Val Robichaux

Homework 4

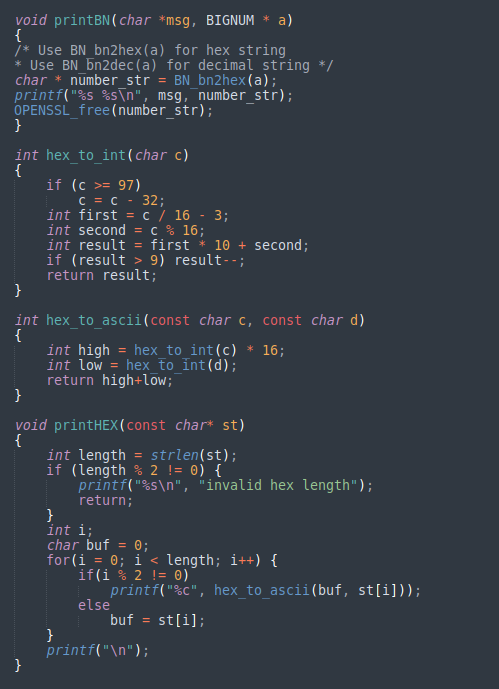
CSCE 465

# Deliverables

\*\* NOTE \*\*

I created four different function in order to produce results for printBN, hex to integer conversion, hex to ascii conversion and printing hex values as strings

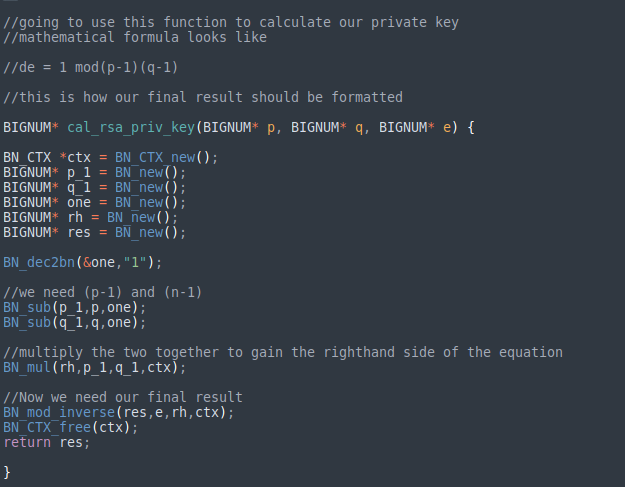
I am going to leave all of them here at the top because they are important to understanding my explanations of the tasks when I call these functions.



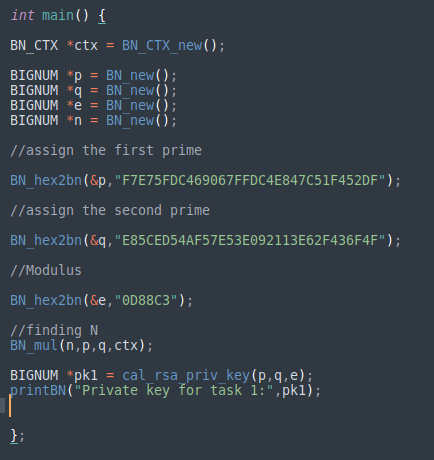
1. Deriving the Private Key



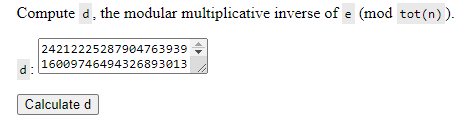
* Calculated private key

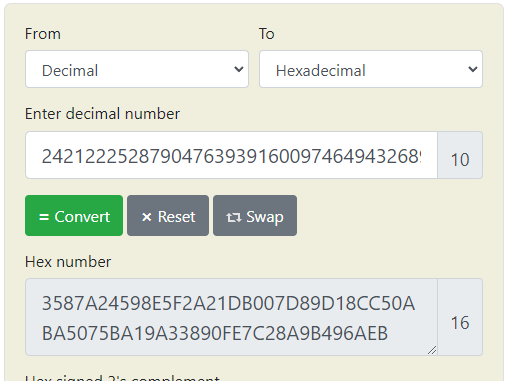


* The method I used where de - 1 mod(p-1)(q-1)



