Міністерство освіти і науки України Національний
технічний університет України «Київський
політехнічний інститут імені Ігоря Сікорського» Фізико-
технічний інститут

**ЛАБОРАТОРНА РОБОТА №4**

з **дисципліни**

**«Криптографія»**

**на тему: «Криптоаналіз афінної біграмної підстановки»**

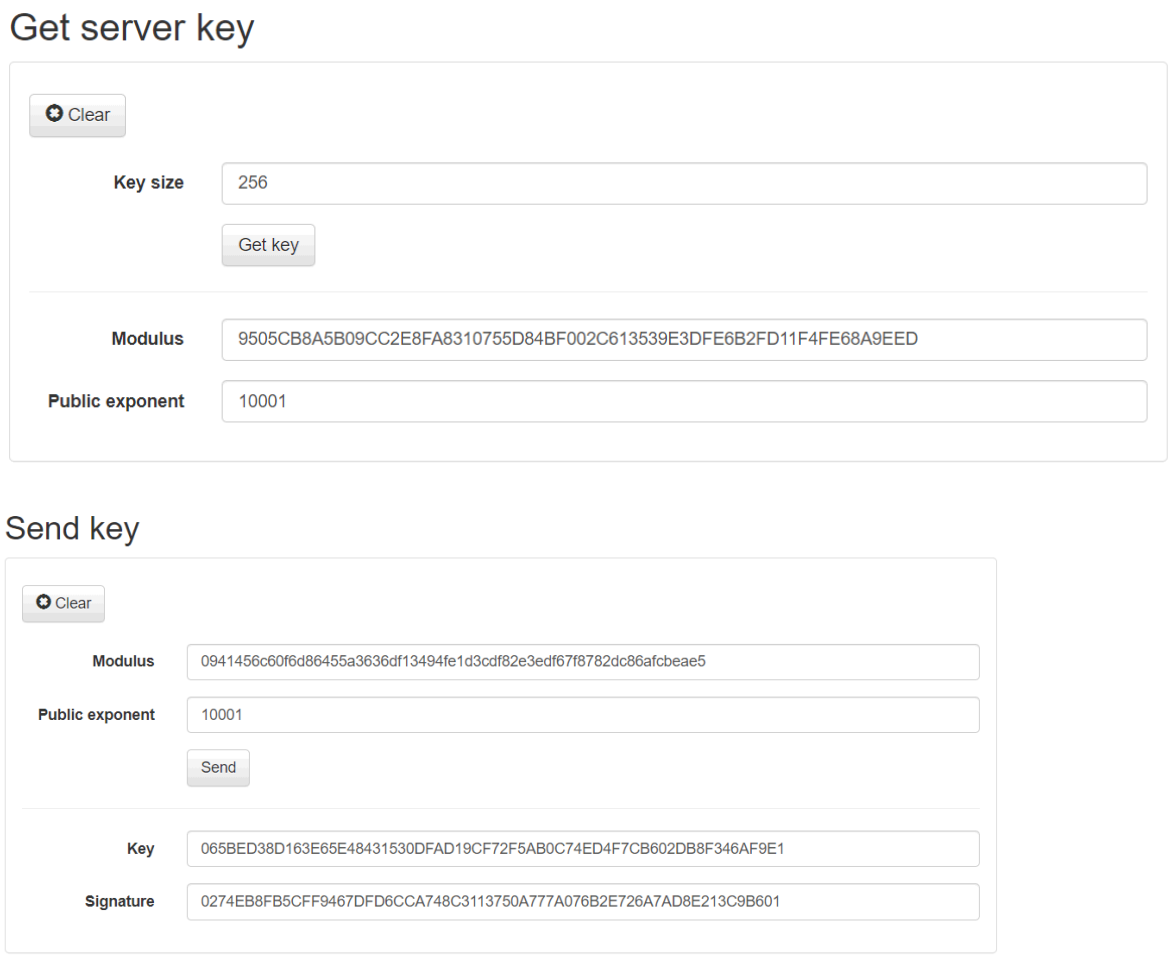
Виконали:
студенти 4 курсу ФТІ
групи ФБ-72
Давидюк Петро, Башкиров Ігор
Перевірили:
Чорний О.
Савчук М. М.
Завадська Л. О.

