КРИПТОГРАФІЯ КОМП'ЮТЕРНИЙ ПРАКТИКУМ №5

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

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 і рекомендації щодо 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 p, q і 1 1 p , q довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq \leq 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 e1 e2 секретні e3 e4 e4 e5.
- 4. 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 и В, перевірити правильність розшифрування. Скласти для А і В 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 роботи

- 1. Написали алгоритм генерації простих чисел на основі тесту Міллера-Рабіна
- 2. Написали функції генерації пари ключів, шифрування розшифрування, верефікації, відправки та отримання ключів

Результати роботи програми

Згенеровано повідомлення M=2334137417967725729855083369190414077067757951271287258509066539971.

А зашифрував повідомлення відкритим ключем В, і отримав C=798644666486311869067162675704970541.

В розшифрував повідомлення своїм закритим ключем і отримав M=233413741796772572985508336919041.

В зашифрував повідомлення відкритим ключем А, і отримав C=636946658832122551241418209061372534.

А розшифрував повідомлення своїм закритим ключем і отримав M=233413741796772572985508336919041.

А підписав повідомлення, і отримав S=289255088984449910430268380685547853766152187498324889889.

В перевірив підпис з результатом True

В підписав повідомлення, і отримав S=194150930795040708781680353397114478375648796775489798386.

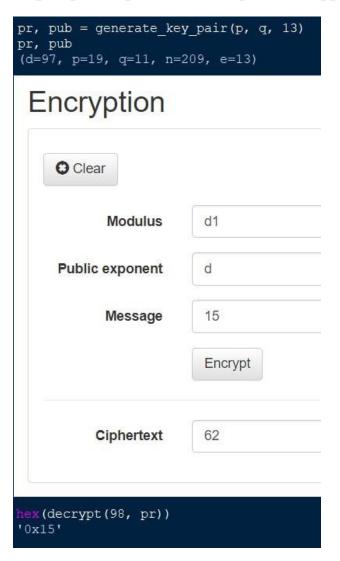
А перевірив підпис з результатом True

А згенерував певний ключ k=2032452553517134347289522865208033821419260088857646748276402143497.

А зформував повідомлення (k1, S1)=(32048323094433966634698376948027583969528795776151516736997.

В отримав повідомлення, після чого знайшов і перевірив ключ k=20324525535171343472895228652080

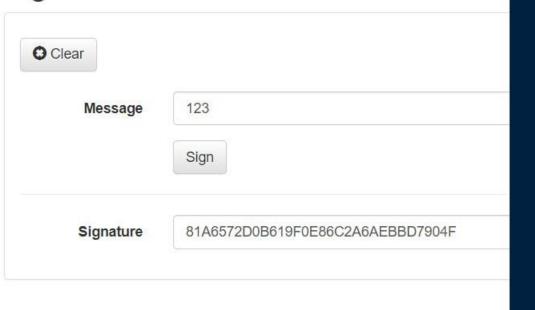
Перевірка коректності операцій шифрування з сервером:



Get server key



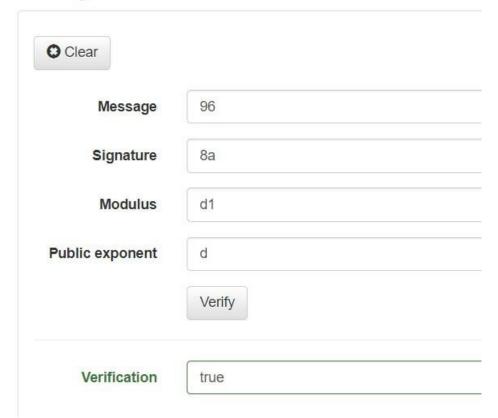
Sign

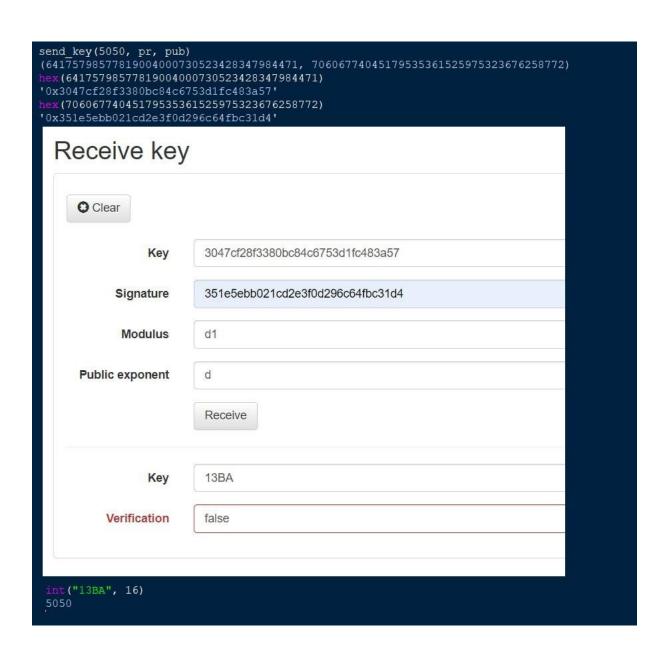


verify((int("123", 16), int("81A6572D0B619F0E86C2A6AEBBD7904F", 16)), pub)
True

```
sign(150, pr)
(150, 138)
hex(150), hex(138)
('0x96', '0x8a')
```

Verify







Висновок: ми ознайомилися з тестами перевірки чисел на простоту та використали тест Міллера-Рабіна. Навчилися генерувати ключі для асиметричної криптосистеми RSA.