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 A, 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 A і B – тобто, створити та зберегти для подальшого використання відкриті ключі (e,n) , (n1, e1) та секретні d і d1 .
1. Написати програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів A 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 A и B, перевірити правильність розшифрування.

Скласти для A 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 x). 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 Alice:

e =

15201884110897490865883561979630034713299638767875461652021825833612458918888289753695324721406286603133136722795116124322685716772420533231701557111819 89

n =

51318408393993789209481864964431101102861924237673728291545632957945059093209123184557040578527389893896348294695670607517020083287575980016779866694004 13

Secret key for Alice:

d =

45205660607741251419933291597050933170440480278843193940900348538233081548329751544203783378237689298457669815781757776895148536965078174912789344930274 69

p =

71803258844346529515225014083482695735913915220461279405209551052827658333947

q =

71470862492802265514062467637545951677637419918222972619441030833536121663879

Keys for Bob:

Open keys for Bob:

e1 =

56151552505414752722220348329595039954390918175090579782928388607461539692877205500324280956901206981748812632803597388040580050099596456657787191912819 31

n1 =

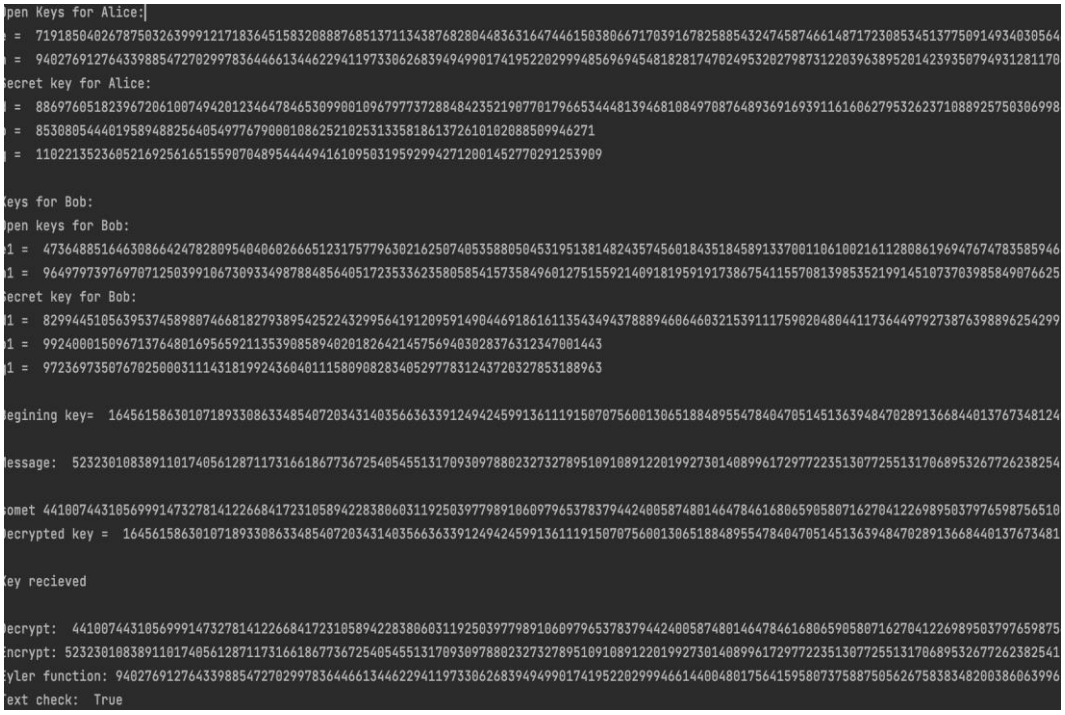
87238483637576645559562588117768137770626718021069400985948450034611395990979394667895468472800024180314391356027548271643806346475115353490198652348235 57

Secret key for Bob:

d1 =
24510736398771363783734496543598067603689683830246898602084590613389233879732867310749090889891501852018021489304629099015167611457832171675621239340390 51

p1 =
81167929424041692171444899034362152914067872376891205535766212507886022717647

q1 =
10747900587905951167110828715141092178963689025833461430857911382027188019953 1



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.