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:
студент групи ФБ-93
Килимчук Денис

Перевірила:
Селюх П.В.

Київ 2021

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.
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 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.

Варіант: 11

Хід роботи

1)

```
p1=61243515839170765062248483224368043173433601463612354184187057703427791701081
q1=69601466102931450588341025905410714147007252475248556046943745068877855632929
p2=10760976229447038473752731723286943036667933186243251340067878491278720118683 3
q2=92157630482613438383463816444422199356328642252664006652348314870248007698701
```

Перевірку простоти виконав за допомогою імовірнісного тесту Міллера-Рабіна

Для першого користуача:

Шифрований текст:

983908546822273468881597664661571983587903959018361064516704193045988617472727640908122059567190907621221114990615133317086415732236614824052698291504 1383

Відкритий текст:

651666599434915132768566483971506290500209852081279016554915698540813042589 61

Підпис:

124953045324156805671801727823198252266094217275429131673976530771232656307 120437018327440052771334851417985484231802019635298037710343634167816580170 9799

Для другого користувача:

Шифрований текст:

508127138571284747495236763645061696326975937874846818595466871832355157158 154536459199874508569306238405654141725144325655769845957750997850481545569 078

Відкритий текст:

651666599434915132768566483971506290500209852081279016554915698540813042589 61

Підпис:

487623624990003507727757872999054263182929181235506047216064570552229173759 353356755516319203620197144868213678040674713400268536296196381068071701425 9096

Перевірка для обох користувачів:

```
Шифрований текст: 323706191915606880603309036176693793795971582379775747411406444258441133366590185553348205760815017369338028210954272206135398593493349558411657969 7202558
Відкритий текст: 651666599434915132768566483971506290500209852081279016554915698540813042589 61
True
Шифрований текст: 3144611889978156841165535678889126675026254368594554766208980422790036978349835641668812302530627210508903435629039407103600211208493038242699501996238738
Відкритий текст: 651666599434915132768566483971506290500209852081279016554915698540813042589 61
True
```

Висновки: у даному практикумі ознайомився з тестами перевірки чисел на простоту імовірнісним методом Міллера-Рабіна, з методами генерації ключів для асиметричного шифрування RSA. Також ознайомився з системою захисту інформації на основі RSA і вивчення протоколу розсилання ключів та цифрових підписів.