**A logo of a university of windsor

Description automatically generated with low confidence**

|  |
| --- |
| **Course Name** |
| Networking and Data Security (COMP-8677) |

|  |
| --- |
| **Document Type** |
| Lab 5 |

|  |
| --- |
| **Professor** |
| Dr. Shaoquan Jiang |

|  |
| --- |
| **Team - Members                               Student ID** |
| Harbhajan Singh                                 110100089 |

**1. Use openssl to generate RSA public/private key**

**We can generate RSA private key (p, q, d) using openssl:**

**$ openssl genrsa –aes128 -out private.pem 1024**

**This will generate a rsa instance (p, q, d, e, n) with p, q of 1024 bits and to prevent leaking the private key, the output private.pem is encrypted by aes128 cipher with password you will be prompted to provide.**

**Now use the above command to generate a rsa private key and save it in file private.pem. Then, extract the public key (e, n) in a file public.pem:**

**$ openssl rsa –in private.pem –pubout >public.pem**

**You can display private key using**

**$openssl rsa –in private.pem –text –noout**

**You also can display public key using**

**$openssl rsa –in public.pem –pubin –text –noout**

**Take screen for the displays for these two files, as evidence of your work.**

**Answer 1: I implemented all the instructions which are required for the problem statement. For the justification/explanation, I have attached the screenshots below.**

