**Лабораторна робота №4**

Вивчення криптосистеми RSA та алгоритму електронного

підпису; ознайомлення з методами генерації параметрів для

асиметричних криптосистем

Виконала: Годлевська Аліса ФБ-11

**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**

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, щобpq < p1q1; p і q –прості числа для побудови ключів абонента А, p1 іq1 – 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.

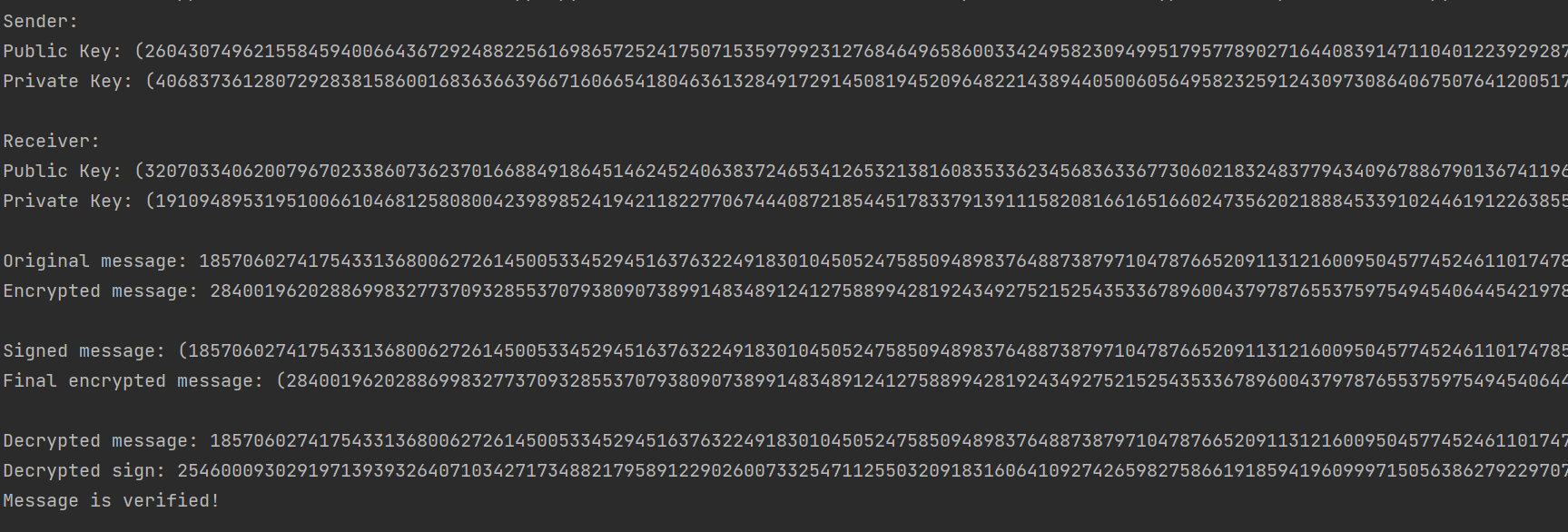
5. За допомогою раніше написаних на попередніх етапах програм організувати роботу протоколу конфіденційного розсилання ключів з підтвердженням справжності по відкритому каналу за допомогою алгоритму RSA. Протоколи роботи кожного учасника (відправника та приймаючого) повинні бути реалізовані у вигляді 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.

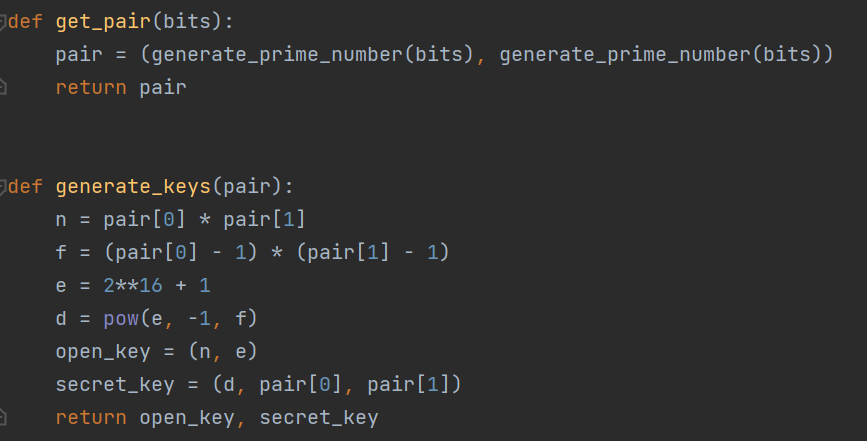
**Хід роботи**

1-2. Числа, що «не проходять» поміщаються у файл rejected\_primes.txt

def generate\_prime\_number(bits):  
 def get\_random\_number(bits):  
 return random.randint(2 \*\* bits, 2 \*\* (bits + 1) - 1)  
  
