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

Перевірила:
Селюх П. В.
_____

Виконали:
студенти III курсу
групи ФБ-95
Гурджия В.
групи ФБ-94
Золотов І.

Київ 2021

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 qp, i11 , qp довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб11qppq ≤ ; p і q – прості числа для побудови ключів абонента A,1p i1q – абонента B.

3. Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ) (qpd) та відкритий ключ),(e, n). За допомогою цієї функції побудувати схеми RSA для абонентів A і B – тобто, створити та зберегти для подальшого використання відкриті ключі),(n, e),(11 ne та секретні d i1d .

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().

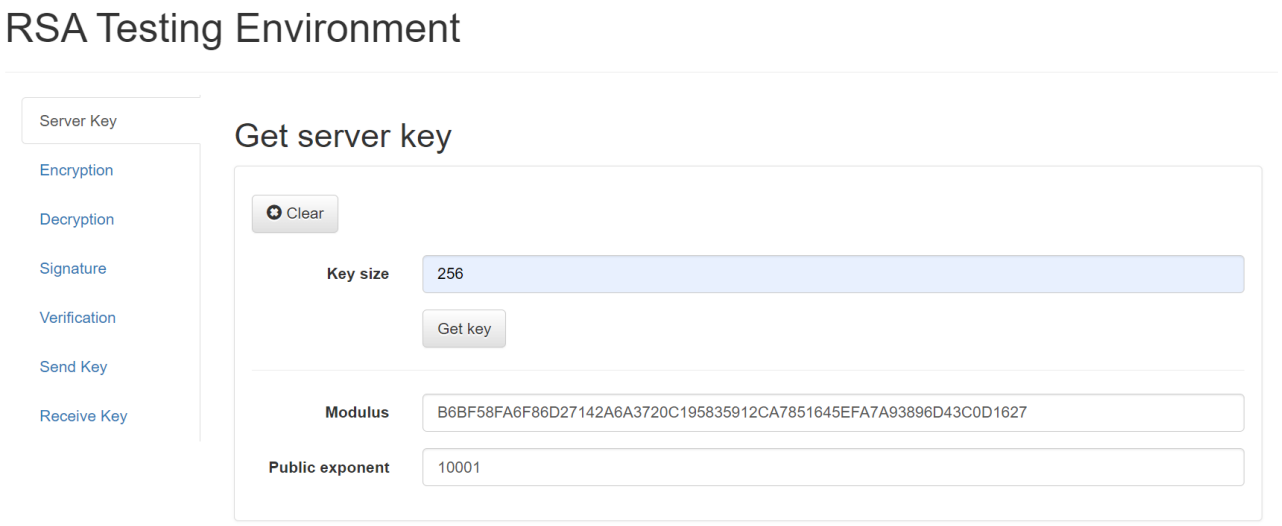
Хід виконання роботи:

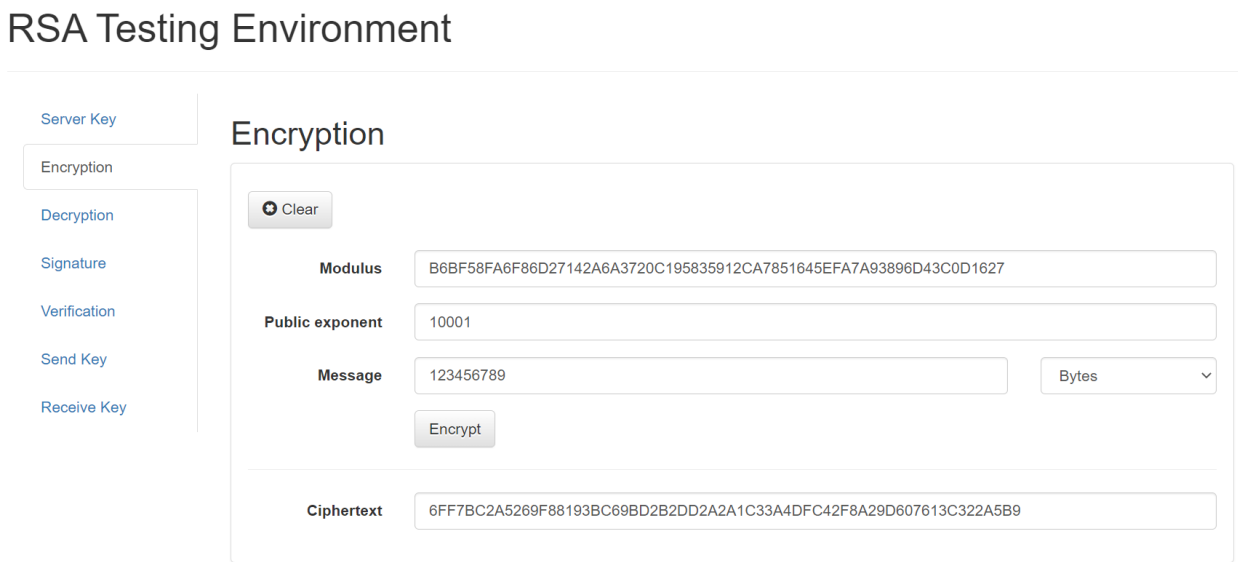
Робота самої програми:

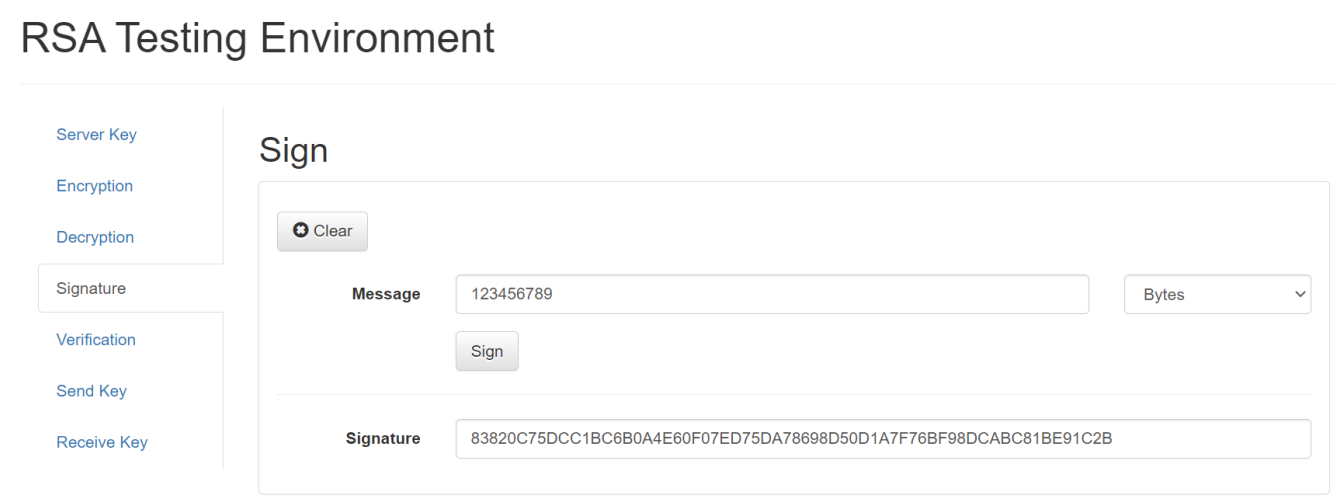
```
                    For A
            private key:
d:  602973640117375771782763910382330161485051027343801324411108716441502291898499551445399638516952646539087138526875039446058140685814479773322185020500902l
p:  72718291972744824538777532333432182808941884357672733152904186093330489550321
q:  88955769248589846188968710732889288229594312404793609568470034061083049157383
            public key:
n:  64687116008790719242461797188219539195918437439394936343506974501863850432797241878005702886246986297708124820001266071875139196441679672368780456271699943
e:  51350743098523607947760495215384221323863158370294431155922158586109397912682263049992134614459546095352785933413243334041566125660937073496977437268578l

                    For B
            private key:
d:  46010856474804926579470952968785633672781821375234187234715059073280902496554365408428110785461823677704951511019769518083200898592342477153054505862034351
p:  95866094104187888817657844814438194115050386349941005387381897087459370924707
q:  70341896766479388095142780808013861067646827605546582753265100020889087133303
            public key:
n:  67434028948823827877205591197669574626803567990481693945368058294976628226504525560415550704334012953616815453791167878009699786496652613610714948852172211
e:  19415501526404279170557129671000314866148458788048336740547792584168290520404455724816021704908386048728869057760535935115928388673783479934695134805521831

Enter Message: 969
Encrypted Text:  241341345141536778217676856207193207002925921943003845565967245590759689010610660565455597730026160803376747798317449788547224604391355245359173979422715551

Signature:
M:  241341345141536778217676856207193207002925921943003845565967245590759689010610660565455597730026160803376747798317449788547224604391355245359173979422715551
S:  384550960292891441542445360656756887416085019610008568530525600256229230860384163502953853234801466245520903195977874904972956974540752814989189954120181
Signature Verified :)
Decrypted Text:  969
```

