Міністерство освіти і науки України Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"

Фізико-технічний інститут

КРИПТОГРАФІЯ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4

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

Варіант 8

Виконали: студенти групи ФБ-01

Курило А. В. і Шевченко Д. М.

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 криптосистеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчення протоколу розсиланя ключів.

Завдання : 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 $0 < k < n$.

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. Відповідно, програмний код повинен містити сім високорівневих процедур: GenerateKeyPair(), Encrypt(), Decrypt(), Sign(), Verify(), SendKey(), ReceiveKey().

Кожну операцію рекомендується перевіряти шляхом взаємодії із тестовим середовищем, розташованим за адресою

http://asymcryptwebservice.appspot.com/?section=rsa.

Наприклад, для перевірки коректності операції шифрування необхідно а) зашифрувати власною реалізацією повідомлення для серверу та розшифрувати його на сервері, б) зашифрувати на сервері повідомлення для вашої реалізації та розшифрувати його локально.

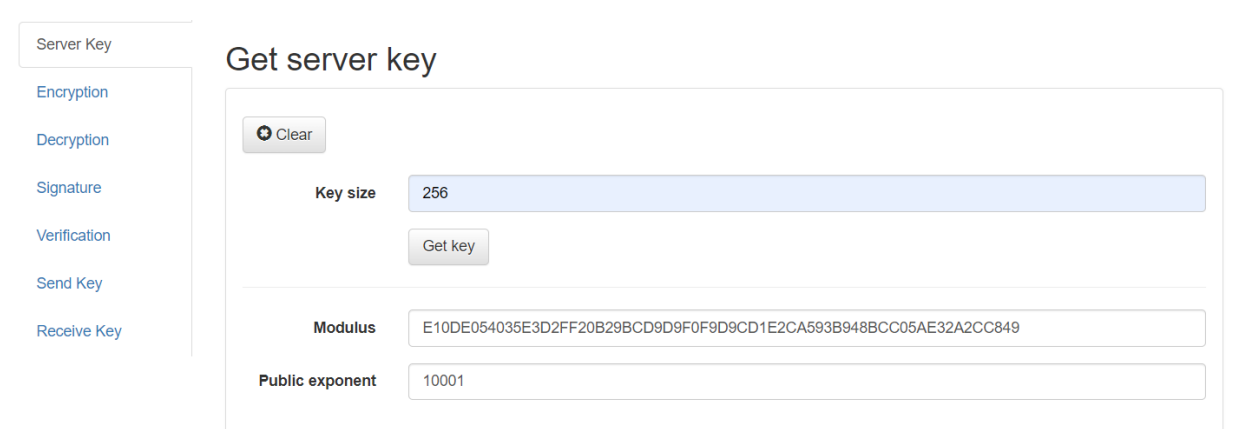
Хід роботи:

crypto_lab4.py – код програми

Робота усіх функцій:

```
C:\Python>py -3 crypto_lab4.py
p = 5893826029072857632768007681099673287412779436766467236049097793880214248768 1
q = 1086208115871085819835516158608749158469005131115213269169100700156492677816 33

p1 = 7060642162396757455039584404327531385077104431074090485691031060957288060697 7
q1 = 9193328532733159024190211338334387517903663413465740905355672245528083277081 7

n = 6401921666311192165455134740622130403297609042283107931105673659330668232163146553477370069344135713441723394571755597673336385538628794871499218900563073
n1 = 6491080305098086133363818888543775526867157815723414399321432771483136338140606639698918148031370277716934856765098741029754670793861589914292675692190209

e = 1997202162119238801079948780021666226301548204509826375433616344668442187709784743378805062993178255850549044046736290731545072752444200083923666792752 23
e1 = 3878069160366915252721223481680936055715312205687615990689749159426215795805481177493154369883967060008048455553680514678655388335861976141110567525247336 3

d = 2448237086734450719937310388584969141387192694987455308324687350175838478042987524883567507193152640291894471715187760481803326094493232877993427851038640 7
d1 = 1211209548427276691359672330714660018098702756923224099618902880013869456641027973657546832501134346981006747727602005591373996387110321482775234313265871 5

M = 5574292196143090188993357472368053548934716086439627208806713966770312847039720020187474968944666110438115707635833862087024296871187927152851615014503732

k = 5160316712965495501546599914104465068261489427983455538157187447873927149170469714508515058738638537085280069568713482313048395629506890799660522248002570

Encrypt M = 3784115092513757378258714070374510978649537233404975351966489696436819014751477409584901180333048649267180999713752784681120458639797660785148212559086551

Decrypt M = 5574292196143090188993357472368053548934716086439627208806713966770312847039720020187474968944666110438115707635833862087024296871187927152851615014503732

A підписав повідомлення S = 1231044435050209261573533590624621825856694049113718144734193468588935540200285361197286559070301183977911109888236797506758350684826278660959624504476419

B перевіряє підпис M = 5574292196143090188993357472368053548934716086439627208806713966770312847039720020187474968944666110438115707635833862087024296871187927152851615014503732

Формування повідомлення (k1, S1), відправка k = 5160316712965495501546599914104465068261489427983455538157187447873927149170469714508515058738638537085280069568713482313048395629506890799660522248002570

S = 9858672469638440809676756068999575407849519747533374994053387123363511991074093657484852308671793412224505557576812975128334701502091025004940854692534579

k1 = 2311741969623345648845709993739638302215978405784247045051545844270533425807027797174339353500621130430025105124926695956347067375586136005060764839454 28

S1 = 1824436925008137150093983556882414597322859609376206751681830984330133737485497870062963230792597022158478586305197881727914496476699709033650051709289 82

Повідомлення сформоване, відправка до B

Отримання повідомлення (k1, S1)

Перевірка k, S (Конфеденційність) k = 5160316712965495501546599914104465068261489427983455538157187447873927149170469714508515058738638537085280069568713482313048395629506890799660522248002570
S = 9858672469638440809676756068999575407849519747533374994053387123363511991074093657484852308671793412224505557576812975128334701502091025004940854692534579

Перевірка підпису A (Автентифікація) k = 5160316712965495501546599914104465068261489427983455538157187447873927149170469714508515058738638537085280069568713482313048395629506890799660522248002570
```