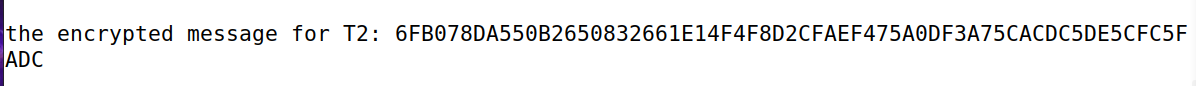
* Setting up the final result





* Finally checking if my math was correct and the two keys match!
* **Success!**

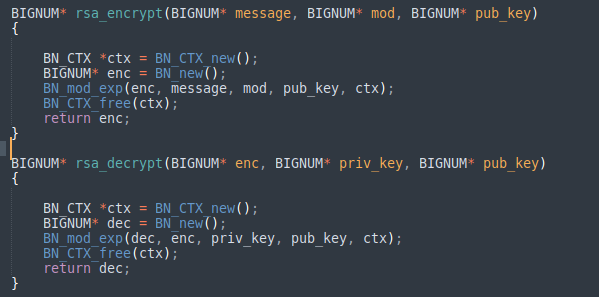
1. Encrypting a Message



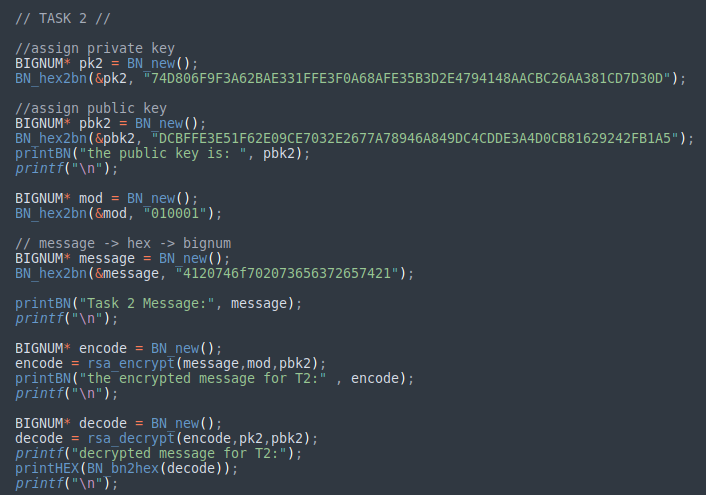
* This is firstly showing the encrypted message for Task 2



* This is the fully decrypted message for Task 2



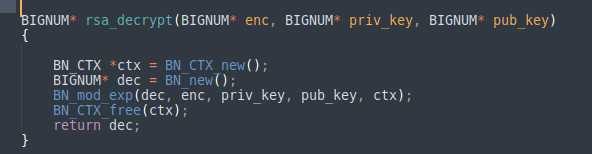
* Here my decryption and encryption and encryption functions are written
  + The encryption function accomplishes the following function using mod\_exp
  + The decryption function accomplishes a similar function but instead the decrypted formula is as follows
  + The decryption method allows the message to be decrypted if the user has access to the private key.

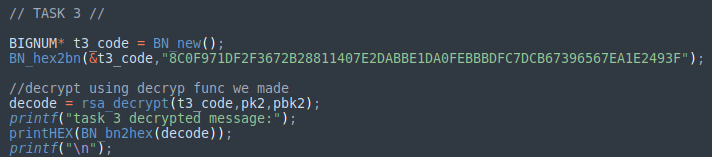


* + - * This is just the basic setup for task 2 using my functions that I created earlier.
      * I encode the message, print the encoded message, then decode the message and print the original message.

