**Міністерство освіти і науки України**

**Національний технічний університет України**

**"Київський політехнічний інститут імені Ігоря Сікорського"**

**Фізико-технічний інститут**

**Криптографія**

**Комп’ютерний практикум №4**

**Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методами генерації параметрів для асиметричних криптосистем**

Варіант 4

Виконали:

Студенти ФБ-05

Береза О. А. Ковбель Д. О.

Київ 2022

**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.

**Постановка задачі:**

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 МіллераРабіна 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 1 1 p , q довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq £ p1q1 ; p і q – прості числа для побудови ключів абонента А, 1 p 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) , (n1, e1 ) та секретні d і d1 .
3. 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.

1. 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. Кожна з наведених операцій повинна бути реалізована у вигляді окремої процедури, інтерфейс якої повинен приймати лише ті дані, які необхідні для її роботи; наприклад, функція Encrypt(), яка шифрує повідомлення для абонента, повинна приймати на вхід повідомлення та відкритий ключ адресата (і тільки його), повертаючи в якості результату шифротекст.

**Хід роботи**

В роботі ми використовували функції з бібліотеки random: randint(), randrange(), аби згенерувати число розміром 256 за завданням. Перевірка на простоту, здійснюється в декілька етапів: ділення на відомі перші 13 простих чисел(якщо випадкове число ділилось на них відповідно воно не було простим); тест Міллера-Рабіна(в якому знайшли розклад нашого числа, а потім перевірка простоти за основою випадково вибраного числа х). Реалізована функція test\_millrab() повертає True/False значення у відповідності чи є число сильно псевдопростим, чи ні.

**Результати виконання:**

Keys for Alice:

Open Keys for Аlice:

e = 1520188411089749086588356197963003471329963876787546165202182583361245891888828975369532472140628660313313672279511612432268571677242053323170155711181989

n = 5131840839399378920948186496443110110286192423767372829154563295794505909320912318455704057852738989389634829469567060751702008328757598001677986669400413

Secret key for Alice:

d = 4520566060774125141993329159705093317044048027884319394090034853823308154832975154420378337823768929845766981578175777689514853696507817491278934493027469

p = 71803258844346529515225014083482695735913915220461279405209551052827658333947

q = 71470862492802265514062467637545951677637419918222972619441030833536121663879 Keys for Bob:

Open keys for Bob:

e1 = 5615155250541475272222034832959503995439091817509057978292838860746153969287720550032428095690120698174881263280359738804058005009959645665778719191281931

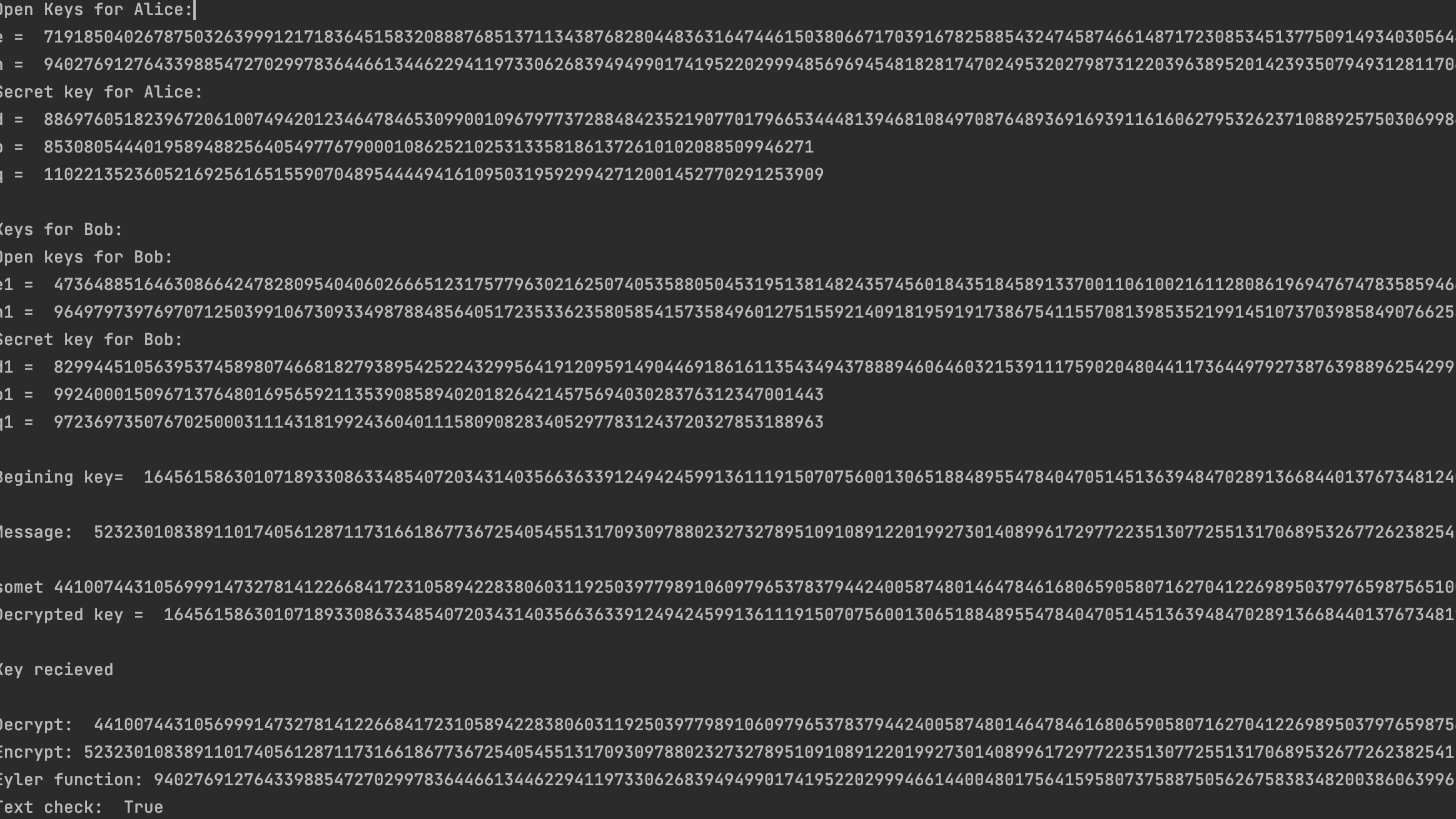
n1 = 8723848363757664555956258811776813777062671802106940098594845003461139599097939466789546847280002418031439135602754827164380634647511535349019865234823557

Secret key for Bob:

d1 = 2451073639877136378373449654359806760368968383024689860208459061338923387973286731074909088989150185201802148930462909901516761145783217167562123934039051

p1 = 81167929424041692171444899034362152914067872376891205535766212507886022717647

q1 = 107479005879059511671108287151410921789636890258334614308579113820271880199531



**Висновок:**

При виконанні лабороторної роботи номер 4, ознайомились з декількома способами генерації простих чисел, що й так вже були відоми, але вперше з практичним використанням. Окрім цього, було засвоєно інформацію про систему захисту інформації на основі RSA, і також не забули про проток розсилання ключів.