Requirement Analysis Document

Note: all the highlighted part should be submitted by November 10th

# **Project name**

Chatting app with encryption

# **Team name**

Coffee talk

# **Team members**

20185083 서영석

20211182 Kornkamol Anasart

# **Scope of the project**

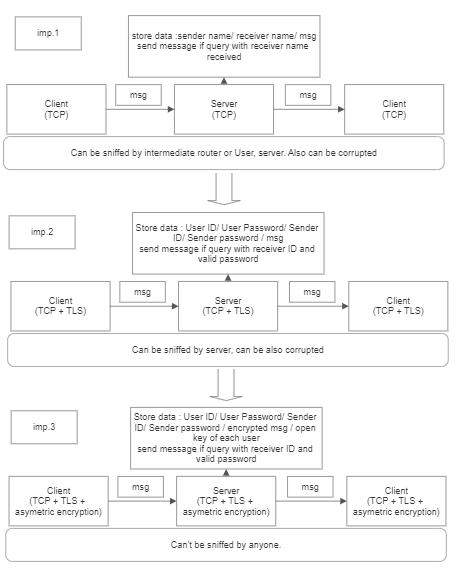
**You will have to describe your project in one paragraph.**

We’ll make program that not to be sniffed by others, including server. First, we’ll make simple chat program with no secure. It will send and store message itself to server. And when host request, it will give message. This can be sniffed by other packet deliver and server itself. Next, we will encrypt host- server connection by TLS. Plus, we will add password-id database at server. This will prevent message to be sniffed by outside packet deliver. But, This can be sniffed by the server itself, cause server knows password. Therefore, we will encrypting message by asymmetric encryption so that only two endpoint user can decrypting it. By this, except for endpoint host, we guarantee that this message can’t be sniffed.

By conduct this project, We expect to learn

1. Server-client model in application
2. Transport layer security
3. ID – Password login system
4. End to end point security
5. Overall encryption on data transfer.

# **Representative diagram to explain your project**



**Tools used (webRTC, socket programming, Tshark, Rest, java, c++, python, NS2, Mininet)**

Development language – python

Library – Socket, TCP, TLS, AES python library, curses – TUI library.

Database - SQL

**Tentative Input**

Messages : text, image, video, etc

**Tentative Output**

For first connection : public key of both endpoint

For extra connection : encrypted message

# **Limitations of your project**

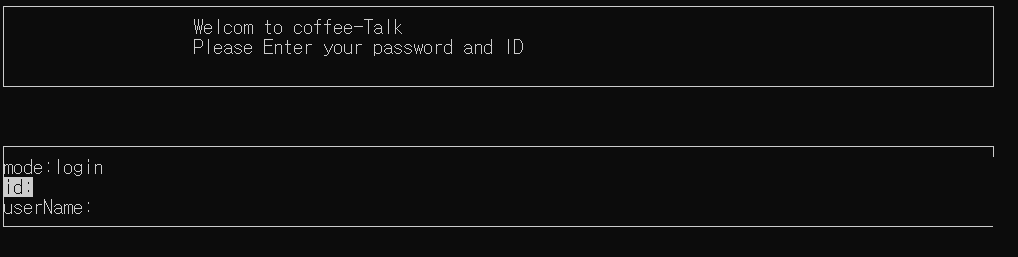
1. Cause we use RSA based private key, It is hard to remember private key by user’s mind.
2. Cause we use RSA key, we can’t make group chat by this algorithm.

# **Sample input and output**

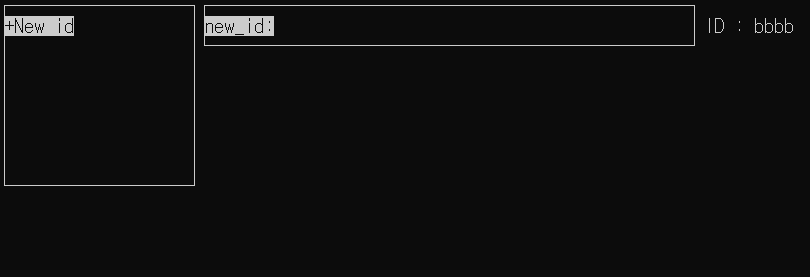
All TUI is implemented from curses library.

Login Window.

