Міністерство освіти і науки України Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Фізико-технічний інститут

ЛАБОРАТОРНА РОБОТА №4

«Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методом генерації параметрів для асиметричних криптосистем»

Виконали: Студенти групи ФБ-04 Музичка-Скрипка Олександра

Кузьмін Гліб Перевірив: Чорний О.

Мета роботи:

Ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практичне ознайомлення з системою захисту інформації на основі криптосхеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчення протоколу розсилання ключів.

Порядок виконання роботи:

1. Написати функцію пошуку випадкового простого числа з заданого інтервалу або заданої довжини, використовуючи датчик випадкових чисел та тести перевірки на простоту. В якості датчика випадкових чисел використовуйте вбудований генератор псевдовипадкових чисел вашої мови програмування. В якості тесту перевірки на простоту рекомендовано використовувати тест Міллера-Рабіна із попередніми пробними діленнями. Тести необхідно реалізовувати власноруч, використання готових реалізацій тестів не дозволяється.

2. За допомогою цієї функції згенерувати дві пари простих чисел p, q і p1, q1 довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq ≤ p1q1; p і q – прості числа для побудови ключів абонента А, p1 і q1 – абонента B.

3. Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ (d, p, q) та відкритий ключ (n, e) . За допомогою цієї функції побудувати схеми RSA для абонентів А і B – тобто, створити та зберегти для подальшого використання відкриті ключі (e, n), (e1, n1) та секретні d і d1.

4. Написати програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів А і B. Кожна з операцій (шифрування, розшифрування, створення цифрового підпису, перевірка цифрового підпису) повинна бути реалізована окремою процедурою, на вхід до якої повинні подаватись лише ті ключові дані, які необхідні для її виконання. За допомогою датчика випадкових чисел вибрати відкрите повідомлення M і знайти криптограму для абонентів А и B, перевірити правильність розшифрування. Скласти для А і B повідомлення з цифровим підписом і перевірити його.

5. За допомогою раніше написаних на попередніх етапах програм організувати роботу протоколу конфіденційного розсилання ключів з підтвердженням справжності по відкритому каналу за допомогою алгоритму RSA. Протоколи роботи кожного учасника (відправника та приймаючого) повинні бути реалізовані у вигляді окремих процедур, на вхід до яких повинні подаватись лише ті ключові дані, які необхідні для виконання. Перевірити роботу програм для випадково обраного ключа 0 < k < n.

Хід роботи:

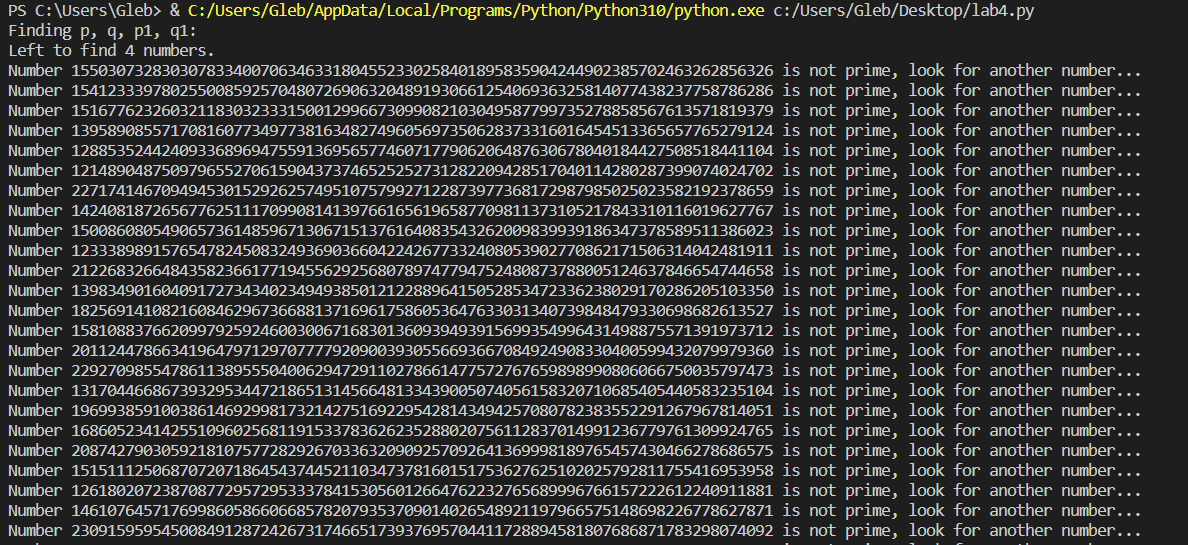
Для виконання першого та другого завдання наша команда брала числа з інтервалу від 2256 - 1 до 2257, після чого перевіряла їх на простоту за допомогою теста Мілера-Рабіна. Для того, щоб ще більше зменшити вірогідність помилки, було вирішено робити перевірку кожного числа 60 разів. Виникли деякі труднощі зі швидкодією програми, бо перевіряти кожне число описаним вище тестом було занадто довго, тому до тесту було додане попереднє ділення на 2, 3, 5 та 7, що дозволило значто підвищити швидкість алгоритму. Всього генерується чотири простих числа (під час пошуку код виводить усі числа, що перевіряються, та пише, чи проходить певне число перевірку, та скільки чисел залишилося знайти), які потім сортуються, після чого два найменші назначаються як p та q, а два найбільші як p1 та q1.

Для виконання третього завдання була реалізована функція GenerateKeyPair, що знаходить, зберігає та повертає значення n, n1, e, e1, fi, fi1, d, d1. Значення е та е1 ми вирішили не генерувати кожного разу окремо, а взяти як 216 + 1 (це значення було описане в методичних вказівках).

