МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет «Чернігівська політехніка»

Навчально-науковий інститут електроних та інформаційних технологій

Кафедра кібербезпеки та математичного моделювання

**ЗВІТ**

про виконання лабораторної роботи

з дисципліни «Основи криптографічного захисту інформації»

Виконав: здобувач 3 курсу, групи КБ-221 Регент Анастасія

Перевірив: Шелест Михайло Євгенович

НУ «Чернігівська політехніка» 2025

Лабораторна робота № 10

**Прикладна криптографія: OpenSSH, Signal, WireGuard**

**Мета роботи:** зробити невеличкі практично орієнтовані завдання, які дозволять закріпити теоретичний матеріал та попрацювати з реальними інструментами OpenSSH, Signal, WireGuard.

**Завдання № 1: Згенерувати ключову пару OpenSSH та підписати файл**

Згенерувати пару ключів SSH. Створіть новий SSH ключ (ключову пару секретного і відкритого ключа):

ssh-keygen -t ed25519 -C [{user@example.com}](mailto:%7buser@example.com%7d)

Можна використати команду ssh-keygen з різними алгоритмами (RSA, ed25519), додати коментар, захистити паролем.

Налаштувати автентифікацію за допомогою ключів. Підключитись до віртуального сервера або локального SSH-серверу без пароля.

Підписати та верифікувати файл з ssh-keygen -Y

Підписати текстовий файл.

Використати allowed\_signers для перевірки підпису.

**Формат виконання:**

Згенерувати ключову пару: ssh-keygen -t ed25519 -C [student@example.com](mailto:student@example.com). Отримав приватний ключ у ~/.ssh/id\_ed25519 та публічний у ~/.ssh/id\_ed25519.pub.

Створити текстовий файл report.txt і підписав його:

ssh-keygen -Y sign -f ~/.ssh/id\_ed25519 -n file report.txt

Згенерувався файл report.txt.sig.

Створити файл allowed\_signers, куди додав свій публічний ключ:

student@example.com namespaces="file" ssh-ed25519 AAAAC3...== student@example.com

Перевірити підпис:

ssh-keygen -Y verify -f allowed\_signers -I student@example.com -n file -s report.txt.sig < report.txt

Перевірка завершилася успішно — підпис дійсний.

