НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

Криптографія

Лабораторна робота №4

Підготував: Літвінчук В.С. ФБ-81
Перевірив: Чорний О.М.

Київ 2020

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

Завдання до 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 p_1, q_1 довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб $pq \le p_1 q_1$; p і q – прості числа для побудови ключів абонента A, p_1 і q_1 – 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), (e_1, n_1) та секретні d і d_1.

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

ABONENT A:

N =
10763776996356808894874748769656895652547108144438688110912836565277774134874677345666244729069968264495115795642950399986533567952191785752595884443505227

E =
10495181008119799317860888377333374563343971793677251957929792205289792859253695887506550039629086365490938208794112204529061510460920414254735848067348637

D =
23420491940702406067406904970578401803208936485462833566029465372642835643480780914576601414868529212982687187883235184167363558209920572879578407217746 67

ABONENT B:

N =
11415837419266846073724383887574032467922323203866497138316345874869147235567819773218002911805884799010307948553549131344741939490586768438246400382710389

E =
53538106159565893026514072343109580035234183156441903730812722302345268337035573070407221274996075320502934424083532470853410035622409610359760530023300 41

D =
18412767106886445781810824372176927033613530024885628140484284365904620732392935885051485102280871435595964100983153125815440310656251265704977109110970 01

SERVER :

E = 65537

N =
90813160292201577307812041242114752175509581095543849925235024507204946782211

SendKey, ReceiveKey (A to B):

```
Sk = SendKey(123,A[2],A[0],B[1],B[0]);
print(ReceiveKey(Sk[0],Sk[1],B[2],B[0],A[1],A[0]))
```

Результат:

```
True
123
```

Encrypt, Decrypt (A to SERVER):

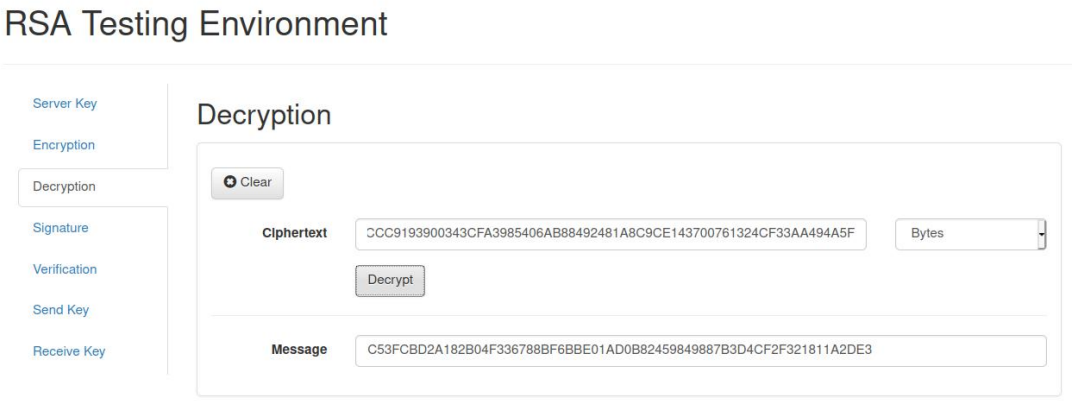
```
server_n =
int("0xC8C66D5F3124CFF18A6D710BA138AB8E8FB29A500368456D3759993F2
7366803",16)
server_e = int("0x10001",16)
print("server_e = "+str(server_e))
print("server_n = "+str(server_n)+"\n")
print(Encrypt(M,server_e,server_n))
```

```
gezter@doby:~/Документы/university/5semestr/Криптографія/Labs GIT/fb-labs-2020/cp_4/cp-4_Litvinchuk_FB
-81$ python3 main.py
M = 892183492690907564833897193328918024617846139486400533405192965786936652712 67

[107637769963568088948747487696568956525471081444386881109128365652777741348746773456662447290699682644951157956429503999865335679521917857525958844435052 27, 104951810081197993178608883773333745633439717936772519579297922052897928592536958875065500396290863654909382087941122045290615104609204142547358480673486 37, -2342049194070240606740690497057840180320893648546283356602946537264283564348078091457660141486852921298268718788323518416736355820992057287957840721774667]
[114158374192668460737243838875740324679223232038664971383163458748691472355678197732180029118058847990103079485535491313447419394905867684382464003827103 89, 535381061595658930265140723431095800352341831564419037308127223023452683370355730704072212749960753205029344240835324708534100356224096103597605300233004 1, 184127671068864457818108243721769270336135300248856281404842843659046207323929358850514851022808718435595964100983153125815440310656251265704977109110970 01]
server_e = 65537
server_n = 908131602922015773078120412421147521755095810955438499252350245072049467822 11

781592514890353670032505649810341428347769554926216913295979636733306256287 67
```

Переводимо останнє число у 16 і розшифровуємо його на сайті



Переводимо message у 10 і отримуємо наше
M=892183492690907564833897193328918024617846139486400533405192965786936652712 67

Висновок

В результаті виконання лабораторної роботи я ознайомився з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практично ознайомився з системою захисту інформації на основі криптосхеми RSA.