

# **Лабораторна робота №4**

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

### **Варіант 16**

#### **Виконали:**

Борщевський Олександр(ФБ-03)

Ржевський Андрій(ФБ-03)

#### **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.

#### **Порядок виконання**

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  $p_1, q_1$  довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб  $pq < p_1q_1$ ;  $p$  і  $q$  – прості числа для побудови ключів абонента А,  $p_1$  і  $q_1$  – абонента В.
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 А і В – тобто, створити та зберегти для подальшого використання відкриті ключі  $(e, n)$ ,  $(e_1, n_1)$  та секретні  $d$  і  $d_1$ .
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  $M$  і знайти криптограму для абонентів  $A$  і  $B$ , 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  $0 < k < n$ .

### Хід роботи

Генеруємо 256-бітне число та перевіряємо його на простоту за допомогою алгоритма Міллера-Рабіна.

Кандидати на  $p$  і  $q$  що не пройшли перевірку були винесені в окремий файл `discarded candidates.txt`

Рахуємо  $n = pq$ , функцію Ойлера  $\phi(n)$  та обираємо  $2 < e < \phi(n) - 1$  таке що  $\gcd(e, \phi(n)) = 1$ . Потім знаходимо секретний ключ  $d$  за формулою  $ed = 1 \bmod \phi(n)$

$n$  –

```
0x14135765b3dc519cbcd0e410bf6aecb865768570ee17ca87e37d603d9d7720a916ef63ccca66641d085047a112851ae6619ffcf241760f99b52e07f1001e872d3eef
```

$e$  –

```
0xc7989dc29b69d2eb9d713d1c04ebc4fa2fe026c75a17683e2a4f6d894ede7e4e11410ae7ede2df319c900ead91229f11f7a41fd00d98d5d989189282dc5406e6c05
```

$d$  -

```
0xa7ba32ea880894da5e84ef6c0de3b78888bdab894695413ab329e0117e3318f829f823a78bcc277a2e5abe0d67fe531d540632c236955d8e65cd1cb5d6429324d4d
```

$n1$  -

```
0x9aaec883778ff000d7c6708cee64cffff5faaa14179cbaa37b36c57c0cbb27e93e3d49dfc824837e7b559555fbcc6e36b191e4931e1166218d2b39f1fce28dd8973
```

$e1$  -

```
0x3ade208365176e0edad1e7beb98623a17ce8d117e4d2cf57548080ecf68043fcc5859d64dbc139038e738d68c1f03bb90ed97b486f60c4013b9b31e96b6ed83a545
```

$d1$  -

```
0x8281a5fb959932907d1743c09e6dc5a71f5523a26fd1274e16c9ef63ef43b98a8c57f64baa171b066ef6b2beecb8d7e07978c4ddf21e5eaaf4266299a8ca369f435
```

Будемо шифрувати один байт:  $M$  - 0x33

Зашифруємо байт за допомогою формули  $C = M^e \bmod n$

encrypted -

0x798f38b969e4264fae8407d94f1a2ffb367c405b6ff9a2cc03f3c49453cdd90a735282  
6645782cd703d4af14b1ff8251981a5bcb23619304418766adead8120647b

encrypted 2 -

0x78571a7c6404185324ea42b9747b271f6cb48cd7e3e4a38811fb2788a485940b467b  
c89a89b9af2e0257852f195ee025752ced01d9f2033d246f571850698e8143dc

Розшифруємо за допомогою формули  $M = C^d \bmod n$

decrypted - 0x33

decrypted 2 - 0x33

Підпишемо повідомлення за допомогою формули  $S = M^d \bmod n$

sign -

0x1a972dad74e4effdbfe8276ccf9b1a0dc60d069263f77c52166589121985d352be303  
cca9eabd5b8e488ee5249db7de0600e3238bcccabf14450a5c50a39010c214

sign 2 -

0x6cbf06953d92ee0398742f14dfb89104e6024a4c387972c4cd542d1611da1ea818708  
f2098fb705ea99730970ccd26557677c75937fe7dcae4cd70f4eae9b2b1f6

Перевіряємо повідомлення (M, S) за формулою  $M = S^e \bmod n$  – перевірка виконана успішно

Для передачі секретного ключа  $k$  спочатку абонент А створює  $k_1$  та  $S_1$  за допомогою формул  $k_1 = k^{e_1} \bmod n_1$ ,  $S_1 = S^{e_1} \bmod n_1$  та  $S = k^d \bmod n$

S -

0x79bf2bbb5c6a43f9c1f75c8df03c2082c6913911b29871e5db6582844dbb1a0b6c32a  
30899bf47e5a55f3487d4b5867b53b481393418831382caf4df7e11ea4e782

k1 -

0x651cd1a78c96bd898cb62edc4f686bec4d3d511368559a81c720d956d4d5285b6c77  
c107a9872ee55ec08d952223aee818ee7828d2c20744bec6e9112cbd00281bd8

S1 -

0x44c9abcb0e31b130f80a0f162cfa1e99bea365101b28222772407379c2b5f0d5cd67e  
9d038ead88213881c7a6b55aec8f6c94fe671c761e0cf1c6409f0050c312972

Потім за допомогою формул  $k = k_1^{d_1} \bmod n_1$ ,  $S = S_1^{d_1} \bmod n_1$  та  $k = S^e \bmod n$

k -

0x8d848e784118ee26cbd2681dd78ae00d9258ae02a2dfec24fbe6f2d8e3d0bfbc880b4  
36b297d6810b8db689bac4e74b79a3b610a89010fedeba523cbaea1b54c46e

S -

0x79bf2bbb5c6a43f9c1f75c8df03c2082c6913911b29871e5db6582844dbb1a0b6c32a30899bf47e5a55f3487d4b5867b53b481393418831382caf4df7e11ea4e782

k -

0x8d848e784118ee26cbd2681dd78ae00d9258ae02a2dfec24fbe6f2d8e3d0bfbc880b436b297d6810b8db689bac4e74b79a3b610a89010fedeba523cbaea1b54c46e

Бачимо що оба ключі однакові отже ми знайшли таємний ключ и перевірили його аутентичність.

**Висновок:** Під час виконання лабораторної роботи ми ознайомились з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практично ознайомились з системою захисту інформації на основі криптосхеми RSA, організували з використанням цієї системи схеми засекреченого зв'язку й електронного підпису, вивчили протокол розсилання ключів і закріпили навички набуті у попередніх лабораторних роботах.