Private Group Chat

Final Project Proposal

COMP 4911 – Computer Networks

Anthony Arseneau

The project aims to create a secure private group chat application using Java as a programming language, incorporating Swing for the user interface, and RSA, AES, and SHA-256 for cryptographic security.

Server:

The server will have an opened socket to receive and send messages to the rest of the users of the group chat. The server is responsible for authenticating users, granting them access to the group chat if authorized, and exchanging AES symmetric keys to authorized clients through RSA asymmetric keys. The server and each client will have RSA public and private keys for the first communications.

User Authentication:

The server will implement user authentication where it will store hashed usernames and passwords of authorized users. The reason to store hashed usernames and passwords of users is to prevent information leaks in the case of an attack on the server. Users will need to input a valid username and password pair to get access when trying to connect to the group chat. The whitelist of authorized members will be preinstalled on the server.

Client Interface:

The user interface of the application will be as intuitive as popular texting applications, such as WhatsApp, iMessage, Messenger, etc. There will only be a desktop version. The UI will be done using Swing in Java for creating an interactive GUI. When starting the application, the user will be prompted to enter a valid username and password pair. Once authorized by the server, basic functions will include a space for writing messages and a button to send to other authorized clients. Text messages from all users will be displayed in chronological order along with the names and the time they were sent.

Things I need to consider:

- To have a reliable group chat, I will need to find a way to properly utilize the cryptographic algorithms. Familiarity with the theory behind using them will not be a problem. Finding a way to implement them in Java could come across as challenging.
- Making sure that chats are synchronized and displayed in the same order for every user will also be necessary.
- Achieve socket programming in Java, like we did in Python, will be essential to learn.

References:

1. Using RSA in Java:
   https://www.devglan.com/java8/rsa-encryption-decryption-java
2. Using AES in Java:
   https://www.baeldung.com/java-aes-encryption-decryption
3. Using SHA-256 in Java:
   https://www.baeldung.com/sha-256-hashing-java
4. Article for starting socket programming in Java:
   https://medium.com/edureka/socket-programming-in-java-f09b82facd0#:~:text=Socket%20programming%20in%20Java%20is,a%20client%20and%20a%20server.