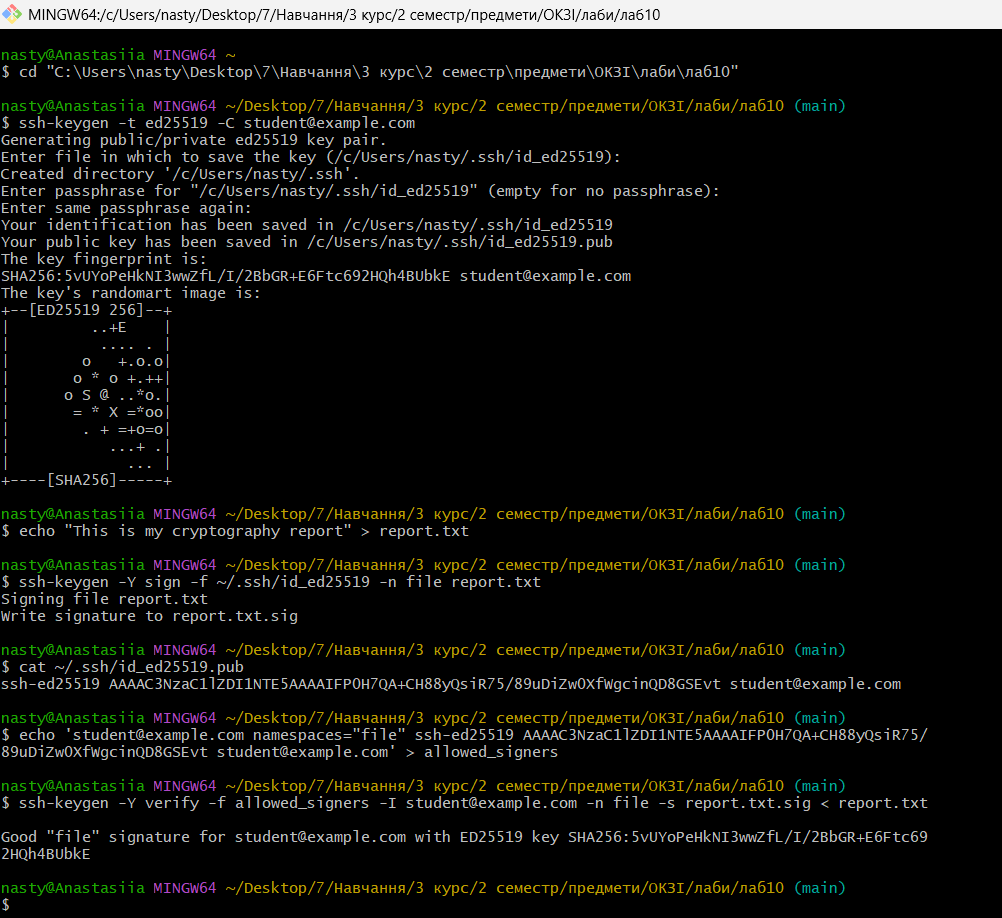


Рисунок 1 – «Процес в Git Bash»

