Kat Pe Benito

Professor Veenstra

CSE13S

13 February 2023

Assignment 5 Design Doc.

Description of Program:

This assignment consists of three programs, keygen, encrypt, and decrypt. Keygen is meant to generate a key, producing SS public and private key pairs. The encrypt program is going to encrypt files using a public key, while the decrypt program will decrypt the encrypted files using a private key using the GNU multiple precision arithmetic library, a library to hold functions relating to mathematics behind SS, and a random state module.

Files to be included in directory "asgn5":

- decrypt.c: contains implementation and main() function for decrypt program
- encrypt.c: contains implementation and main() function for encrypt program

  -i : specifies the input file to encrypt (default: stdin).

  • -o : specifies the output file to encrypt (default: stdout).

  • -n : specifies the file containing the public key (default: ss.pub).

  • -v : enables verbose output.

  • -h : displays program synopsis and usage

- keygen.c: contains implementation and main() function for keygen program

  • -b : specifies the minimum bits needed for the public modulus n.

• -i : specifies the number of Miller-Rabin iterations for testing primes (default: 50).

• -n pbfile : specifies the public key file (default: ss.pub).

• -d pvfile : specifies the private key file (default: ss.priv).

• -s : specifies the random seed for the random state initialization (default: the seconds since the

UNIX epoch, given by time(NULL)).

• -v : enables verbose output.

• -h : displays program synopsis and usage.

- numtheory.c: contains implementations of the number theory functions
- numtheory.h: specifies interface for number theory functions
- randstate.c: contains implementation of the random state interface for the SS library and number theory functions
- randstate.h: specifies the interface for initializing and clearing the random state.
- ss.c: contains implementation of SS library
- ss.h: specifies the interface for the SS library

- Makefile: Makes and cleans all c files, compiles and formats them as well, pkg-config to locate compilation and include flags for GMP library
- README.md: Describes how to use script and Makefile in Markdown syntax, explains command-line options that program accepts. Lists and bugs or errors if any
- DESIGN.pdf: Describes design for program thoroughly with pseudocode and descriptions.

- WRITEUP.pdf: What was learned from this assignment, applications of public-private cryptography and how it influences the world today, one way in which you personally take advantage of it on a day to day basis.

Pseudocode:

- **decrypt.c**
- headers, define options,
- int main(), set files to their defaults
- for each case open and read or write to files.
- read private key, print stats of private key
- decrypt file, and close files
- **encrypt.c**
- headers, define options
- int main() set files to their defaults open
- for each command line case open and read or write to appropriate files
- set buffer for username
- read input for public key, print stats of public key
- convert username string to mpz_t
- verify signature and encrypt input
- close files

- **keygen.c**
- headers, define options

- open pub and priv
- set defaults for keygen,iters and seed
- initialize random state var
- create public keys
- create priv key and print stats
- close files
- **numtheory.c**
    - include appropriate headers
    - calculate power mod of a^d and mod n, return void take in
- **randstate.c:**
- initialize state variable with seed, then clear it

Credit:

- I often like to watch youtube videos beforehand on the topic I'll be coding about just to get some insight. I watched "Public Key Cryptography" by channel, "Computerphile".