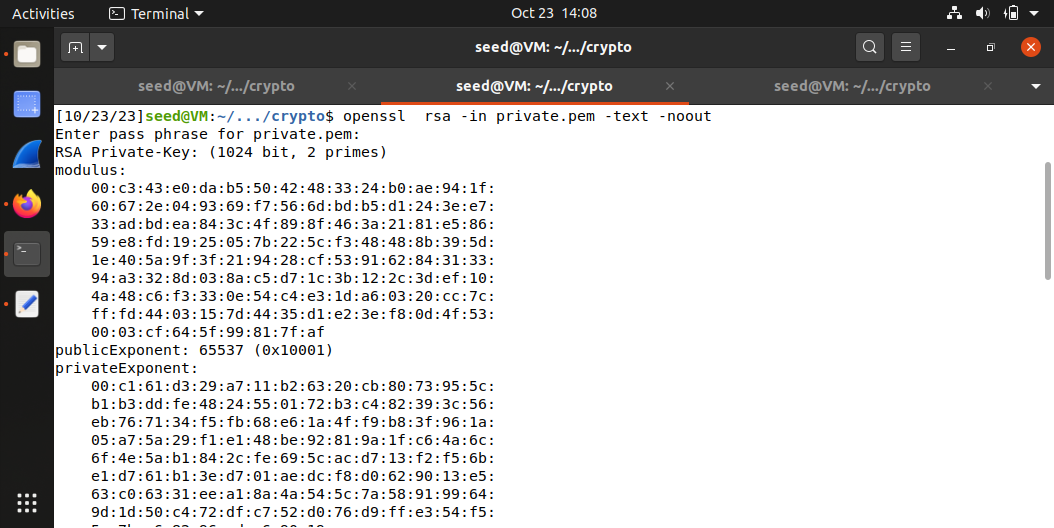
**Step 1:**

Screenshot 1

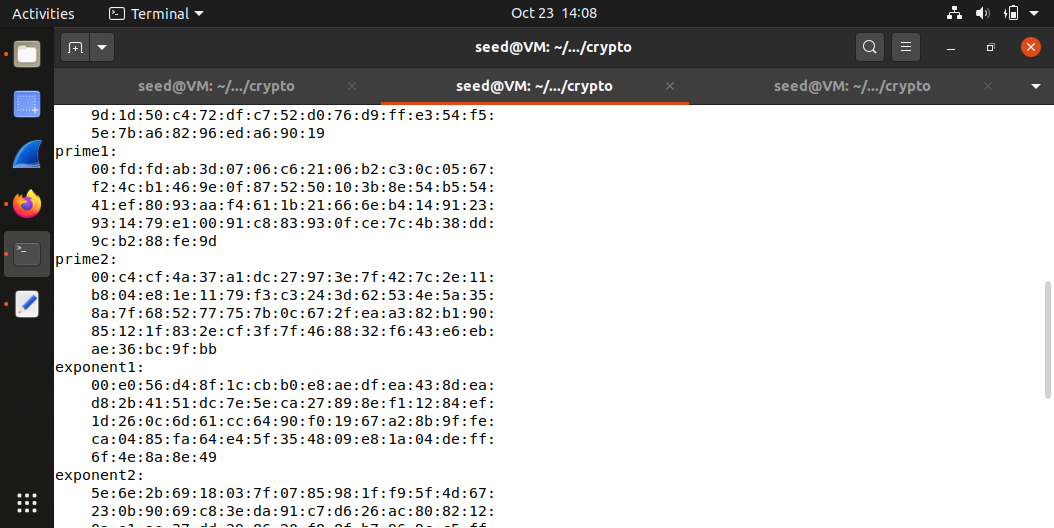
**Step 2:**

Screenshot 2

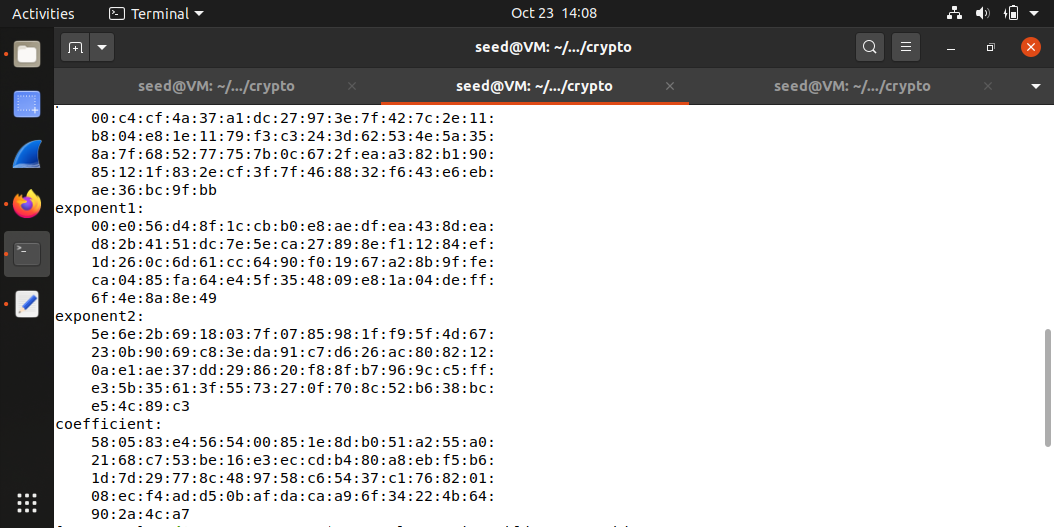
**Step 3:**



Screenshot 3

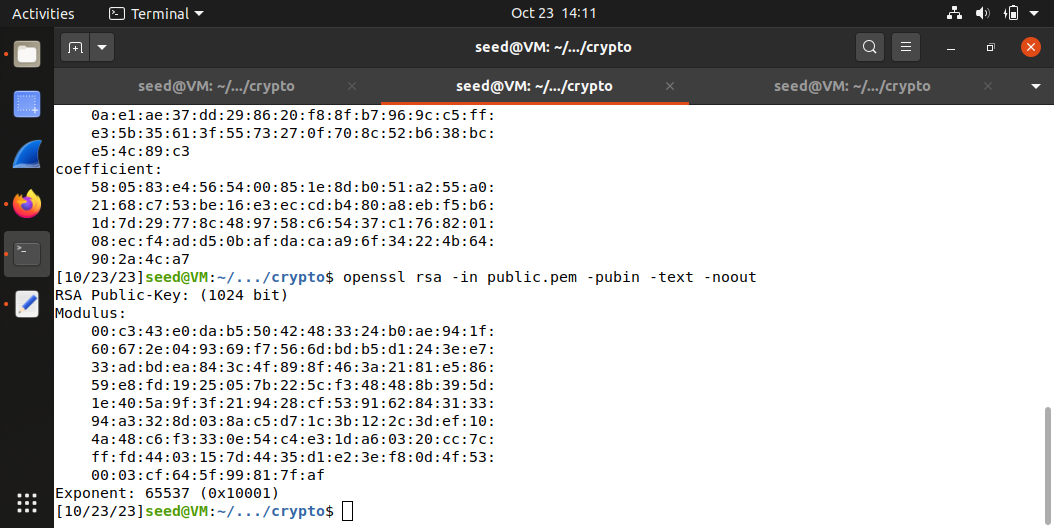


Screenshot 4



Screenshot 5

**Step 4:**



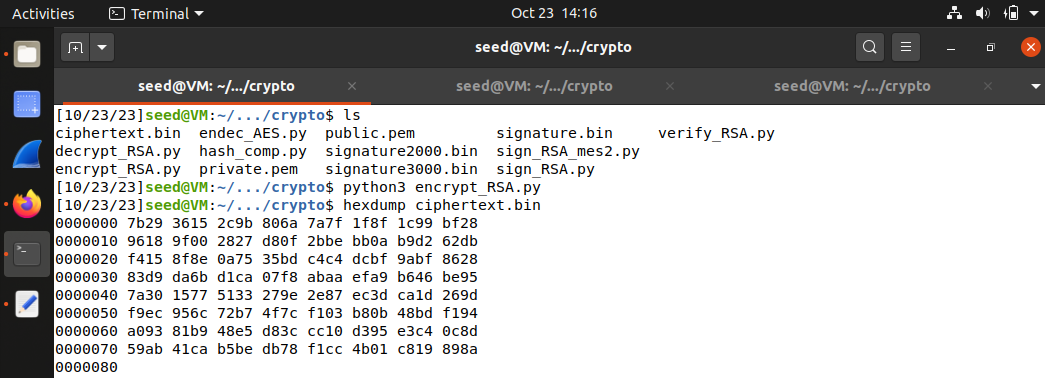
Screenshot 6

**2. In this problem, you need to practice RSA encryption and decryption.**

**A) Encrypt messages using PKCS1\_OAEP, which is an implementation of RSA. Use the key RsaKey derived above to do the encryption. The functions are described as follow. •Cipher=PKCS1\_OAEP.new(RsaKey):oFor the encryption, RsaKey is a public-key. Return an encryption object Cipher. •Cipher.encrypt(message):oThis returns ciphertext of message (byte string) under encryption object Cipher. Encrypt message=’your name and ID’ and save ciphertext into a file. Take a screen shot for hexdump of your ciphertext ($hexdump -C filename). Ref. encrypt\_RSA.py**

