Hongyi Zhang
860976097
cs165: assn2

Server:

In my function run_server_connect() function, I first setup the server connection using the command line argument --port=XXXX (X={1,9}, ex: 1234). Once the client connect, the server accepts and does a handshake with client. Then, I open up public.pem and private.pem and uses the encrypted random_int sent by the client. Using the private key(rsa_private), the server decrypts the encrypted random_int. After, the server create a hash_buf where the server encrypts it with its rsa_private and then sent it to the client. Then the server waits for the client, if everything is working on the client side, the client would sent to the server the option of --send or --receive. If its --send, then the decrypt and write the contain of the file into a temp file called SERVER_HOLD.enc. If --receive is selected from the client, then the server would open SERVER_HOLD.enc and write it to the client. If any of the steps mentioned above fail or does not match, then the function run_server_connect() return -1.

Client:

In my run_connect() function, I pass in the server_address, port, send_receive and the file from command line arguments. The first thing in this function is that I establish the connection using server_address and port [server_addres:port] and do a handshake with the server. Then I upload the public.pem (rsa_public) and encrypt a random_int generated by (rand()%10000)+33 with rsa_public key. Then, sent this encrypted key over to the server, and then the server would decrypt it and verify it. To verify if my client and server encrypt and decrypt correctly, i output the random number before its encrypted in the client, and outputted the random number after the server decrypts it. After, it would generate its open hash table and then it would compare hash to the received server hash after decrypting the server has. If any error occurs, run_connect() would return -1 and exit the program. Then based off the command line option, if --send is selected, the client would open the file and send it over to the server where the server saves it in SERVER_HOLD.enc. If --receive is selected, then the server would send SERVER_HOLD.enc info over to the client. To verify it, my client outputs the received file content in out.dcry.

Instructions:

to run server: ./server --port=1234
to run client(send): ./client --serveraddress=localhost --port=1234 --send sample.txt
to run client(receive):./client --serveraddress=localhost --port=1234 --receive sample.txt

If the server and client program does what it suppose to, then they both would output “success”.

To save to the server(when sending), my_server.c writes to SERVER_HOLD.enc.
To compare the correctness of sample.txt, my program outputs the received file to out.dcry file.

*make clean will remove all the files that was added.