

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"  
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

КРИПТОГРАФІЯ  
Комп'ютерний практикум

Робота №4

Виконали:  
Сернова А.Р., Колесник А.М.  
студенти групи ФБ-93

Перевірила:  
Селюх П.В.

Київ-2021

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

**Мета 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.

Постановка задачі:

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.
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.
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  $0 < k < n$ .

**Перевірка за допомогою сайту:**

Шифруємо згенерований нашою програмою ключ за допомогою сайту та розшифровуємо самі. Результати зюігаються:

```
C:\Users\koles\AppData\Local\Programs\Python\Python39\python.exe C:/Users/koles/OneDrive/Документы/GitHub/crypto-F
encrypted 3C4060F9A767336FBC1A91118C6AF76E17E26225C7A5DFAD56C122618F690C535063F74984E9B96454BF84561339AA1128989FB6
decrypted hello
```

## Encryption

Modulus

8979534148FA4715556C962854C377897267DE83D4365E15BEB

Public exponent

86479834A6A3521B60DD269BECD6CB59E6BE02C0E12044629C

Message

hello

Text

Ciphertext

3C4060F9A767336FBC1A91118C6AF76E17E26225C7A5DFAD56

Перевірка ЦП:

- 1) Згенерований на сайті

## Sign

Message

signcheck

Text

Signature

A440107EE7450AA139F2AA975906D4C1AAB4221E830100F4708664DBCD054E8D

```
C:\Users\koles\AppData\Local\Programs\Python\Python39\python.exe C:/Users/koles/OneDrive/Документи/GitHub/crypto-FB-9/cp4/kolesnyk_fb-93_sernova_fb-93_cp4/main.py
A440107EE7450AA139F2AA975906D4C1AAB4221E830100F47086640BCD054E8D
signature: signcheck

Process finished with exit code 0
```

## 2) Згенерований програмою

### Verify

Message	verificationcheck	Text
Signature	67435326D80887A4AFA83BEAD2C7D0796E4DFE70E2C8BDC767EAECD5D686B98C6632B4E576CBCBF36E991	
Modulus	8979534148FA4715556C962854C377897267DE83D4365E15BEB39A5097795FA0BF540057D69F912A459D702	
Public exponent	86479834A6A3521B60DD269BECDC6B59E6BE02C0E1204462902CFF4FD6AA340716430C47C56D76EE6841E	
	<input type="button" value="Verify"/>	
Verification	true	

```
C:\Users\koles\AppData\Local\Programs\Python\Python39\python.exe C:/Users/koles/OneDrive/Документи/GitHub/crypto-FB-9/cp4/kolesnyk_fb-93_sernova_fb-93_cp4/main.py
verificationcheck
modulus 8979534148FA4715556C962854C377897267DE83D4365E15BEB39A5097795FA0BF540057D69F912A459D702233200910E04B79CA05837F6C7CB91440E2C0830B
expon 86479834A6A3521B60DD269BECDC6B59E6BE02C0E1204462902CFF4FD6AA340716430C47C56D76EE6841863F977876909A5FF038A70797986878DA06957AFB075
signature 67435326D80887A4AFA83BEAD2C7D0796E4DFE70E2C8BDC767EAECD5D686B98C6632B4E576CBCBF36E99DDF187BB49F656ED3160DE5ACD101F2E5AF363E3B8072

Process finished with exit code 0
```

**Висновки:** виконавши лабораторну, ми практичним шляхом ознайомилися з асиметричною криптографією на прикладі RSA, дізналися про різні тести перевірки на простоту, реалізували тест Мілера-Рабіна. Було організовано протокол конфіденційного розсилання ключів із підтвердженням справжності по відкритому каналу.