Генеруємо значення p1 q1 I p2 q2, далі передаємо у функцію RSA і за допомогою неї ми отримуємо приватні та відриті ключі. Потім проходить шифрування і розшифрування і перевірка цифрового підпису, і як бачимо перевірка пройшла успішно.

Наступний пункт буде перевірка через сайт коректності роботи програми.

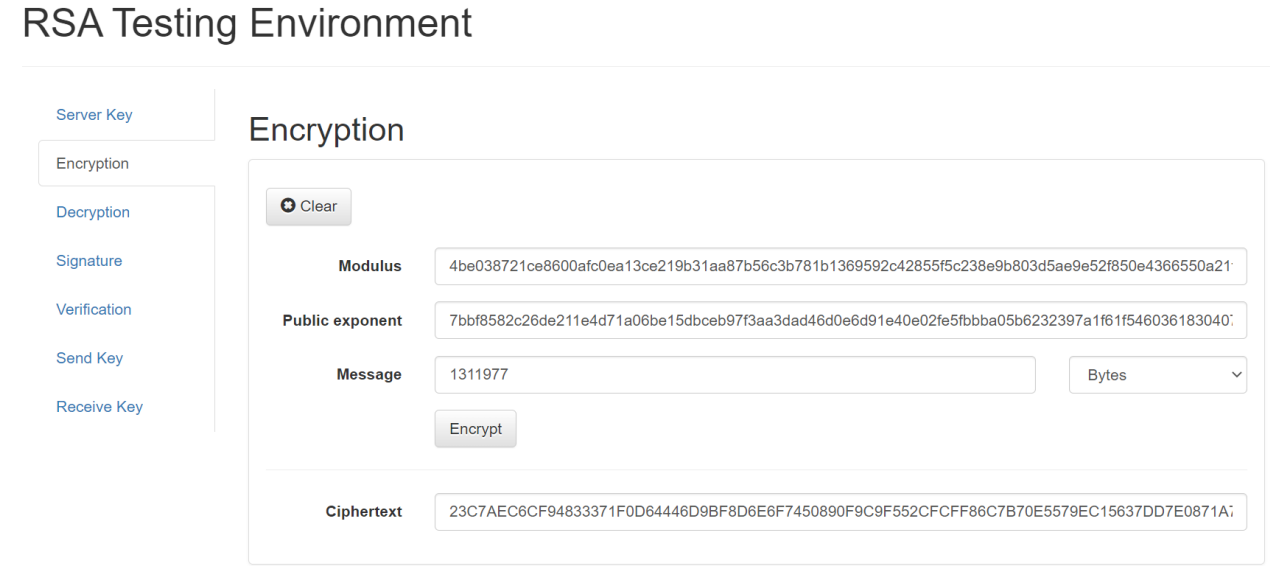


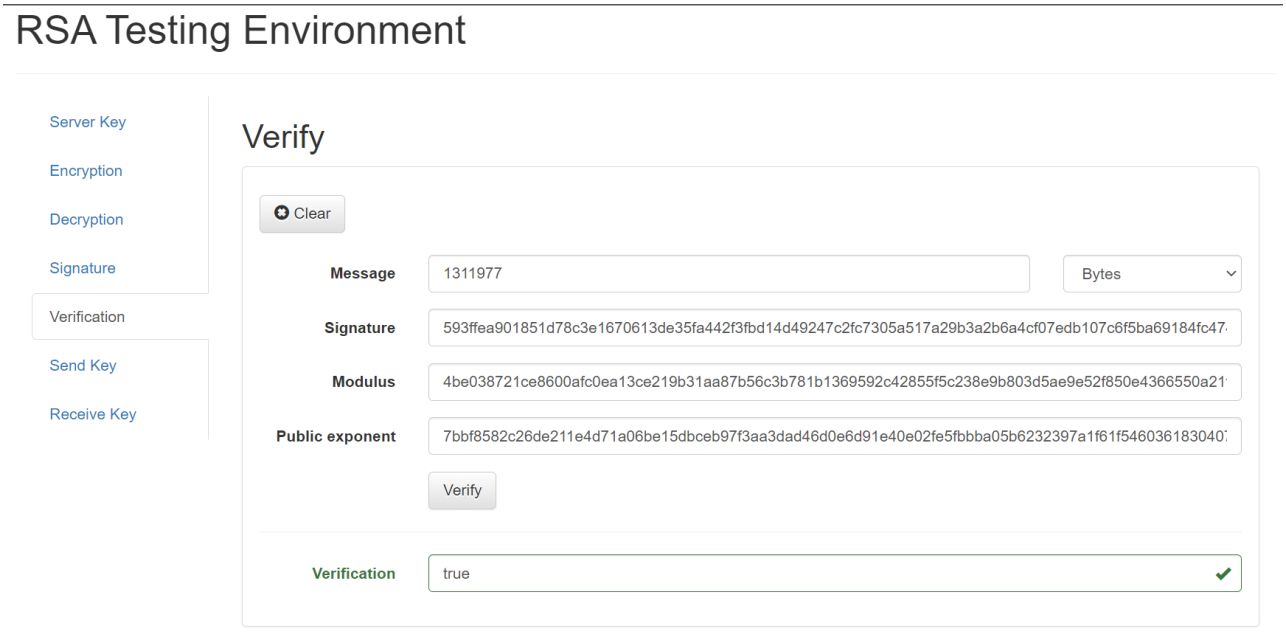


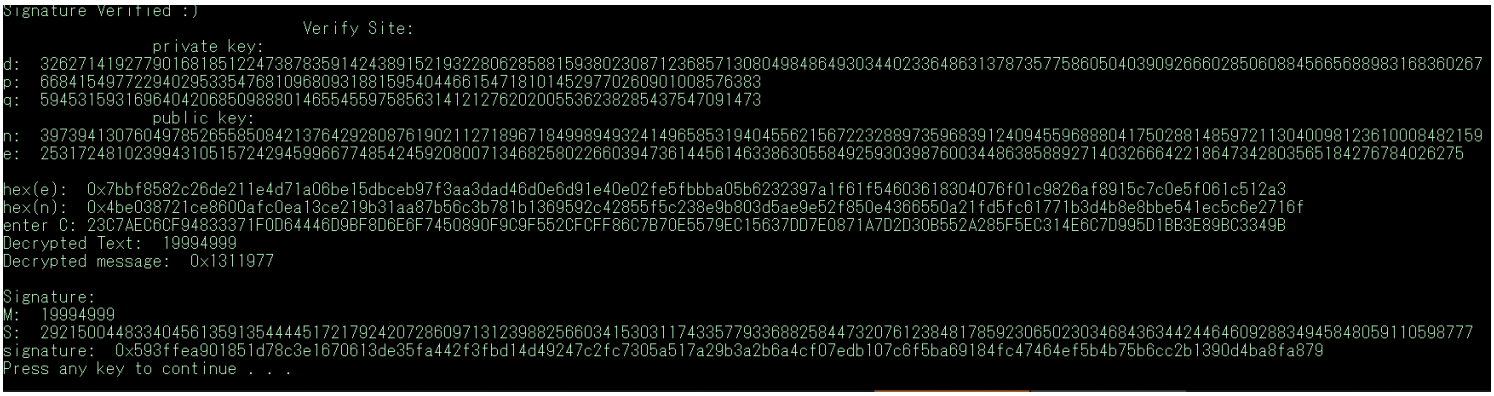



```
                              Verify Site:
Encrypted Text:  506444360870529884306199123325730884920473652575779715529232392886155656984 89
Encrypted message: 0x6ff7bc2a5269f88193bc69bd2b2dd2a2a1c33a4dfc42f8a29d607613c322a5b9
Sign: 0x83820c75dcc1bc6b0a4e60f07ed75da78698d50d1a7f76bf98dcabc81be91c2b
Signature Verified :)
```

Наступна перевірка через сайт.







Висновок:

Ми навчились використовувати RSA для шифрування і дешифрування текстів і повідомлень а також використовувати і перевіряти цифрові підписи.
Також отримали навички роботи з алгоритмом Міллера-Рабінна і схеми Горнера.