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

КРИПТОГРАФІЯ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4

Експериментальна оцінка ентропії на символ джерела відкритого тексту

Виконали:
Студенти 3 курсу
Остапова О. А.
Литвин М. Р.

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.

Постановка задачі

Реалізувати криптосистему RSA на мові Python, яка забезпечує безпечний обмін інформацією між двома абонентами. Система повинна виконувати наступні функції: генерацію великих 256-бітних простих чисел із застосуванням тесту пробних ділень та тесту Міллера-Рабіна; формування ключових пар для абонентів A та B; шифрування та розшифрування повідомлень; створення та перевірку цифрового підпису із використанням хеш-функції SHA-256; а також реалізацію протоколу конфіденційної передачі ключів із автентифікацією для забезпечення достовірності та конфіденційності обміну.

Варіант 10

Хід роботи

Основні формули, які використовувалися при виконані роботи:

1. Генерація простих чисел:
   - шукаємо p, q такими що:
     $n = p \cdot q$
     p, q — прості
   - перевірка: тест Міллера-Рабіна

2. Функція Ейлера:
   $\phi(n) = (p - 1)(q - 1)$

3. Вибір відкритого експонента:
   $2 < e < \phi(n)$
   $\gcd(e, \phi(n)) = 1$

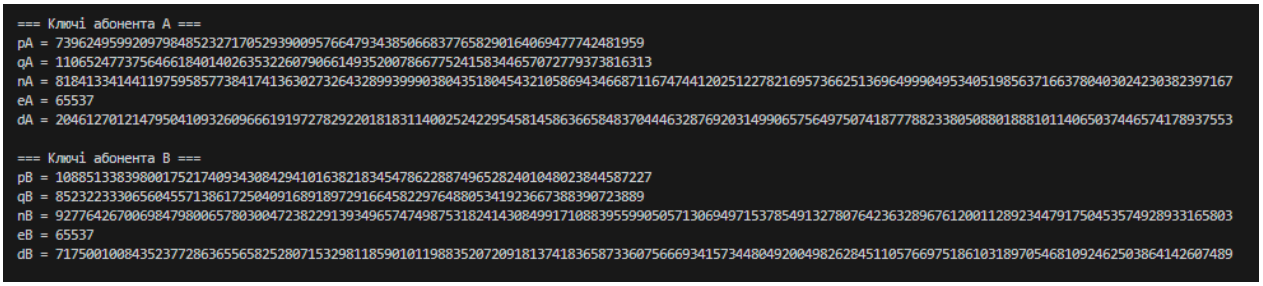
4. Секретний експонент:
   $d \equiv e^{-1} \pmod{\phi(n)}$
   $e \cdot d \equiv 1 \pmod{\phi(n)}$

5. Відкритий ключ: (n, e)
   Приватний ключ: (d, p, q)

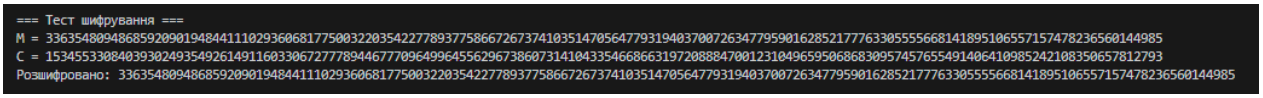
Генерація двох ймовірних прости числа p і q за допомогою тесту Міллера-Рабіна:

```
Число 64008251336882740207846060329148097258456861435818959414505150783227163059847 не пройшло тест Міллера-Рабіна
Число 75304900295845273038318311793080285360215180764304146168572453454785100000833 не пройшло тест Міллера-Рабіна
Число 10671762230933393754235486174915329942885009951212951668048920577394458214229 7 не пройшло тест Міллера-Рабіна
Число 75577176074447184103126670417990673703079476102787561773651532317071230872733 не пройшло тест Міллера-Рабіна
Число 10052330682914625594722827583891598422980089161491335967355750060941493266308 9 є ймовірно простим (пройшло Міллер-Рабін)
Число 11212556849712993524974382829045008984396009778668284890214216079532016654999 9 не пройшло тест Міллера-Рабіна
Число 67520388863873814305604770459775274490353681819309789274082633628090243603009 не пройшло тест Міллера-Рабіна
Число 93353745628309843510583617024147563207296909345817155591100318608611375338197 не пройшло тест Міллера-Рабіна
Число 59763449617338679511788724901360182215185035395417037354383294576307068917373 є ймовірно простим (пройшло Міллер-Рабін)
Число 11041987906363192741479137855335484805086948862557717006185519725949520925250 1 не пройшло тест Міллера-Рабіна
Число 62142947553432873966091371032767514633643498580929973656531447899923964427287 не пройшло тест Міллера-Рабіна
```

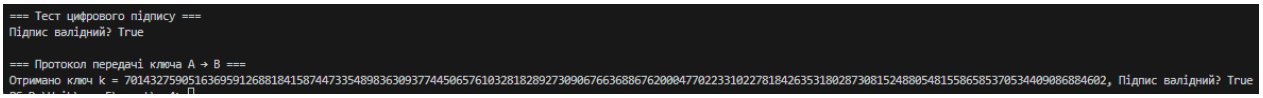
Генерація ключових параметрів криптосистеми RSA для абонентів A та B:



Шифрування та дешифрування повідомлення:



Перевірка цифрового підпису:



Висновок

Метою роботи було створення криптосистеми RSA на Python для безпечного обміну інформацією між двома абонентами. Було опрацьовано теоретичні основи, зокрема піднесення до степеня за модулем за схемою Горнера, поняття псевдопростих чисел та ймовірнісні тести Ферма, Соловея-Штрассена і Міллера-Рабіна. На їх основі реалізовано генерацію 256-бітних простих чисел і функції для створення ключових пар RSA, що забезпечують шифрування та розшифрування повідомлень.

Крім того, реалізовано цифровий підпис із SHA-256 для автентифікації та перевірки цілісності повідомлень, а також протокол конфіденційної передачі ключів, який поєднує шифрування та перевірку підпису для підтвердження справжності відправника. Практична реалізація охопила всі ключові компоненти RSA, включно з перевіркою чисел на простоту, генерацією ключів, шифруванням і розшифруванням, підписом та протоколом безпечної передачі ключів, що продемонструвало ефективне застосування асиметричної криптографії.