ITSEC

LAB REPORT: 2

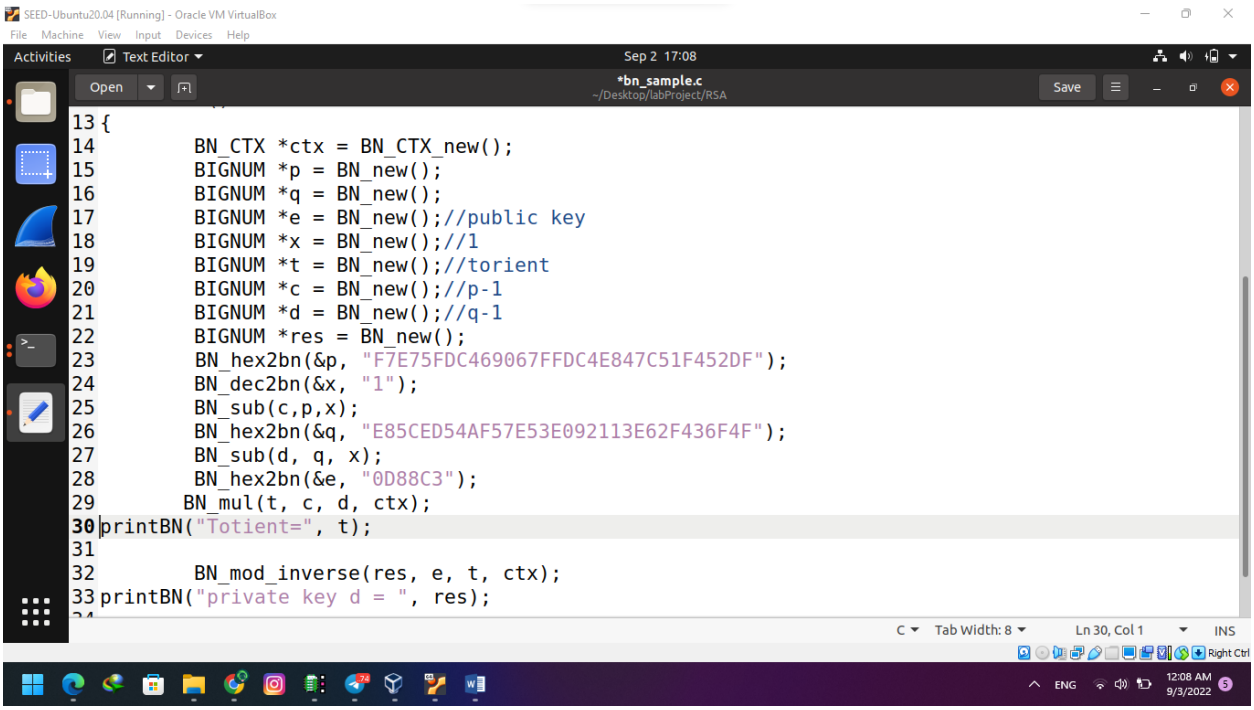
BY – AMANUEL LEGESSE SOFTWARE ENG. EXT

ID – ATE/3628/1

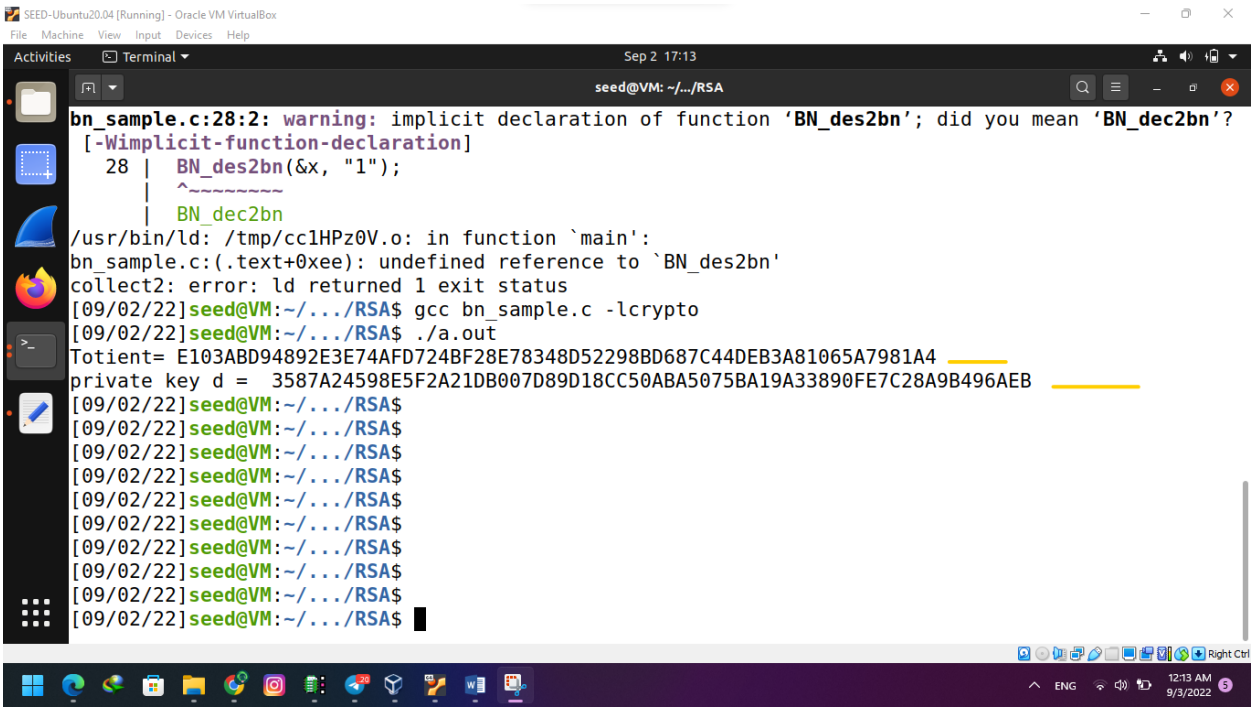
RSA Encryption and Signature Lab

Task -1 Deriving the Private Key

- ✓ I compiled the program, bn_sample.c, given in the project description. I ran " gcc bn_sample.c -lcrypto " to compile the program using the crypto library. The following was the result from running the program.

