COMPUTER NETWORK

ASSIGNMENT 2

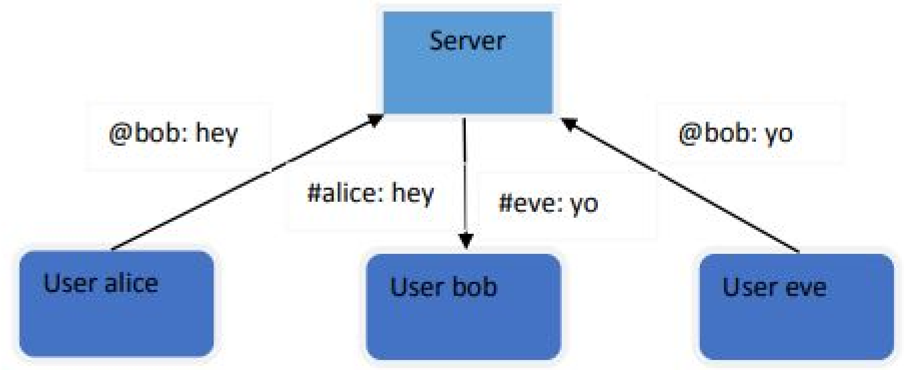
15 September 2019

AKSHAY GUPTA
MCS192556

PRITESH KUMAR SRIVASTAVA
MCS192658

Overview

In this assignment, we will build a chat application that allows users to do an encrypted chat with one another, which cannot be decrypted by the chat server. Users can direct messages to other users through the format given @[recipient name]:[message], and the server needs to forward these messages to the intended recipient. The message itself would be encrypted between any pair of users so that the server cannot read the messages, the server can only infer that a communication is happening between the given pair of users. This is pretty much what Whatsapp's architecture is all about, so here is our small whatsapp which will be running on our terminal.



Goals

1. To successfully send the data from one client to another.
2. If user wants the data to be encrypted then it will ask user if he wants data to be encrypted and then it will be encrypted and will only be decrypted only at the final destination, the server can't read the encrypted text as will be encrypted end to end and can only be read by the corresponding clients.
3. We will verify the sender using the signature and provide the best possible security to the user.

Milestones

I. Registering the user

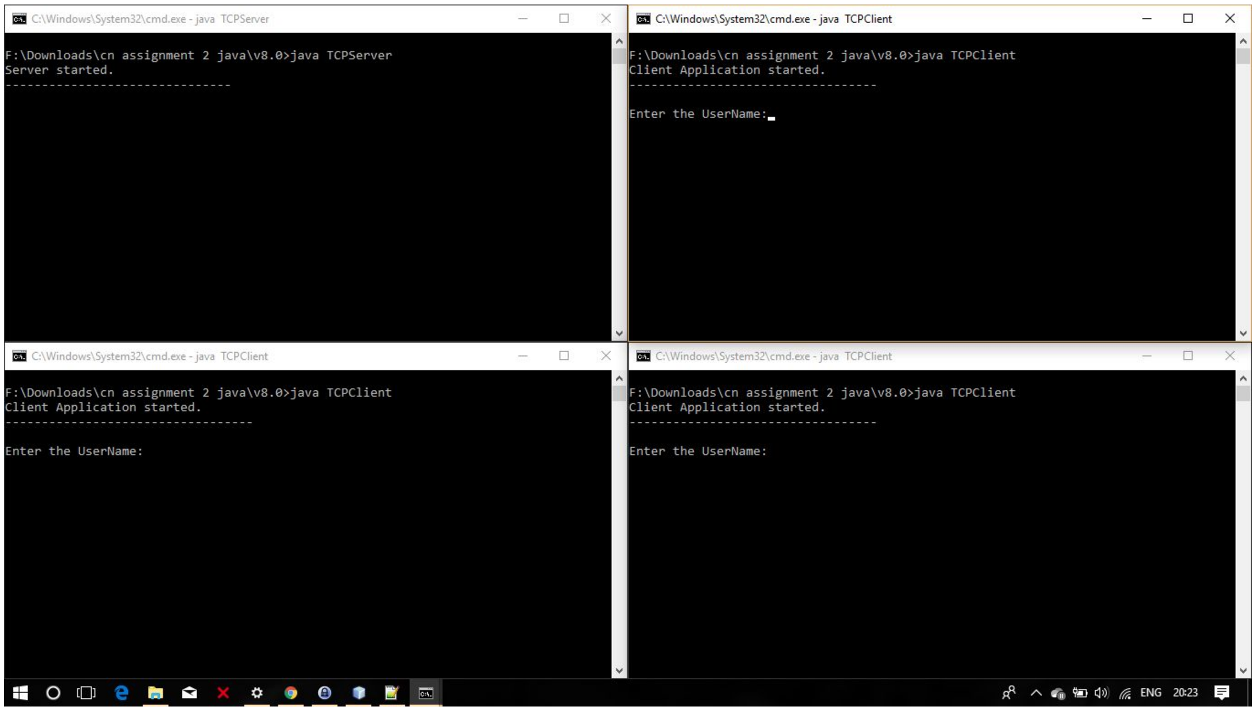
II. Sending data from one Client to Another

