Національний технічний університет України

«Київський політехнічний інститут»

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-32

Грабовецький Микита

Драбок Алла

Київ - 2025

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.

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 $pq \leq p1q1$; p і q – прості числа для побудови ключів абонента A, p1 і 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 $0 < k < n$.

Варіант: 13

Хід роботи:

1. Генерація простих чисел:

Для генерації простих чисел p та q використовувався імовірнісний підхід. Спочатку генерувалося випадкове непарне число заданої довжини біт. Першим етапом число проходило тест пробних ділень на малі прості числа (3, 5, 7...). Якщо число проходило цей етап, до нього застосовувався тест Міллера-Рабіна з кількістю раундів k=40, що забезпечує ймовірність помилки не більше 4^(-40).

2. Реалізація RSA:

- Генерація ключів: n = p*q, функція Ойлера φ(n) = (p-1)(q-1). Випадкове число e=2^16+1 (зазвичай обирається). Секретний ключ - для e обернений за modφ(n) елемент d: ed ≡1(modφ(n)) => d = e^(-1) (mod φ(n)).
- Шифрування/Розшифрування: C = M^e (mod n), M = C^d (mod n).

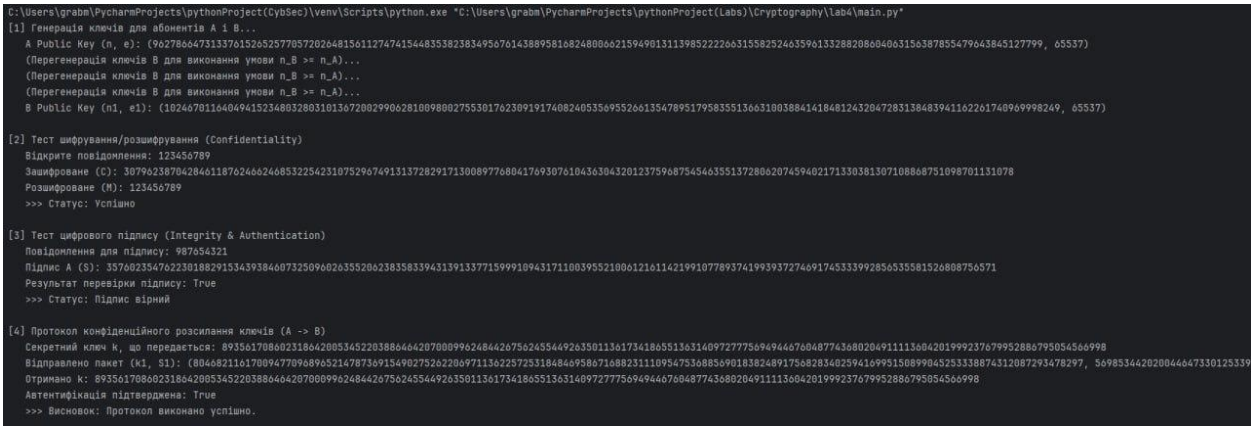
3. Цифровий підпис:

S = M^d (mod n), перевірка M = S^e (mod n).

4. Протокол конфіденційного розсилання ключів:

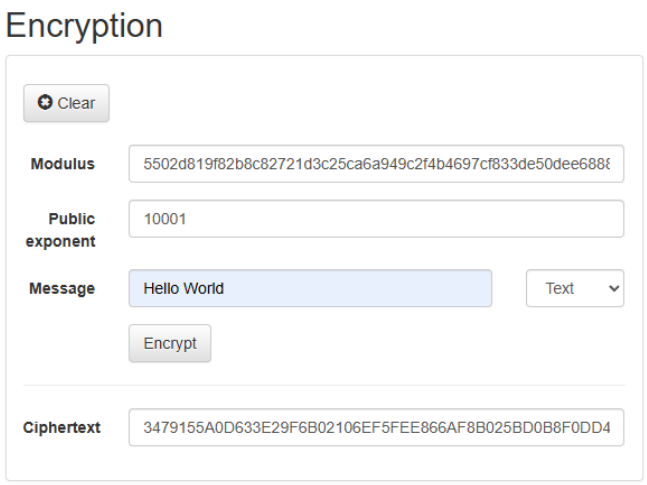
Схема передачі секретного ключа k від користувача A до B:

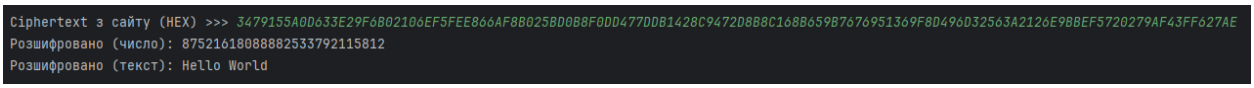
- Перед початком перевіряється умова n_B >= n_A. Це необхідно, щоб при операціях за модулем n_B не відбулася втрата інформації, якщо значення зашифрованого повідомлення або підпису перевищуватиме n_B.
- Крок відправлення (A):
    1. Абонент A підписує ключ k своїм секретним ключем: S = k^d_A (mod n_A)
    2. Абонент A шифрує ключ k відкритим ключем B: k1 = k^e_B (mod n_B).
    3. Абонент A шифрує свій підпис S відкритим ключем B: S1 = S^e_B (mod n_B) (забезпечення конфіденційності підпису).
    4. Відправляється пакет (k1, S1).
- Крок отримання (B):
    1. Абонент B розшифровує k1 своїм секретним ключем: k = k1^d_B (mod n_B).
    2. Абонент B розшифровує S1 своїм секретним ключем: S = S1^d_B (mod n_B).
    3. Абонент B перевіряє автентичність ключа k, використовуючи відкритий ключ A: k = S^e_A (mod n_A).



Перевірка роботи алгоритму

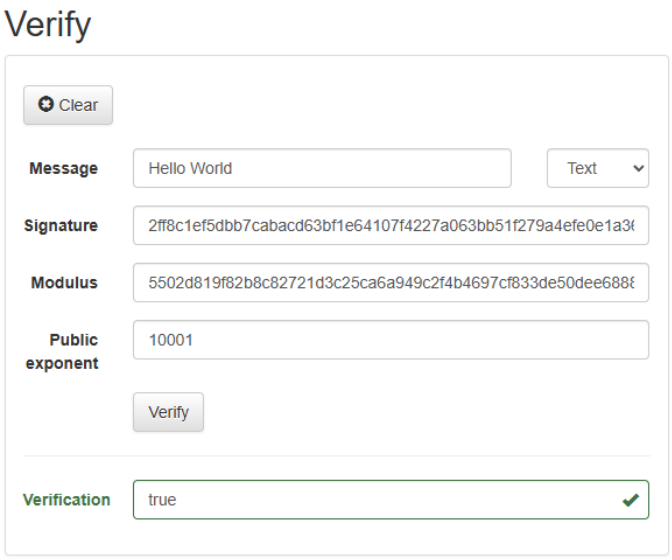
Перевірка відбувається за допомогою згенерованих ключів у скрипті та за допомогою сайту asymcryptwebservice.appspot.com





Публічний ключ «передається» сайту, де ми шифруємо повідомлення Hello World за його допомогою. Отриманий шифротекст вставляється у код й розшифровується за допомогою приватного ключа.

```
    [2] ПЕРЕВІРКИ ПІДПИСУ
Message (Text): Hello World
Signature: 2ff8c1ef5dbb7cabacd63bf1e64107f4227a063bb51f279a4efe0e1a36cf24c72fb06f2e0fe110d426495b059f2b51062298d518e7e77cf0d80b8e2d5fad494d
Modulus: 5502d819f82b8c82721d3c25ca6a949c2f4b4697cf833de50dee688862af12e889f986fc5d0be3ebc40cc6611a1baff98c332f49963ba0fb8e17547d5267e823
Public Exponent: 10001
```



Для перевірки підпису, код проводить математичне перетворення повідомлення за допомогою приватного ключа й разом з публічним ключем передається на сайт.

Зі скріншоту можна зробити висновок, що підпис дійсний й код успішно пройшов тестування.

Висновки:

Було реалізовано криптосистему RSA з повним набором необхідних процедур: генерація ключових пар, шифрування та розшифрування повідомлень, створення і перевірка цифрового підпису, а також протокол конфіденційної передачі ключів між абонентами.

На тестових повідомленнях програма коректно виконала всі операції: повідомлення 123456789 після шифрування та розшифрування співпадає з оригіналом, а підпис для 987654321 успішно підтверджено відкритим ключем.

Протокол передачі секретного значення k також виконано успішно – отриманий ключ співпав із відправленим, а автентичність підтверджено.