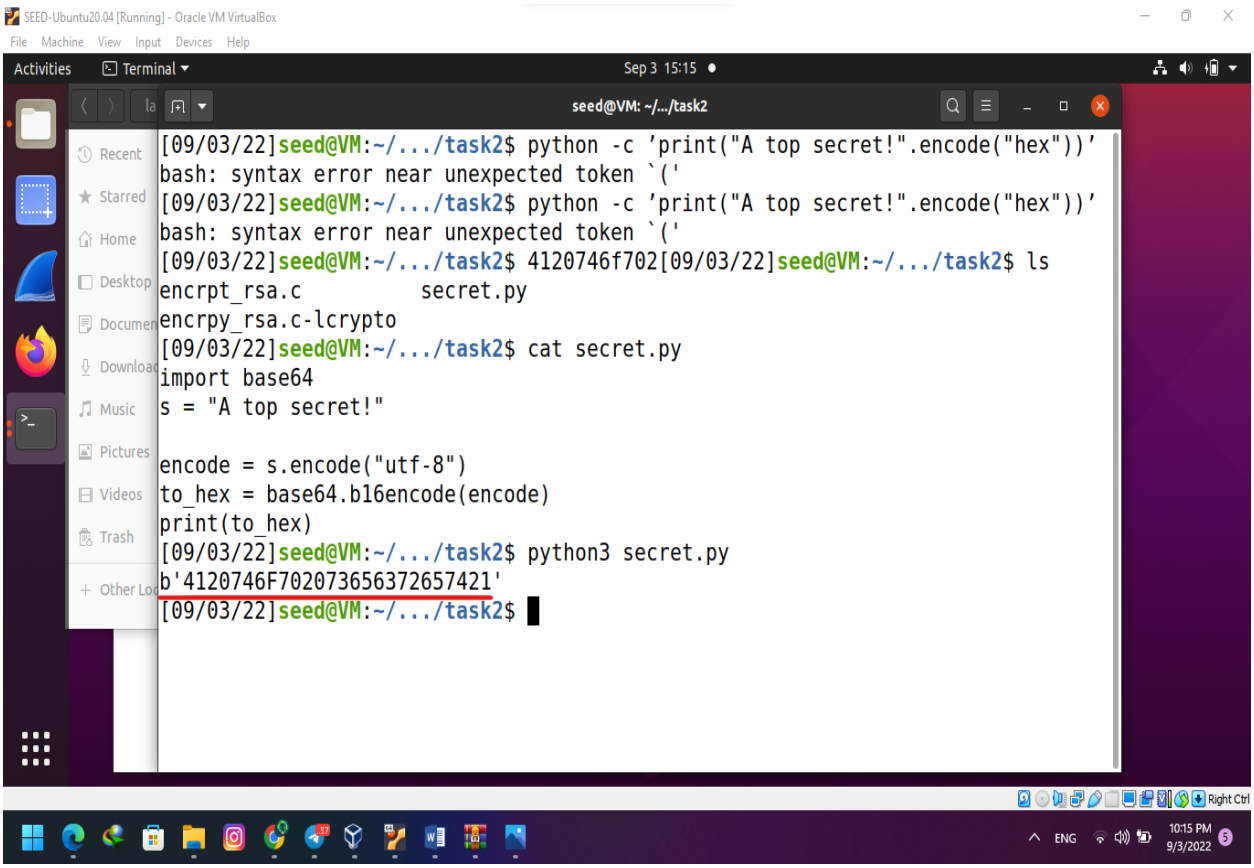


- I calculated n by multiplying p * q and To find ϕ(n) , I calculated (p – 1)*(q – 1).

