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

Виконав:
Іванілов Ігор
Березовський Максим
Варіант 12
Група:
ФБ-06

Київ - 2022

Мета та основні завдання 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.

Хід 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.

Ось які числа вдалось створити так перевірити за допомогою тесту Міллера-Рабіна:

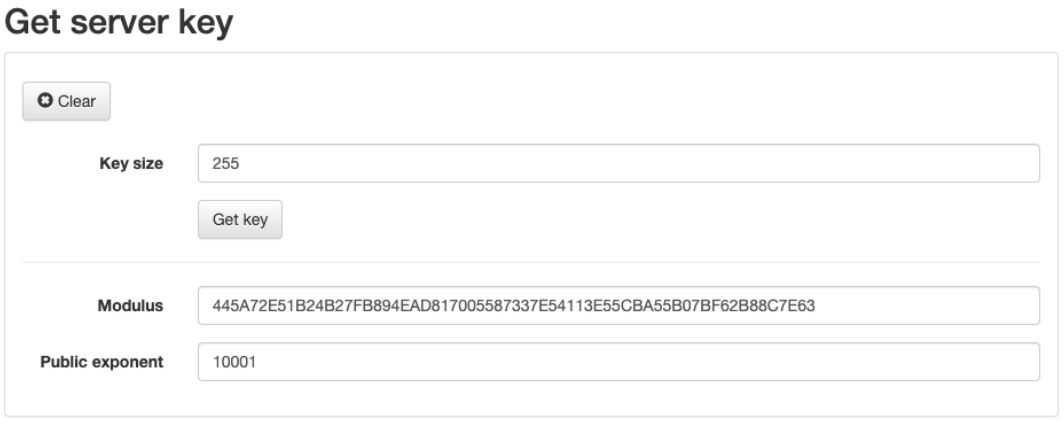
```
p1 76359642116807025490125207355526350059270391187441455021752396388666507470793
q1 58309526249452777231739214274484874053966235044824936805329617731296510417127
p2 10930800474410717722097494484735661637919731861514723356832669639847750695401l
q2 89914520439454877512413578544902776404014761058769747551219190452650951080269
```

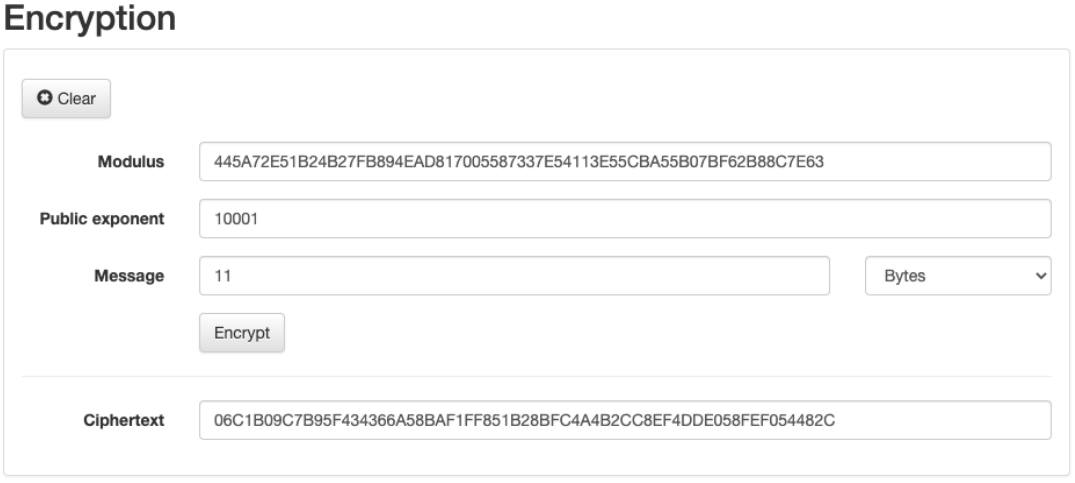
3. Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ (d, p,q) та відкритий ключ (n,e) . За допомогою цієї функції побудувати схеми RSA для абонентів A і B – тобто, створити та зберегти для подальшого використання відкриті ключі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

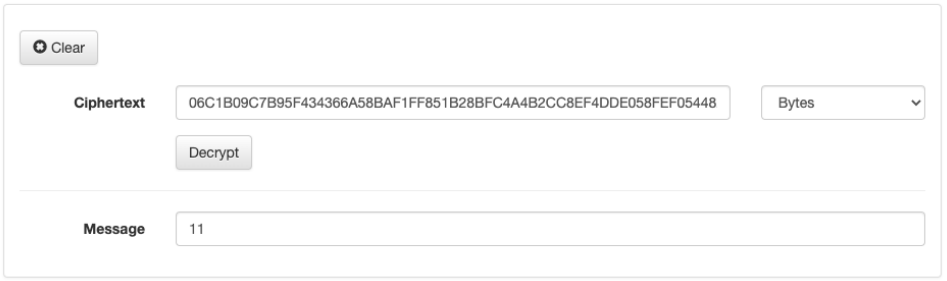
Скріни працездатності коду:

Роботи з сайтом:

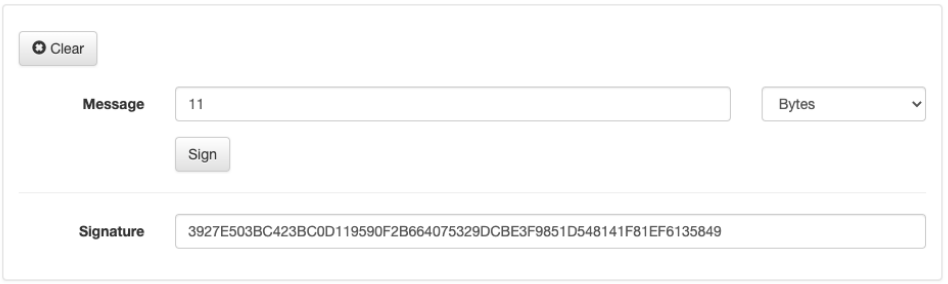




Decryption



Sign



Verify


```
n_server = 309170829152620181535734839201869644879561775870769734083177385232905187160 03
e_server = 65537
message_server = 17
encrypt_server = encrypt(message_server, e_server, n_server)
print(encrypt_server)

sign_server = 258523200053917538452522560839979890284060971769295407336426515733876029092 57

verify_server = check_sign(message_server, sign_server, e_server, n_server)
print(verify_server)
```

```
305609750112564106119501831471556229316815013344181321367818139129589820830 0
Верифікація пройшла успішно
```

Висновки

Впродовж цієї роботи ми навчилися працювати з великими простими числами, а саме перевіряти її простоту та використовувати її при подальших діях.Також познайомилися на практиці з RSA та створили свій приклад роботи захищеного трафіку. Ці знання допоможуть нам у подальшому розвитку як спеціалістів, а також роботі чи навчанні.