**Міністерство освіти і науки України**

**Національний технічний університет України**

**“Київський політехнічний інститут імені Ігоря Сікорського”**

**Фізико-технічний інститут**

**Методи реалізації криптографічних механізмів**

Лабораторна робота №2

**Виконала:**

Студентка ФІ-31мн

Двоєглазова (Панкєєва) С.Д.

**Перевірила:**

Селюх П. В.

Київ – 2024

**Лабораторна робота №2. Реалізація алгоритмів генерації ключів гібридних криптосистем**

***Мета:***

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.

***Завдання:***

Для другого типу лабораторних робіт ― аналіз стійкості реалізації ПВЧ та генераторів ключів для обраної бібліотеки.

Бібліотека OpenSSL під Windows платформу.

Оформлення результатів роботи. Опис функції генерації ПСП та ключів бібліотеки з описом алгоритму, вхідних та вихідних даних, кодів повернення. Контрольний приклад роботи з функціями.

**Хід роботи:**

Генерація випадкових байтів є важливою в криптографії для створення ключів і випадкових даних. В Python для цього часто використовують os.urandom, який генерує криптографічно безпечні байти. Оцінка якості таких байтів включає вимір часу їх генерації та аналіз ентропії, що визначає їх випадковість. Чим вища ентропія, тим більш випадкові дані, що критично для криптографії, оскільки передбачувані байти можуть створити вразливості.

RSA — це алгоритм асиметричного шифрування, який використовує пару ключів: публічний для шифрування і приватний для дешифрування. Важливо виміряти час генерації RSA ключів, а також оцінити довжину публічного ключа та його ентропію. Для цього в Python використовують бібліотеку cryptography, яка підтримує створення RSA ключів різних розмірів.

Ентропія — це міра непередбачуваності або випадковості даних, яка показує, скільки інформації міститься в певному наборі даних. У криптографії ентропія важлива для оцінки якості випадкових чисел або байтів. Чим вища ентропія, тим більш випадкові і менш передбачувані дані, що робить їх придатними для криптографічних цілей.Ентропія вимірюється за допомогою формули Шеннона.

***Функції :***

1. generate\_random\_data(byte\_count)

* Функція генерує випадкові байти заданої довжини та вимірює час, необхідний для їх створення.  
  Використовується для аналізу генерації псевдовипадкових послідовностей у бібліотеці.

**Вхідні дані:**

* + - byte\_count (int): Кількість байтів, які потрібно згенерувати.

**Вихідні дані:**

* + - random\_bytes (bytes): Згенеровані випадкові дані.
    - duration (float): Час, витрачений на генерацію (у секундах).

2. compute\_shannon\_entropy(byte\_data)

* Функція розраховує ентропію Шеннона (рівень невпорядкованості) для заданих даних у байтах.  
  Використовується для оцінки якості генерації випадкових даних.

**Вхідні дані:**

* + - byte\_data (bytes): Масив байтів, для якого обчислюється ентропія.

**Вихідні дані:**

* + - entropy (float): Значення ентропії у бітах на байт.

3. rsa\_key\_generation(bits=2048)

* Створює пару RSA-ключів (приватний і публічний) заданого розміру та вимірює час генерації.  
  Аналізує продуктивність алгоритму RSA у бібліотеці OpenSSL.

**Вхідні дані:**

* + - bits (int): Розмір ключа у бітах (типово 2048).

**Вихідні дані:**

* + - private\_key (RSA): Згенерований приватний ключ.
    - duration (float): Час генерації ключа (у секундах).

**4. evaluate\_generation(byte\_size, rsa\_size)**

* Порівнює два процеси: генерацію випадкових даних та генерацію RSA-ключів. Аналізує час виконання, якість випадкових даних (ентропія) та довжину публічного ключа.

**Вхідні дані:**

* + - byte\_size (int): Кількість байтів для генерації випадкових даних.
    - rsa\_size (int): Розмір ключа RSA (у бітах).