Перевірка з http://asymcryptwebservice.appspot.com/?section=rsa



From

Decimal

To

Hexadecimal

Enter decimal number

384210381293129 10

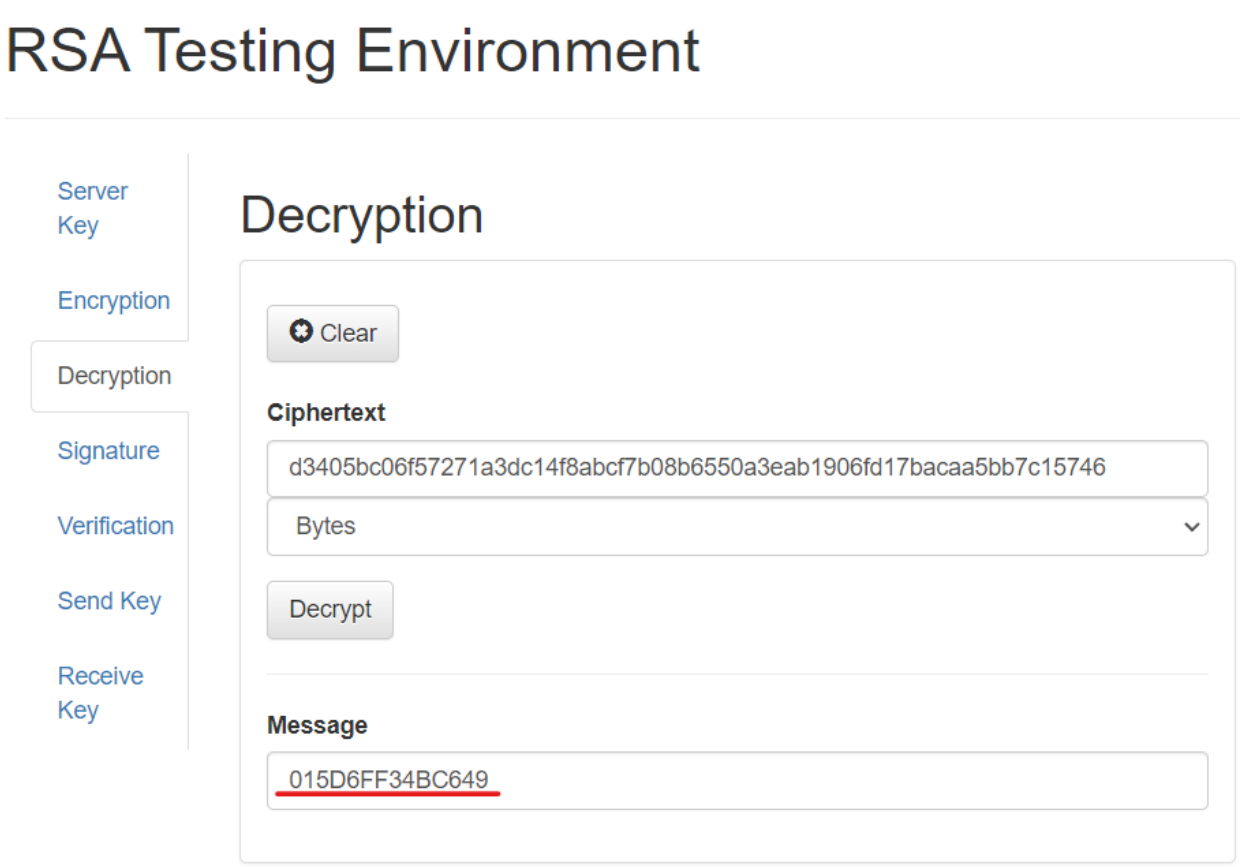
= Convert × Reset ⇅ Swap

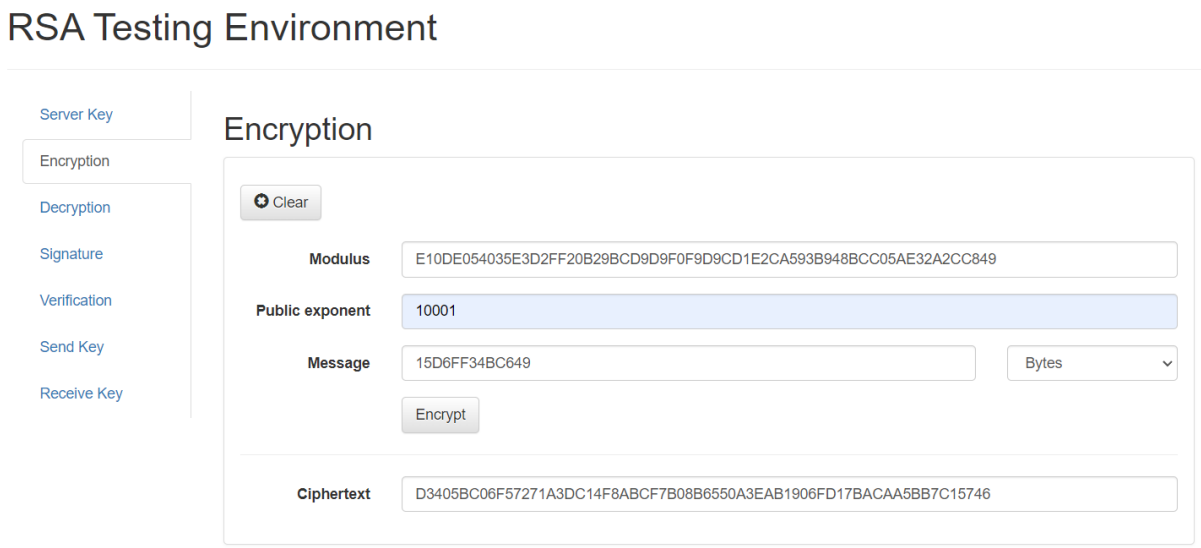
Hex number

15D6FF34BC649 16

```
M = int('15D6FF34BC649', 16)
e = int('10001', 16)
n = int('E10DE054035E3D2FF20B29BCD9D9F0F9D9CD1E2CA593B948BCC05AE32
A = encrypt(M, e, n)
print(hex(A)[2:])
```

```
C:\Python>
C:\Python>
C:\Python>
C:\Python>
C:\Python>
C:\Python>
C:\Python>
C:\Python>
C:\Python>
C:\Python>
C:\Python>
C:\Python>py -3 crypto_lab4.py
d3405bc06f57271a3dc14f8abcf7b08b6550a3eab1906fd17bacaa5bb7c15746

C:\Python>
```





Висновки : під час виконання лабораторної роботи ми отримали навички реалізації та розуміння роботи критпосистеми RSA, алгоритму електронного підпису та генерації ключів та простих чисел, їх перевірка на простоту ймовірнісним тестом Міллера-Рабіна.