**Applied Cryptography (UE20CS314)**

**Lab 4**

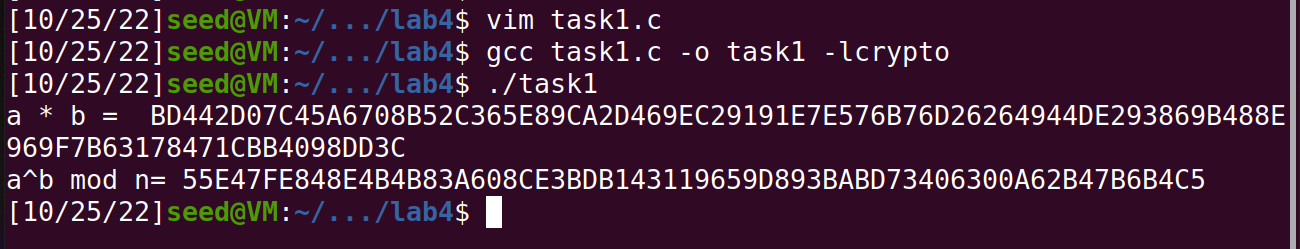
**Name: Vishwa Mehul Mehta**

**SRN: PES2UG20CS389**

**Section: F**

Task 1:

Screenshot:

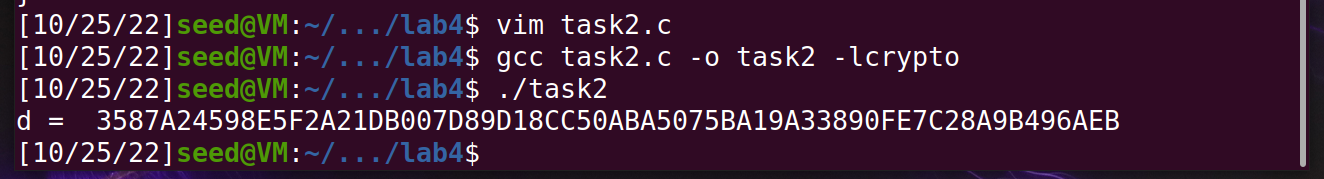


Observation:

We use the <openssl/bn.h> library for BIGNUM calculations. Here, we see the multiplication and modular exponentiation functions.

Task 2:

Screenshot:



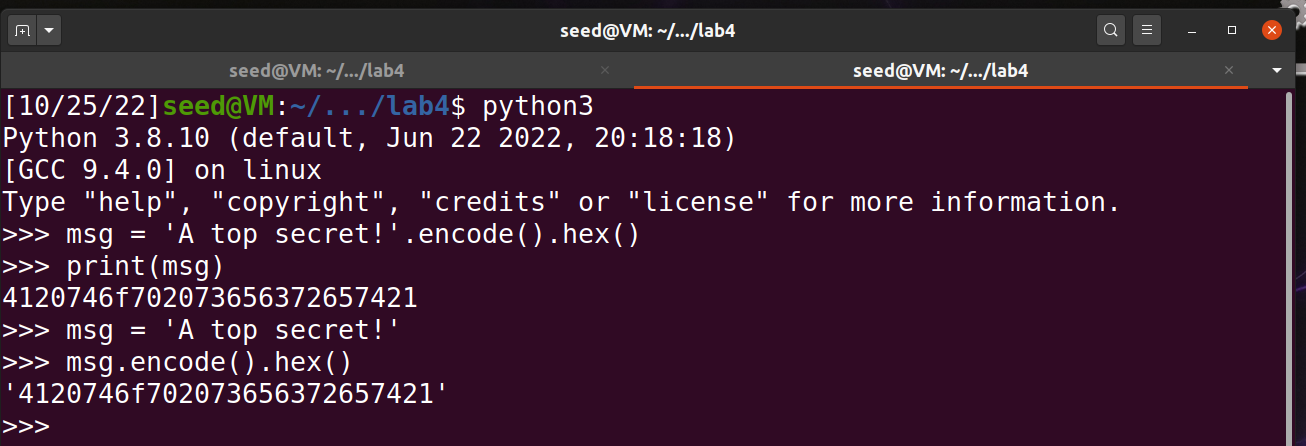
Observation:

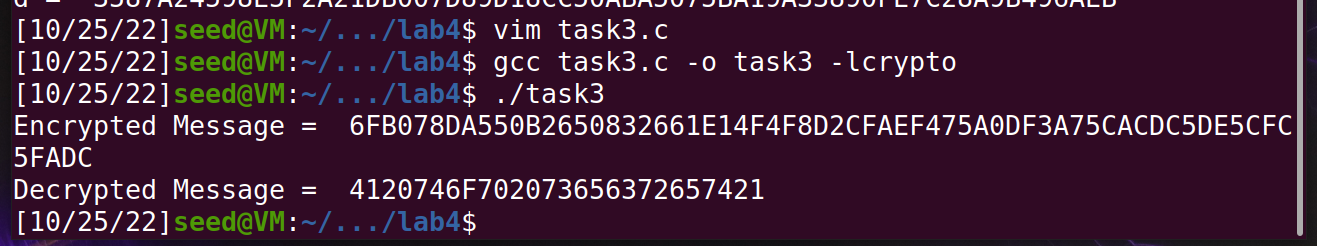
We now calculate the private key using the same library functions given 2 hexadecimal prime numbers p, q, and the public key (e, n)

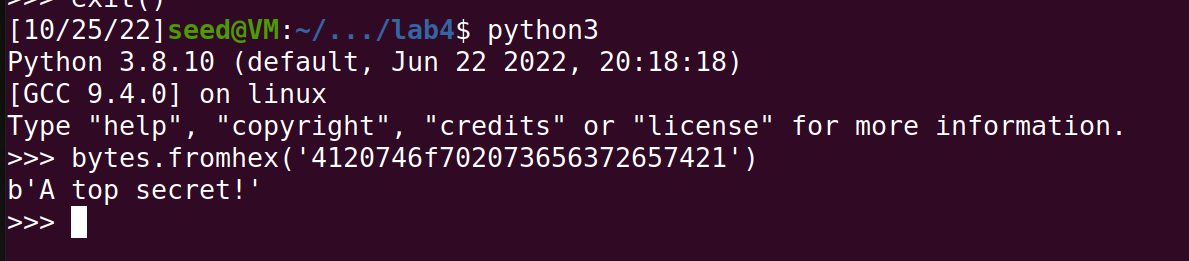
The **res1** variable takes the value **p-1** and **res2** takes the value **q-1**. These values are used to calculate f(n) (Totient Function) which is **res3 = res1 \* res2**. Then we find the value of private key **d** **= e-1 mod res3.**

Task 3:

Screenshot:







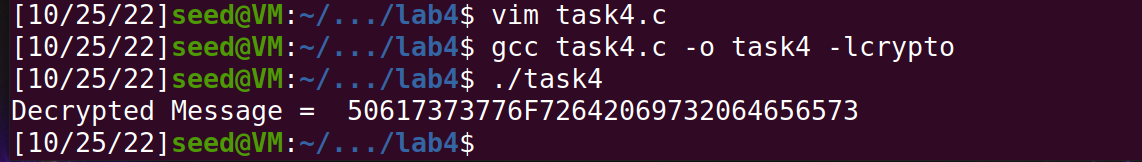
Observation:

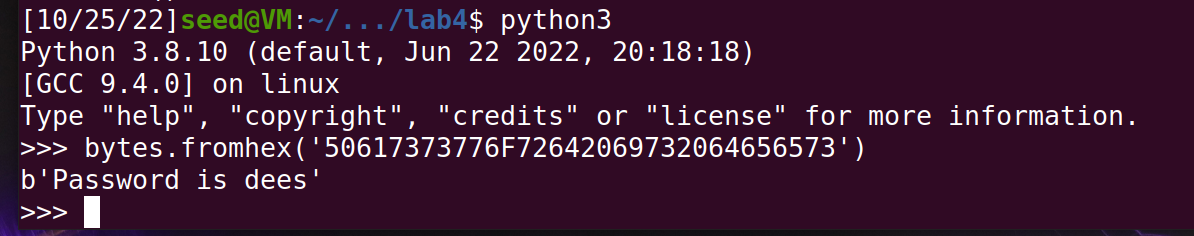
Task 1 is generating the hex code for the secret message that is used as the message in the encryption and decryption for Task 2.

In Task 2, we encrypt using the formula **enc = m^e mod n**, where **m** is the message, e and n are part of the public key. Decryption is done using the formula **dec = enc^d mod n**, where d is the private key and enc is the encrypted message.

