Міністерство освіти і науки України

Національний технічний університет України

Київський політехнічний інститут імені Ігоря Сікорського

Фізико-технічний інститут

**ЛАБОРАТОРНА РОБОТА №4**

**з дисципліни**

**Криптографія**

Виконали:

студенти 4 курсу ФТІ

групи ФБ-72 та ФБ-73

Морозов Артур

Сницін Максим

Перевірили:

Чорний О.

# Криптоаналіз афінної біграмної підстановки

**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:**

*Абонент А:*
q = 3004447814223547511943204080090079454 79
p = 3275299377421010327997695139411987837 31

публічний ключ (exponent):
10001

modulus:
d98f1069d6ee7e2ef4e9e61e449f26cbc3b329261da5bc62781f54235851f7a5

секретный ключ (private exponent):
5a6fd127de13b7da772ebfdcf7339d61346e218696b45117764d4ac21ac808e5

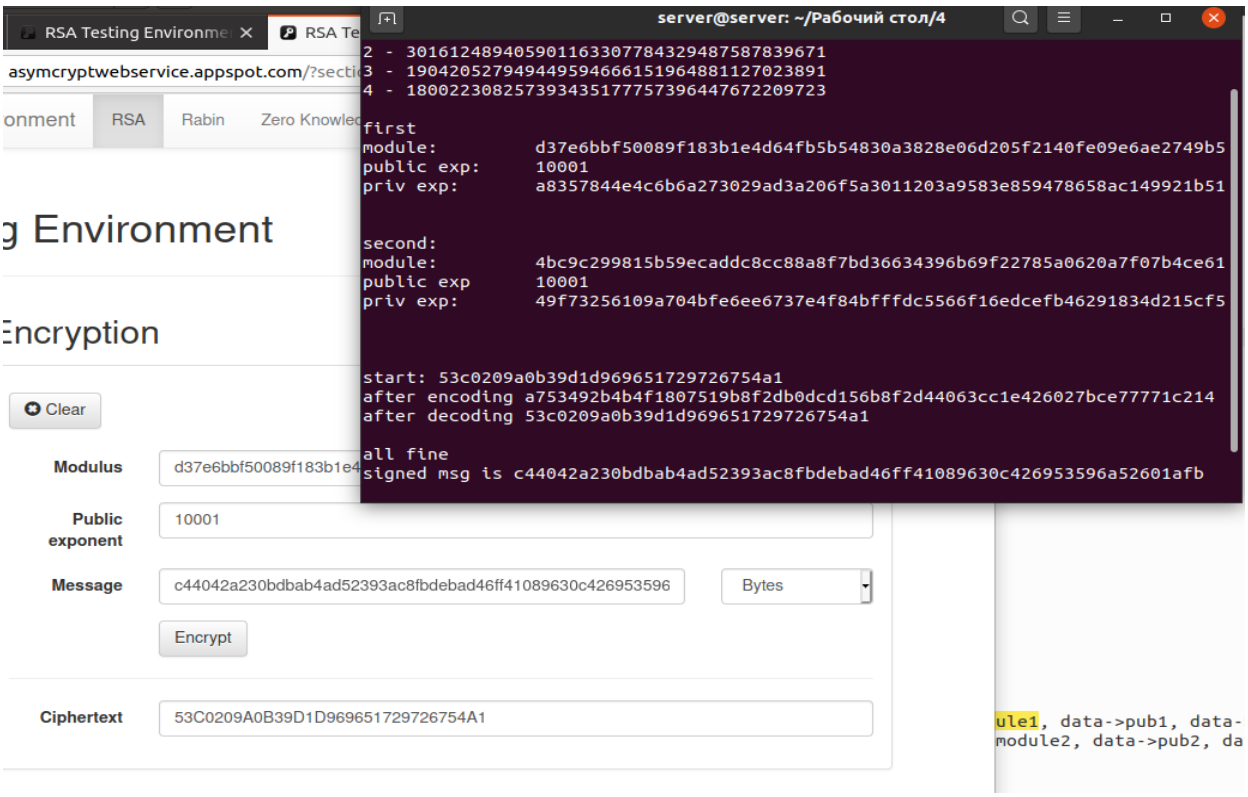
*Абонент В:*
q = 2304136934202476451712383661774933267 07
p = 2658411594517872129006989529333032924 63