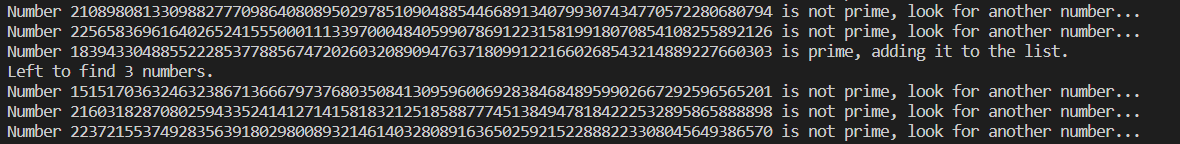
Для виконання четвертого та п'ятого завдань була реалізована функція Horners\_method, яка далі використовуватиметься в алгоритмах зашифровування (Decrypt), дешифровування (Encrypt), створення цифрового підпису (Sign), перевірки цифрового підпису (Verify), конфіденційного розсилання ключів (SendKey) та отримання ключів (ReceiveKey)

Приклад роботи програми:

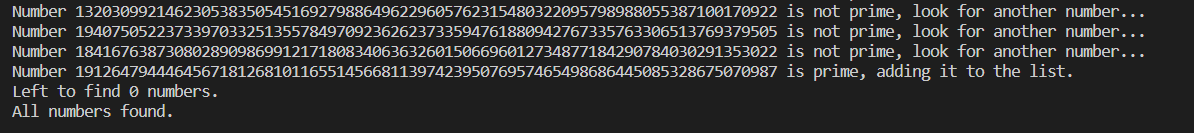
Спочатку програма знаходить значення p, q, p1, q1, для чого за допомогою вбудованого генератору псевдопростих чисел перевіряє числа на простоту. Усі перевірки виводяться у термінал:



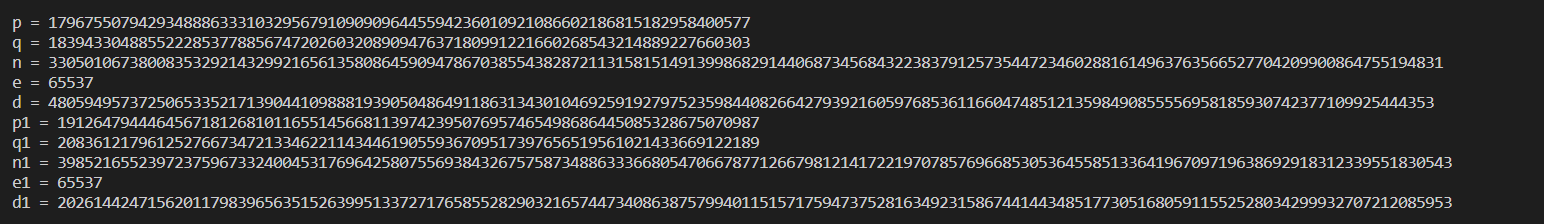
Коли програмі вдається знайти просте число, вона звітує про це, після чого видає кількість простих чисел, які ще потрібно знайти, і продовжує пошук:



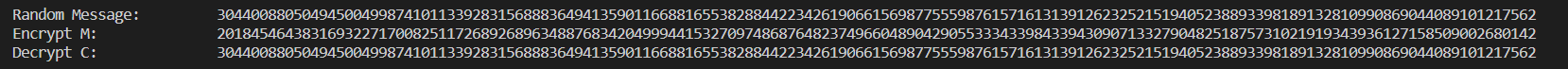
Коли усі чотири числа знайдено, програма сповіщає про це, після чого завершує пошук:



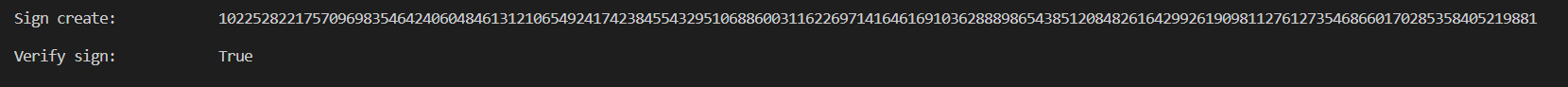
Коли усі прості числа знайдені, вони сортуються, після чого два найменші стають p, q, а два найбільші- p1, q1. Далі за формулами, зазначеними у методичних вказівках, знаходяться значення n, n1, fi, fi1, d, d1. Коли усі значення знайдені, програма виводить їх на екран:



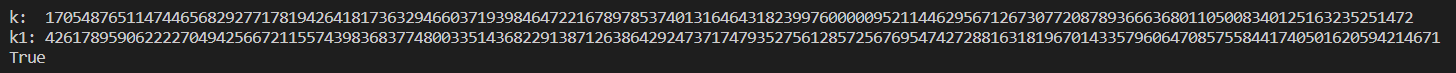
Далі програма генерує випадкове повідомлення М, виводить його, зашифровує, знов виводить, після чого розшифровує та виводить втретє. Як можна побачити, повідомлення до шифрування співпадає з повідомлення після дешифрування, з чого можна судити про коректність роботи алгоритмів шифрування та дешифрування:



Далі програма за допомогою схеми Горнера створює та перевіряє цифровий підпис:



У кінці програма знаходить випадкове число k, з якого формує повідомлення k1 за допомогою функції SendKey, після чого за допомогою ReceiveKey знаходить початкове повідомлення та перевіряє підпис. Якщо усі перевірки пройдені коректно, програма повертає значення True:



Висновки: під час виконання лабораторної роботи наша команда ознайомилася з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA, на практиці ознайомилися з системою захисту інформації на основі криптосхеми RSA, організували з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчили та реалізували протокол розсилання ключів.