Task 4:

Screenshot:



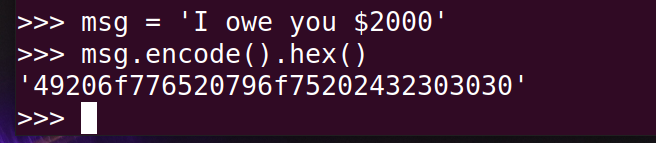


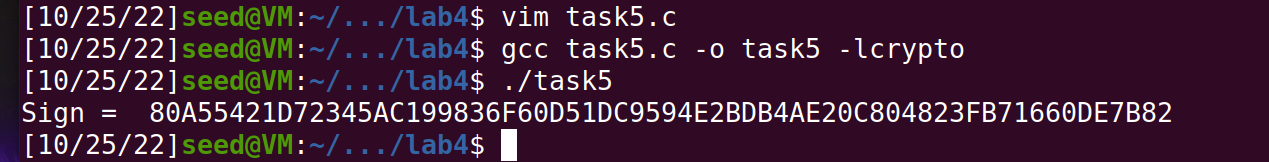
Observation:

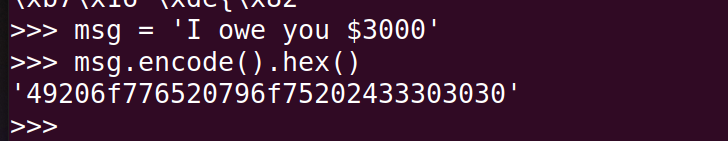
We decrypt a cipher text **c** and find the message using the decryption half of the code in Task 3.

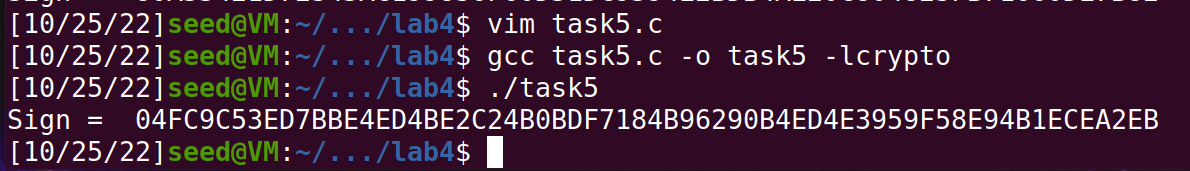
Task 5:

Screenshot:







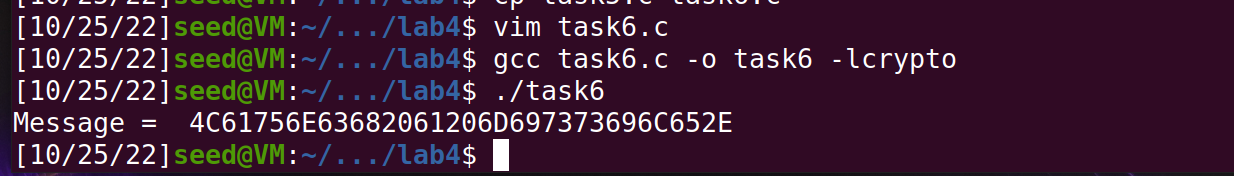


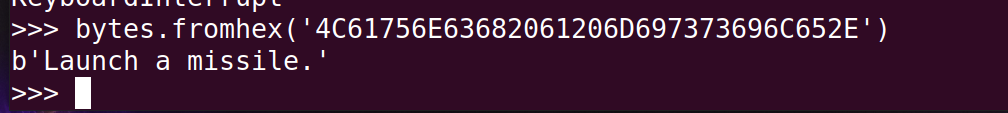
Observation:

We sign a message using the formula **sign = m^d mod n**, which gives a unique value for every message, a change in a single bit causes a change in multiple bits in the sign, as seen in the example where we change 2000 to 3000 and the signs are completely different.

