



Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Фізико-технічний інститут

## **ЛАБОРАТОРНА РОБОТА №4**

З предмету «Криптографія»

Виконала:

студентка 3 курсу ФТІ

групи ФБ-84

Даневич А.С.

Перевірив:

Чорний О.М.

## 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  $p, q$  і  $1 < p, q$  довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб  $1 < p < q$ ;  $p$  і  $q$  – прості числа для побудови ключів абонента А,  $1 < p < q$  – 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  $(e, n)$ ,  $(, )$  і  $n$  і секретні  $d$  і  $d$ .
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  $M$  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  $n < k < 0$ .

## Опис кроків протоколу

1. Створюємо відкритий та закритий ключі для абонента Аліса (функція `GenerateKeyPair()`), для цього генеруємо два простих числа довжиною 256 біт (функція `ChooseRandomNumber()`), потім знаходимо  $n$  за формулою,  $pq=n$ , далі рахуємо функцію Ойлера  $\varphi(n)=(p-1)(q-1)$ . Потім А обирає випадкове число  $e$ ,  $2 \leq e \leq \varphi(n) - 1$  таке, що  $\gcd(e, \varphi(n)) = 1$ , і знаходить для  $e$  обернений за  $\text{mod} \varphi(n)$  елемент  $d$  (функція `evk()`).

KeyA

d

1083576947286865914346266757837224866769416442832846446291467560818442138936517114052032069780758016558786241915153788651056516234382014878987516005852219

p 12542251002476200039828874090002144476941270939343795371820000047748366257471

q 93234869415884562231140041127186465361852437477448512441325008817785300351179

n

1169375134397115753889131639727302557316729168122277439421914630739023663038709638419317808455692018057516217685967811010330953645606955912022490232408309

e

30945248756386385567727070707117924109629180326172807044869981426651458014423431385990856881522258908157694447017639375205830195423599965805031845726019

2. Робимо те саме для абонента Боб, але щоб задовольнялась умова, що

$n_{\text{Аліси}} \leq n_{\text{Боба}}$

KeyB

d

1347620703301499837886320214952191016112291363594633008102894656298108883817764099060511339114515063934014334695980066353893648544973569261459218588924821

p 48252690735032940938470530081923506008861924231557980948932786541278780643019

q 43817159759091548569126680090252016860035540752090841165984560325081928569591

n

2114295858742974976021879084841052745603730115473037978654939572245771732889906494461866004821949522954307667843682463794442449281958202395218205869835229

e

1059297815974719671194660468423485224578290694540744740984187797815911630846355403399043758172263956824513211942309602037609180995085298388150545531629901

Всі числа, що проходили перевірку на простоту, і чому вони не підійшли можна знайти у файлі `test.txt`.

3. Аліса і Боб обмінюються своїми відкритими ключами (функція `KeyExchange()`)

Інформація яку знає:

Аліса	Боб
d 1083576947286865914346266757837224866769416442832846446291467560818442138936517114052032069780758016558786241915153788651056516234382014878987516005852219	d 1347620703301499837886320214952191016112291363594633008102894656298108883817764099060511339114515063934014334695980066353893648544973569261459218588924821
p 1254225100247620003982887409000214447694127093934379537182000047748366257471	p 48252690735032940938470530081923506008861924231557980948932786541278780643019
q 93234869415884562231140041127186465361852437477448512441325008817785300351179	q 43817159759091548569126680090252016860035540752090841165984560325081928569591
n 1169375134397115753889131639727302557316729168122277439421914630739023663038709638419317808455692018057516217685967811010330953645606955912022490232408309	n 2114295858742974976021879084841052745603730115473037978654939572245771732889906494461866004821949522954307667843682463794442449281958202395218205869835229

e	e
309452487563863855677270707071179241096291803261728070448699	1059297815974719671194660468423485224578290694540744740984
814266514580144234313859908568815222589081576944470176393752	1877978159116308463554033990437581722639568245132119423096
0583019542359965805031845726019	02037609180995085298388150545531629901
nb	na
211429585874297497602187908484105274560373011547303797865493	1169375134397115753889131639727302557316729168122277439421
957224577173288990649446186600482194952295430766784368246379	9146307390236630387096384193178084556920180575162176859678
4442449281958202395218205869835229	11010330953645606955912022490232408309
eb	ea
105929781597471967119466046842348522457829069454074474098418	3094524875638638556772707070711792410962918032617280704486
779781591163084635540339904375817226395682451321194230960203	9981426651458014423431385990856881522258908157694447017639
7609180995085298388150545531629901	375205830195423599965805031845726019

4. Аліса створює повідомлення, у якому передає шифрований текст, для цього обирається число  $k$  від 0 до  $n-1$ , та шифрує його за допомогою відкритого ключа Боба (функція `Encrypt()`), також повідомлення містить цифровий підпис, для створення якого використовує свій таємний ключ та відкритий ключ Боба (функція `Sign()`).

