Jonathan Nguyen
jnguy330
Assignment 5

Assignment 5
Writeup
Public Key Cryptography
November 6, 2022

In assignment 5, I programmed six files named: randstate.c, numtheory.c, rsa.c, keygen.c, encrypt.c, and decrypt.c. Using all these files together I was able to create my own RSA encryption algorithm.

In randstate.c, I created two functions that initialized my seed and state used for the random functions that I used in the rest of my files.

In numtheory.c, I created five number theory functions that did all the mathematical calculations used to create my private and public keys. To make sure these functions worked correctly, I used online calculators and did some manual calculations myself to ensure my functions' outputs were precise.

In rsa.c, I used the numtheory.c functions to create 12 functions that created public and private keys, encrypted files, decrypted files, created signatures, and verified signatures. To test these functions I created a main function inside rsa.c that gave me my function outputs, then I manually did the calculations to make sure they matched correctly. To test my decrypt and encrypt function, I encrypted a file with a message, and if my decrypt outputted a file with the same message I knew my functions were working.

In keygen.c, I created a main function that used functions from rsa.c to create public and private keys and put them in their respective files. I was able to make sure keygen.c was working correctly by running my keygen.c and then running the example

binaries: encrypt-dist and decrypt-dist. If the output from decrypt-dist was the same as my input in encrypt-dist, I knew my keygen.c was working as intended.

In encrypt.c, I created a main function that used functions from rsa.c to encrypt an input file using a public key file and then output the cipher text to another file. To test encrypt.c, I ran the example binary, keygen-dist, to create keys, then I used my encrypt.c to encrypt a file. After I ran the example binary, decrypt-dist, and if the output was the same as my input to encrypt.c I knew my file was working correctly.

In decrypt.c, I created a main function that used functions from rsa.c to decrypt an input file using a private key file and then output the decrypted text into another file. To test decrypt.c, I ran example binaries: keygen-dist and encrypt-dist. After, I ran my decrypt.c file and if the output was correct I knew decrypt.c was working as intended.