Project Title : Java Chat Application

- *By Jatin Dnyaneshwar Gangare*

Introduction

The Java Encrypted Chat Application is a real-time peer-to-peer communication system built using Java socket programming and multithreading.
The project enables multiple users to connect to a central server, send messages, and interact through a GUI built with JavaFX.
The system incorporates AES encryption to ensure that messages are transmitted securely between clients, making it suitable for private and group communication.
This project demonstrates concepts such as network programming, multithreading, event-driven GUI design, and cryptography.

Abstract

The primary goal of this project is to develop a secure and user-friendly chat platform where multiple users can connect simultaneously to exchange messages.
The server acts as a central hub for all communication, managing connected clients, broadcasting messages, and handling private chat requests.
Key features of the system include:

- AES encryption for secure communication.
- Private and group messaging.
- Typing indicators to show when users are typing.
- Dynamic user lists that display all connected clients.
- Guest name auto-assignment (Guest1, Guest2, etc.) for users who do not provide nicknames.

This project highlights how Java networking, concurrency, and cryptographic techniques can be combined to build a robust, real-time communication tool.

Tools Used

- Programming Language: Java (Java 11 and above).
- Networking: Socket Programming (Socket and ServerSocket).
- Concurrency: Multithreading using Java Thread and Runnable.
- GUI Framework: JavaFX for building the client-side chat interface.
- Encryption: AES (Advanced Encryption Standard) for secure data transmission.
- IDE: IntelliJ IDEA / Eclipse / VS Code.
- Libraries/Utilities: Java I/O streams, Collections framework, and JavaFX components.

Steps Involved in Building the Project

1. Planning and Design
    - Identified the need for a client-server architecture.
    - Designed the structure with the following core components: Server.java, Client.java, AESEncryption.java, PrivateChatManager.java, PrivateChatWindow.java.
2. Server Implementation
    - Created the Server.java file that uses ServerSocket to listen for incoming connections on a fixed port (12345).
    - Implemented ClientHandler threads to manage each connected client separately.
    - Added nickname verification and auto-assignment of guest names (Guest1, Guest2, etc.) if the nickname is empty or already in use.
    - Implemented a broadcast feature to send messages to all connected users and a private message feature for individual users.
    - Added user list broadcasting so clients can see all online users in real time.
3. Client Implementation
    - Developed a JavaFX GUI in Client.java for the user interface, which includes:
        - A scrollable chat window with chat bubbles.
        - A text input box for message sending.
        - A typing indicator (User is typing...).
        - A ListView of online users to start private chats.
    - Established a socket connection with the server and implemented encryption for sending/receiving messages.
    - Created PrivateChatManager and PrivateChatWindow to handle private messages between two users.
4. AES Encryption
    - Implemented AESEncryption.java to encrypt outgoing messages and decrypt incoming messages using a shared secret key.
    - Integrated encryption at both server and client sides to ensure privacy.
5. Testing and Debugging
    - Tested the server and client by running multiple client instances on the same network.
    - Verified message broadcasting, private messaging, guest name assignment, and typing indicators.
    - Debugged edge cases like disconnection handling, duplicate nicknames, and message encryption errors.

Conclusion

The Java Encrypted Chat Application successfully demonstrates the concepts of Java socket programming, concurrency, encryption, and GUI development.
It allows secure, real-time communication between multiple users, supporting both group and private chats.

- *Jatin Dnyaneshwar Gangare*