**MiniRSA**

**COLLABROTION**

Sheng Huang and Yao Chen made this RSA-based chatting program. We mainly worked together in a working mode called pair programming. And this is the github address for our work: <https://github.com/yaochen1025/MiniRSA> .

**WHAT IS IT?**

This program is a mini chat application using RSA cryptosystem. It consists of 3 parts: RSA cryptosystem, terminal version of server and client, GUI version of chat program.

-RSA package contains a java class of tools for encrypting and decrypting with unit test for them. It also has three classes with main function to simulate encryption, decryption, and cracking the key.

-Terminal version asks user to input information and calculates both public and private key. It then sets up a connection between server and client and allows users chat with each other.

-GUI version generates user interface when program starts and allows visualized operations by user.

**Structures and Approaches**

**Packages:**

**-gui**

This package contains two java class. GUI.java sets up gui and acts as message sender. The Reader.java acts as another thread working to receive messages. All the chatting content will reflect in the chattingArea.

Another thing to mention is that, there is no distinction between a client and a server in this class. The object will first try to act as a client, and if there is no server available, it will become a server automatically, listening for others. **So the firstly-run program will be server, and the secondly-run program will become the client.**

**-modular**

This package contains two java class that have main method: Server.java and Client.java. Others are supportive. Both Server.java and Client.java extends the ChatProgram class which is a wrapper of setting up connection as well as sending-key and receiving-key. Two class: MessageSender and MessageRecver extends Thread, so that sending and receiving messages can be asynchronously. And inside these two class, they communicate via a full duplex socket which provides input stream and output stream. The encryptor class is just a wrapper for encrypting and decrypting.

**-rsa**

This class is the rsa system. According to the specification in the assignment, you can run Encryptor, Decryptor, and Cracker to see whether the whole cryptosystem works. RSA.java is a full box of methods, and RSATest is the unit test for it.

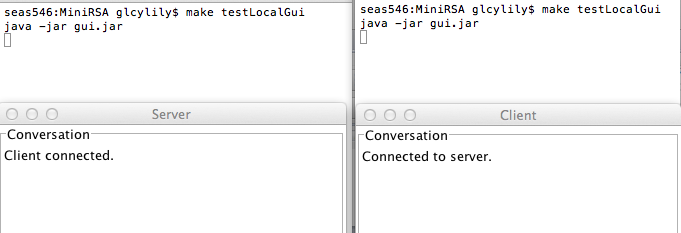
**Sample Run:**

1.  ***make build***  (prepares all the jar files needed)

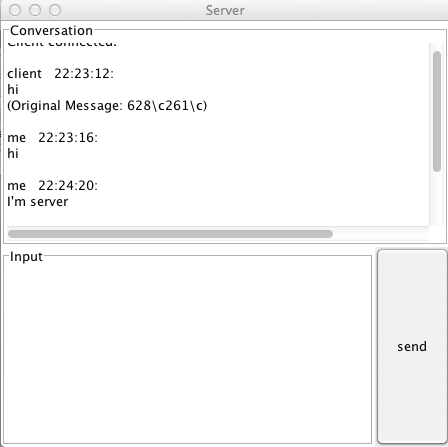


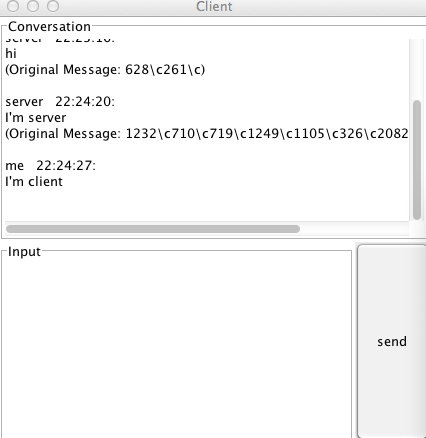
2. To test both end of GUI on local host:

In our GUI version, there is no difference between server and client. User can simply open two terminals and input command line twice: ***make testLocalGui.*** The one runs first would automatically be the server and latter one would be client.



GUI:



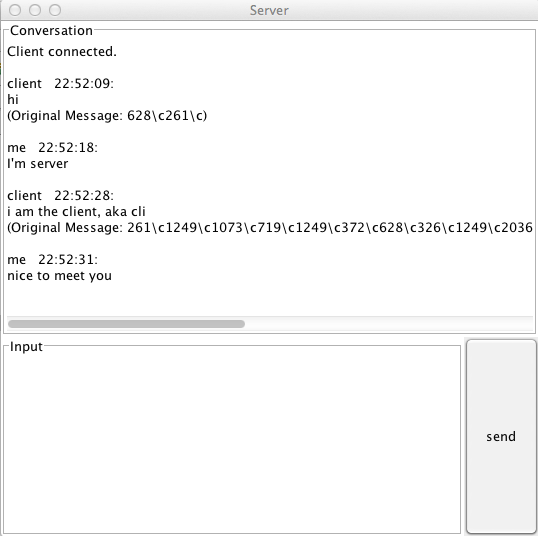


3. To test remote chatting:

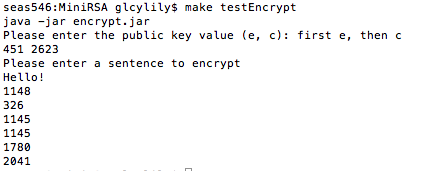
For one end, ***java –jar gui.jar <ipaddress> <port>***

For the other end, ***java –jar gui.jar <ipaddress> <port>***

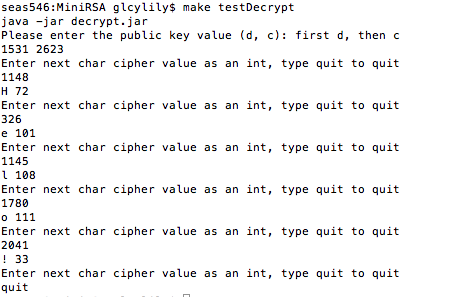
The port number should be the same



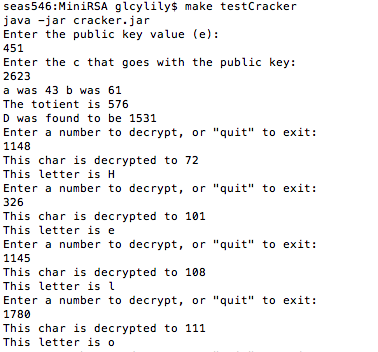
4. To test encrypt:



5. To test decrypt:



6. To test cracker



7. To test terminal version:

***make runTerminalServer***

***make runTerminalClient***