III. Encrypting the User Data

IV. Setting up the Signature to verify the sender

Scenario description

Server up


C:\Windows\System32\cmd.exe - java TCPServer

```
F:\Downloads\cn assignment 2 java\v8.0>java TCPServer
Server started.
-------------------------------
```

C:\Windows\System32\cmd.exe - java TCPClient

```
F:\Downloads\cn assignment 2 java\v8.0>java TCPClient
Client Application started.
----------------------------------

Enter the UserName:
```

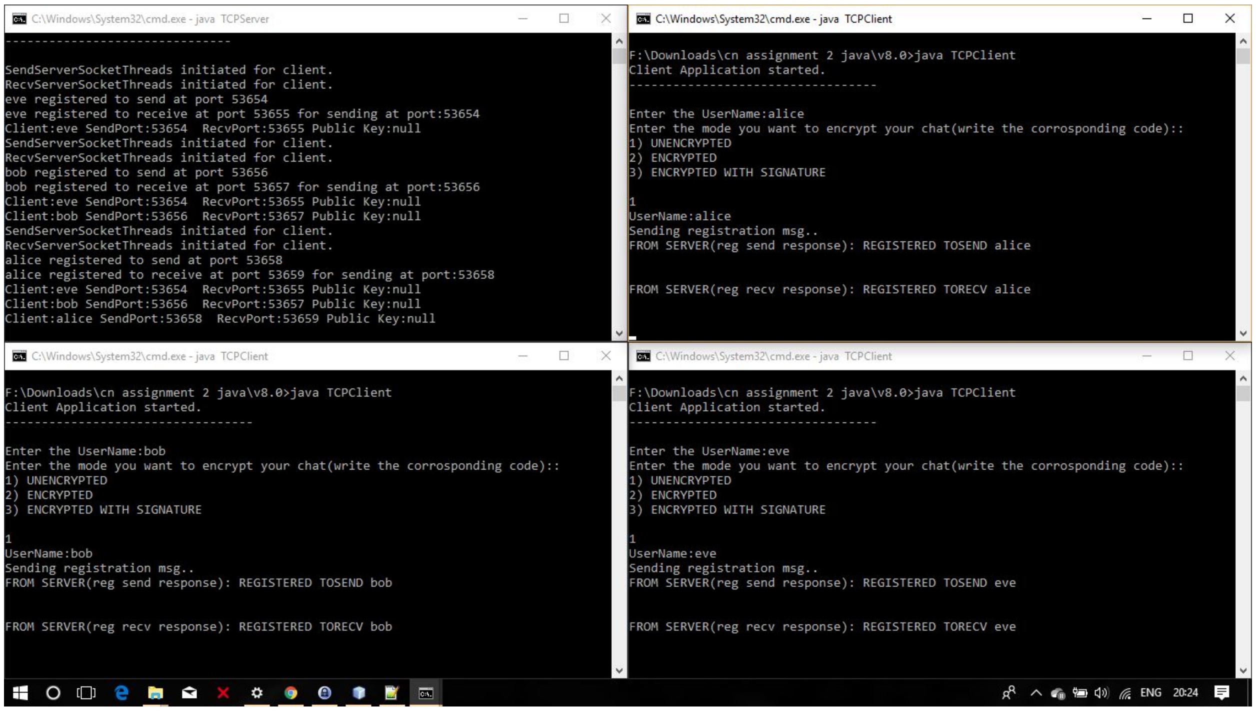
C:\Windows\System32\cmd.exe - java TCPClient

```
F:\Downloads\cn assignment 2 java\v8.0>java TCPClient
Client Application started.
----------------------------------

Enter the UserName:
```

