Key-p it Together

Angela Shen

February 2023

1 Abstract

Information in general has become more widely available through the internet, but some might want certain information to only be accessible by certain people. Many forms of digital communication rely on encryption so that people outside the conversation cannot eavesdrop. Social media is a good example of this, since many of them have encryption for direct messages. For me, Discord is the one I use on a day to day basis, so having my messages to other people encrypted greatly benefits me.

2 Introduction

Makefile - Compiles executables for keygen.c, encrypt.c, and decrypt.c and creates multiple object files (ss.o, numtheory.o, randstate.o, and the .o file of the respective c file). For a list of these c files, .c below.
README.md - Details the process of building any necessary files, the command line options for any executables, and any errors or bugs.
DESIGN.pdf - Contains the pseudo code and descriptions of each c file.
WRITEUP.pdf - This document. Describes the assignment in its entirety and discusses the results.
keygen.c - Generates a public key and a private key and writes them into separate files
encrypt.c - Takes input from a file, encrypts the message using the public key, and outputs them into a file.
decrypt.c - Takes input from a file, decrypts the message using the private key, and outputs it into a file.
randstate.c - Contains functions to initalize and clear a random state for generating random mpz_t integers.
numtheory.c - Contains mathematical functions used in Schmidt Samoa.
ss.c - Contains functions necessary to implement Schmidt Samoa.

3 Learning

3.1 Math

I put a lot of effort into understanding the math behind the numtheory.c functions, since I primarily relied on the pseudocode rather than implementing it on my own. I think writing out all the formulas was interesting and good practice, though maybe not the most efficient way to go about the assignment. It was really interesting to see pretty simple math used to implement more complicated formulas, and how the functions of numtheory.c contributed to the implementation of Schmidt Samoa.

3.2 mpz_t

I was unsure of what to expect when Professor Long described gnu multi-precision library as terrible, but I have to agree. While very useful practically, programming and debugging . I implemented numtheory.c entirely in C code first, and managed to get it functioning correctly fairly easily. The mpz_t equivalent took hours to debug because there were multiple typos (like putting the wrong variable as the parameter for a function, or forgetting to clear a variable) that were easy to miss.

3.3 Debugging

While I have used gdb, valgrind, and scan-build in other assignments, this assignment definitely helped me understand their utility much better. For example, I had an issue with a floating point exception in keygen, which was quite worrying because I had debugged my numtheory functions, and therefore had no clue where the error might be. Then gdb came in and saved my life, instantly pointing to the place where the exception occurred.

Valgrind and scan-build also came in handy because there were places where I was freeing pointers or clearing mpz_t after return values. Most numtheory.c function do not return anything, so I was clearing mpz_t variables at the end of the file. However, is_prime returns a boolean and has multiple return statements, so clearing at the end of the file resulted in thousands of blocks of uncleared heap memory, which I was able to check with valgrind. Scan-build caught some things that valgrind did not, since most of them did not cause segmentation faults. I forgot that since I have a return after calling the help function for my executables, any pointers initialized at the start of main need to be freed. I also learned what "garbage variables" are. They are uninitialized variables, which I had multiple of for parsing getopt(), but it is probably better to initialize them to ensure that I know what the values of the variables are.

4 Conclusion

I have noticed a pattern of assignments with a lot of math taking much longer than other assignments, since I get bogged down trying to understand exactly how it works. This is extremely concerning, but I will strive to do better and learn more. And also manage my time better. Overall, I think I learned many important skills and lessons from this assignment.

5 Credit and Notes

All numtheory.c functions were based on pseudocode from the assignment 5 spec. This is acknowledged in the comments of the functions.