомлень. Протокол IKEv2 встановлює операції обміну ключами та процедури налаштування безпечного каналу. Для аутентифікації використовуються асиметричні криптографічні ключі RSA довжиною 2048 або 3072 біти. Система сертифікації стандарту X.509 виконує аутентифікацію клієнта і сервера за допомогою сертифікатів, що походять з внутрішніх або зовнішніх центрів сертифікації. Для підвищення безпеки слід суворо уникати використання слабких криптографічних алгоритмів, включаючи DES і 3DES, а також застарілих хеш-функцій, таких як SHA-1. Для мінімізації ризиків під час тривалого використання однієї сесії слід встановити автоматичне переузгодження ключів через певні проміжки часу або за обсягом переданих даних.

3.3. Впровадження TLS-сертифікатів для внутрішніх веб-ресурсів

Використання сертифікатів TLS дозволяє шифрувати мережеві дані як для

аутентифікації, так і для захисту трафіку. Організація може отримати сертифікати двома основними способами.

- Самопідписані сертифікати. Основна мета цих сертифікатів – захист внутрішніх мереж з некомерційними та закритими ресурсами. Перевага: прості у створенні. Користувачі отримують попередження браузера, коли стикаються з сертифікатами цього типу, оскільки браузери не довіряють їм.
- Безкоштовні та широко прийняті сертифікати Let's Encrypt найкраще підходять як для публічних ресурсів, так і для підприємств будь-якого розміру. Основним недоліком є необхідність періодичного поновлення кожні 90 днів.

Впровадження сертифікатів Apache TLS у системах Windows вимагає виконання таких кроків:

1. Відкрити термінал (cmd або Git Bash) з правами адміністратора та перейти у папку, де знаходиться OpenSSL - xampp\apache\bin.
2. Виконати команду:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/myserver.key -out /etc/ssl/certs/myserver.crt
```

- req -x509 – генерує сертифікат формату X.509.
- nodes – не шифрувати приватний ключ (для спрощення запуску Apache).
- days 365 – сертифікат буде дійсним 365 днів.
- newkey rsa:2048 – створює новий приватний ключ (RSA, 2048 біт).
- keyout та out – вказують файли для збереження ключа та сертифіката.

3. Переміщення сертифікатів до відповідних директорій Apache. Попередня команда згенерувала myserver.crt та myserver.key у папці /apache/bin. Тепер потрібно ці файли перенести у:

- myserver.key до apache\conf\ssl.crt\
- myserver.key до C:\xampp\apache\conf\ssl.key\

Це стандартні директорії XAMPP для зберігання SSL-сертифікатів і ключів. Стандартний віртуальний хост Apache з TLS виглядає так:

```
</VirtualHost>
<VirtualHost *:443>
    ServerName yourdomain.com
    DocumentRoot /var/www/html

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/myserver.crt
    SSLCertificateKeyFile /etc/ssl/private/myserver.key

    <Directory /var/www/html>
        Options -Indexes +FollowSymLinks
        AllowOverride All
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

4. Відкрити файл конфігурації TLS – apache\conf\extra\httpd-ssl.conf та замінити відповідні рядки на:

```
SSLCertificateFile "conf/ssl.crt/myserver.crt"
SSLCertificateKeyFile "conf/ssl.key/myserver.key"
```

Браузер може показати попередження, бо сертифікат самопідписаний (не довірений глобальними центрами сертифікації).