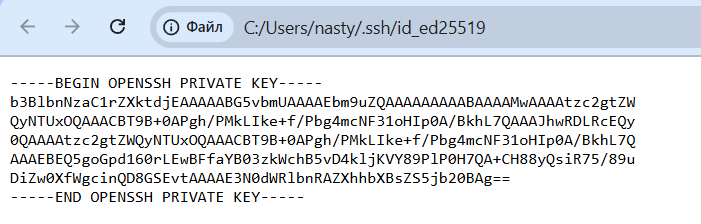


Рисунок 2 – Приватний ключ

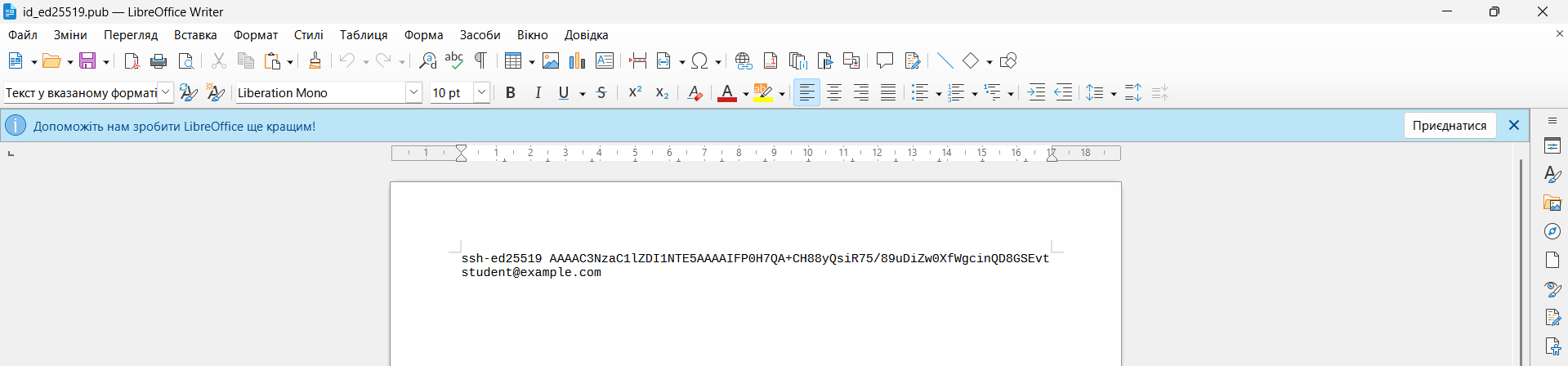


Рисунок 3 – Публічний ключ

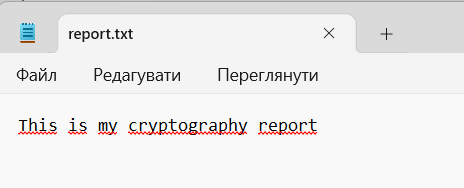


Рисунок 4 – Файл з текстом для підпису

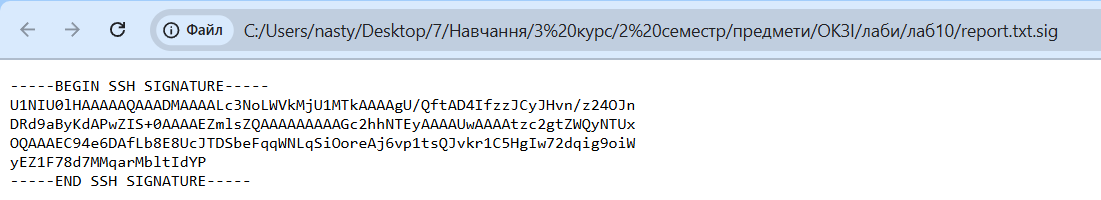


Рисунок 5 – Підпис файлу

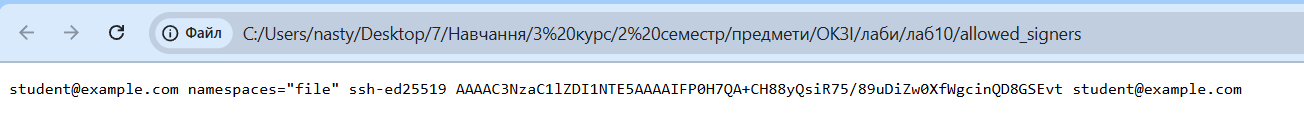


Рисунок 6 – Перевірка підпису

**Завдання № 2: Сценарій наскрізного зашифрованого обміну повідомленнями (Signal)**

Проаналізувати X3DH-протокол (на прикладі)  
Невелике завдання з розбором обміну ключами між Алісою та Бобом (у вигляді діаграми або JSON).

Знайти або змоделювати реалізацію Double Ratchet

Візуалізувати KDF-ланцюг (можна у Python).

Обґрунтувати forward secrecy.

Знайти бібліотеку libsignal на GitHub та розібрати її структуру  
Наприклад: які частини відповідають за X3DH, які — за Double Ratchet.

**Формат виконання:**

Описати алгоритм роботи системи, у якій двоє користувачів — Alice та Bob — обмінюються повідомленнями через протокол Signal.

При першому з'єднанні відбувається X3DH-обмін: сторони генерують довготривалі (static) та короткотривалі (ephemeral) ключі, формуючи спільний секрет.

**Після встановлення спільного секрету запускається Double Ratchet:**

Diffie-Hellman Ratchet — оновлення root key.

Symmetric-key Ratchet — кожне повідомлення шифрується унікальним ключем.

При кожному повідомленні:

Змінюється DH-ключ (обмін новими відкритими ключами).

Ключі не повторюються — забезпечується пряма секретність.

Таким чином, навіть при компрометації поточного стану, попередні повідомлення залишаються захищеними.

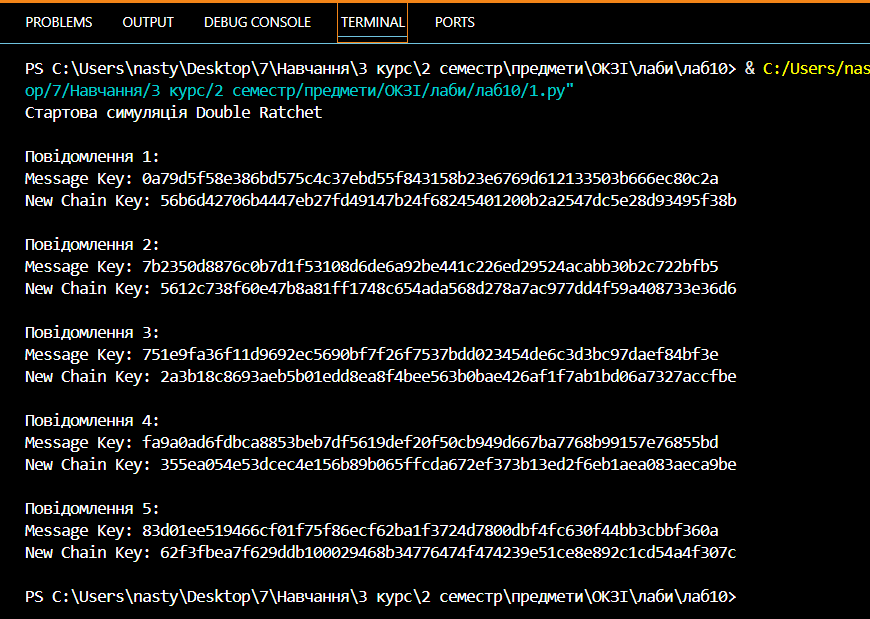


Рисунок 7 – Результат в терміналі

**Контрольні питання**

**SSH**

**Управління ключами**

1. Які типи ключів використовуються у SSH?