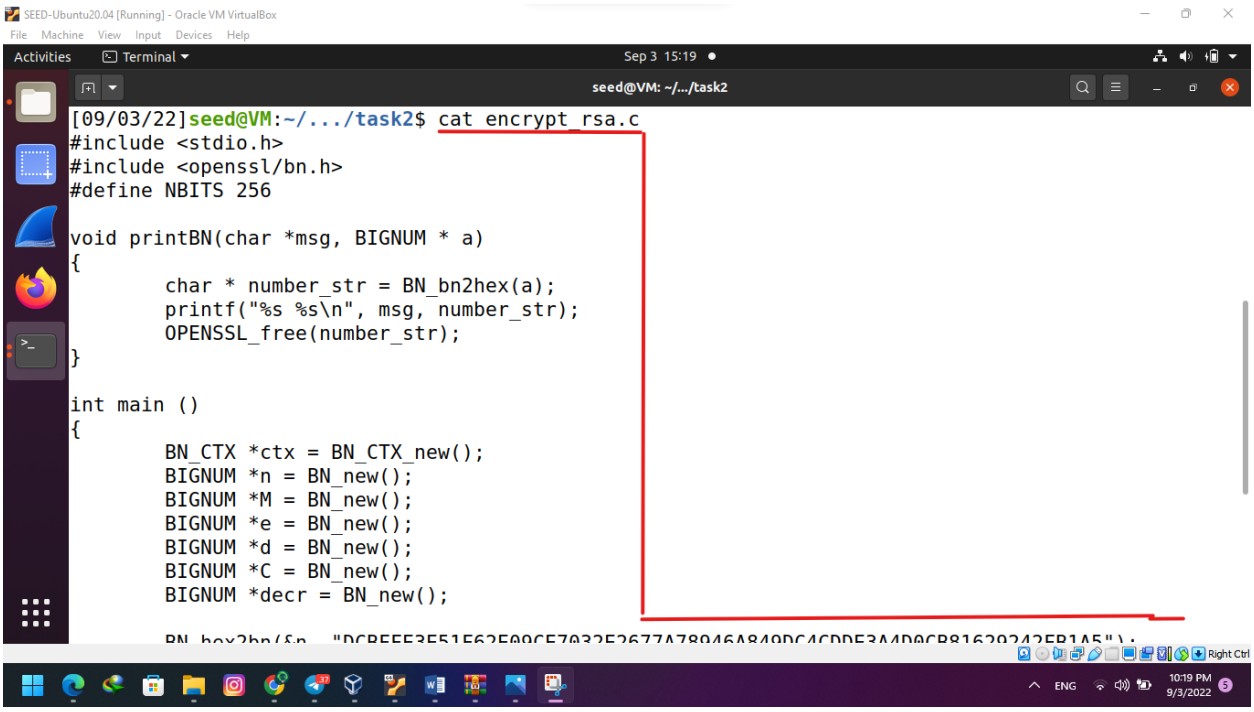


Task -2 Encrypting a Message

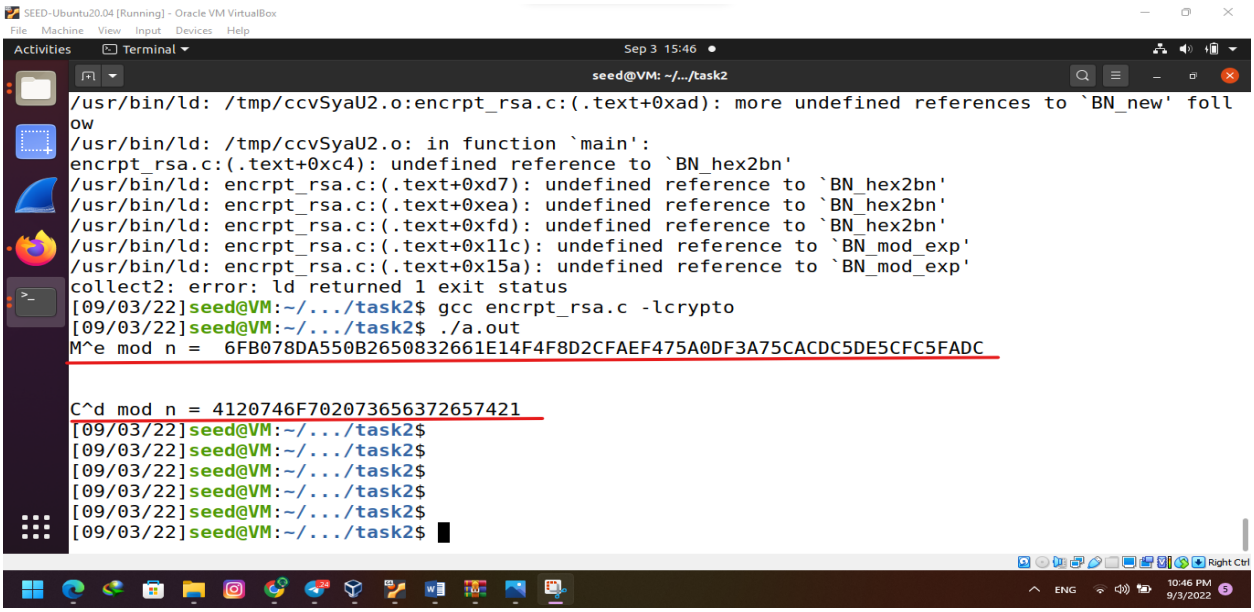
- ✓ In order to encrypt the message “A top secret!” first I converted it into hex string then to a BIGNUM using hex-to-bn. Using s python command



➢ Next I compiled the encryption to C.