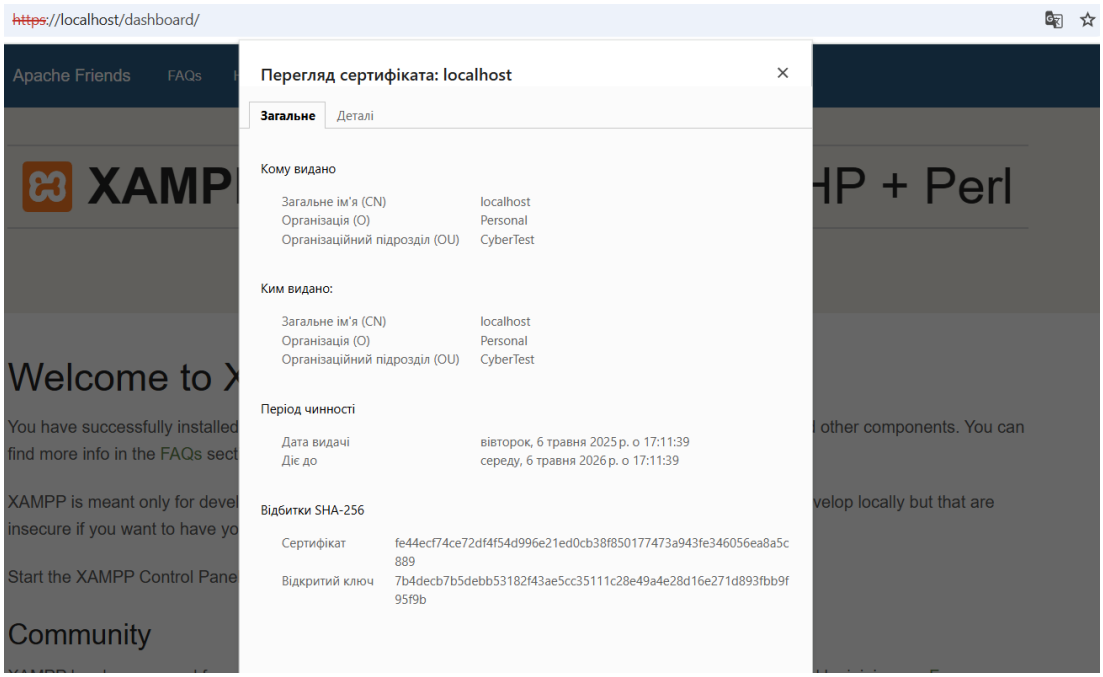

Рис. 3.2. Результат створення самопідписаного TLS-сертифікату в Apache на Windows

Впровадження TLS через Let's Encrypt.

1. Для початку треба зареєструвати доменне ім’я http://www.letsencryptest.ct.ws/?i=1, яке буде використовуватися для тестування впровадження TLS-сертифіката.