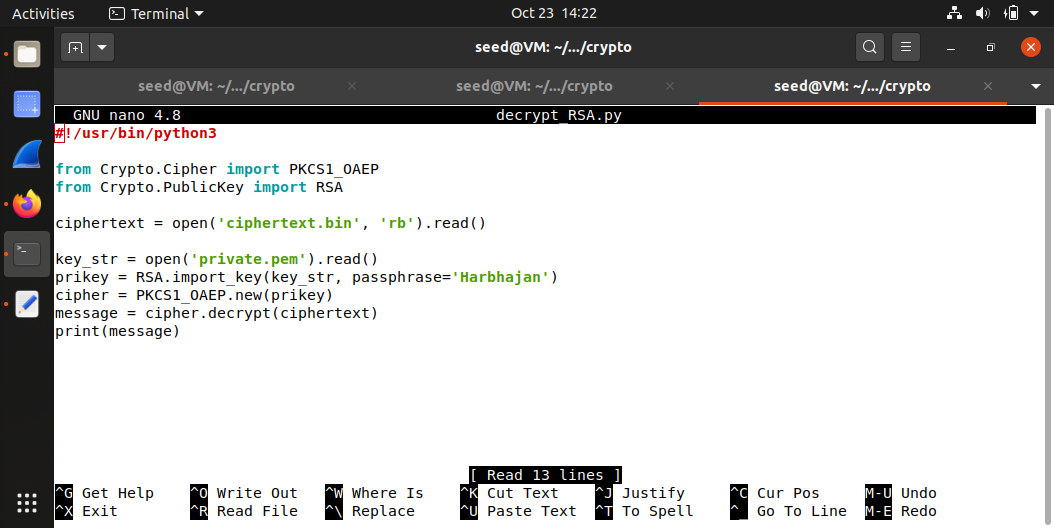
**ANSWER 2A:**

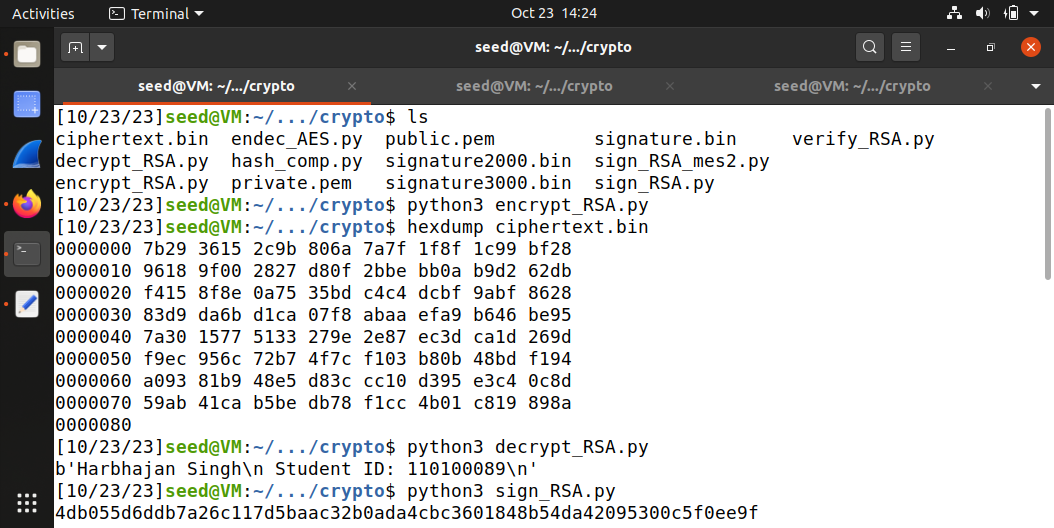




**B)Decrypt the ciphertextin (a). The functions are described as follow. •Cipher=PKCS1\_OAEP.new(RsaKey):oFor the decryption, RsaKey is a private-key. Return an decryption object Cipher. •Cipher.decrypt(ctxt):oThis returns message=’your name and ID’ under decryption object Cipher. Take a screen shot for your decryption. Ref. decrypt\_RSA.py.**