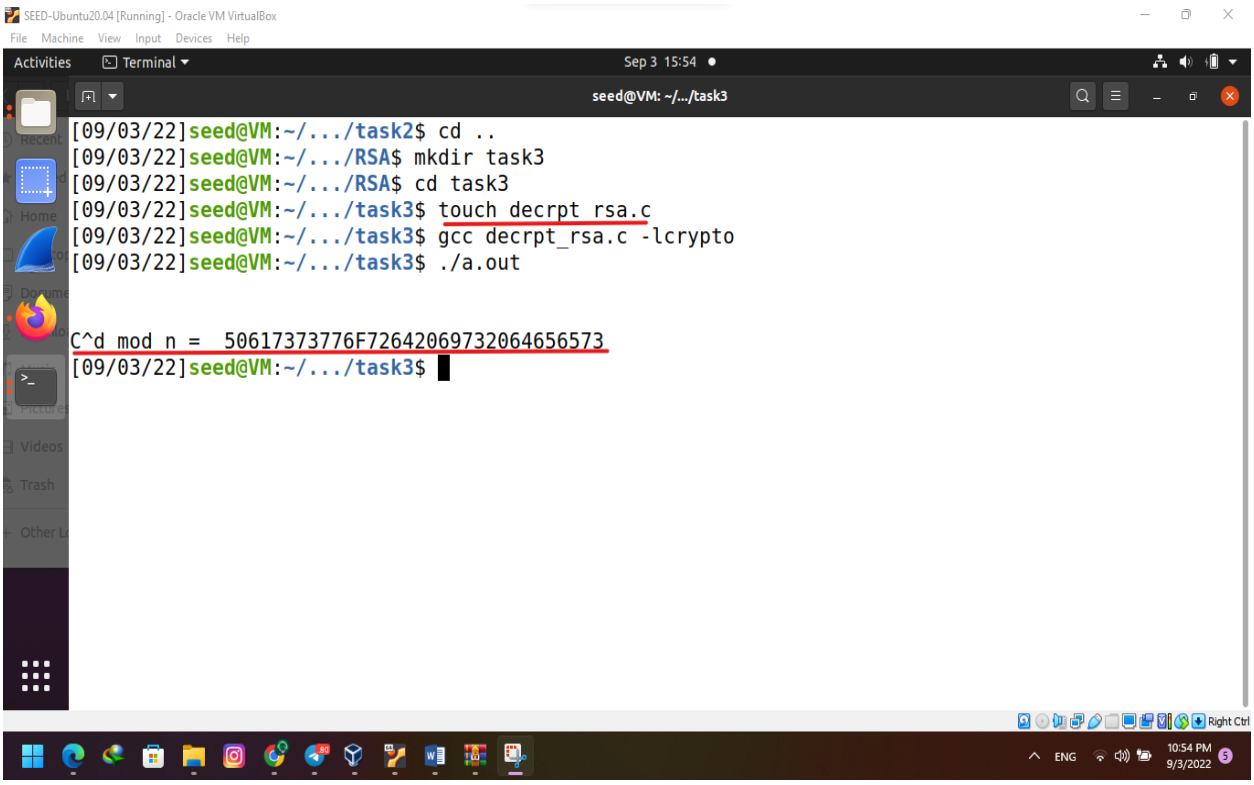


✓ After running the C program encrpt_rsa.c I got the encrypted and decrypted message in hexDecimal.



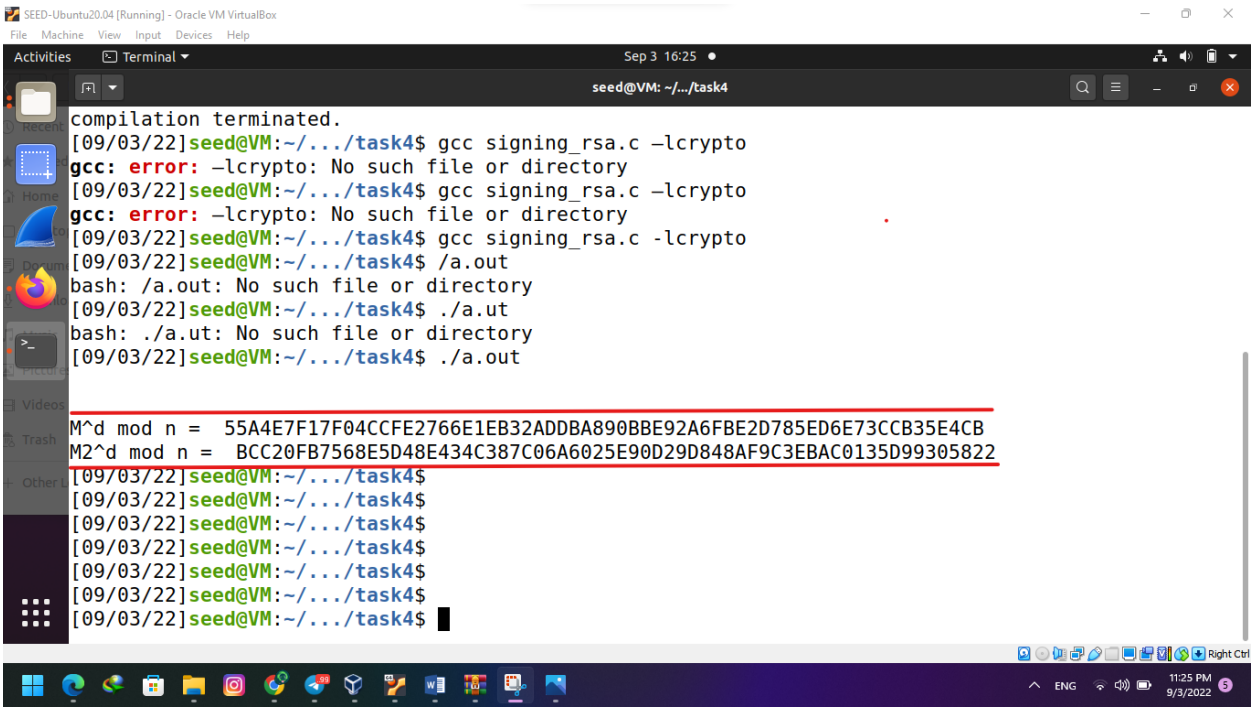
Task 3: Decrypting a Message

- ✓ Previously I already implemented the decryption equation, I used this code to decrypt the ciphertext provided by creating a decrpt_rsa.c file with an out put if the decrypted number.



