МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4

З дисципліни «Криптографія»

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

5 варіант

Виконали:
Правдива Тамара ФБ-02
Бобер Наталія ФБ-05

Перевірила:
Селюх П. В.

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 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) , (,) 1 n1 e та секретні d і d1 .

4. 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.

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:

Наша програма має такі основні функції:

Generate_Key_Pair - генерує відкритий і закритий ключ користувачам A і B;

Miller_Rabin - тест Міллера-Рабіна з попередніми розподілом;

Choose_Random_Prime – це функція для вибору випадкового простого числа з певного інтервалу;

gcd - пошук найбільшого загального дільника;

Find_Mod_Inverse - пошук зворотного по модулю;

Encrypt — шифрування;

Decrypt — розшифрування;

Sign - цифровий підпис;

Verify - перевірка цифрового підпису;

SendKey - відправлення ключа з підтвердженням справжності;

ReceiveKey - отримання ключа і перевірка на справжність відправника.

Труднощі при виконанні:

Труднощі виникли на моменті зашифрування, коли нам потрібно було перетворювати рядок в число. Проблема була вирішена взяттям значення ASCII кожного символу і конкатенацією цих значень.

Значення вибраних чисел p, q, q_1, p_1 із зазначених кандидатів, що не пройшли тест перевірки простоти і параметрів криптосистеми RSA для абонентів A і B:

q користувача user= 0xe73b198352cc0eff525d38438de963be6684967685f4aec3b9929655e2b6a24b

p користувача user = 0xabd053899779ed17c538102c51df369e152b9472958a2c234448066ffec1b9fd

n користувача user =0x9b30a583e3b9cc12e7817a948b6ee5aac054f0cc88602483712c7bf52f12584b7ab05aede0783bc670723cdb1b6f3b09ac721032ad8b0e7e0849e4f2ce51971f

e користувача user = 0x10001

d користувача user=0x64e64216f84eeac237c6d55e1331ed1d11a3a854332edf4e9e9c9f9390214b49f1d427dda5f491b77dfd5ae5058996da43e32d22a254508e997834e310b34431

n_1 сайта = 0xba955cfbfcb62141aa1e2e2cbbec3a4ffd8f7f95477f25c60e4f d1cb5a0 1bacca6eceb5c82599acc9c4f8d069f0b8538f57c1ca9e968931ff3e14b633be3611

e_1 сайта = 0x10001

Опис кроків протоколу конфіденційного розсилання ключів з підтвердженням справжності, чисельні значення характеристик на кожному кроці:

Процедура SendKey приймає на вхід відкритий ключ отримувача, зберігає його в змінні n_1, e_1:

n_1 = 0xba955cfbfcb62141aa1e2e2cbbec3a4ffd8f7f95477f25c60e4f d1cb5a0 1bacca6eceb5c82599acc9c4f8d069f0b8538f57c1ca9e968931ff3e14b633be3611

e_1 = 10001

В змінні n, e зберігає відкритий ключ відправника:

n = 0x9b30a583e3b9cc12e7817a948b6ee5aac054f0cc88602483712c7bf52f12584

b7ab05aede0783bc670723cdb1b6f3b09ac721032ad8b0e7e0849e4f2ce51971f

e = 10001

Генерує випадкове число k в межах від 1 до n1 – 1:

k=0x52aadc57687385aeaba13f68b5fd7c6427755c1bcd4f2c1d5b2ec86e25217c213ddf8bb40a51edbac20b6ef6bd2ffc97138c05c1d5589574f217a8d9c65666cc

Зашифровує його за допомогою відкритого ключа отримувача і зберігає у змінну k1:

k1=0x6c335621bceb7db4ab7830982d3acf4db82c17430578eb9cc60e8e3cda2fe3f833faccf58adf19366852bd2a9611717e9f304fdb9b8cc6c6fd98bf74913c9d07

Підписує k за допомогою секретного ключа та модуля відправника та зберігає в змінну s:

s=0x2144bcc35acb8c1be35f2983938a4853553d8db9343ab09b3e98956132094cac9e32519e74f3b4c2bef2c4d9f23bdb16ca3992663836df130bf4b575a50873ce

Підписує s за допомогою відкритого ключа отримувача та зберігає в s1:

s1=0x9d25269822656511b57f766c5297cc0709da4c5d31213fb8b5baea1bd6ac0298c1a36d172480c1cc9e76756eacdebb3f48826b2d3722b97a897cb3f4ebc3c055

Повертає масив [k1, s1]

```
відповідь ReciveKey сайту:
{"key":"52AADC57687385AEABA13F68B5FD7C6427755C1BCD4F2C1D5B2EC86E25217C213DDF8BB40A51EDBAC20B6EF6BD2FFC97138C05C1D5589574F217A8D9C65666CC","verified":true}
>>>
```

Висновок

В ході виконання лабораторної роботи ми ознайомились з тестом перевірки числа на простоту, методами генерації ключів для асиметричної криптосистеми типу RSA, системою захисту інформації на основі криптосхеми RSA, організували з використанням цієї системи засекречений зв'язок і електронний підпис. А також вивчили протокол розсилання ключів.