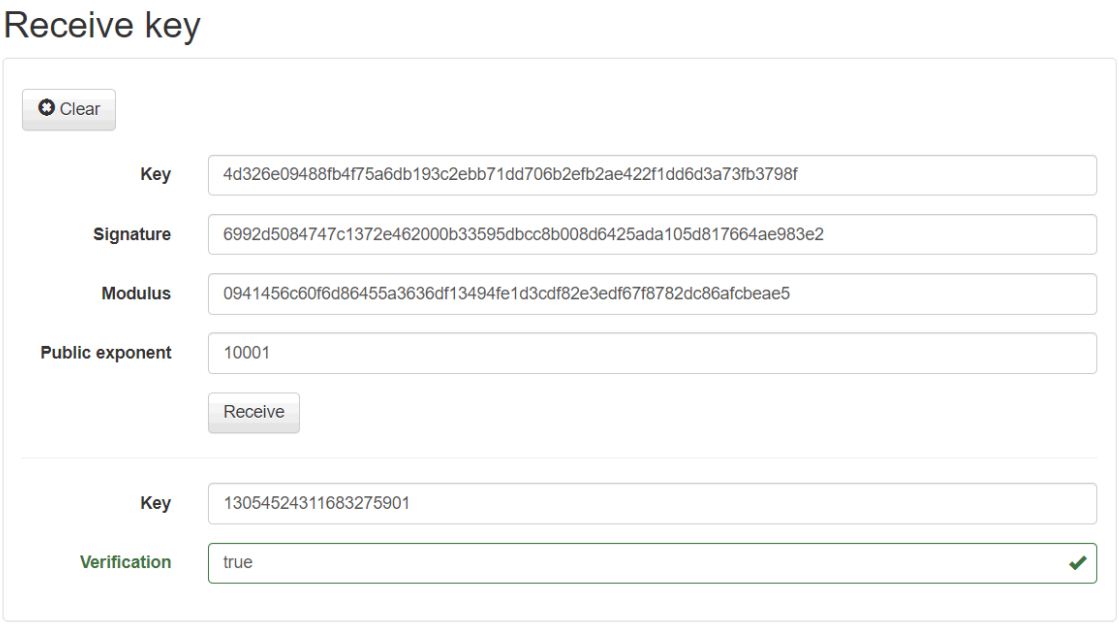
**Мета роботи:**
Ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практичне ознайомлення з системою захисту
інформації на основі криптосхеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, 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 МіллераРабіна 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 p1, q1 довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq p1 q1 ; p і q – прості числа для побудови ключів абонента A, p1, q1 – абонента B.
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 A і B – тобто, створити та зберегти для подальшого використання відкриті ключі (e,n) , (e1, n1) та секретні d і d1.
4. Написати програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів A і B. 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 и 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. 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:

```
C:\Users\Xiaomi\Desktop\KPI studying, lol\Kurs 3 sem 1\Kripta\lab4\ConsoleApp3\ConsoleApp3\bin\Debug\netcoreapp3.1>ConsoleApp3.exe
p and q generated
p = 6fd36d47f7544f651be15a8ef10635bb
q = 152fe8afd4aa92e32f9982fad39107df
module = 0941456c60f6d86455a3636df13494fe1d3cdf82e3edf67f8782dc86afcbeae5
public exp = 10001
private exp = 24540563743896496251171088446751392601543118706967694713067704716045843215 3
p and q generated
p = 33e0abc8649b4452cfba515e2d059feb
q = 7b3f07b7a07be7ee4188e594e00dd463
module = 18f9b863e97a2a598f5e70b55dccda045d381aa1f4feee7ed8c254c6d08a73e1
public exp = 10001
private exp = 749006867168351277984535308767554694411168233541756135674135266706567190227 7
0a9ba0d
20d37d6cff11081b0a3910222b4a04e9dd9c78a42e3001d5f11df65ae887b0d
0a9ba0d
08dd9b6ccc4f96aa9089cf6cfcd31c8e15c3e19b95b320ffc11c8c1220201d0e
True
Enter modulus from site
9505CB8A5B09CC2E8FA8310755D84BF002C613539E3DFE6B2FD11F4FE68A9EED
Enter Key from site
065BED38D163E65E48431530DFAD19CF72F5AB0C74ED4F7CB602DB8F346AF9E1
Enter signature from site
0274EB8FB5CFF9467DFD6CCA748C3113750A777A076B2E726A7AD8E213C9B601
Verification Success
Key = 13054524311683275901
Enter modulus from site
9505CB8A5B09CC2E8FA8310755D84BF002C613539E3DFE6B2FD11F4FE68A9EED
Enter Key to send
13054524311683275901
Key = 4d326e09488fb4f75a6db193c2ebb71dd706b2efb2ae422f1dd6d3a73fb3798f
Signature is 6992d5084747c1372e462000b33595dbcc8b008d6425ada105d817664ae983e2
```






## Висновок

Під час виконання лабораторної роботи №4 ми ознайомились з асиметричною криптосистемою RSA, генерацією ключів для цієї криптосистеми, а також з тестами перевірки чисел на простоту. У ході роботи реалізували функції шифрування, розшифрування, підпису, перевірки підпису для RSA. RSA алгоритм з відкритим ключем, що часто використовується в криптографічних застосунках. Складність задачі цього алгоритму полягає у великих цілих простих числах.