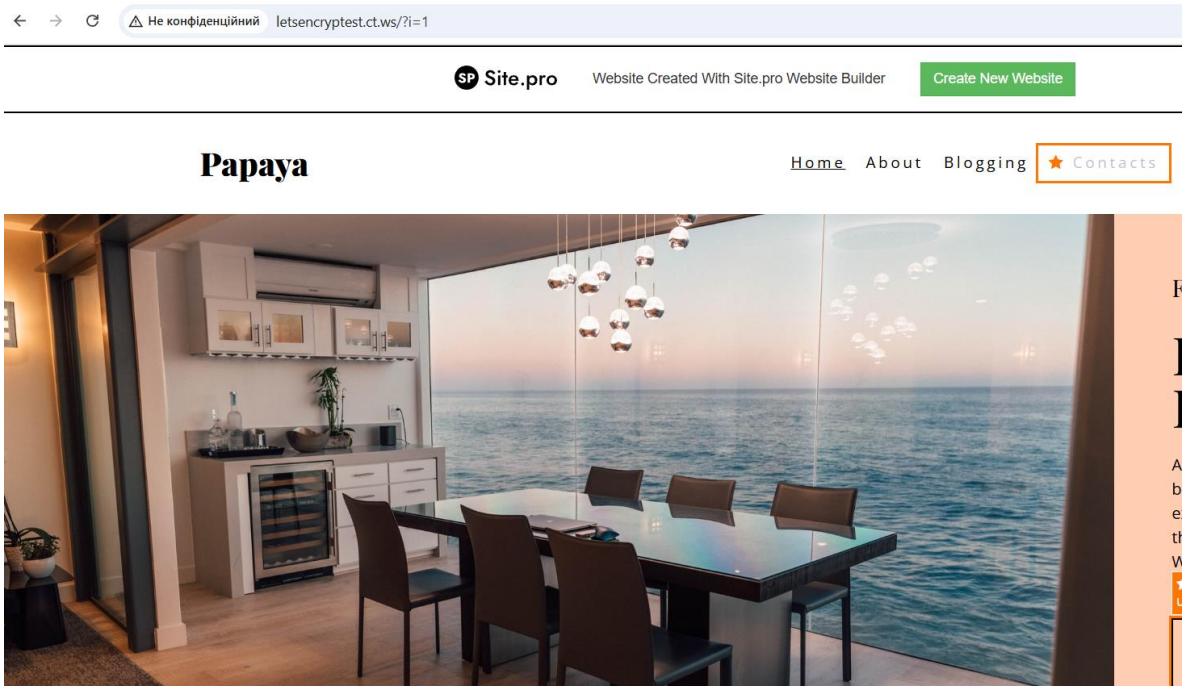

Рис. 3.3. Домен без TLS сертифікату

2. Наступним етапом було розпочато процес створення сертифікатів TLS для призначеного домену. Для цього було використано безкоштовну службу видачі сертифікатів. ZeroSSL, Get HTTPS for free, Certbot та Buypass Go SSL – можливі та рекомендовані варіанти.

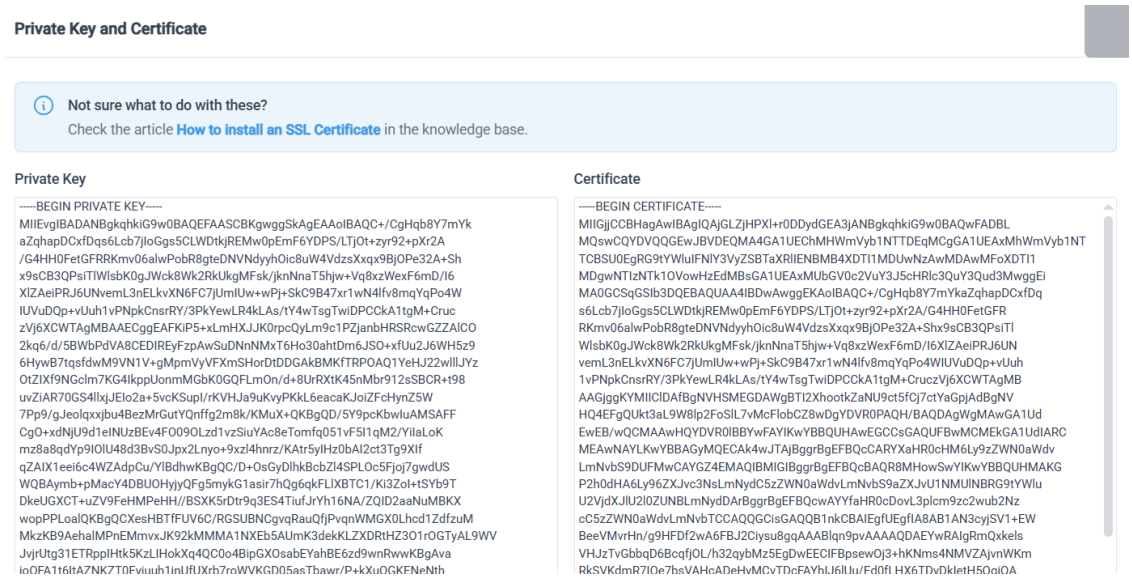

Рис. 3.4. Успішно створений сертифікат

Більшість хостинг-провайдерів надають вбудовані інструменти, які

дозволяють генерувати та розгортати сертифікати TLS без використання зовнішніх утиліт, що спрощує реалізацію захисту криптографічного з'єднання.

3. Користувачам потрібне HTTPS-посилання з оновленою адресою для доступу до веб-ресурсів, на яких увімкнено сертифікати TLS. Безпечне з'єднання підтверджує, що процес інтеграції цифрового сертифіката було успішно завершено. Наприклад, після встановлення сертифіката доступ до сайту здійснюється за адресою: https://letsencryptest.ct.ws/?i=1.