C:\Windows\System32\cmd.exe - java TCPClient

```
F:\Downloads\cn assignment 2 java\v8.0>java TCPClient
Client Application started.
----------------------------------

Enter the UserName:
```

Clients up and getting registered with the server


C:\Windows\System32\cmd.exe - java TCPServer
C:\Windows\System32\cmd.exe - java TCPClient
C:\Windows\System32\cmd.exe - java TCPClient
C:\Windows\System32\cmd.exe - java TCPClient
ENG
20:24

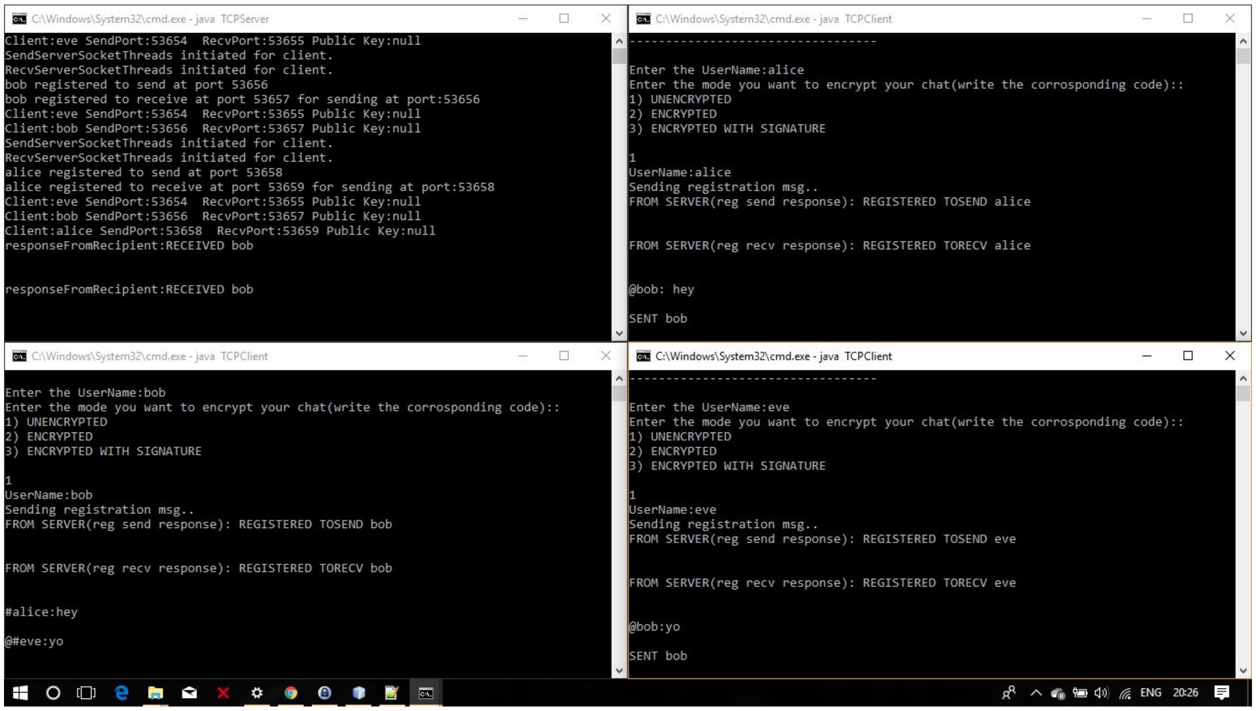
```
------------------------------
SendServerSocketThreads initiated for client.
RecvServerSocketThreads initiated for client.
eve registered to send at port 53654
eve registered to receive at port 53655 for sending at port:53654
Client:eve SendPort:53654  RecvPort:53655 Public Key:null
SendServerSocketThreads initiated for client.
RecvServerSocketThreads initiated for client.
bob registered to send at port 53656
bob registered to receive at port 53657 for sending at port:53656
Client:eve SendPort:53654  RecvPort:53655 Public Key:null
Client:bob SendPort:53656  RecvPort:53657 Public Key:null
SendServerSocketThreads initiated for client.
RecvServerSocketThreads initiated for client.
alice registered to send at port 53658
alice registered to receive at port 53659 for sending at port:53658
Client:eve SendPort:53654  RecvPort:53655 Public Key:null
Client:bob SendPort:53656  RecvPort:53657 Public Key:null
Client:alice SendPort:53658  RecvPort:53659 Public Key:null
```

```
F:\Downloads\cn assignment 2 java\v8.0>java TCPClient
Client Application started.
----------------------------------

Enter the UserName:alice
Enter the mode you want to encrypt your chat(write the corrosponding code)::
1) UNENCRYPTED
2) ENCRYPTED
3) ENCRYPTED WITH SIGNATURE

1
UserName:alice
Sending registration msg..
FROM SERVER(reg send response): REGISTERED TOSEND alice


FROM SERVER(reg recv response): REGISTERED TORECV alice
```

```
F:\Downloads\cn assignment 2 java\v8.0>java TCPClient
Client Application started.
----------------------------------

Enter the UserName:bob
Enter the mode you want to encrypt your chat(write the corrosponding code)::
1) UNENCRYPTED
2) ENCRYPTED
3) ENCRYPTED WITH SIGNATURE

1
UserName:bob
Sending registration msg..
FROM SERVER(reg send response): REGISTERED TOSEND bob


FROM SERVER(reg recv response): REGISTERED TORECV bob
```

```
F:\Downloads\cn assignment 2 java\v8.0>java TCPClient
Client Application started.
----------------------------------

Enter the UserName:eve
Enter the mode you want to encrypt your chat(write the corrosponding code)::
1) UNENCRYPTED
2) ENCRYPTED
3) ENCRYPTED WITH SIGNATURE

1
UserName:eve
Sending registration msg..
FROM SERVER(reg send response): REGISTERED TOSEND eve


FROM SERVER(reg recv response): REGISTERED TORECV eve
```