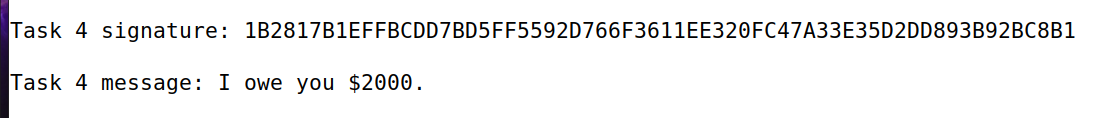
1. Decrypting a Message

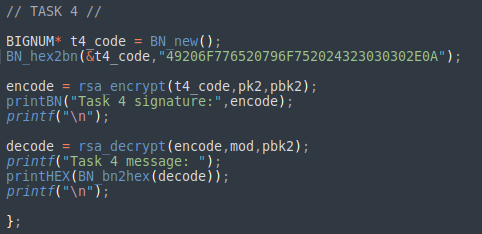
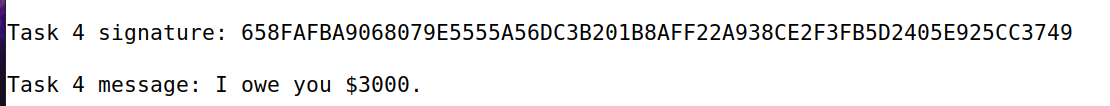






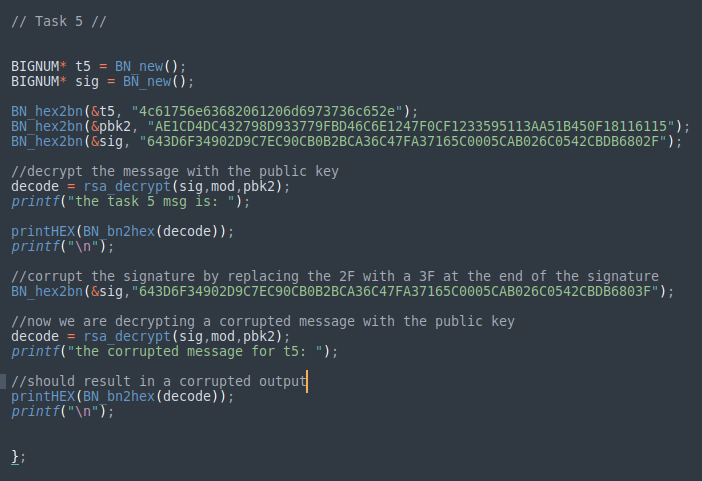
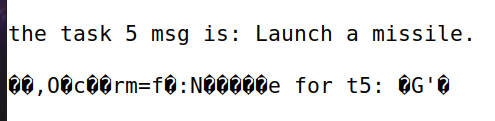
* + - * Since I already created a decryption function this task was relatively trivial.
      * It displayed the message “Password is dees”
      * This is funny because it is the password of the virtual machine that I am using.
      * Yet again, since the private key and public keys are the same as in task 2, the decryption is allowed since I have access to the private key

1. Signing a Message



* + - * Here I show the signed message in both cases
      * Case 1
        1. The calculation to get the signed message is as follows
        2. The encode variable that I use in this case signs the message with the encryption function because I am using the private key to write the message’s signature.
        3. I then use the public key to check if it is actually my message that I signed
      * Case 2
        1. Case 2 is a bit similar, but changing the message will change the signature slightly but it still will function the same because I will still check the message if it is my own.

1. Verifying a Signature



**Walking through my code**

* I assign the task 5 message, public key and the signature.
* I decrypt the message first off by using Alice’s signature, the modulus provided and the public key, and I can see that the message “Launch as missile’ is correct.
* Now I corrupt Alice’s signature by replacing some bits on the end of the string and decrypting it again using the same parameters as before
* This is now interesting because when I print out the message again it becomes corrupted
* I think the reasoning for this is because Alice signed the message thus making the decryption of that message reliant upon a correct signature
* This is how the system will verify that the message is in fact Alice’s original message
* Since we change the signature in the last step, the system can not verify that it is Alice’s message correctly and displays a corrupted message.