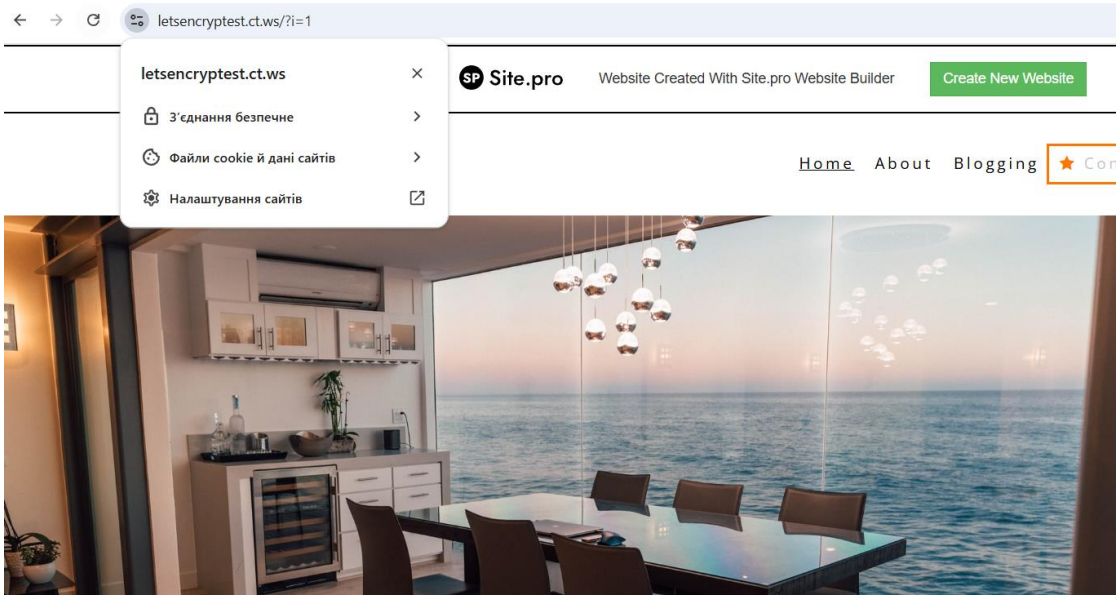

Рис. 3.5. Встановлений сертифікат

Під час впровадження захисту веб-ресурсів TLS було розглянуто два основні типи сертифікатів: самопідписані сертифікати та сертифікати публічного центру сертифікації Let's Encrypt. Інструмент OpenSSL створив самопідписані сертифікати, які потім були інтегровані в конфігурацію веб-сервера Apache. Цей метод можна використовувати у внутрішній мережі та тестових або обмежених середовищах, де офіційна перевірка не потрібна. Самопідписані сертифікати забезпечують просте створення та гнучке налаштування, але браузери показуватимуть попереджувальні повідомлення, оскільки вони не визнаються надійними, якщо не додані до сховищ сертифікатів. Було зроблено спробу впровадження безкоштовного сертифіката Let's Encrypt через автоматизований публічний центр сертифікації. Процес перевірки власності домену

використовував DNS через записи CNAME. Більшість браузерів визнають ці сертифікати надійними, що робить їх ідеальними для веб-сервісів, доступних для широкого загалу. Головною перевагою є те, що клієнтські пристрої можуть інтегруватися автоматично, без додаткових кроків. Основним недоліком є обмежений термін дії (90 днів), що вимагає періодичного поновлення або налаштування автоматизації.

ВИСНОВКИ

Сучасні методи криптографічного захисту повинні бути впроваджені організаціями, які стикаються з розширенням кіберзагроз, оскільки вони зменшують ризик витоку даних і захищають цілісність та конфіденційність інформації, блокуючи несанкціонований доступ до конфіденційних даних та забезпечуючи безпеку обміну даними між внутрішніми мережами та зовнішніми користувачами і службами. Організації можуть виконати поточні міжнародні стандарти, впровадивши ці заходи, які запобігають фінансовим збиткам, репутаційній шкоді та юридичним санкціям від кібератак.

