ASSIGNMENT 2

BY: SHIVAM GOYAL (2017CS10377)
MEENAL MEENA (2017CS10351)

In this assignment, we have built our own whatsapp like architecture for communication using multiple clients which can connect to a multithreaded server. Protocol is same as HTTP, but ours preserve state. Client need not register to the server again and again before sending a message. Further encryption has been used for secure communication and signature to ensure message integrity and that the message has not been tampered.

Extension Part

-Now extend this with defining your own UNREGISTER message, for when users go offline.
Answer part: The user will give input to client to go offline(e.g.: "#off"). Upon getting this input from the user, the client will send an unregister request to the server(e.g.: UNREGISTER [username]\n). On getting such a request from the client, the server will remove the client's entry from its hash table and close its sending and receiving socket.
-Users may also disconnect arbitrarily by pressing Ctrl-C and not send an UNREGISTER message. How would you deal with such a scenario?
Answer part: For the case when users disconnect by pressing Ctrl+C, server can keep refreshing its hashmap which stores all the connected clients by sending some packet sending and receiving acknowledgement scheme. As soon as the client disconnects, during refreshing it will be removed from the hashmap and close its socket, hence that user gets deregistered. Now when anyone tries to send to someone which is not registered, server will inform the client.

-Also think and describe how you would extend the client and server applications to deal with offline users? Similar to the single and double checks in Whatsapp, a sender should be able to send messages to offline recipients and get notified later whenever the messages are delivered to them.
Answer part: We can use a similar protocol like SMTP. While sending a message to the client which is online, the message will be sent and an acknowledge will be heard by the server. For an offline user, we can maintain a buffer which will store the incoming messages until the user turns online again. For the case of whatsapp, two acknowledgements can be designed, one for the message have reached till buffer or not and other for whether the message has been received by the receiver.

-You should not send the public keys or encrypted data in a binary format in various REGISTER and SEND methods. Think why this is not desirable.
Answer part: As text data is easier to handle and transmit as compared to binary data, the binary data is generally encoded using Base64 into text characters. A Base64 encoded message of the

binary data is sent in for registering and sending data because converting the resulting bytes(from hashes like SHA/MD5) into Base64 makes it much easier to display the hash as well as comparing a checksum for integrity.

Certificates for encryption

In cryptography, a **public key certificate,** also known as an **identity certificate** is an electronic document used to prove the ownership of a public key. The certificate includes information about the key, information about the identity of its owner, and the digital signature of the one that has verified the certificate's contents (called the issuer).

Apart from these protocols, one can design peer to peer which is a secure way of not handling the privacy issues with the server. One can directly communicate without central agency.

Length of a message

As we are using key size of 512, we can only encrypt/decrypt messages of size <= 53 [(512/8) - 11]. Longer messages could have been handled by splitting up the message, but this was not the concern for this assignment.

Exceptions handled

1. If a client with ill-formed username is trying to register, server will ask to connect with a well-formed username.
2. If a client connected to server is trying to send a message to user not connected with the server or any ill-formed username then server will request to send the message again saying the client with username you sent the message to does not exist.
3. If we open a new client and give it the username with is already registered with the server, then the server will inform to try another username!
4. The format of sending a message by the client is well checked. If the client tries to send in an invalid format then server will inform it that the format is wrong! Some invalid format includes: not sending by typing '@', no message typed after username.
5. User sending messages without registration is also handled.