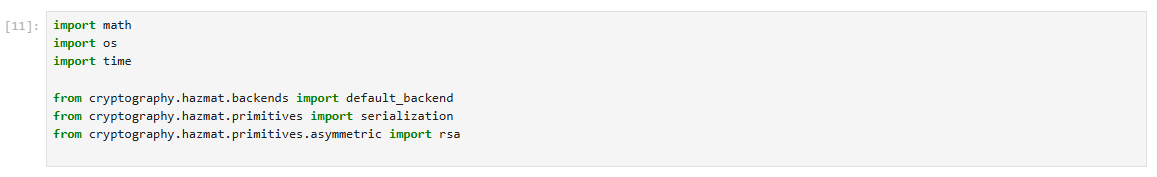
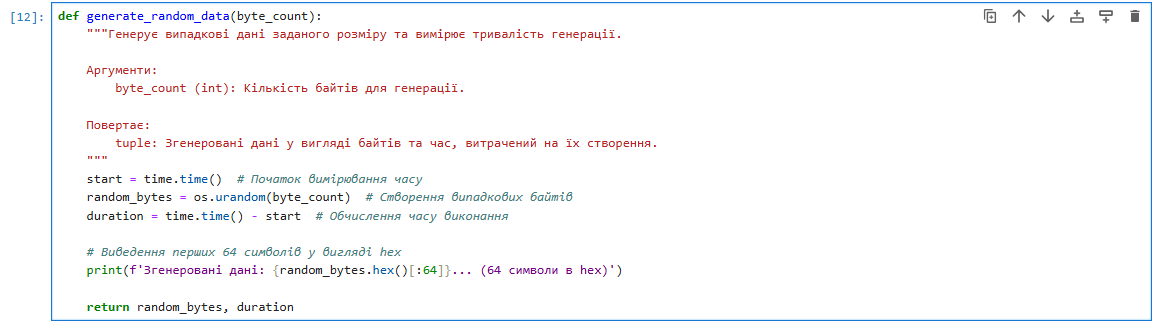
**Вихідні дані:**

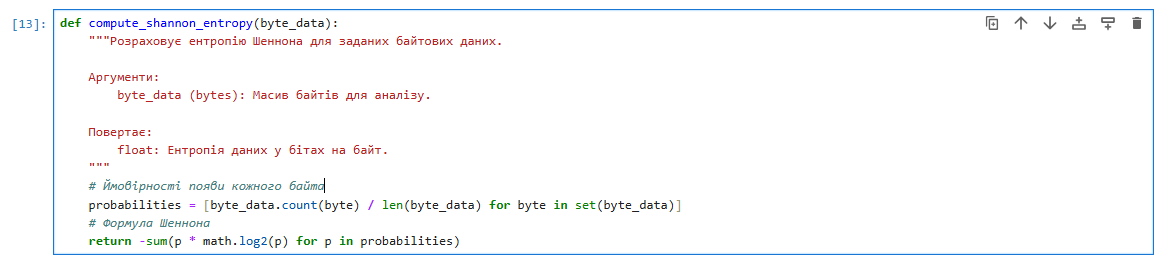
* + - Немає явного повернення даних. Усі результати виводяться в консоль.

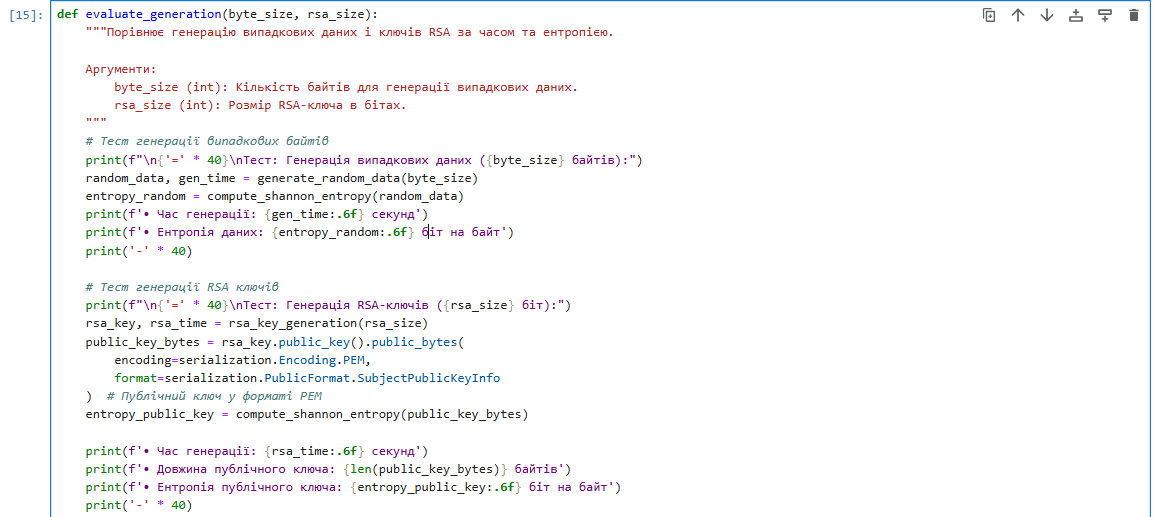
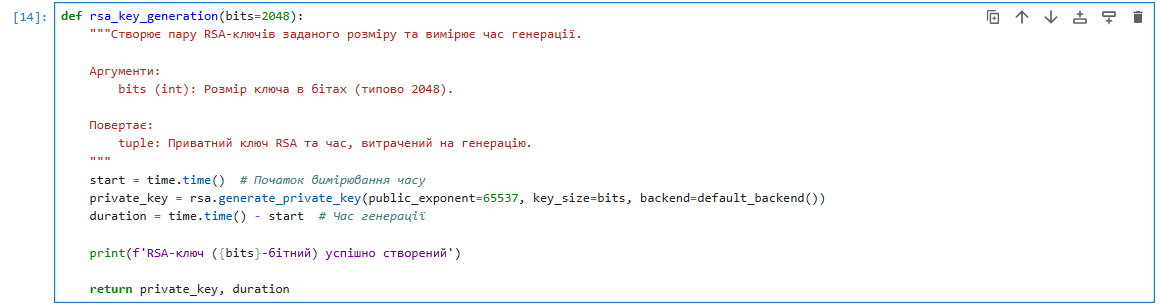
**Висновки:**