**ANSWER 2B:**





**3. In this problem, you practice RSA signature: generation and verification.**

**A. Generate RSA based signature. The functions are described as follows.**

**•Signer=pss.new(RsaKey):**

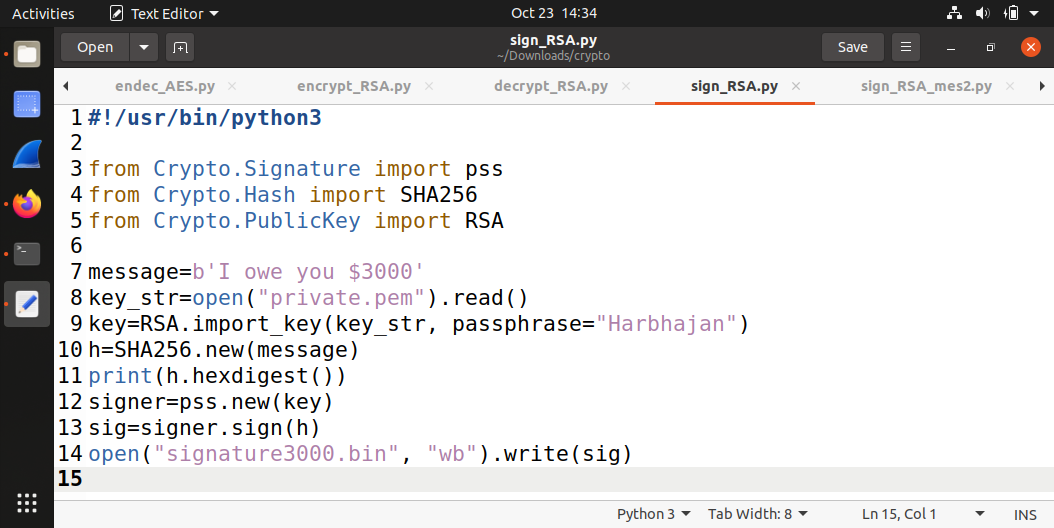
* **This defines a signing object signer with RsaKey (imported from your RSA private key file).**

