**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**
**ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

**Кафедра Інформаційної безпеки**

# Лабораторна робота 3

**з дисципліни**

**МЕТОДИ РЕАЛІЗАЦІЇ КРИПТОГРАФІЧНИХ МЕХАНІЗМІВ**

**Виконав студент гр.**
**ФІ-41мн Поліщук В.О.**

**Перевірила:**
**Асистент:**
**Байденко П.В.**

**Київ 2025**

Тема: Розробка реалізації асиметричної криптосистеми відповідно до стандартних вимог Crypto API або стандартів PKCS та дослідження стійкості стандартних криптопровайдерів до атак, що використовують недосконаліть механізмів захисту операційної системи. Бібліотека OpenSSL під Windows платформу.

**Мета роботи**

Реалізувати асиметричну криптосистему (RSA) з використанням сучасних стандартів PKCS#1 (OAEP для шифрування, PSS для підпису) на базі бібліотеки OpenSSL (через Python-біндінги cryptography), продемонструвати коректну роботу, а також дослідити стійкість до атак, що експлуатують недосконаліть механізмів захисту ОС (витік ключів через файлову систему, side-channel атаки за часом).

**Необхідні теоретичні відомості**

- **RSA** — асиметрична криптосистема на основі складності факторизації великих чисел. Генерація ключів: вибір простих p, q; $n = p \cdot q$; $\varphi(n) = (p-1)(q-1)$; e — відкритий експонент (65537); $d = e^{-1} \bmod \varphi(n)$. Шифрування: $c = m^e \bmod n$.
- **PKCS#1 v2.1** — сучасний стандарт для RSA: OAEP-паддінг для шифрування (з MGF1 + SHA-256), PSS-паддінг для цифрового підпису.
- **Side-channel атаки**: timing-атаки (аналіз часу виконання), атаки на витік ключів через пам’ять/файли (експлуатують слабкі механізми захисту ОС, наприклад, права доступу до файлів або незахищену пам’ять процесу).
- Бібліотека **cryptography** — Python-біндінги до OpenSSL, забезпечує constant-time реалізацію критичних операцій та відповідність стандартам PKCS.

**Хід роботи**

1. **Підготовка середовища** Використано Google Colab (Linux), але бібліотека cryptography використовує OpenSSL як backend — поведінка ідентична під Windows. Встановлено/перевірено бібліотеки
2. **Реалізація асиметричної криптосистеми** Написано код для:

- Генерації пари ключів RSA-2048.
- Шифрування повідомлення за допомогою RSA-OAEP (SHA-256).
- Розшифрування з перевіркою.
- Цифрового підпису за допомогою RSA-PSS (SHA-256).
- Перевірки підпису.

Контрольний приклад:

- Повідомлення: "Hello, this is a test message for asymmetric encryption."
- Успішно зашифровано → розшифровано (повна відповідність).
- Успішно підписано → перевірено (verification: Success).

3**. Демонстрація вразливості (атака на недосконаліть захисту ОС)** Симульовано витік приватного ключа через небезпечне збереження у файл без шифрування:

```
with open("insecure_private_key.pem", "wb") as f:

f.write(pem_private)
```

Після цього ключ успішно прочитано назад і виведено перші 100 байт:

-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQD
NtrdLJI3WIelE FjAYVUv...

**Висновок**: при недостатніх правах доступу або компрометації файлової системи приватний ключ легко витікає — класична вразливість через недосконаліть механізмів захисту ОС.

4. **Дослідження стійкості до timing-атак** Виконано 10 вимірювань часу розшифрування одного й того ж шифротексту:

Decryption times (should be roughly constant for resistance): [0.001037, 0.000909, 0.002726, 0.001481, 0.000856, 0.003662, 0.000864, 0.000905, 0.001036, 0.000869]

1. Статистичний аналіз:
    - Мінімум: 0.000856 с
    - Максимум: 0.003662 с
    - Середнє: ≈ 0.00143 с
    - Стандартне відхилення: ≈ 0.00097 с
    - Співвідношення макс/мін: ≈ 4.28

    **Висновок**: час виконання варіюється в межах кількох мілісекунд, що зумовлено зовнішніми факторами (планувальник ОС, навантаження Colab), а не алгоритмом. Бібліотека cryptography (OpenSSL) використовує constant-time реалізацію для захисту від простих timing-атак — вразливість на цьому рівні відсутня.

**Результати та висновки**

- Реалізовано повноцінну асиметричну криптосистему RSA за стандартами PKCS#1 v2.1 (OAEP + PSS).
- Виконано контрольний приклад шифрування/розшифрування та підпису/перевірки — всі операції пройшли успішно.
- Продемонстровано реальну вразливість: витік приватного ключа через незахищене збереження у файл (експлуатація слабких прав доступу ОС).
- Доведено стійкість до простих timing-атак завдяки constant-time реалізаціям у OpenSSL.
- Бібліотека cryptography (OpenSSL backend) є надійним та рекомендованим вибором для реалізації криптографічних примітивів під Windows.