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»

Виконали:

студенти групи ФБ-93

Флекевчук Д.І

Перевірила:

Селюх П.В.

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.

Завдання

- 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 лозволяється.
- За допомогою цієї функції згенерувати дві пари простих чисел p0, q0 і p1, q1 довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб p0q0p1q1, де p0, q0 прості числа для побудови ключів абонента A, p1, q1 абонента B.
- Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ (d, p, q) 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 (e0, n0), (e1, n1) та секретні d0 і d1.
- Написати програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів А і В. 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 М і знайти криптограму для абонентів А і В, перевірити правильність розшифрування. Скласти для А і В повідомлення з цифровим підписом і перевірити його.
- 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.

Хідроботи:

Частина 1:

Для генерації чисел використовую бібліотеку random мови python завдяки ній отримаю рандомні числа заданого розміру. На сильну простоту превіряю тестом Міллера-Рабіна, для пришвидшення роботи алгоритму, було використано постулат Бертрана.

Частина 2:

Створив клас User, який на даному етапі розробки зберігає закритий ключ та своє імя.

За допомогою методу для генерації чисел тимчасово надаю користувачам ключі.

Створюю клас RSA, клас RSA дозволяє зберігати відкриті ключі за іменами користувачів, а також породжувати нові оюєкти класу User, та надавати цим екземплярам класу їх закриті ключі та імена. Також користувач може змінити ключ на ключ меншого розміру за потреби(для протоколу обміну ключами).

Частина 3:

Створюю клас RSA, клас RSA дозволяє зберігати відкриті ключі за іменами користувачів, а також породжувати нові оюєкти класу User, та надавати цим екземплярам класу їх закриті ключі та імена. Також користувач може змінити ключ на ключ меншого розміру за потреби(для протоколу обміну ключами).

Частина 4:

Реалізую функції шифрування розшифрування та для верифікації за цифровим підписом.

Демонстрація роботи

Шифрування та розшифрування.

```
PROBLEMS 2 OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\Данил\Desktop\crypto-FB-9\cp4\flekevchuk_fb-93_cp4> python main.py
test message
PS C:\Users\Данил\Desktop\crypto-FB-9\cp4\flekevchuk_fb-93_cp4>

PS C:\Users\Данил\Desktop\crypto-FB-9\cp4\flekevchuk_fb-93_cp4>
```

Верифікація за підписом

```
RSA_SERVER = RSAma()

Alice = RSA_SERVER.addUser("Alice")

Bob = RSA_SERVER.addUser("Bob")

184

185  print(Alice.verify)

186  Bob.sign(1111), Bob.name

187  )

188 )

PROBLEMS (2 OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Users\Данил\Desktop\crypto-FB-9\cp4\flekevchuk_fb-93_cp4> python main.py

True

PS C:\Users\Данил\Desktop\crypto-FB-9\cp4\flekevchuk_fb-93_cp4> [
```

Протокол обміну ключами.

Тести на сайті

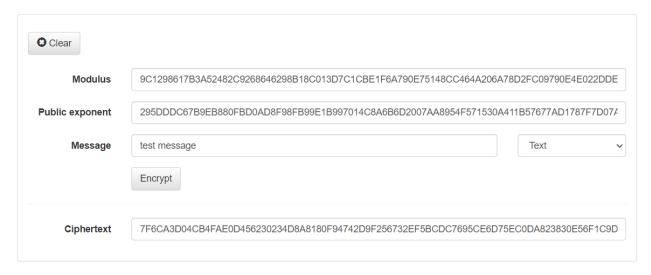
Генерую ключі для одного з абонентів на сайті



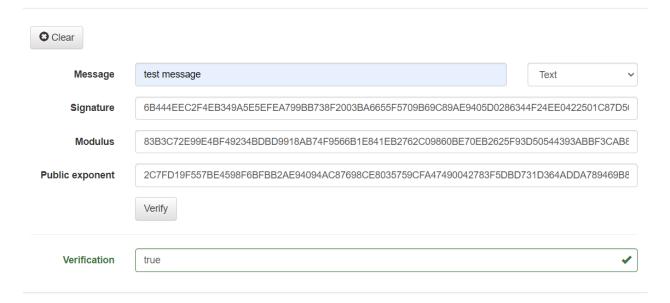
I) Я зашифрував повідомлення(test message) для абоненту з сайту та віддав на сайт ШТ



ІІ)Зашифрував повідомлення на сайті та розшифрува у себе у себе



III) Згенерував підпис та надав його разом з відкритими ключами абоненту на сайті



IV)Згенерував підпис абонента на сайті, перевірив підпис у себе.



Висновки

В ході виконання цієї лаю. роботи я навчився перевіряти числа на сильну простоту, з методами розподілення та генерації ключів в протоколі RSA. Після отримання

теоретичних відомостей я розробив на основі RSA систему, організував засекрчений зв'язок, та електронний підпис та протокол передачі ключів .