RSA, ECDSA, Ed25519, DSA (застарілий).

2. Як відбувається генерація SSH-ключів?

ssh-keygen -t ed25519 -C "your\_email@example.com"

3. Що таке "authorized\_keys" і яку функцію він виконує?

Це файл на сервері (~/.ssh/authorized\_keys), що містить публічні ключі користувачів, яким дозволено підключення. При з'єднанні SSH-клієнт використовує свій приватний ключ для автентифікації.

**Криптографія SSH-протоколу**

1. Які криптографічні алгоритми використовуються у SSH?

1. Обмін ключами: ECDH, Curve25519, Diffie-Hellman;

2. Шифрування: AES, ChaCha20;

3. Хешування: SHA-2, HMAC-SHA256;

4. Підписи: RSA, Ed25519.

2. Як працює обмін ключами за алгоритмом Diffie-Hellman у контексті SSH?

Клієнт і сервер обмінюються частинами даних, на основі яких обчислюється спільний секретний ключ, не передаючи його напряму.

3. Як забезпечується цілісність даних під час сеансу SSH?

Використовується HMAC (Hash-Based Message Authentication Code), що дозволяє виявити зміну або пошкодження даних.

**Практичне застосування**

1. Як налаштувати аутентифікацію за допомогою SSH-ключів на сервері?

1. Згенерувати ключі;

2. Передати публічний ключ на сервер;

3. Увійти без пароля.

2. Як можна обмежити доступ до SSH-сервера для підвищення безпеки?

1. Вимкнення root-доступу (PermitRootLogin no);

2. Дозвіл тільки для певних користувачів (AllowUsers);

3. Зміна порту (наприклад, з 22 на 2222);

4. Використання файрволів та Fail2ban.

**Signal Protocol**

**Властивості безпеки**

1. Які основні властивості безпеки забезпечує Signal Protocol?

1. End-to-End шифрування;

2. Forward Secrecy (пряма секретність);

3. Post-Compromise Security;

4. Аутентифікація користувачів.

2. Як Signal захищає повідомлення від атак посередника?

Кожен користувач перевіряє "безпековий код" (fingerprint) іншого. Всі ключі автентифіковані.

**Криптографія протоколу Signal**

1. Як використовуються короткострокові та довгострокові ключі у протоколі Signal?

1. Довгострокові: Identity Key (Ed25519);

2. Середньострокові: Signed PreKey;

3. Одноразові: One-time PreKeys.

2. Що таке PreKey, і яку роль він відіграє у Signal?

Це наперед згенеровані ключі, які дозволяють встановити захищене з'єднання, навіть якщо одержувач офлайн.

**Double Ratchet алгоритм**

1. Що таке Double Ratchet алгоритм, і як він забезпечує пряму секретність?

Механізм оновлення ключів після кожного повідомлення.

**Забезпечує:**

1. Forward Secrecy;

2. Post-Compromise Security.

2. Як відбувається синхронізація ключів у Double Ratchet?

Diffie-Hellman Ratchet — оновлення ключа після кожного обміну DH.

Symmetric-key Ratchet — оновлення симетричного ключа після кожного повідомлення.

**WireGuard VPN**

**Криптографічні протоколи**

1. Які основні криптографічні алгоритми використовує WireGuard?

1. Шифрування: ChaCha20-Poly1305;

2. Обмін ключами: Curve25519;

3. Хешування: BLAKE2s;

4. MAC: Poly1305;

5. KDF: HKDF.

2. Як забезпечується автентифікація та шифрування трафіку в WireGuard?

Використовується криптографія на основі публічних ключів. Трафік між пірів шифрується і автентифікується з використанням попередньо обмінених ключів.

**Приклад конфігурації**

1. Які ключові кроки необхідно виконати для налаштування WireGuard VPN?

1. Генерація ключів;

2. Створення конфігурації;

3. Запуск.

2. Як перевірити працездатність конфігурації WireGuard після налаштування?

1. wg — показує статус інтерфейсу;

2. ping — перевірка зв’язку;

3. ifconfig або ip a — перевірка мережевого інтерфейсу.