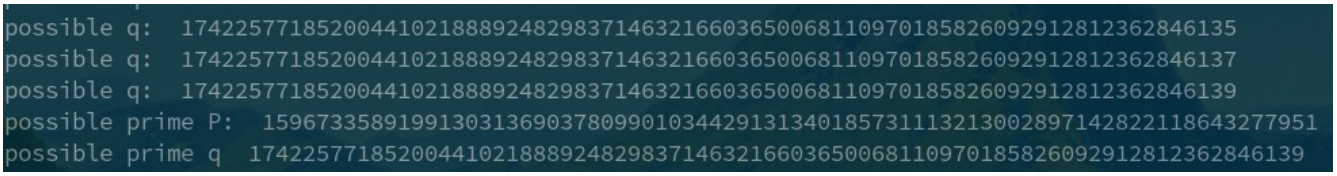
Грипас Владислав ФБ-12
Варіант 15
Лабораторна робота №4
Тема:

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

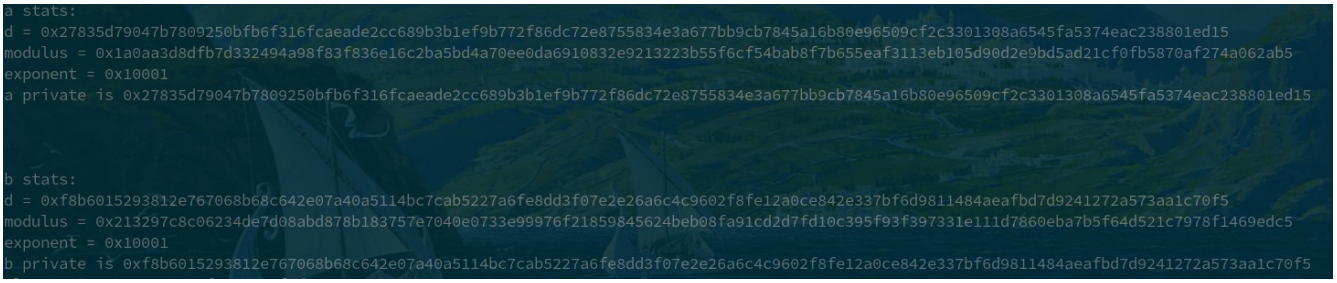
Мета:

Ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практичне ознайомлення з системою захисту інформації на основі криптосистеми RSA, організація з використанням цієї системи засекреченого зв’язку й електронного підпису, вивчення пртоколу розсилання ключів

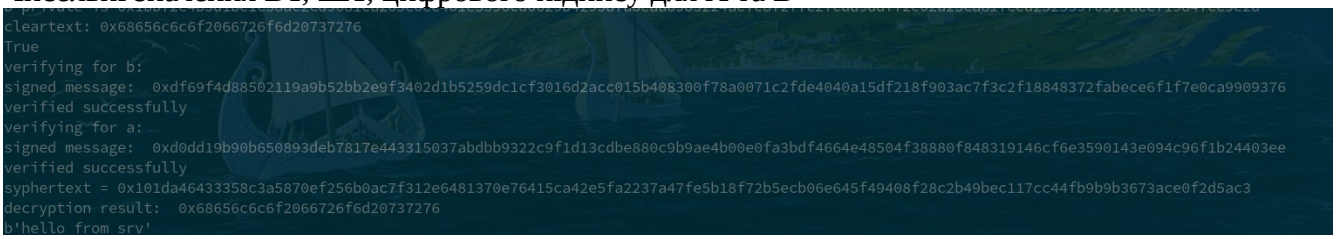
Враховуючи те, що було перебрано велику кількість чисел, що не пройшли тест перевірки простоти, надаю останні декілька строк:



Параметри криптосистеми RSA:



Чисельні значення ВТ, ШТ, цифрового підпису для А та В



Чисельні значення характеритсик на кожному кроці для протоколу конфіденційного розсилання ключів

