The following contains the Server and Client code for a CLI based chatbox. It allows multiple users to connect to the server and interact among themselves. The code has been developed to function over the localhost only. But the same can be extended to function over the internet.

SALIENT FEATURES

Multi user interface :

A large no. of users can connect to the server provided they have the Client code and interact amongst themselves. There are NO restrictions on the no. of users who can connect at the same time in the program(Although being a multithreaded program, with the no. of threads increasing with the no. of new users, the memory of the computer on which the server is running might be a restricting factor)

Flexibility :

Users can enter and leave as per there convenience and it won't affect the functioning of the program. Although the users who entered late won't have any access to the previous messages shared among the users.

Supported in all OS :

The program can be perfectly run in all operating systems as long as Python 3 is installed. NO extra libraries are required.

Encrypted Messages :

The messages are encrypted before sending. Hence in case of a "Man in the middle" attack, no data loss will occur. The encryption procedure is very simple. A random no. (the private key) in the range 0 to 255 is generated as the server starts running. This no. is sent to the client as soon as the connection is established. The ascii value of every character is XORed with the _key and sent to the server, and the encrypted message is broadcasted to all the other clients. The recipients follow the same procedure of XORing to get the message back.

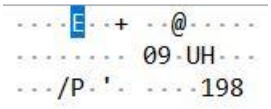
FOLLOWING ALONG WITH THE PACKET CAPTURE :

The .pcapng file includes the traffic capture corresponding to the following communications between port 52684 and 52565.

52684 : So it starts!
52684 : Right?

52565 : Yep!
52565 : Here we go!

The 1st three packets correspond to the TLS handshake with port 52565 of localhost and the next two packets correspond to the sharing of the key with port 52625.



```
····E··+ ··@·····
········ 09·UH···
···/P·'· ····198
```

It can be observed that the key sent was 198. And the XORing occurs using the private key

198 – 10 = 188 as per the client program (Yeah! I know it's a very stupid relation between the private and public key!)

In a similar manner the next 5 packets correspond to the TLS handshake and key sharing with port 52684 of localhost.

Remaining packets (except last two) correspond to the conversation. (Try extracting the messages by following any one TCP stream and filtering the packets with PSH flag. It shouldn't be possible unless you know the key, as it's encrypted!)

The last two packets (RST flags) correspond to the abrupt closing of the connection by closing the server down!

DISADVANTAGES :

Restriction on message length :

It can be seen in the program that the buffer size is 1024. Therefore there will be a limitation to the size of the messages. However that problem won't occur upto a stretch of atleast 1000 characters, therefore you're safe as long as you're not planning on sending a really long essay or something! Even if you are, just break the message. That's it!

Only Text Messages :

You can send only text messages, no multi media files. (But hang on to your hats, work's on progress!)