

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

Виконали  
студенти групи ФБ-11  
Комар Анастасія та Сергеев Максим

Київ - 2023

## 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 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.

На наступному зображенні наведено лише перші 5 чисел, які не пройшли перевірку (з рештою можна ознайомитись у файлі `not_prime.txt`):

```
1 71874997168256727025792994743330363452216136234090062908303335826025316268828
2 80230443528002765483126815768613376902746128745051615005817518459913667325897
3 65755694777918090261789808387952896591017620956219251750875169612677414661842
4 89979486158345431823752512976471437112056306893903411107386765513654293078125
5 106128478343114239808285222598653013291787848086835922167574476173002070215751
```

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  $pq \leq p_1q_1$ ;  $p$  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  $d_1$ .

```
Користувач А
-----
p = 86488600325159026677673396820065874999842948295813757719623949609959115180209
q = 59021894165996824752805627415433999171245557810786360963960688430086547645973
публічний ключ: (5104721014956734637003682149222970797280566996676986794703022913143762895734772114902647538179590609217564682528852346708278611893465884217491058628148357, 65537)
секретний ключ: 4632703538559618322551657890799619056561275054722632155706725877062599535370987175952269971950684042045083537494125107864116220521670610984436393154072769

Користувач В
-----
p = 60344932879038940942813042488201054808415899603033897519928033331147196075379
q = 871476113243562108026576594589542033498484789193328649029287948638108414999073
публічний ключ: (5258916755936849448922026184009969916972337283320889297917341228061125625145738303153180328647056991543016279911107014004879931977038364748340763523123667, 65537)
секретний ключ: 1059855844979384279435465795480471835197989392833091320122865601724695168483802484497663416659384386945829699229689291317346396341576524358854309420668417
```

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 А и В, перевірити правильність розшифрування. Скласти для А і В повідомлення з цифровим підписом і перевірити його.

Повідомлення: 2648844340165118743947750326754952145513710738440268820533557409050236641167436614750201289494999787541553192741067249891835680750305824763326371566593099

Зашифроване повідомлення за допомогою відкритого ключа A: 2826044530700854446706646615840999575958389645195906124008117534825225866416091214428042538837173414724647657976839345548065497477295321137582304488999809

Розшифроване повідомлення за допомогою секретного ключа A: 2648844340165118743947750326754952145513710738440268820533557409050236641167436614750201289494999787541553192741067249891835680750305824763326371566593099

Повідомлення з цифровим підписом: (2648844340165118743947750326754952145513710738440268820533557409050236641167436614750201289494999787541553192741067249891835680750305824763326371566593099, 384653117830915568231775165018173327711252864700726444510690071597214718195957908264429654073975054295809857628904619280329765337490640831571660839722317)

Цифровий підпис підтверджено, повідомлення не спотворено.

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  $0 < k < n$ .

Абонент  $A$  формує повідомлення  $(k_1, S_1)$  і відправляє його  $B$ , де

$$k_1 = k^{a_1} \bmod n_1, \quad S_1 = S^{a_1} \bmod n_1, \quad S = k^d \bmod n$$

$k = 4193394661591558627436448326013257137231961820769564364244148110833594745352777730156494829859735301352904499712161775204419339944097607759501446679553829$   
 $S = 2784086794673326855329492571509674861971981450955880282262620592627575639235159703392846184240407657382630492344405356856635999645151697338300991637720409$   
 $S_1 = 13428893995068955639057013581742491034653112614857477682555353972804158731409785087486166948134906007242426198590218605017121751419192870825925636966364$

Абонент  $B$  за допомогою свого секретного ключа  $d_1$  знаходить (конфіденційність):

$$k = k_1^{d_1} \bmod n_1, \quad S = S_1^{d_1} \bmod n_1,$$

і за допомогою відкритого ключа  $e$  абонента  $A$  перевіряє підпис  $A$  (автентифікація):

$$k = S^e \bmod n.$$

$k_1 = 2572363348145140522016036250475077580055405183144527798719823338426511363331393471185434739357552346922143534483245577198595181991381941944460718247729891$   
 $k = 4193394661591558627436448326013257137231961820769564364244148110833594745352777730156494829859735301352904499712161775204419339944097607759501446679553829$   
 $S = 2784086794673326855329492571509674861971981450955880282262620592627575639235159703392846184240407657382630492344405356856635999645151697338300991637720409$   
 Автентифікація пройдена.

## Висновки

Під час лабораторної роботи ми ознайомились з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA. Крім того, ми на практиці ознайомились з системою захисту інформації на основі криптосхеми RSA, організували з використанням цієї системи засекречений зв'язок й електронний підпис, вивчивши протокол розсилання ключів.