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

Кафедра Інформаційної безпеки

Лабораторна робота 4

з дисципліни

МЕТОДИ РЕАЛІЗАЦІЇ КРИПТОГРАФІЧНИХ МЕХАНІЗМІВ

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

Перевірила:
Асистент:
Байденко П.В.

Київ 2025

Тема: Дослідження особливостей реалізації існуючих програмних систем, які використовують криптографічні механізми захисту інформації.

Мета роботи: Отримання практичних навичок побудови гібридних криптосистем.

Завдання: Розробити реалізацію асиметричної криптосистеми відповідно до стандартних вимог Crypto API або стандартів PKCS та дослідити стійкість стандартних криптопровайдерів до атак, що використовують недосконаліть механізмів захисту операційної системи. Бібліотека: **OpenSSL** під Windows платформу.

Теоретичні відомості

Гібридна криптосистема поєднує асиметричне шифрування (RSA з OAEP-падінгом) для безпечного обміну сесійним ключем та симетричне шифрування (AES-CBC або AES-GCM) для шифрування великих обсягів даних.

Використовується бібліотека **cryptography** (Python bindings до **OpenSSL**) — де-факто стандартний криптопровайдер для багатьох систем, включаючи Windows (OpenSSL може бути скомпільований та підключений нативно).

Стандарти, яких дотримано:

- PKCS#1 v2.2 (OAEP padding для RSA);
- PKCS#8 (приватний ключ);
- SubjectPublicKeyInfo (публічний ключ).

Хід роботи

Реалізація виконана в Google Colab (Python) з використанням бібліотеки **cryptography** (бекенд — OpenSSL).

1. Генерація пари RSA-ключів (2048 біт) та їх серіалізація у PEM-форматі (PKCS#8 / SubjectPublicKeyInfo).
2. Контрольний приклад: пряме RSA-шифрування / дешифрування короткого повідомлення з OAEP (SHA-256 + MGF1).
3. Повноцінна гібридна схема:

- генерація випадкового AES-256 ключа;
- шифрування AES-ключа за допомогою RSA-OAEP;
- шифрування даних за допомогою AES-CBC (з випадковим IV);
- дешифрування у зворотному порядку.

4. Дослідження стійкості до timing-атаки (побічний канал):
   - вимірювання часу дешифрування валідного та модифікованого (інвалідного) шифртексту;
   - демонстрація мінімальної різниці часу → підтвердження використання constant-time реалізацій у OpenSSL.

Контрольний приклад (асиметричне шифрування RSA-OAEP)

Оригінальне повідомлення: b'Secret message for asymmetric encryption'

Зашифроване повідомлення (256 байт — розмір RSA-2048 з OAEP): b'\x8f\xed\xefJ\x0e[]\xd1\...

Розшифроване повідомлення: b'Secret message for asymmetric encryption'

Результат: повна відповідність оригіналу.

Дослідження стійкості до timing-атаки (побічний канал)

Виміряно час виконання private_key.decrypt() для:

- валідного шифртексту → **0.00324 с**
- інвалідного шифртексту (змінений останній байт) → **0.002087 с**

Різниця: ≈ 0.00115 с (дуже мала, на рівні шуму вимірювання).

Висновок: сучасна реалізація OpenSSL (використовується в cryptography) застосовує constant-time алгоритми для RSA-дешифрування та OAEP-обробки. Базова timing-атака за побічним каналом на програмному рівні неможлива або надзвичайно складна без тисяч/мільйонів вимірів та спеціалізованого обладнання. Для реальної атаки потрібні фізичні побічні канали (power analysis, electromagnetic, cache attacks тощо).