$k_1$

12644040015729198529298573452754729853279973338782363063375382326046955527130364718349  
93100043473816198932282166333646738200123879140420998428326445213696

$S_1$

56218967298848408970022418873637154574607367114194668365447203627548473506167297976597  
0970244285310347067367156066940359412995278035982227300821536827042

5. Боб отримує повідомлення від Аліси і за допомогою свого відкритого ключа розшифровує повідомлення  $k$  (функція `Decrypt()`), а за допомогою свого таємного та Алісиного відкритого ключів перевіряє підпис.

$C$

123456789

$S^e \bmod(n)$

123456789

Оскільки  $k = S^e \bmod(n)$  то автентифікація пройдена успішно.

## Робота з сайтом

1. Надішлемо сайту запит на отримання його відкритого ключа розміром 512 біт, отримуємо:

Get server key

Clear

Key size

Get key

---

Modulus

Public exponent

2. Генеруємо ключ довжиною 256 біт, так щоб  $n \leq \text{Modulus}$ .
3. Інформація, яка нам відома

d

0x14b069ee28addf7084b0bb26d617202dec286aa7143c476310576bd1ffaa72bf0c709015f50b4237306b00295440092cd5a7a7344fe132bc56f94af741c0243b

p 0x1bbaa9999e56254fc8104d9d776bf8bdd53afb18e2253ac42f9922c5a998cd3f

q 0xce2110e6f599a52f486fb00834f2c374423e8916de2f8bcf33285acde41c78cb

n

0x1653c95e291efdce75c63ce4b0374d36488e05f2ccad78d917500c6f968619f20f91d0b65b31c6efe176615abd23757c888215351d01993685c25f32ab4448f5

e

0x9741d9817da69cd3b3b017afd2ed7c143fcf64b8c7b7592810aafc387c7f87f6bc4374ee2911cb50aa09563849b3d5f6c25f23a18fea6788e23d03cdcf0743

Modulus

0x9dae58a50dfe0eb760377ab7ee3d7f7923683dc9f92aea498d6ffe7a9b9cad53fb3e715d2a8ec7aa19dc1057a65bfbedc40befba9ac05b414ac7f4012020817b

Public exponent

0x10001

4. Створюємо повідомлення, у якому передаємо шифрований текст, для цього обирається число  $k$  від 0 до  $n-1$  (наприклад 123456789), та шифруємо його за допомогою відкритого ключа сайту (функція `Encrypt()`), також повідомлення містить цифровий підпис, для створення якого використовуємо свій таємний ключ та відкритий ключ сайту(функція `Sign()`).

Отримаємо:

$k_1$

0x603794a75d9fdb797846d46377270c4b694091b473dd11635028bf2e52498ea5a5da29806c9676405dba35065211aa76f655a51b4865b758668e566bf6ef2311

$S_1$

0x40afbfe5969c53adbdcc79167837c2dd6587170d0eefab98c986c4ce472ba94ef58ffaeb55bbd66c44857411a3538585fc7559056cd65a81b9a3da34ba5b446

5. Вводимо ці дані та свій відкритий ключ на сайті

## Receive key

✖ Clear

Key

603794a75d9fdb797846d46377270c4b694091b473dd11635028bf2

Signature

40afbfe5969c53adbdcc79167837c2dd6587170d0eefab98c986c4c

Modulus

1653c95e291efdce75c63ce4b0374d36488e05f2ccad78d917500c61

Public exponent

9741d9817da69cd3b3b017afd2ed7c143fcf64b8c7b7592810aafc38

Receive

Key

075BCD15

Verification

true

✓

**Висновок:**

У ході виконання комп'ютерного практикуму №4, я ознайомилась із поняттям псевдопростих чисел, тестами перевірки на простоту, був реалізований тест Міллера-Раббіна (функція `test()`). Також практично реалізувала протокол передачі ключів RSA із використанням функції генерації ключів, цифрового підпису, за шифрування та розшифрування повідомлення.