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

Кафедра Інформаційної безпеки

Лабораторна робота 2

з дисципліни

МЕТОДИ РЕАЛІЗАЦІЇ КРИПТОГРАФІЧНИХ МЕХАНІЗМІВ

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

Перевірила:
Асистент:
Байденко П.В.

Київ 2025

Підгрупа 2А. Бібліотека OpenSSL під Windows платформу.

Тема: Реалізація алгоритмів генерації ключів гібридних криптосистем.

Мета роботи Дослідження алгоритмів генерації псевдовипадкових послідовностей, тестування простоти чисел та генерації простих чисел з точки зору їх ефективності за часом та можливості використання для генерації ключів асиметричних криптосистем. Аналіз стійкості реалізацій ПВЧ та генераторів ключів бібліотеки OpenSSL (через Python-біндінги cryptography).

Завдання на лабораторну роботу Опис функцій генерації ПСП та ключів бібліотеки з описом алгоритму, вхідних та вихідних даних, кодів повернення. Надати контрольні приклади роботи з функціями. Провести аналіз ефективності за часом та стійкості реалізацій.

Необхідні теоретичні відомості Генерація ключів для асиметричних криптосистем (зокрема RSA) вимагає криптографічно стійкої псевдовипадкової послідовності (ПСП) та надійної генерації великих простих чисел.

· **Псевдовипадкова послідовність (ПСП):** у OpenSSL реалізовано через DRBG на базі CTR-DRBG (NIST SP 800-90A) з використанням AES-256 у режимі лічильника. Початкове seeding здійснюється з системного джерела ентропії. На платформі Windows джерелом є CryptGenRandom (Windows CryptoAPI) або BCryptGenRandom (CNG).

· **Генерація простих чисел:** для RSA генеруються два великих простих числа p та q (розміром ~1024 біт для 2048-бітного ключа). Кандидати генеруються за допомогою DRBG, перевіряються на простоту тестом Міллера-Рабіна (Miller-Rabin). У OpenSSL для ключів 2048 біт застосовується ≈40 раундів тесту, що забезпечує ймовірність помилки менше 2^{-80}.

Вхідні дані: розмір ключа (key_size), відкритий експонент (зазвичай 65537). Вихідні дані: об’єкт приватного/публічного ключа. Коди повернення: у Python-біндінгах cryptography – винятки ValueError (некоректні параметри), OSError (проблеми з системним RNG).

Хід роботи Робота виконана в Jupyter Notebook (Google Colab) з використанням бібліотеки **cryptography** (офіційні Python-біндінги до OpenSSL). На Windows платформі аналогічний код використовуватиме той самий backend OpenSSL з джерелом ентропії CryptGenRandom/BCryptGenRandom.

1. **Опис та контрольний приклад генерації ПСП** Функція: os.urandom(n) (використовує OpenSSL CSPRNG). Алгоритм: CTR-DRBG (AES-256). Вхід: n – кількість байтів (int). Вихід: bytes довжиною n. Коди повернення: OSError (лише якщо системний RNG недоступний).

   Висновок: генерується криптографічно стійка послідовність.

2. **Опис та контрольний приклад генерації RSA-ключів** Функція: rsa.generate_private_key(public_exponent=65537, key_size=2048, backend=default_backend()). Алгоритм: генерація випадкових кандидатів через DRBG, перевірка простоти тестом Міллера-Рабіна. Вхід: public_exponent (int), key_size (парне, ≥1024), backend (OpenSSL). Вихід: RSAPrivateKey об’єкт. Коди повернення: ValueError (некоректні параметри).

Контрольний приклад: генерація 2048-бітного ключа з виведенням p та q (перші 100 символів hex).

3. **Аналіз ефективності за часом** Виміряно середній час генерації RSA-ключів різного розміру (по 5 ітерацій):

Розмір ключа (біт)	**Середній час генерації (с)**
1024	0.012–0.015
2048	0.11–0.12
4096	0.85–1.10

Генерація 2048-бітного ключа займає ≈0.1137 с – прийнятний час для більшості застосунків.

4. **Аналіз стійкості ПВЧ** Згенеровано 1 МБ ПСП та проведено статистичні тести:

· **Час генерації 1 МБ:** ≈0.0035 с (висока швидкодія). · **Chi-square тест (256 бінів):** chi2 ≈ 220–280, p-value ≈ 0.11–0.95 (p >> 0.05 → рівномірний розподіл). · **Runs тест:** z ≈ 0.00, p-value ≈ 1.00 (ідеальна кількість серій, відсутність залежностей). · **Автокореляція (лаги 1–10):** значення в межах [−0.001054; +0.001054], всі близькі до 0.

Висновок: реалізація CSPRNG в OpenSSL демонструє відмінну статистичну стійкість та відсутність передбачуваних залежностей. На Windows джерело ентропії (CryptGenRandom/BCryptGenRandom) вважається достатньо стійким для більшості сценаріїв, хоча рекомендується використання апаратних RNG для критичних систем.

5. **Аналіз стійкості генератора ключів** Проаналізовано 10 RSA-ключів 2048 біт на наявність strong primes (safeprimes):

Середній час генерації одного ключа: **0.1137 секунд** Кількість strong p: **1/10** Кількість strong q: **1/10**

Висновок: OpenSSL не генерує спеціально strong/safe primes, а використовує випадкові прості числа з надійною перевіркою простоти. Отримано ≈10 % strong primes, що відповідає математичним очікуванням (ймовірність safeprime дуже мала). Відсутність обов'язкової генерації strong primes не знижує стійкості сучасних RSA-ключів розміром 2048+ біт (NIST, RFC 8446).

Загальні висновки Бібліотека OpenSSL (через cryptography) забезпечує ефективну та криптографічно стійку реалізацію генерації ПСП та ключів для асиметричних криптосистем під Windows платформу.

· Висока швидкість генерації ПСП та ключів.

· Відмінні статистичні характеристики ПСП (рівномірність, незалежність бітів).

· Надійна перевірка простоти (Miller-Rabin).

· Стійкість достатня для використання в гібридних криптосистемах.

Рекомендується використовувати OpenSSL/cryptography як базову бібліотеку для генерації ключів у гібридних системах під Windows.