



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 III курсу ФТІ
групи ФБ-94\ФБ-96
Солопенко Борис
Бутко Максим
Перевірила:
Селюх П.В

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 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 і довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб p і q – прості числа для побудови ключів абонента А, і – абонента В. $q \cdot p$, $1 \cdot 1$, $q \cdot p$, $1 \cdot 1$, $q \cdot p$, $r \cdot q$, $1 \cdot p$, $1 \cdot q$
3. Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ та відкритий ключ. За допомогою цієї функції побудувати схеми RSA для абонентів А і В – тобто, створити та зберегти для подальшого використання відкриті ключі, та секретні і.), (, (qpd), (en), (ne), (1 1 n e d 1 d
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 і 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


Для виконання поставлених завдань було реалізовано функцію генерації чисел заданої довжини та функцію перевірки отриманого числа на простоту. За допомогою цих функцій було отримано дві пари ключів p і q . Також було реалізовано функцію отримання секретного та публічного ключів. Далі було реалізовано функції шифрування і дешифрування, отримання цифрового підпису а також його верифікації. Після цього було створено два абоненти та функції отримання та надсилання ключа.

Результати:

```
P * Q 28925017287374420376016195091614286268727351285956929548859837884906475568696724313243627774380974315772415780439616365138164918374346716805903417938066291
P1 * Q1 45238403631692361163637664204262661346698907656978256479596250403857784892781049008181866666135387422686950473758873269916995807978658264813780556014934453
Ваше сообщение: 86885263700952550173549362185149553604953220028219851919769110603772306934827
C1: 12976392415616273838699858686543235156925255366319936644063723443314698026470305819189801917150960891806192746852636305447912029374271200538469746718875803
S1: 3353764134590625090056122914253731630893310021665700006423411742720605137186140073744948452713218137770242530316485321078050984635860044842721051915617681
Получено: 86885263700952550173549362185149553604953220028219851919769110603772306934827
Подтверждено? - True
```

```
Open key ['10001', '85918D25C824C820D8A82F4256BCD88170971F9D2108D1B1280BA40F4094AF5D']
Message 'hochveshku'
Ciphertext 21573bc7b41eacd583d5adc2f89c765dac43bf76dabc59f660b6ead82ddfb9b2
sign is True
```

Get server key

 Clear

Key size

256

Get key

Modulus

85918D25C824C820D8A82F4256BCD88170971F9D2108D1B1280BA40F4094AF5D

Public exponent

10001

Encryption

✖ Clear

Modulus

85918D25C824C820D8A82F4256BCD88170971F9D2108D1B128C

Public exponent

10001

Message

hochueshku

Text ▼

Encrypt

Ciphertext

21573BC7B41EACD583D5ADC2F89C765DAC43BF76DABC59F6A

Decryption

✖ Clear

Ciphertext

21573BC7B41EACD583D5ADC2F89C765DAC4

Text ▼

Decrypt

Message

hochueshku

Sign

✖ Clear

Message

hochueshku

Text ▼

Sign

Signature

36337E1F0B1CEBDC4BCE8A39073AF2A3252236BC211594C327

Verify

✖ Clear

Message

hochueshku

Text ▼

Signature

36337E1F0B1CEBDC4BCE8A39073AF2A3252236BC211594C327

Modulus

85918D25C824C820D8A82F4256BCD88170971F9D2108D1B128C

Public exponent

10001

Verify

Verification

true

✓

Висновок: в ході виконання цього комп'ютерного практикуму мною був досліджений метод побудови криптосистеми RSA.

Для досягнення результату були реалізовані ймовірнісні тести на простоту чисел, реалізовані методи вибору ймовірно простих чисел у проміжку. Були вдосконаленні знання щодо організації засекреченого зв'язку й електронного підпису за допомогою RSA, були проаналізовані протоколи розсилання ключів.