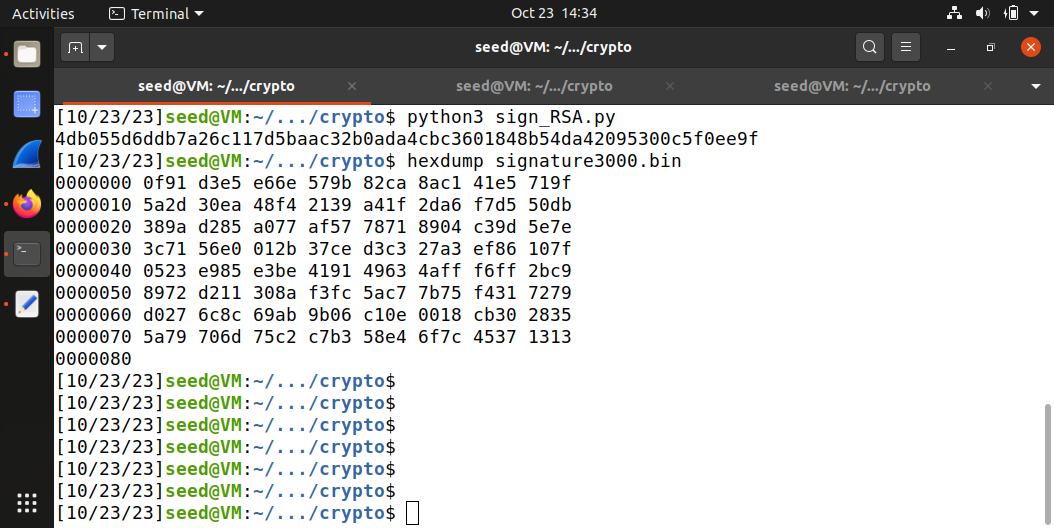
**• Signer.sign(hashedmessage):**

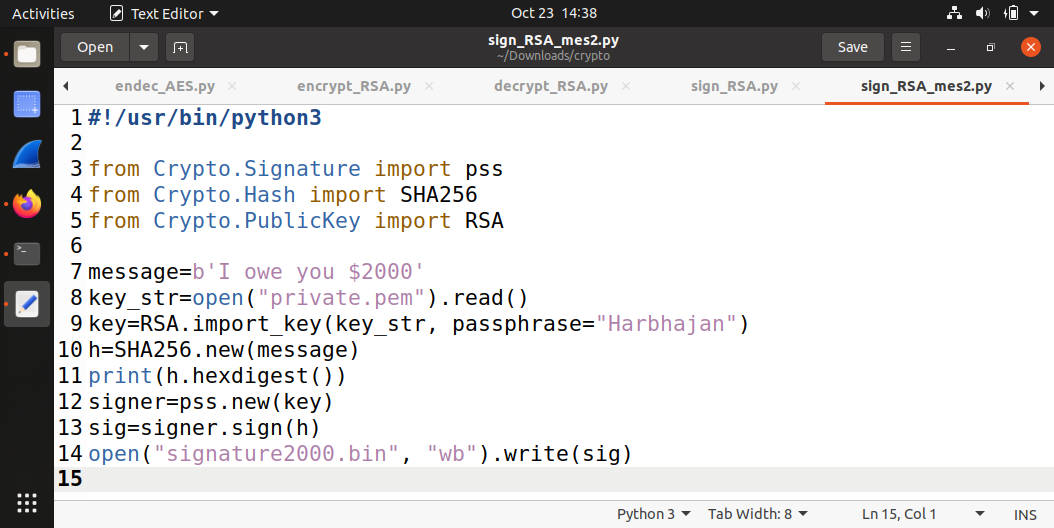
* **This generates the RSA signature of the hashed message. Here you can use SHA512 to generate the hash value of your message.**

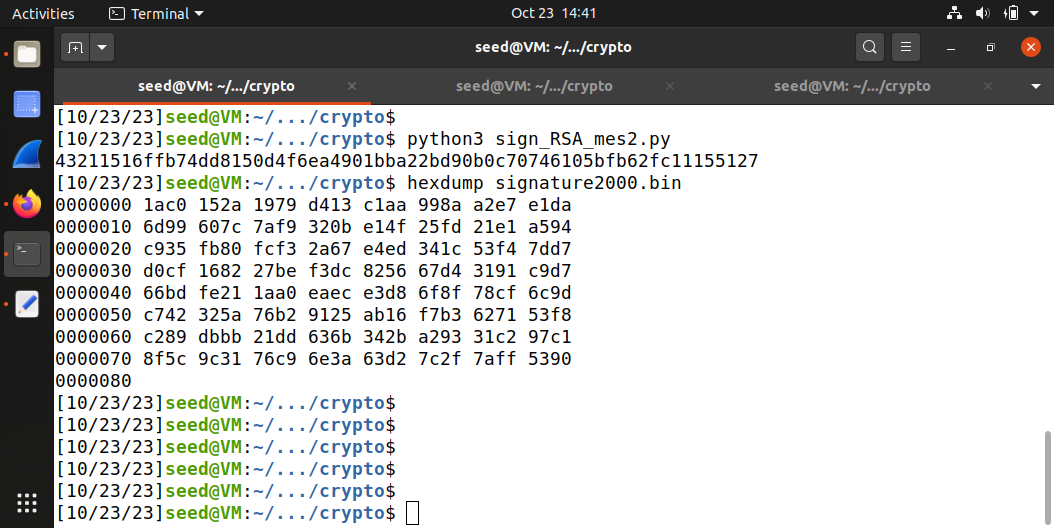
**M = “I owe you $2000”. Change $2000 to $3000 and sign the modified message. Compare both signatures. Are they similar? Save your signature into a file. Take a screen shot for your file content (using hexdump). Ref. sign\_RSA.py**