1. У дослідженні вивчалися як теоретичні аспекти, так і сучасні застосування криптографічного захисту в локальних мережах організацій. Організаціям необхідний ефективний захист інформації для збереження конфіденційності, цілісності та автентичності даних в умовах зростаючих кіберзагроз.
2. Інформація захищається трьома типами криптографічних алгоритмів: симетричними алгоритмами (AES), асиметричними алгоритмами (RSA, ECC) та хеш-функціями (SHA). Різні типи організацій можуть використовувати ці криптографічні методи для конкретних організаційних завдань і вимог.
3. У дослідженні розглянуто три широко використовувані протоколи VPN, а саме IPSec, OpenVPN та WireGuard. IPSec забезпечує гнучкість та підтримку апаратного забезпечення, що робить його придатним для великих організацій, тоді як OpenVPN найкраще підходить для середніх підприємств завдяки простоті налаштування та широким можливостям підтримки, а WireGuard для малих підприємств завдяки простоті розгортання, високій надійності та високій продуктивності.
4. У цьому дослідженні було проаналізовано TLS-сертифікати в локальних мережах шляхом оцінки самопідписаних сертифікатів та сертифікатів Let's

Encrypt. Були представлені рекомендації щодо конфігурації веб-сервера Apache для досягнення оптимального захисту внутрішніх веб-ресурсів.

5. Набір практичних рекомендацій окреслює кроки для встановлення безпечних VPN-з'єднань в локальних мережах організації за допомогою протоколів IPSec та OpenVPN. Документ містить стандартну конфігурацію мережі з покроковими інструкціями щодо налаштування всіх необхідних параметрів, методів шифрування та методів автентифікації.
6. Впроваджені рекомендації підвищують рівень інформаційної безпеки, але не забезпечують повного захисту від загроз безпеки. Ці рекомендації дозволяють фахівцям з інформаційної безпеки розробляти ефективні заходи реагування на потенційні кіберінциденти, одночасно зменшуючи ймовірність витоку конфіденційних даних.
7. Впровадження регулярних аудитів криптографічних засобів захисту слід поєднувати з моніторингом актуальності алгоритмів і протоколів та оновленнями міжнародних стандартів (ISO, NIST). Необхідно регулярно проводити навчання та підвищення кваліфікації персоналу, який працює з криптографічними засобами захисту.

Рекомендації, наведені в цьому документі, призначені для фахівців з інформаційної безпеки, системних адміністраторів та керівників IT-відділів для побудови безпечних сучасних систем інформаційної безпеки в різних локальних мережах організацій.

ПЕРЕЛІК ПОСИЛАНЬ

1. Starlings, W. (2017), Cryptography and Network Security: Principles and Practice. URL: (дата звернення: 21.03.2025).

2. Menezes A.J., van Oorschot P.C., Vanstone S.A. (1996), Handbook of Applied Cryptography. URL: https://cacr.uwaterloo.ca/hac/ (дата звернення: 21.03.2025).

3. Singh S. (2000), The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography. URL: https://www.simonandschuster.com/books/The-Code-Book/Simon-Singh/9780385495325 (дата звернення: 21.03.2025).

4. Rivest R.L., Shamir A., Adleman L. (1978), A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.

URL: https://www.gartner.com/reviews/market/endpoint-detection-and-response-solutions (дата звернення: 21.03.2025).

5 Williams, E. (2020), Cryptography 101: Symmetric Encryption**.** URL: https://medium.com/@emilywilliams_43022/cryptography-101-symmetric-encryption-444aac6bb7a3 (дата звернення: 26.03.2025).

6. Gupta, N., Symmetric vs. Asymmetric Encryption – What are differences? URL: https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences (дата звернення: 26.03.2025).

7. Shreyansh, S. (2021), Hashing Algorithms**.**

URL: https://medium.com/nybles/hashing-algorithms-d10171ca2e89 (дата звернення: 26.03.2025).

8. What is a LAN. URL: https://www.cisco.com/c/en/us/products/switches/what-is-a-lan-local-area-network.html (дата звернення: 26.03.2025).