Task 6:

Screenshot:





Observation:

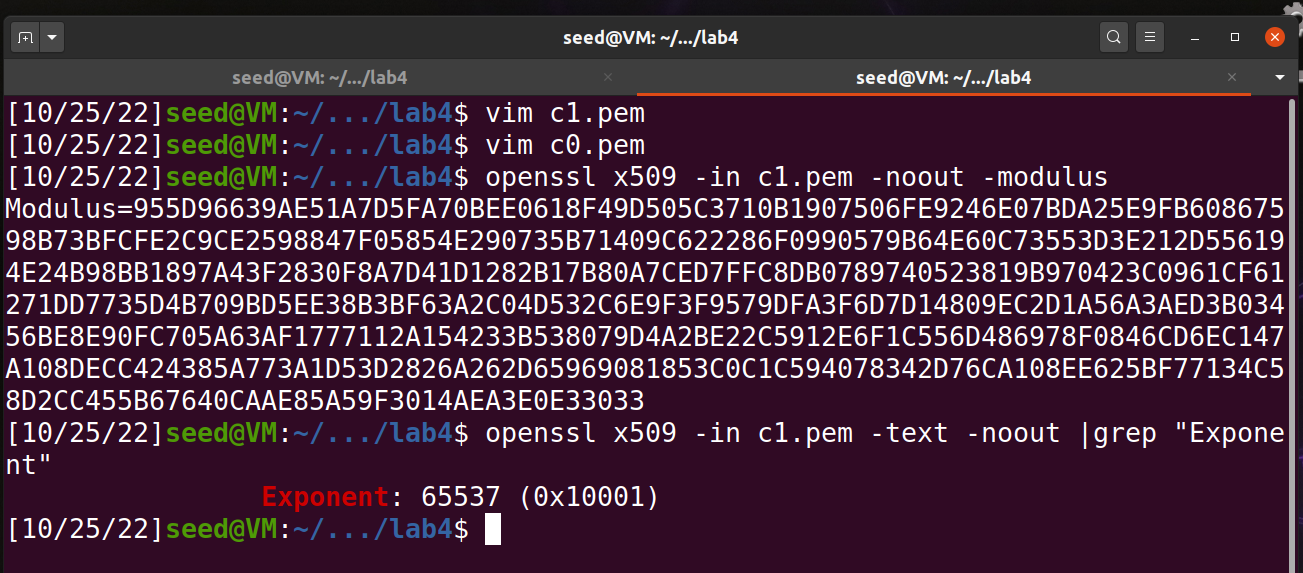
We now verify the sign by getting the message using the formula **message = s^e mod n**, and convert to byte form from hex.

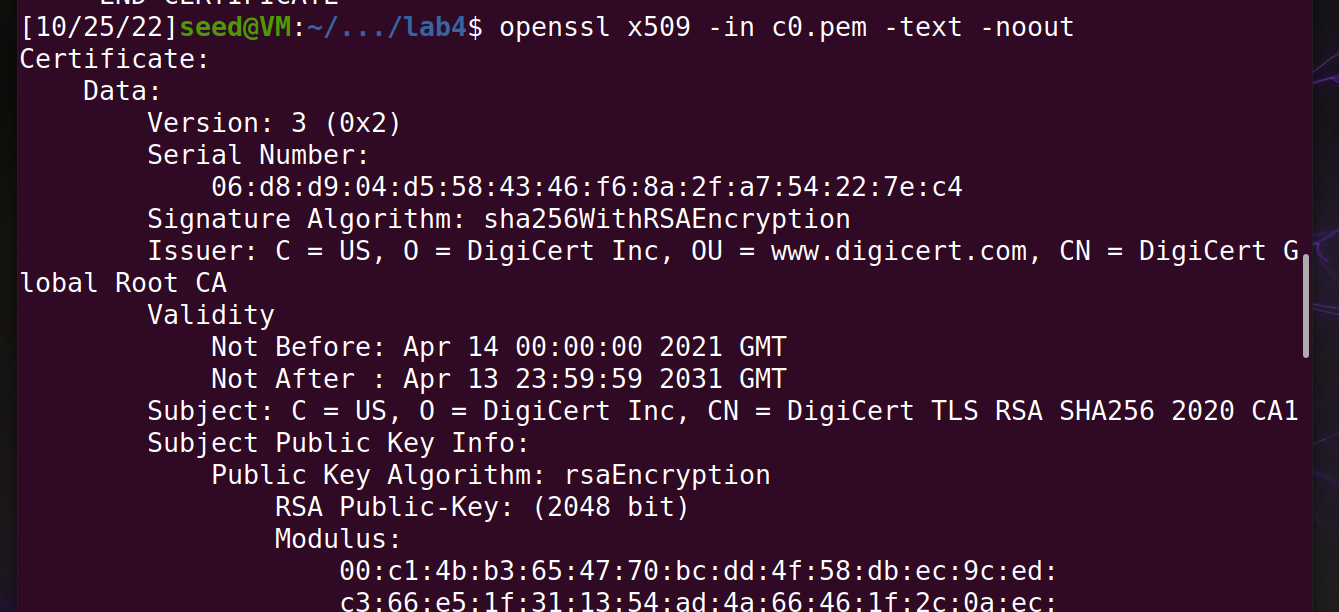
Task 7:

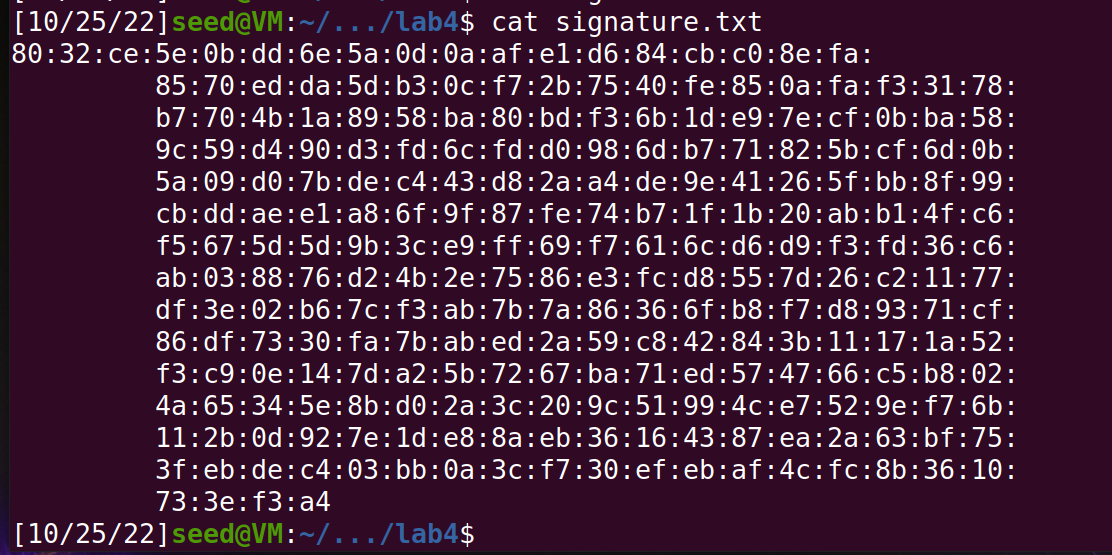
Screenshot:



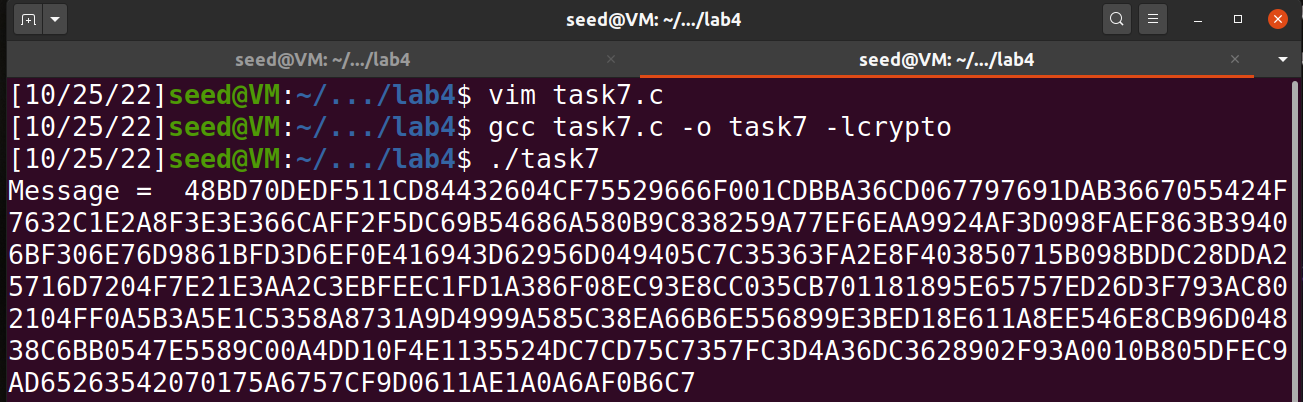












Observation:

We verify a certificate manually by first getting the user and the server certificates first along with the **modulus**, **exponent, and signature** to get back a message.