публічний ключ (exponent):
10001

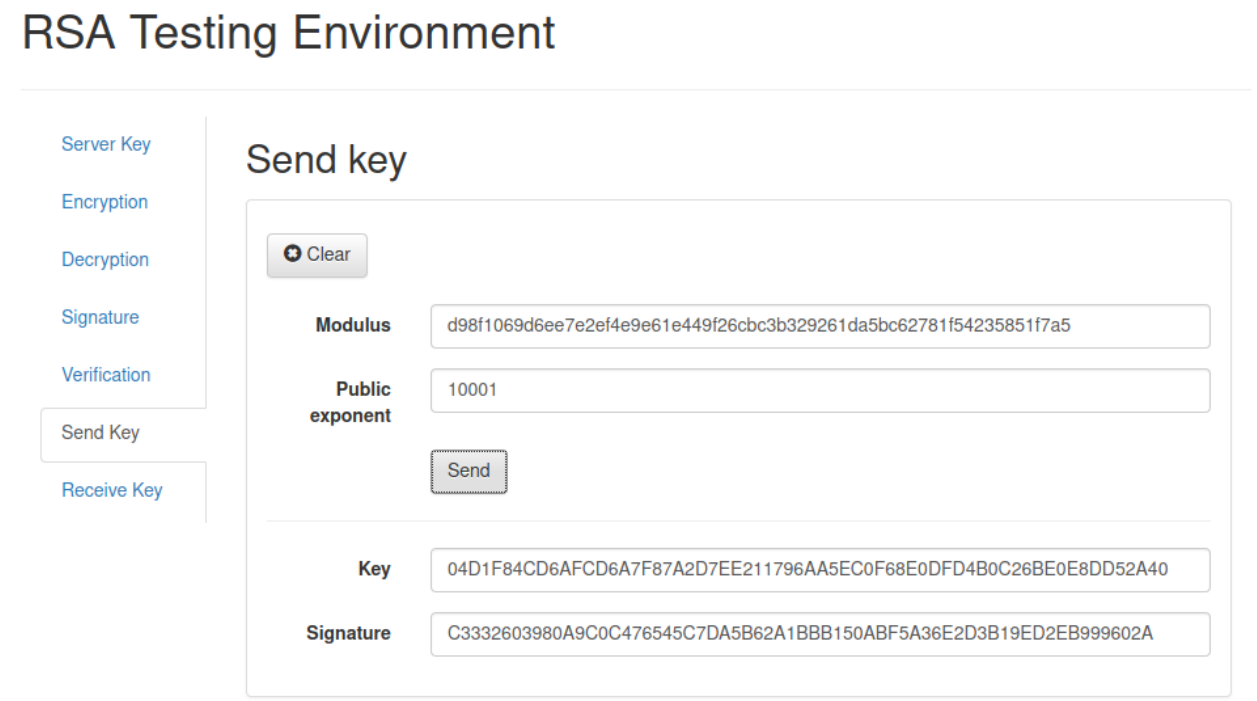
modulus:
876c386a8f93d5c1f1d3db50256931df0af9d1967ad72e0973bc45bccfd1a01d

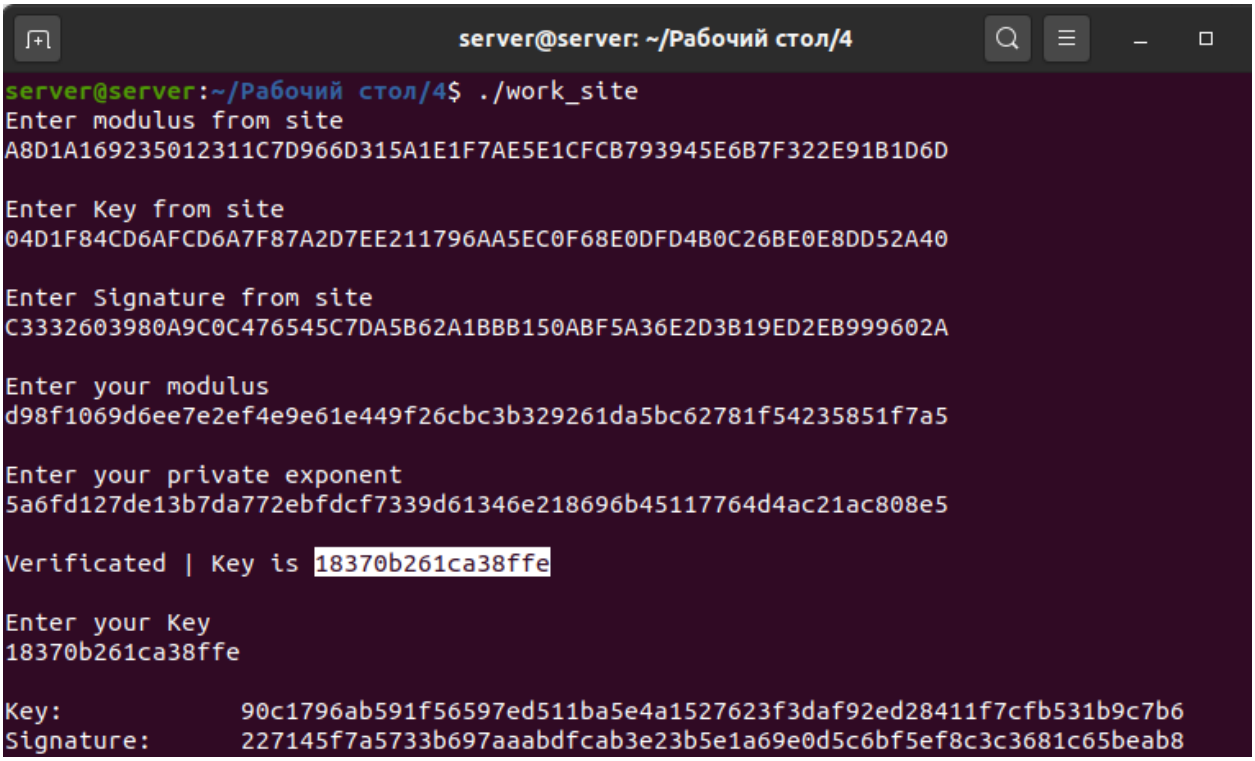
секретный ключ (private exponent):
4fc93ee3e97b9c310c1d59d53a7563dfbd4d74f25fe73ff97e0adab8483076b1