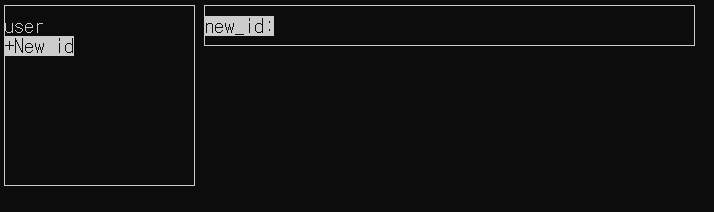
we can register or login at here.



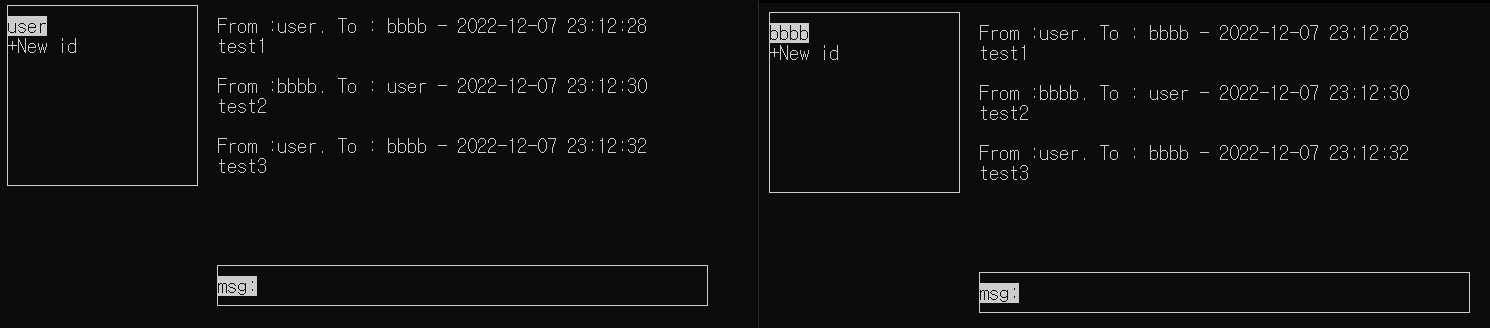
At first login it is looked like this. We can add friend from +New\_id.



After we type friend ID, it is added to friend window.

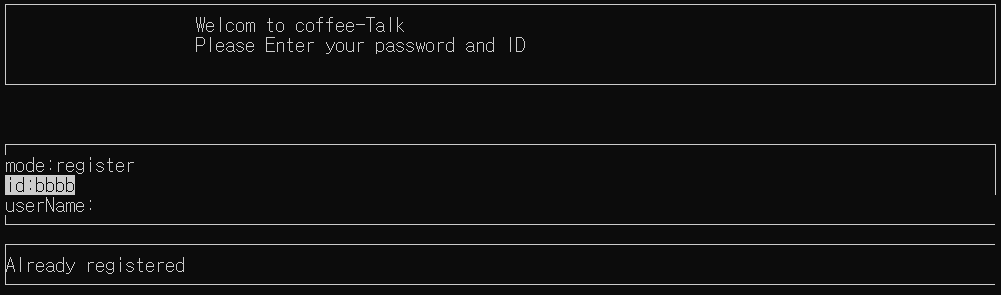


By selecting friend name and type messages, we can send to friend.

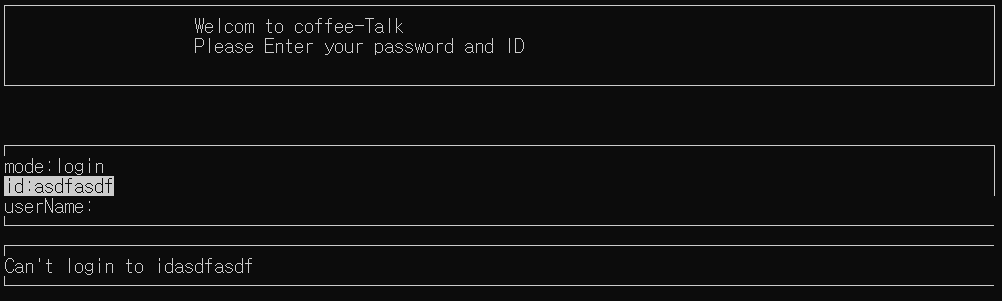


(left one is screen for userID:bbbb and right one is screen for userID: user)

Sample Error case1) try to add already existing ID



Sample Error case2) try to login with unvalid ID.



Sample Error case3) try to add friend who is not exist. 