Task 4: Signing a Message

- ✓ Created 2 python scripts with “I owe you $2000” and “I owe you $3000 to sign and change it from $2000 to $300 by using a complied C program signing_rsa.c. The result is as shown below.

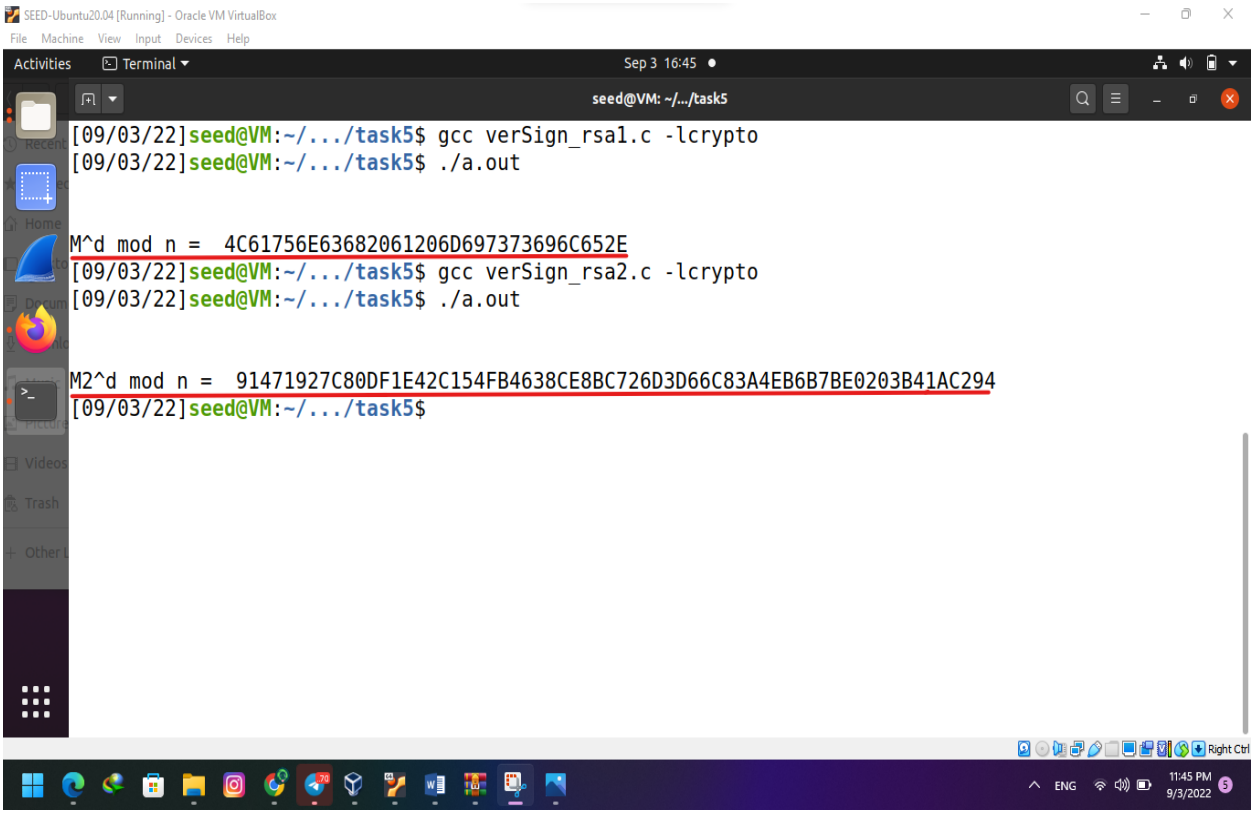


✓ I converted the message to hexadecimal and ran the digital signature equation on it.

➢ The original hexadecimal message encodings only differed by one byte but, the digital signatures of the two messages are different.

Task 5: Verifying a Signature

✓ To verify that a signature is the correct message, I used the following formula to get the hexadecimal representation of the message: Signature^e mod n. My program returned the hex string.



- After I compiled the C program receiving the first output I modified the signature from 2F to 3F and ran the program again. I got the second hex string displayed above.