Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”
Фізико-технічний інститут

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4

з предмету «Криптографія»

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

Виконав:
Студент 3 курсу,
ФТІ, групи ФБ-05
Савченко Ярослав

Київ – 2022

Мета та основні завдання 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 і рекомендації щодо 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 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.

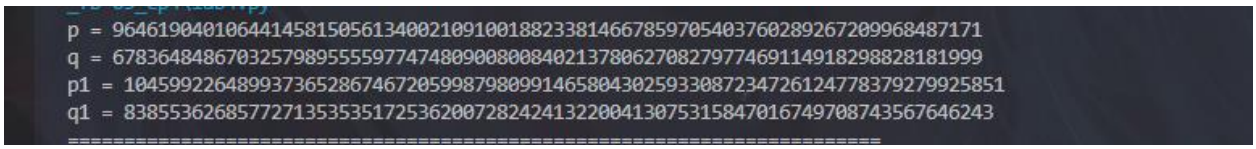
За допомогою датчика випадкових чисел вибрати відкрите повідомлення M і знайти криптограму для абонентів A и B, перевірити правильність розшифрування. Скласти для A і B повідомлення з цифровим підписом і перевірити його.

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.

Хід роботи

Створимо функцію що буде брати випадкове число заданої довжини і якщо воно не виявится простим то візьме інше випадкове число цієї довжини. Звісно створимо функцію для визначення простоти числа використовуючи метод Міллера-Рябіна описаний в методичці.

Для наступного завдання створив функцію що використовуючи функції минулого завдання складає дві пари чисел.



Власне сгенерував їх:

p = 964619040106441458150561340021091001882338146678597054037602892672099684871 71
q = 678364848670325798955559774748090080084021378062708279774691149182988281819 99
p1 = 104599226489937365286746720599879809914658043025933087234726124778379279925851
q1 = 838553626857727135353517253620072824241322004130753158470167497087435676462 43

Звісно вони виконують умову pq<=p1q1

Тепер створимо функцію що генерує наші RSA ключі

Для A

e = 1286703187130306547411584022937342928579376175667791652733057844072551243967857574640280108978148112178607385018442839217700486312418173269599624675595571
n = 6543636491663210923262994325949724193420476654657570522201719242781130797694906503367433311507935809093404378533223754059877160338819514523009390284634829
d = 1674832595352593078357870084825849886626005282155600700036560852460166161921488965964935664642952779496747703269379217494818388830240265662619564413171 91
p = 964619040106441458150561340021091001882338146678597054037602892672099684871 71
q = 678364848670325798955559774748090080084021378062708279774691149182988281819 99

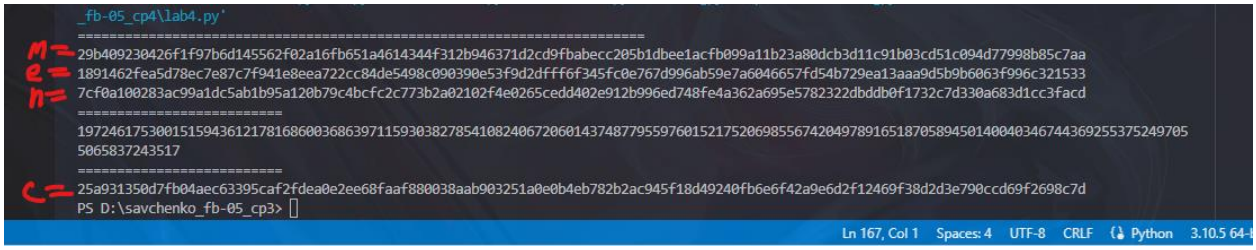
I для B

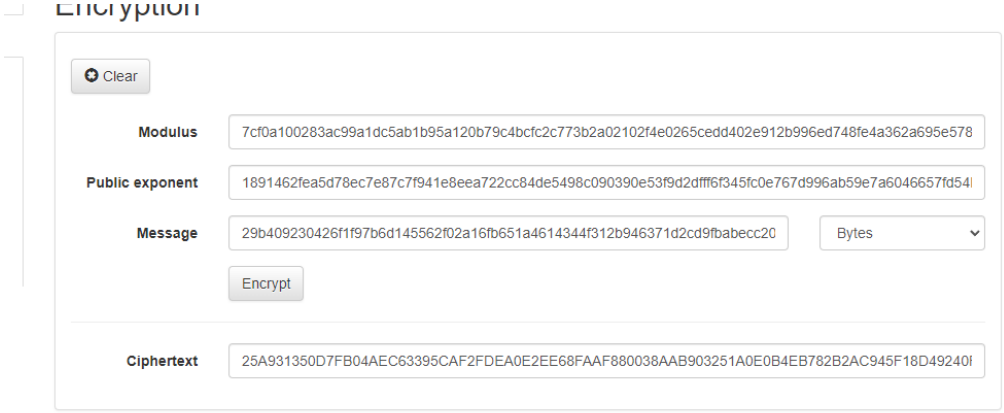
e = 3729987443220244214247463398815233288821137123839120487457072700787954034885560336891095924483582897829600959018572425946696242389492272 14686769854924093 3
n = 8771206073964982507149570623899445520020906584868332149226622176972242559689799879472089066782191573537772647116117311465277677167005467092294247638727793
d = 5170491599282747278183166900796686568093857572649585638798361513209743126261110004445232215369590222096030861896920955578902051101826446117727789239499397
p = 104599226489937365286746720599879809914658043025933087234726124778379279925851
q = 838553626857727135353517253620072824241322004130753158470167497087435676462 43

