CSE13s Fall 2022
Assignment 5
Professor Miller
George Sono

Writeup

For this assignment, I tested my code on all functions individually before using them all in conjunction in keygen.c, encrypt.c and decrypt.c.

Regarding numtheory, I tested each individual numtheory function with valgrind by adding a main function in numtheory.c (which I later removed). I compiled by adding another option in the Makefile.

For the RSA library functions, I tested each individual part by doing the same thing with numtheory and by having an own main function in rsa.c. Again, using valgrind to make sure I had no memory leaks or errors at runtime.

Putting it all together, I tested my keygen, decrypt and encrypt programs by first generating the key pairs in keygen, then encrypting a plain text file (with the public key) and then finally decrypting the cipher text with the private key. To make sure that my code was properly working with stdin and stdout, I used the pipe operator in the UNIX terminal.

```
george@george-VirtualBox:~/cse13s/ysono/asgn5$ echo I LOVE CSE13S!
 | ./encrypt | ./decrypt
I LOVE CSE13S!
george@george-VirtualBox:~/cse13s/ysono/asgn5$
```

As seen in the code above, after generating the keys with keygen, I echoed “I LOVE CSE13S!” to stdout and used the pipe operator to convert stdout into stdin for the encrypt executable. I then repeated this process for the decrypt executable, until it ultimately prints the decrypted message into stdout (as seen in the final line). We know that this works as the original echo message is the same as stdout in the end.

Regarding file permissions, this was tested through viewing the file permissions of private and public key files after keygen produced them. It was shown that the private key file has only user read/write permissions while others have no access.