**ANSWER 3A:**









**B. Verify the signature in (a).The functions are described as follows.**

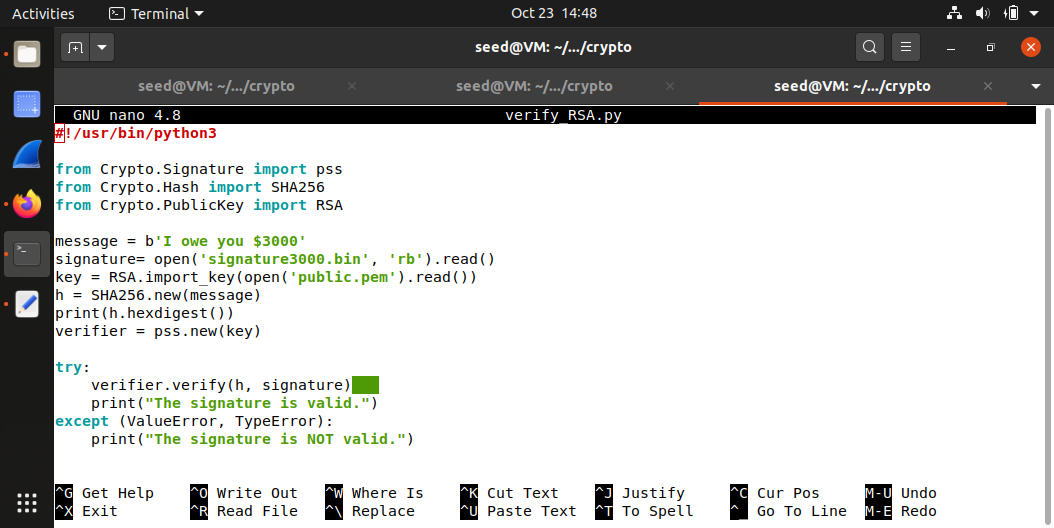
**• Signer=pss.new(RsaKey):**

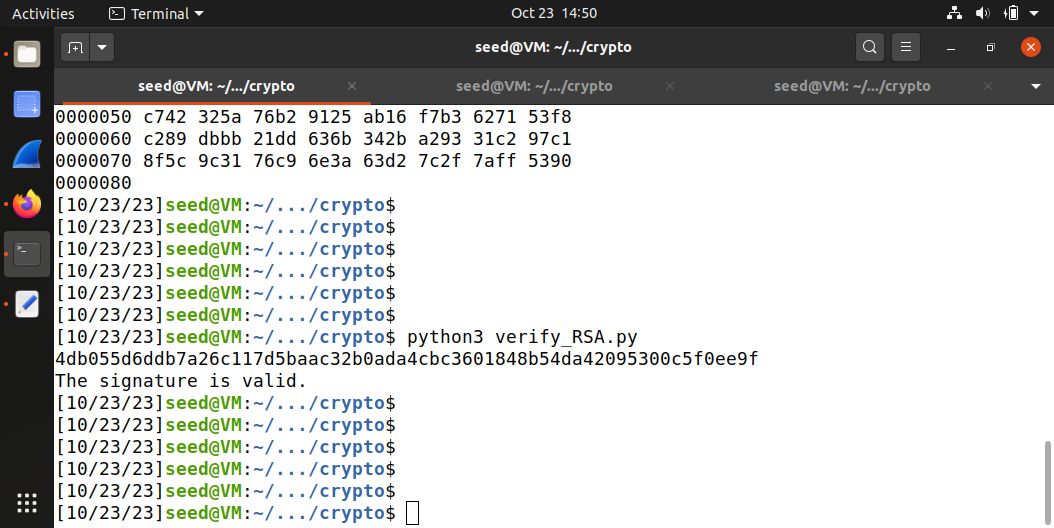
* **This defines a signing object signer with RsaKey (imported from your RSA public key file).**