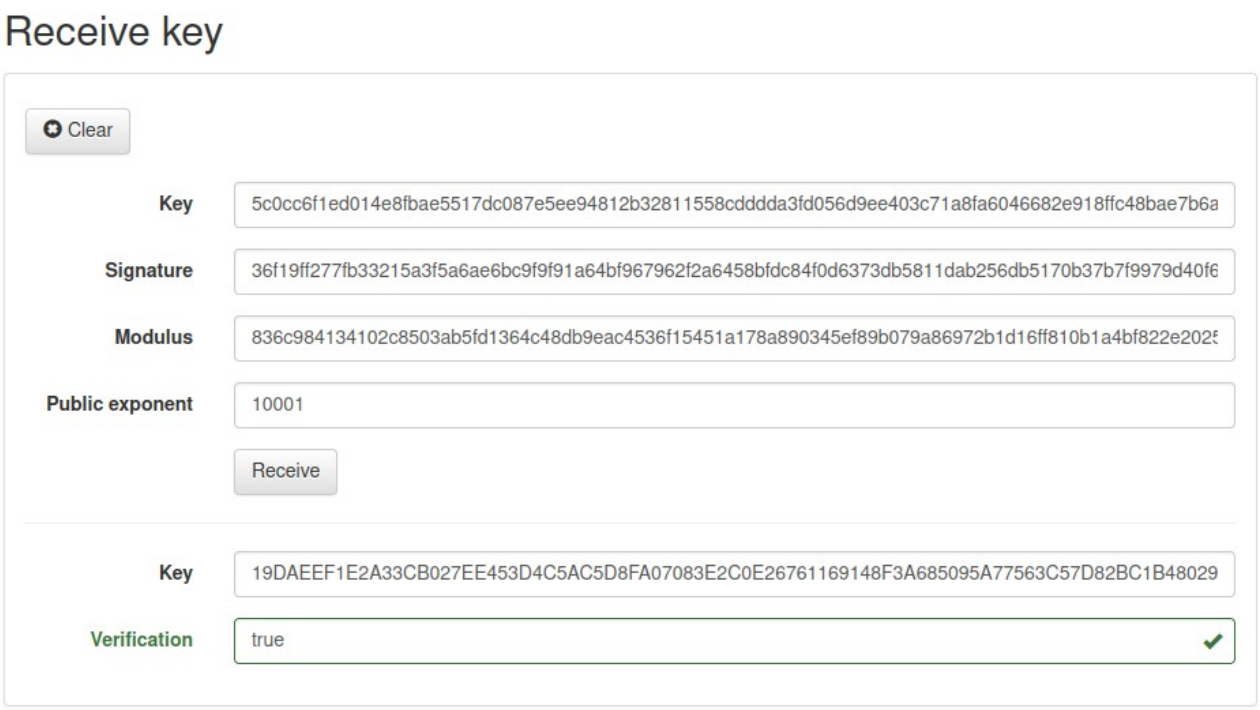
Код протоколу конфіденційного розсилання ключів має такий вигляд;

```
def SendKey(self):
    k = random.randint(1, self.public_key.n-1)
    print("generated k: ", hex(k))
    k1=  pow(k, self.friend_public_key.e, self.friend_public_key.n)
    S = pow(k, self.d, self.public_key.n)
    print("generated S: ", hex(S))
    S1 = pow(S, self.friend_public_key.e, self.friend_public_key.n)
    print("sending S1: ", hex(S1))
    print("sending k1: ", hex(k))
    return k1, S1

def RecieveKey(self, k, S1):
    k = pow(k, self.d, self.public_key.n)
    S = pow(S1, self.d, self.public_key.n)
    print("decrypted k: ", hex(k))
    print("decrypted S: ", hex(S))
    return k == pow(S, self.friend_public_key.e, self.friend_public_key.n)
```

Відповідно, в моєму випадку абонент А відпавляє ключ, та цифровий підпис до В, в результаті вдалої перевірки повертається значення True

Результати перевірки SendKey та RecieveKey в RSA testing env:



В результаті виконання даної лабораторної роботи я ознайомився з тестами перевірки чисел на простоту для асиметричної криптосистеми RSA, та її імплементації на мові програмування Python