 def test\_miller\_rabin(p):  
 if p % 2 == 0 or p % 3 == 0 or p % 5 == 0 or p % 7 == 0 or p % 11 == 0:  
 with open("rejected\_primes.txt", "a") as file:  
 file.write(f"{p}\n")  
 return False  
  
 s, d = 0, p - 1  
  
 while d % 2 == 0:  
 d //= 2  
 s += 1  
 assert (p - 1 == d \* (2 \*\* s))  
  
 x = random.randint(2, p - 2)  
  
 if math.gcd(x, p) > 1:  
 with open("rejected\_primes.txt", "a") as file:  
 file.write(f"{p}\n")  
 return False  
  
 if pow(x, d, p) == 1 or pow(x, d, p) == -1:  
 return True  
  
 for \_ in range(1, s - 1):  
 x = (x \* x) % p  
 if x == -1:  
 return True  
 if x == 1:  
 with open("rejected\_primes.txt", "a") as file:  
 file.write(f"{p}\n")  
 return False  
 with open("rejected\_primes.txt", "a") as file:  
 file.write(f"{p}\n")  
 return False  
  
 num = get\_random\_number(bits)  
 while not test\_miller\_rabin(num):  
 num = get\_random\_number(bits)  
  
 return num

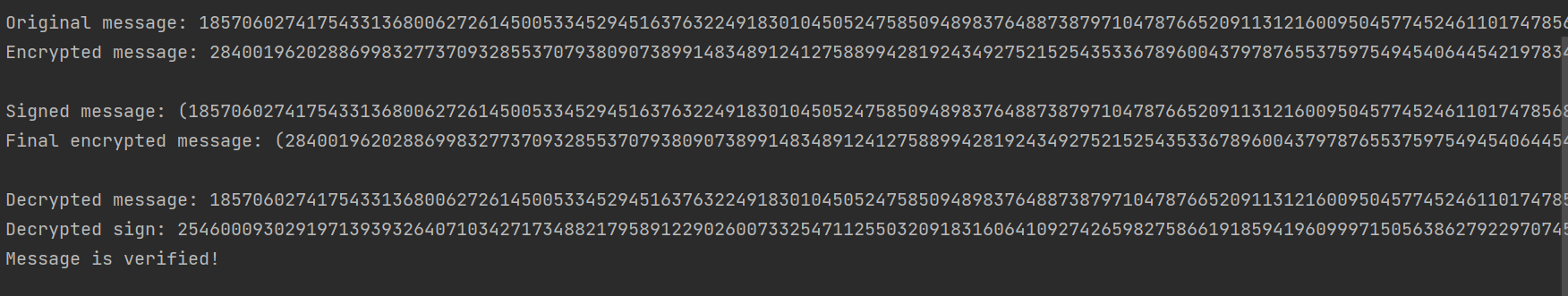
3.

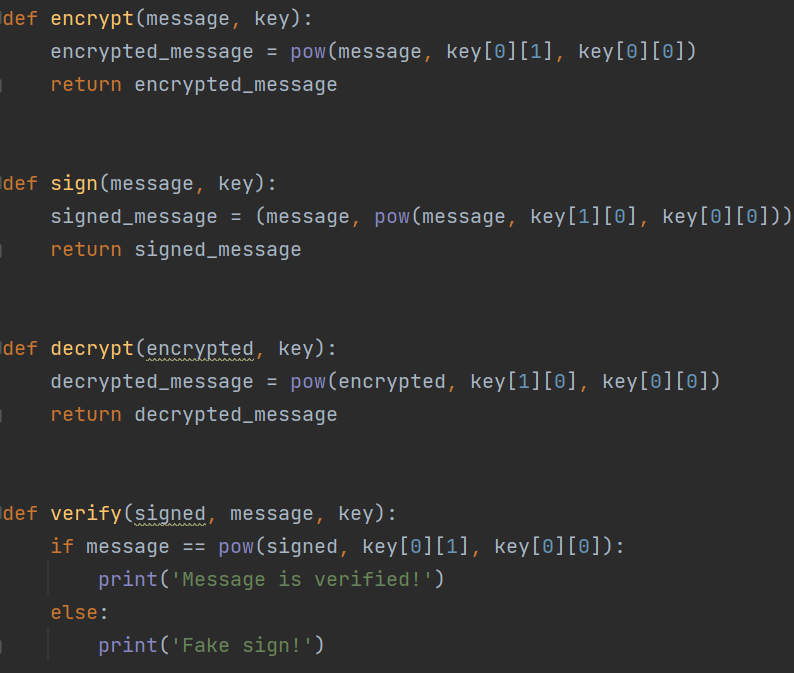




Так було створено та збережено для подальшого використання відкриті ключі (e, n), (e1, n1) та секретні d і d1.

4-5.





**Висновки**: під час виконання комп’ютерного практикуму я ознайомилась із методами генерації ключів для асиметричної криптосистеми RSA; практично використала знання: краще ознайомилась із поняттями засекреченого зв'язку й електронного підпису.