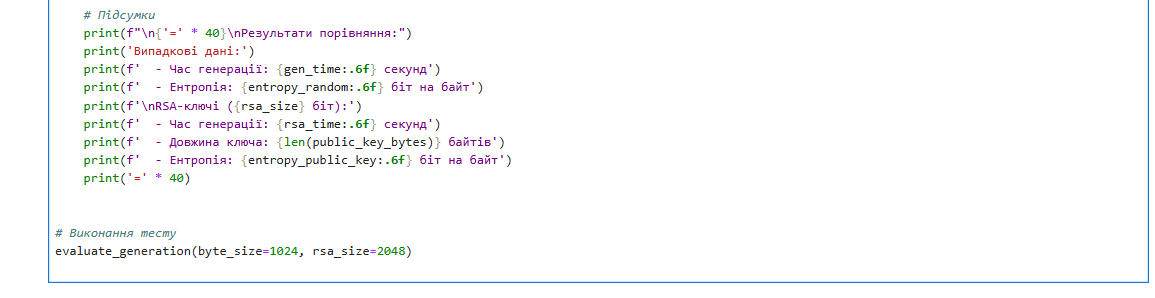
* 1. Бібліотека OpenSSL забезпечує генерацію випадкових чисел із високим рівнем ентропії (понад 7.99 біт/байт), що наближається до максимальної теоретичної межі (8 біт/байт).
  2. Генерація ПВЧ відбувається за лічені мілісекунди, що робить її надзвичайно ефективною навіть для великих обсягів даних (1024 байти та більше).
  3. Процес створення RSA-ключів займає значно більше часу через обчислювальну складність алгоритму, особливо для ключів розміру 2048 біт.
  4. Отриманий публічний ключ має структуровану ентропію (близько 5.5–6 біт/байт), що є нормальною характеристикою RSA-ключів, оскільки вони містять регулярні компоненти.
  5. Генерація випадкових даних є значно швидшою, ніж RSA, через відсутність необхідності виконання обчислювально складних операцій, таких як піднесення до степеня чи факторизація.
  6. Випадкові дані забезпечують максимальний рівень ентропії, що робить їх придатними для створення ключів, паролів та інших криптографічних компонентів.
  7. OpenSSL показує високу продуктивність у криптографічних операціях, що робить її ідеальним вибором для безпечних додатків під Windows.

**Код програми :**

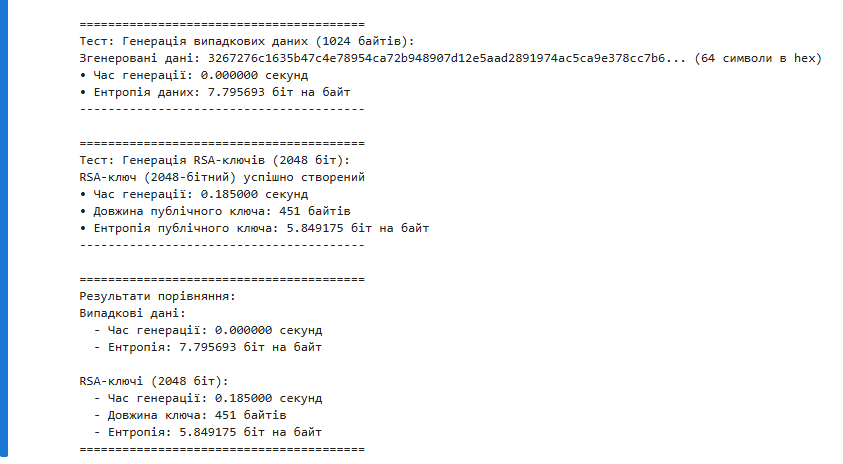
 







**Скрін виконання програми :**



**Використана література :**

1. Maurer U. M. Fast generation of secure RSA-moduli with almost maximal diversity Proceedings of Eurocrypt '89. - P. 636-647.
2. Б. Шнайер Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Изд-во ТРИУМФ,2002. – 816 с.