МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФІЗИКО- ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра інформаційної безпеки

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4

Варіант 2

з дисципліни

Криптографія

З теми: « Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методами генерації параметрів для асиметричних криптосистем »

Перевірила:

Селюх П.В

Виконав студент групи ФБ-92

Андрієвич Дмитро Юрійович

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.

Завдання:

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 1 p , q довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq ≤ p1q1 ; p і q – прості числа для побудови ключів абонента A, 1 p 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) , (,) 1 n1 e 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().

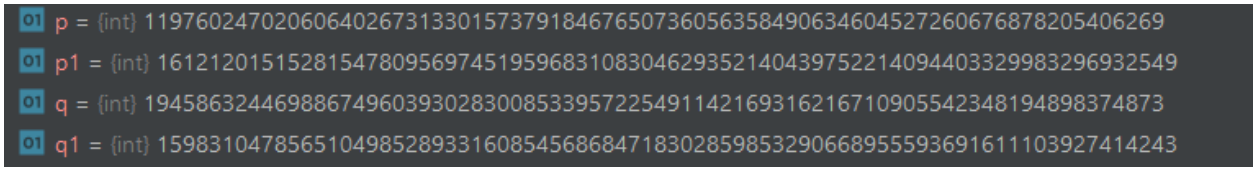
Код програми знаходиться у файлі **main.py**.

Хід роботи:

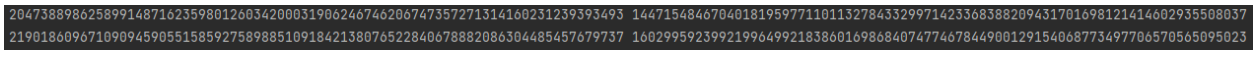
Спочатку я реалізував функцію пошуку простих чисел з заданого інтервалу, після цього за допомогою неї згенерував дві пари p та q. Далі я реалізував функцію генерації пар ключів зі знайдених раніше p та q. Далі мною було реалізовано функції шифрування та дешифрування, також було зроблено функцію, яка відповідає за створення цифрового підпису, та ще одну, за допомогою якої цей підпис можна підтвердити. Після цього були створені два абоненти, на яких і було перевірено функціонування програми.

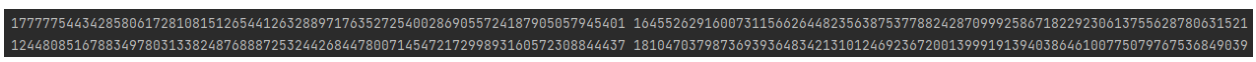
Результати:

P, q, p1, q1:

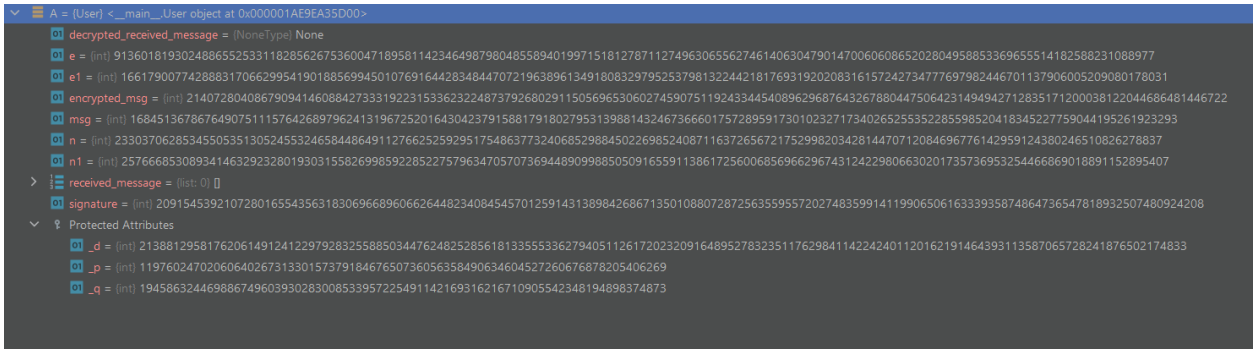


Кандидати, що не пройшли перевірку:

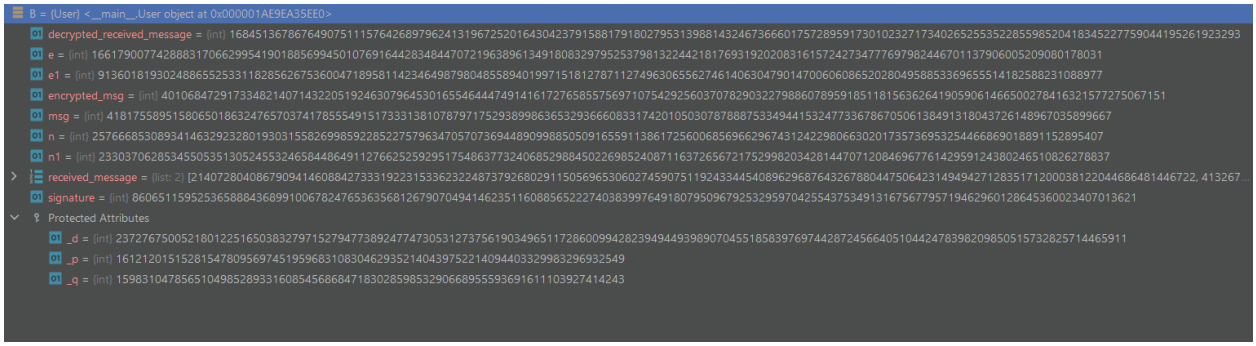




Параметри абонента A:



Параметри абонента B:



Вхідний текст A:

168451367867649075111576426897962413196725201643042379158817918027953139881432467366601757289591730102327173402652553522855985204183452277590441952619232 93

Шифрований текст A:

214072804086790941460884273331922315336232248737926802911505696530602745907511924334454089629687643267880447506423149494271283517120003812204468648144672 2

3445408962968764326788044750642314949427128351712000381220446864814467 22

Цифровий підпис A:

209154539210728016554356318306966896066264482340845457012591431389842686713501088 07

2872563559557202748359914119906506163339358748647365478189325074809242 08

Вхідний текст B:

418175589515806501863247657037417855549151733313810787971752938998636532936660833 1

7420105030787888753349441532477336786705061384913180437261489670358996 67

Шифрований текст B:

401068472917334821407143220519246307964530165546444749141617276585575697107542925 6

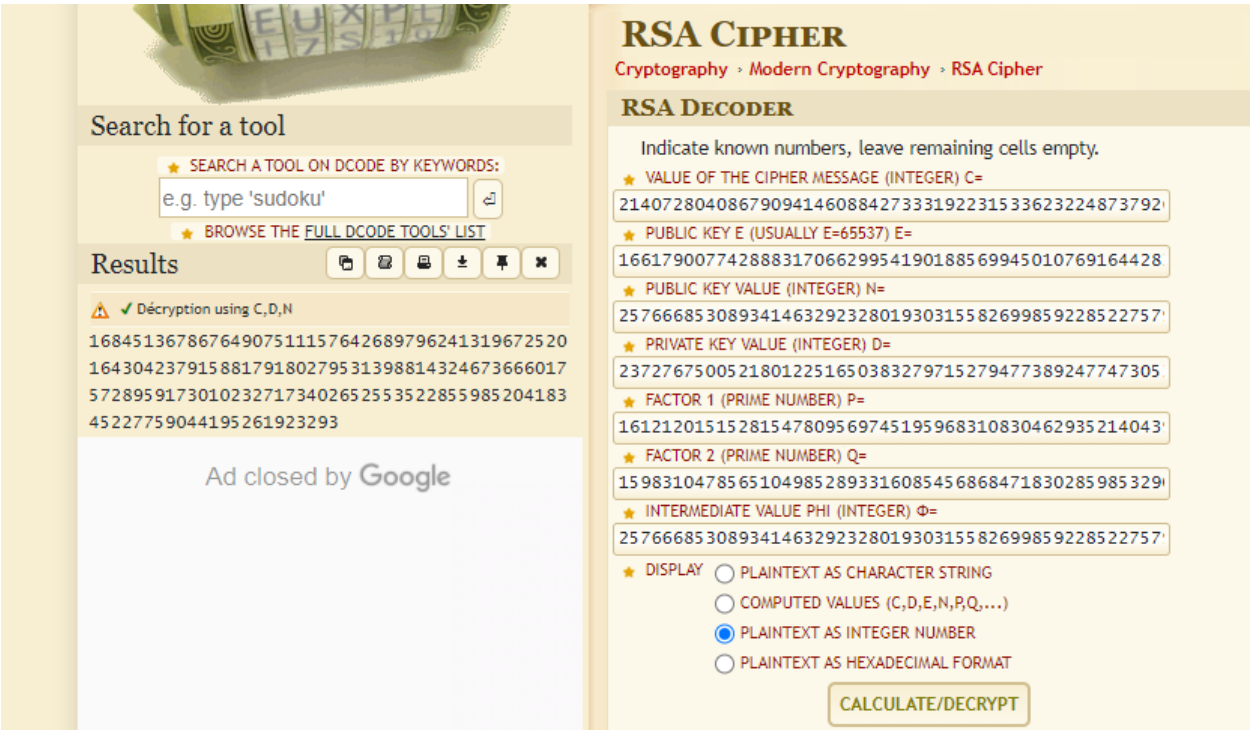
037078290322798860789591851181563626419059061466500278416321577275067151

Цифровий підпис B:

860651159525365888436899100678247653635681267907049414623511608856522274038399764 9

180795096792532959704255437534913167567795719462960128645360023407013621

Перевірка роботи програми на прикладі вхідного тексту A (абонент A відправляє абоненту B текст, який було зашифровано відкритим ключем абонента B) за допомогою сайту https://www.dcode.fr/rsa-cipher:



Результат дешифрування співпадає з вхідним текстом абонента А.

Висновки: під час виконання даної лабораторної роботи я ознайомився з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA. Також, я ознайомився з криптосистемою RSA та реалізував засекречений зв'язок з використанням цієї системи.