9. Top 10 Security Risks. URL: https://owasp.org/www-project-top-ten/ (дата звернення: (дата звернення: 29.03.2025).

10. Design Principles of Security in Distributed System. URL: https://www.geeksforgeeks.org/design-principles-of-security-in-distributed-system/ (дата звернення: 29.03.2025).

11. What is 'defense in depth'?

URL: https://www.cloudflare.com/learning/security/glossary/what-is-defense-in-depth/ (дата звернення: 29.03.2025).

12. Local area network. URL: https://en.wikipedia.org/wiki/Local_area_network (дата звернення: 29.03.2025).

13. Gamma, M. (2024), What is Zero Trust and why you need it. URL: https://www.softwareone.com/en-gb/blog/articles/2021/06/17/zero-trust-security (дата звернення: 02.04.2025).

14. NIST SP 800-77 Rev. 1 – Guide to IPsec VPNs.

URL: https://csrc.nist.gov/publications/detail/sp/800-77/rev-1/final (дата звернення: 02.04.2025).

15. RFC 4302 – IP Authentication Header (AH).

URL: https://datatracker.ietf.org/doc/html/rfc4302 (дата звернення: 02.04.2025).

16. RFC 4303 – IP Encapsulating Security Payload (ESP). URL: https://datatracker.ietf.org/doc/html/rfc4303 (дата звернення: 02.04.2025).

17. IPsec (Internet Protocol Security).

URL: https://networklessons.com/security/ipsec-internet-protocol-security (дата звернення: 04.04.2025).

18. OpenVPN Docs. URL: https://openvpn.net/as-docs/general.html (дата звернення: 04.04.2025).

19. WireGuard. URL: https://www.wireguard.com/ (дата звернення: 04.04.2025).

20. RFC 6101 – The Secure Sockets Layer (SSL) Protocol Version 3.0. URL: https://datatracker.ietf.org/doc/html/rfc6101 (дата звернення: 04.04.2025).

21. RFC 5246 – The Transport Layer Security (TLS) Protocol Version 1.2. URL: https://datatracker.ietf.org/doc/html/rfc5246 (дата звернення: 07.04.2025).

22. RFC 8446 – The Transport Layer Security (TLS) Protocol Version 1.3. URL: https://datatracker.ietf.org/doc/html/rfc8446 (дата звернення: 07.04.2025).

23. Gupta, N., TLS 1.3 Protocol Released – Move Ahead to Advanced Security and Privacy. URL: https://www.ssl2buy.com/wiki/tls-1-3-protocol-released-move-ahead-to-advanced-security-and-privacy (дата звернення: 10.04.2025).

24. Cloudflare: Understanding TLS 1.3. URL: https://blog.cloudflare.com/rfc-8446-aka-tls-1-3/ (дата звернення: 10.04.2025).

25. RFC 5280 – X.509 PKI Certificate and CRL Profile. URL: https://datatracker.ietf.org/doc/html/rfc5280 (дата звернення: 10.04.2025).

26. OWASP: TLS Cheat Sheet. URL: https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html (дата звернення: 10.04.2025).

27. Cloudflare: Why TLS 1.3 is Faster and More Secure. URL: https://blog.cloudflare.com/rfc-8446-aka-tls-1-3/ (дата звернення: 11.04.2025).

28. FIPS PUB 197: Specification for the AES. URL: https://csrc.nist.gov/publications/detail/fips/197/final (дата звернення: 11.04.2025).

29. PKCS #1: RSA Cryptography Standard. URL: https://datatracker.ietf.org/doc/html/rfc8017 (дата звернення: 12.04.2025).

30. SHAttered attack on SHA-1 – Google & CWI. URL: https://shattered.io (дата звернення: 20.04.2025).

31. NIST FIPS PUB 180-4: SHA-2 Standard. URL: https://csrc.nist.gov/publications/detail/fips/180/4/final (дата звернення: 24.04.2025).

32. NIST FIPS PUB 202: SHA-3 Standard. URL:

https://csrc.nist.gov/publications/detail/fips/202/final (дата звернення: 25.04.2025).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)