Alice and eve sending message to the bob



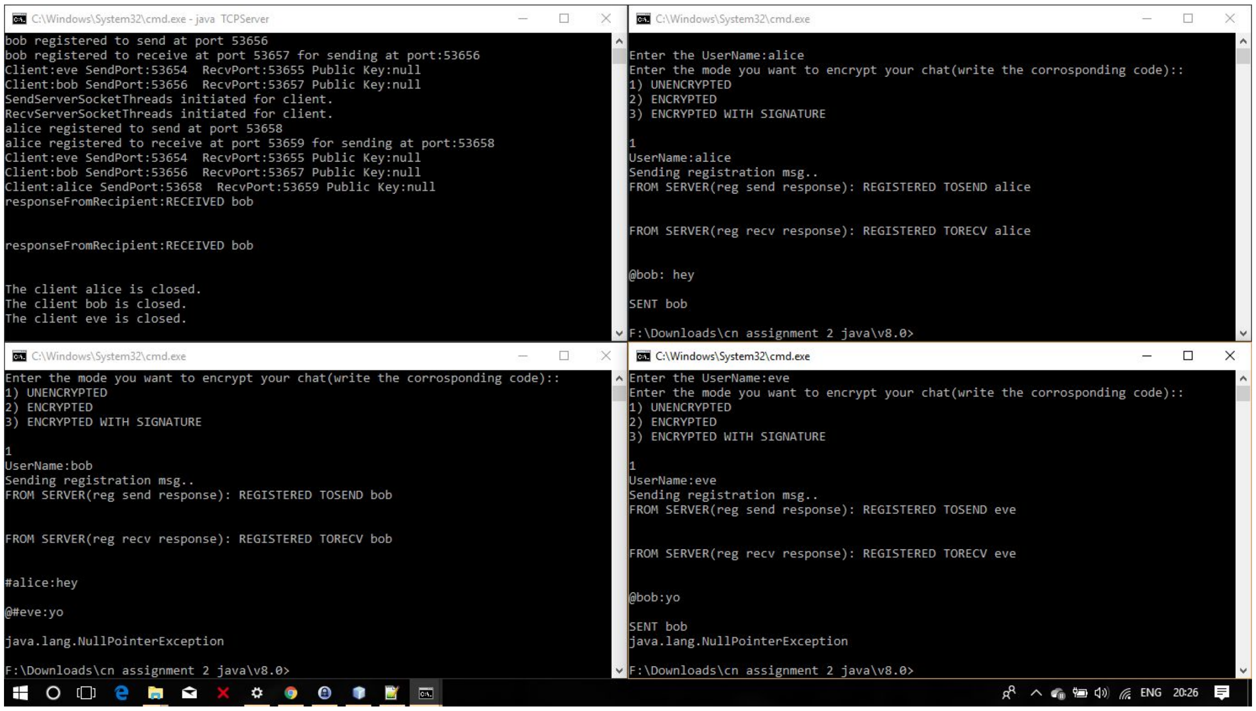
```
C:\Windows\System32\cmd.exe - java TCPServer
Client:eve SendPort:53654  RecvPort:53655 Public Key:null
SendServerSocketThreads initiated for client.
RecvServerSocketThreads initiated for client.
bob registered to send at port 53656
bob registered to receive at port 53657 for sending at port:53656
Client:eve SendPort:53654  RecvPort:53655 Public Key:null
Client:bob SendPort:53656  RecvPort:53657 Public Key:null
SendServerSocketThreads initiated for client.
RecvServerSocketThreads initiated for client.
alice registered to send at port 53658
alice registered to receive at port 53659 for sending at port:53658
Client:eve SendPort:53654  RecvPort:53655 Public Key:null
Client:bob SendPort:53656  RecvPort:53657 Public Key:null
Client:alice SendPort:53658  RecvPort:53659 Public Key:null
responseFromRecipient:RECEIVED bob


responseFromRecipient:RECEIVED bob
```

```
C:\Windows\System32\cmd.exe - java TCPClient
----------------------------------

Enter the UserName:alice
Enter the mode you want to encrypt your chat(write the corrosponding code)::
1) UNENCRYPTED
2) ENCRYPTED
3) ENCRYPTED WITH SIGNATURE

1
UserName:alice
Sending registration msg..
FROM SERVER(reg send response): REGISTERED TOSEND alice


FROM SERVER(reg recv response): REGISTERED TORECV alice


@bob: hey

SENT bob
```

```
C:\Windows\System32\cmd.exe - java TCPClient

Enter the UserName:bob
Enter the mode you want to encrypt your chat(write the corrosponding code)::
1) UNENCRYPTED
2) ENCRYPTED
3) ENCRYPTED WITH SIGNATURE

1
UserName:bob
Sending registration msg..
FROM SERVER(reg send response): REGISTERED TOSEND bob


FROM SERVER(reg recv response): REGISTERED TORECV bob


#alice:hey

@#eve:yo
```

```
C:\Windows\System32\cmd.exe - java TCPClient
----------------------------------

Enter the UserName:eve
Enter the mode you want to encrypt your chat(write the corrosponding code)::
1) UNENCRYPTED
2) ENCRYPTED
3) ENCRYPTED WITH SIGNATURE

1
UserName:eve
Sending registration msg..
FROM SERVER(reg send response): REGISTERED TOSEND eve


FROM SERVER(reg recv response): REGISTERED TORECV eve


@bob:yo

SENT bob
```