**• Signer.verify(hashedmessage, signature):**

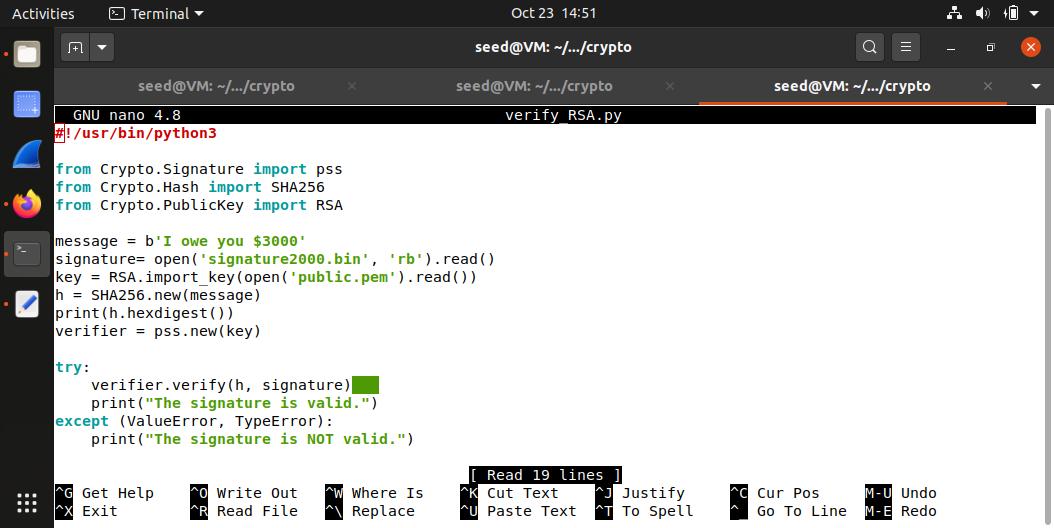
* **This verifies if signature is consistent with the hashed message. Take a screen shot for the output result. Ref. verify\_RSA.py**

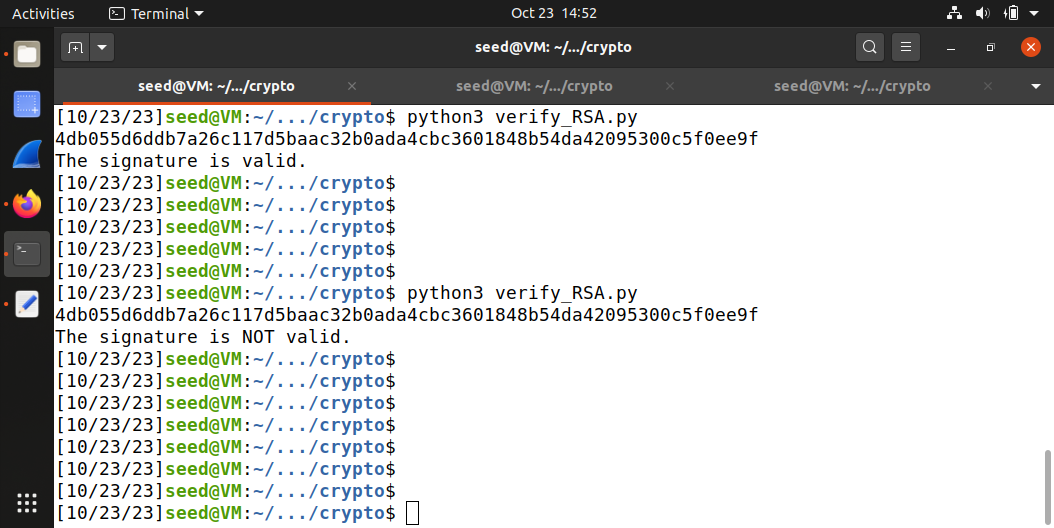
**ANSWER 3B:**





**NOT VALID:**





**4. In this problem, you will use Diffie-Hellman with authentication to protect the client-server communication. Implement the following functionalities.**

**a. Create two files: TCP client and TCP server, capable to chat with each other using socket.**

**ANSWER 4A:**

**References**

1. **Week 4 – Class 4 Notes**
2. **Week 4 – Class 4 Instruction Document**
3. **Code files provided by professor.**