Для наступного завдання реалізуємо необхідні функції та перевіримо їх користуючись запропонованим сервісом

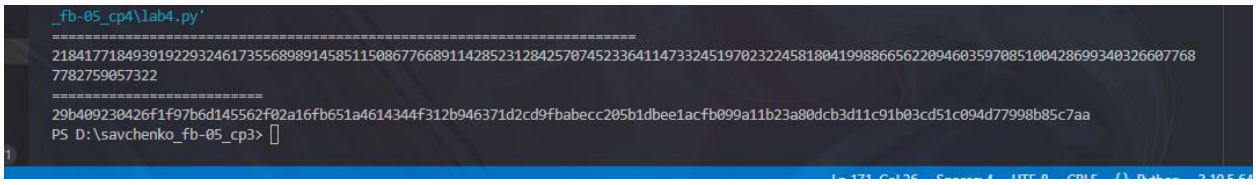
Шифрування:

На цьому етапі зіткнувся з проблемою що на сайті усі ключі та тексти перевіряються з 16 системою HEX чисел, а я маю всі дані в 10 ссистемі. Тому переведу свої значення в HEX значення і зашифрую повідомлення на своєму комп'ютері



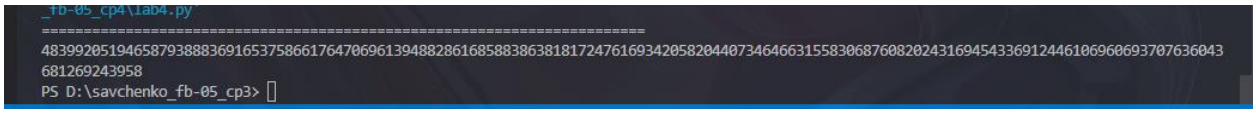


Працює відмінно. Перевіримо наступні функції. Дешифрування. Передамо в неї наш зашифрований текст і секретні ключі.



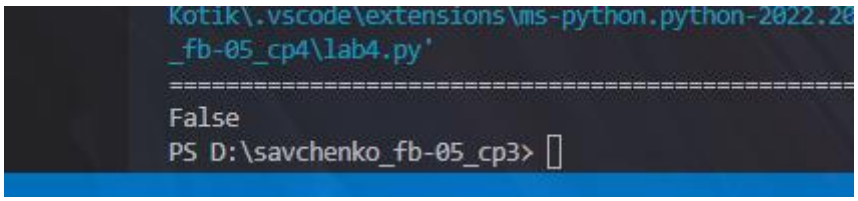
Отримали початковий текст 😊

Підпис:



Верифікація:

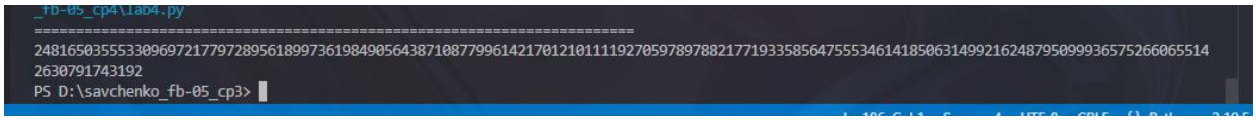
Спробуємо ввести ключ для користувача B



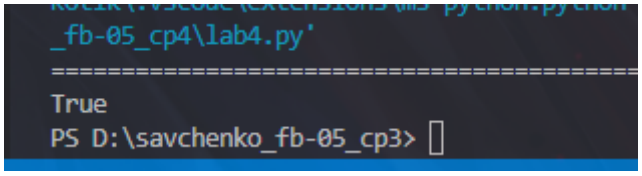
Працює

І останнє завдання. Передача ключів.

Згенеруємо якесь випадкове k і запишемо для подальшої роботи з ним



Запишемо фукнції що використовують метод з методички, та виведемо повідомлення чи є підпис А істинним



Висновки: на цій лабораторній роботі я отримав дуже корисні навички роботи з генеруванням простих чисел, ключів для RSA і реалізацією простої системи шифрування на основі RSA з підписами і перевіркою іх. А також з протоколом розсилання іх. В цій роботі в мене майже не виникало проблем і на мою думку це була найлегша з робіт.