Closing the clients


C:\Windows\System32\cmd.exe - java TCPServer
C:\Windows\System32\cmd.exe
C:\Windows\System32\cmd.exe
C:\Windows\System32\cmd.exe
ENG
20:26

```
bob registered to send at port 53656
bob registered to receive at port 53657 for sending at port:53656
Client:eve SendPort:53654  RecvPort:53655 Public Key:null
Client:bob SendPort:53656  RecvPort:53657 Public Key:null
SendServerSocketThreads initiated for client.
RecvServerSocketThreads initiated for client.
alice registered to send at port 53658
alice registered to receive at port 53659 for sending at port:53658
Client:eve SendPort:53654  RecvPort:53655 Public Key:null
Client:bob SendPort:53656  RecvPort:53657 Public Key:null
Client:alice SendPort:53658  RecvPort:53659 Public Key:null
responseFromRecipient:RECEIVED bob


responseFromRecipient:RECEIVED bob


The client alice is closed.
The client bob is closed.
The client eve is closed.
```

```
Enter the UserName:alice
Enter the mode you want to encrypt your chat(write the corrosponding code)::
1) UNENCRYPTED
2) ENCRYPTED
3) ENCRYPTED WITH SIGNATURE

1
UserName:alice
Sending registration msg..
FROM SERVER(reg send response): REGISTERED TOSEND alice


FROM SERVER(reg recv response): REGISTERED TORECV alice


@bob: hey

SENT bob

F:\Downloads\cn assignment 2 java\v8.0>
```

```
Enter the mode you want to encrypt your chat(write the corrosponding code)::
1) UNENCRYPTED
2) ENCRYPTED
3) ENCRYPTED WITH SIGNATURE

1
UserName:bob
Sending registration msg..
FROM SERVER(reg send response): REGISTERED TOSEND bob


FROM SERVER(reg recv response): REGISTERED TORECV bob


#alice:hey

@#eve:yo

java.lang.NullPointerException

F:\Downloads\cn assignment 2 java\v8.0>
```

```
Enter the UserName:eve
Enter the mode you want to encrypt your chat(write the corrosponding code)::
1) UNENCRYPTED
2) ENCRYPTED
3) ENCRYPTED WITH SIGNATURE

1
UserName:eve
Sending registration msg..
FROM SERVER(reg send response): REGISTERED TOSEND eve


FROM SERVER(reg recv response): REGISTERED TORECV eve


@bob:yo

SENT bob
java.lang.NullPointerException

F:\Downloads\cn assignment 2 java\v8.0>
```