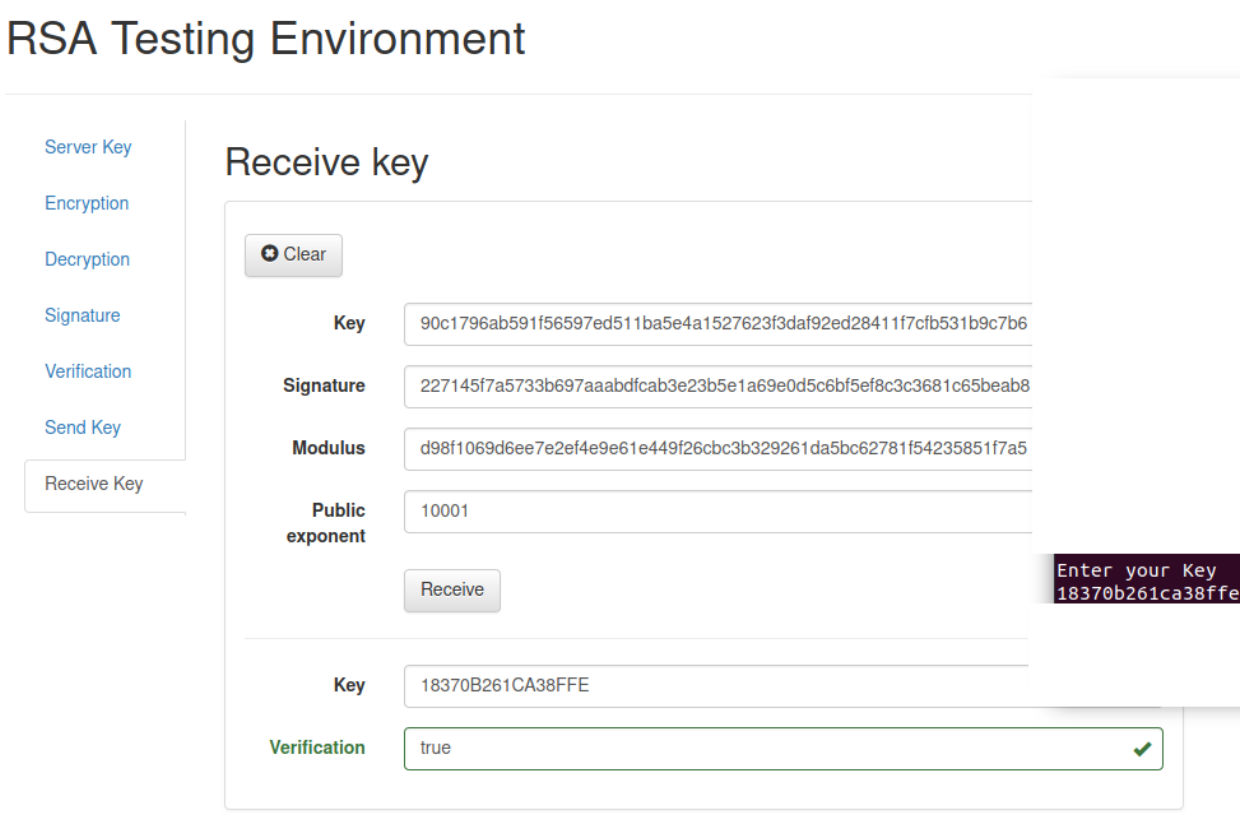
1. Перевірка зашифрованого повідомлення:



2. Відправка та отримання ключа:






---

**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.