Luca Schram
lschram
Assignment 5

Overall Takeaways:

This assignment was an interesting mixture of several different kinds of challenges, such as file management, formatting, and learning how to use the GNU Multiprecision Library (GMP). Using GMP and treating all variables as pointers was a difficult task, as temp variables had to be constantly allocated and deallocated to avoid changing original values passed in to functions. Furthermore, understanding how to encrypt and decrypt and encrypt entire files was extremely difficult.

Testing Methodology:

Most of this assignment's difficulty came from the number theory calculations that were initially implemented within numtheory.c, such as pow_mod and mod_inverse. They were somewhat confusing to implement with how random numbers were generated for them and with how mpz_t variables worked, as all of them functioned as pass by reference instead of pass by value. To test the validity of my functions outputs, I employed the use of online calculators to test multiple different values passed in to the functions. Furthermore, I tested the difference between my pow_mod and the gmp implementation of pow_mod to confirm its validity. Then, for the generation of public and private keys, I simply tested whether write_pub and write_priv were writing the correct bit-sized values to files. From there, I tested encrypt and decrypt_file using a tester main function to encrypt and decrypt a file I hardcoded into the main file. Finally, I wrote the executable files that handled key generation, encryption, and decryption. I debugged them primarily through the usage of the binary executables provided in piazza for each of those files. I gleaned the proper option ranges and help messages from each executable from each file. I also tested my keygen time complexity for high bit numbers, as well as the functionality of encrypt and decrypt when not provided file names. Most of my testing time was spent trying to improve the efficiency of my keygen executable to pass the Git tests. I think the most effective change I made was implementing short circuit calculations when making my prime numbers so that numbers that were below the required bit amount or even numbers were never checked with is_prime, saving me valuable runtime.