МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ 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

Виконали:

Орлов Дмитро ФБ-14

Макуха Андрій ФБ-14

Перевірила

Селюх П. В.

Київ 2023

Мета

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.

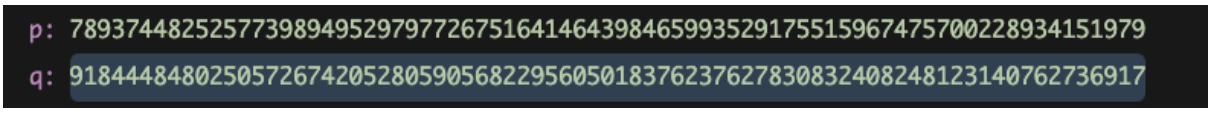
Порядок виконання:

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.

Написали метод для знаходження довільних випадкових простих чисел з перевіркою на тест Міллера-Рабіна, а також на ділення на не досить великі прості числа.

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.

Згенерували пари чисел для двох абонентів перевіряючи кожен раз на умову pq(A) ≤ pq(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 .

```
User: UserA
PrivateKey:(432790543137233881389162189684352440508989734198197593199177352797982100927980589621914453107757227907131534066575399355442874908495457441066418139882153, 723851400481687979580453686629500724167792807724249306253585533572817102103663, 995795511008978692384596529721880113087811120737585532291688003948270223606290)
PublicKey:(720807975237227366825959909157392780016204833802980321867712482092029299835266596775376851096785354322993992794932961313939248408819170507065629612788402 7, 65537)
q:995795511008978692384596529721880113087811207375855322916880039482702236062 9
p:723851400481687979580453686629500724167792807724249306253585535728171021036 63

User: UserB
PrivateKey:(909507916769684042977635772474975035712871467725891358185668015851351214500062627918163825927683526918944786635584970941993124245451279787072862767215 53, 795847326235018293057505226915322687486847426578603989550288510505805636690 11, 108577562934672825084932082663799054434613129478429435071775118807976685319653)
PublicKey:(864111631506737940339595761390148433104007790379120642815513551099594151126296062472998882989106729955039512281873740640002892301346595188817392702537318 3, 65537)
q:108577562934672825084932082663799054434613129478429435071775118807976685319653
p:795847326235018293057505226915322687486847426578603989550288510505805636690 11
```

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.

Далі наведено приклад шифрування та дешифрування за допомогою пар:

```
Original message:21106
Encrypted by B:428804195169024095249218397734670709747768635670465032437118146787199739566746417384946370374910589360061960517787372141659678975085190007951011018565551
Decrypted by A:21106
Sign:187588606561109510607363673148977882310829410474506319700189834492008427112871954292211625098763685499036694092238085853909523105238974856471660345590441 1
Sign is correct.
```

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 $0 < k < n$.

Беремо рандомні значення k з інтервалу, та намагаємось відправити ключі (другий приклад навмисно провалено, для перевірки коректності):

```
Original k = 962765088521775166630109687871788634663960911490960646938078620189784219360667157007255970480561359389857545643143002071834773039766545906382680213723795
Sent........
S1 = 500007902742548175263622038016522450770999903943965821563370702780882497182729837656565241111948904000787119586400363482820602891262910033827853251117828 94
k1 = 660412765843862482259589429734767033908085263880779323110981003187641976230706247136955106944583918908888449973688917747913239195660121101291042229164502 85
S = 642753689944956390929190534027136280560816803377060869907972922661526803217641806044443240416978407791408475126164972525248956816150555976408705855074181

Received.......
k = 962765088521775166630109687871788634663960911490960646938078620189784219360667157007255970480561359389857545643143002071834773039766545906382680213723795
S = 642753689944956390929190534027136280560816803377060869907972922661526803217641806044443240416978407791408475126164972525248956816150555976408705855074181
Sign is correct.

You were Authenticated

Original k = 425849003119456756506236589638977719734577060716358799872431787155022332368051024655444144707228716845540987937987797277521425541028994524219837665477 4329
Sent........
S1 = 81748695580728634097015002909486611786591187042456852049473688272429845136559302434770733324868448721531748999245795068674952332964763472811841948070335 47
k1 = 489470348765170096950236722564295250226039768544612858681235325240286989891349257319965629396076496261724450411819807524530490168406839986045967685784804 2
S = 438056968567180181032618827438927615929457873462898504879928887641128281049330317764302399815271612409907432410343558691246925887942239516477482394537642 4

Received.......
k = 629570965986041044532574084793422892149419149287629348962680324274914372622464048768764447461438393451283433451755104248306591598056117792500599202805 36
S = 438056968567180181032618827438927615929457873462898504879928887641128281049330317764302399815271612409907432410343558691246925887942239516477482394537642 4
Sign is incorrect!.

Authentication failed
```

Висновки:

В даній лабораторній роботі ми на практиці змогли перевірити роботу криптосистеми RSA та розглянути наочно його працездатність. В результаті отримали передачу ключів, успішне шифрування та дешифрування даних між двома абстрактними користувачами.