Questions and Answers

Q1. How we will be dealing with the Scenario where the clients disconnect arbitrarily by pressing Ctrl-C or just exiting the Terminal?

Ans. When the client disconnects arbitrarily by pressing Ctrl-C or just exiting the Terminal server will detect that the client is closed now and will deregister the client from the registered list and also close the respective sockets.
At the server end we will catch the java.net.SocketException and will handle it gracefully by closing the respective sockets of the client and deregistering it from the server.

Q2. How to deal with the offline users?

Ans. In this case let us first set the scenarios:

1. At first there are two clients say Alice and Bob and Alice wants to communicate with the Bob.
2. Then Alice will send the msg to the Bob. At first the message will go to the server and server will send the Acknowledgement to Alice that the message has been reached (similar to the single check in Whatsapp).
3. Now Bob can be online or offline so we will be taking the two cases.
    - If the Bob is online then the Server will forward message to the and then Bob will send acknowledgement to the server and server will forward it to the client resulting in Double Checks in Whatsapp.
    - Now consider the case when Bob is offline. In this case the Server will first try to send the message and then will start the timer of say 5secs(timeout time) and expect the acknowledgement from the Client and when the timer will goes out then it will store the data in buffer with some additional information and then will retry after some time and for that time period handle the requests of other users.

Q3. How to identify if the two clients have two different sockets and not having the same socket number assigned?

Ans. Socket clientSocketSend = new Socket(serverIP, serverPortSend);

This will call a socket class and create the object of Socket class and pass the arguments like server port and server ip to the object and assign the port to it. This port is assigned by checking which ports on the system are free and after assigning it making it busy so that it can' t be used any further.

Q4. What is actually going on in our code while sending the message from one client to another?

Ans. For this we have to understand what actually should be done while writing the code. At first we will be having the ip and port of the server. Now at first we will define the data input and output stream which will help us sending data to the server and read the data coming from the server. Now when the client arrives it will first register itself to the server and the server will save it in the list it will be maintaining for every Client. The client will also be asked if he want the data to be encrypted with signature or encrypted or unencrypted and after that when we want to send data to some other client then at first we will have to write it in the format of @[username]:[text] and then this text will be sent first to the server and the acknowledgement to the client and then the data will be sent to the recipient and then the recipient will first send the acknowledgement to the server and the server will forward it to the sender.

Q5. How will the server know to which port it should redirect the message?

Ans. To confirm this we have used the concept of username i.e the username should be unique and we will be saving the port of that username then at the server end we will extract that port using getPort method defined in our code and then redirect the message to the recipient.

Q6. What if user try to send the data to some client which is not yet registered?

Ans. In this case the sender will get the message that the username is invalid and can’t be found.

Q7. How the encryption of data will be done?

Ans. When the sender first input the data then the hash of the data will first be calculated if selected at first and then the data will be encrypted by rsa algorithm and the public key will be available at the server side of every user and can be fetched by the user using FETCHKEY method. After this the data is formatted in some specified order and then the formated message is sent to the server. Now as the server will not be having the private key so it can’t decrypt the message but it can read the header and find out the recipient if exists and then forward it to the recipient and the recipient will then read the msg decrypt his text and the hash if available and then calculate the hash for the data and verify if the data received is correct or is manipulated by some third party.

References

1. https://www.geeksforgeeks.org

2. 2_chat_application.pdf

3. https://docs.oracle.com/javase/7/docs/api/index.html?java/security/package-summary.html

4. https://docs.oracle.com/javase/7/docs/api/javax/crypto/package-summary.html

5. https://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#KeyPairGenerator