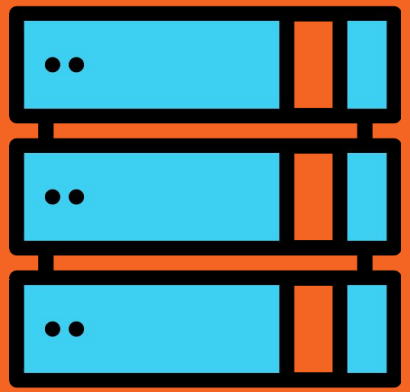
AnonChat

A Desktop Based End-2-End Encrypted Chat App



Made By

1. Shoray Singhal
2. Ayusman Samal
3. Udbhav Misra

Source Code

Idea

To create a truly end-2-end encrypted chat app which can be used by people to communicate with others without being tracked or their data being stolen.

How will this benefit society

- You can bypass surveillance from governments and ISPs.
- There is no risk of Data leak as the servers won't contain any logs.
- As the messages are encrypted, there is no risk of man in the middle attack.
- Key pairs are generated client side and private keys are never transmitted.

Features to Implement in the Future

- Improve E2EE from 1024 bit to 2048 bit.
- Option for the user to join specific server
- On Demand Servers
- Web Functionality
- Media transfer Functionality
- Emoji and Sticker Support
- Cross Platform Support

Features

- Use of Java Swing , Java AWT , Java Crypto , Java Security , Java Net
- Follows Hub and Spoke or Star topology.
- Clients join the server with an Alias Username
- Generates a private and unique avatar for every new user
- Group Chats
- Personal Chat with more than one people simultaneously
- Notification if someone joins or exits the Chat
- Display of all Currently online users
- Use of Java RMI
- Use of full 1024 bit End-to-End RSA encryption

Work Distribution

- Ayusman :
  - Implementation of E2EE in group chat
  - Implementation of E2EE in private chat
- Shoray:
  - UI design and implementation of Client GUI
  - Implementation of RMI
- Udbhav:
  - Implementation of Server Logic
  - Integration of E2EE on the Server Side
- All:
  - Creation of diagrams and documentation

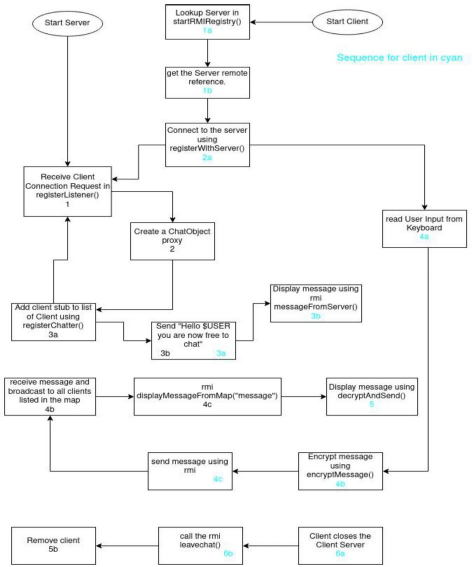
—

What is Remote Method Invocation?

Remote Method Invocation (RMI) is an API that allows an object to invoke a method on an object that exists in another address space, which could be on the same machine or on a remote machine. Through RMI, an object running in a JVM present on a computer (Client-side) can invoke methods on an object present in another JVM (Server-side).

RMI creates a public remote server object that enables client and server-side communications through simple method calls on the server object.

Flowchart to implement RMI



What is End-to-end Encryption?

End-to-end encryption (E2EE) is a method of secure communication that prevents third parties from accessing data while it's transferred from one end system or device to another.

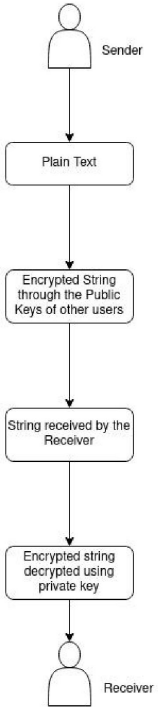
In E2EE, the data is encrypted on the sender's system or device, and only the intended recipient can decrypt it. As it travels to its destination, the message cannot be read or tampered with by an internet service provider (ISP), application service provider, hacker or any other entity or service.

How did we implement End-to-end Encryption?

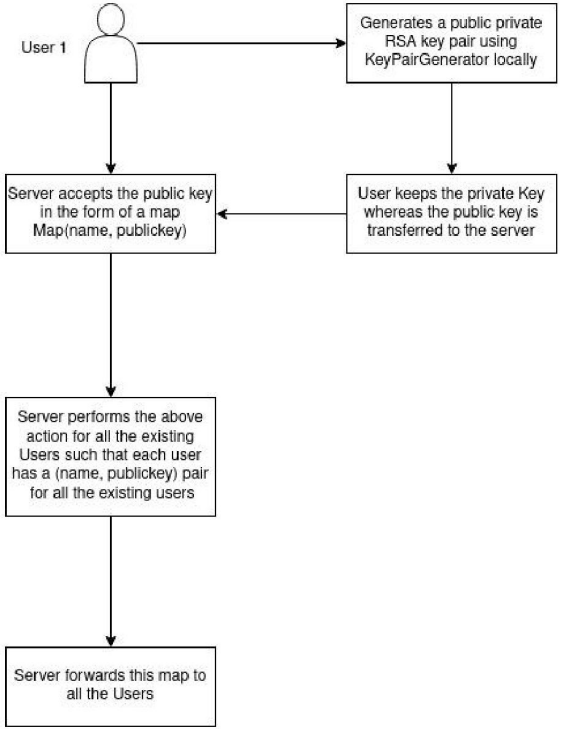
We used RSA(Rivest-Shamir-Adleman) 1024 -bits encryption to implement E2EE in our chat client which is a type of asymmetric encryption. It generates a public-private key pair locally and then transfers the public via the server to other clients which is then used to encrypt the plain text which can only be decrypted using the private key stored locally.

Flowcharts to show the working of E2EE

Very Basic Explanation of End-to-end Encryption

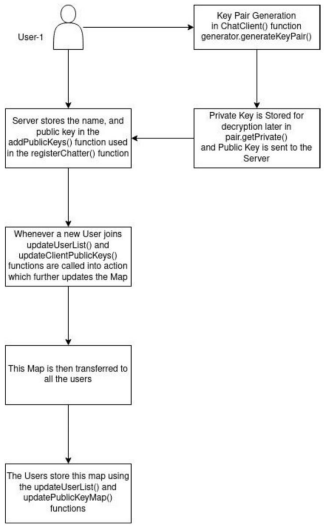


How the Key Exchange Happens

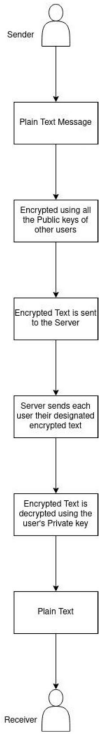


Flowcharts to show the working of E2EE

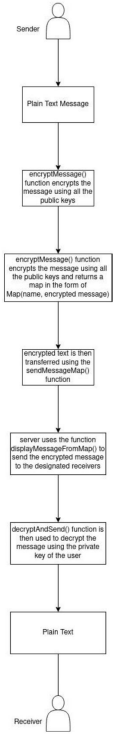
How the key exchange happens as shown in code



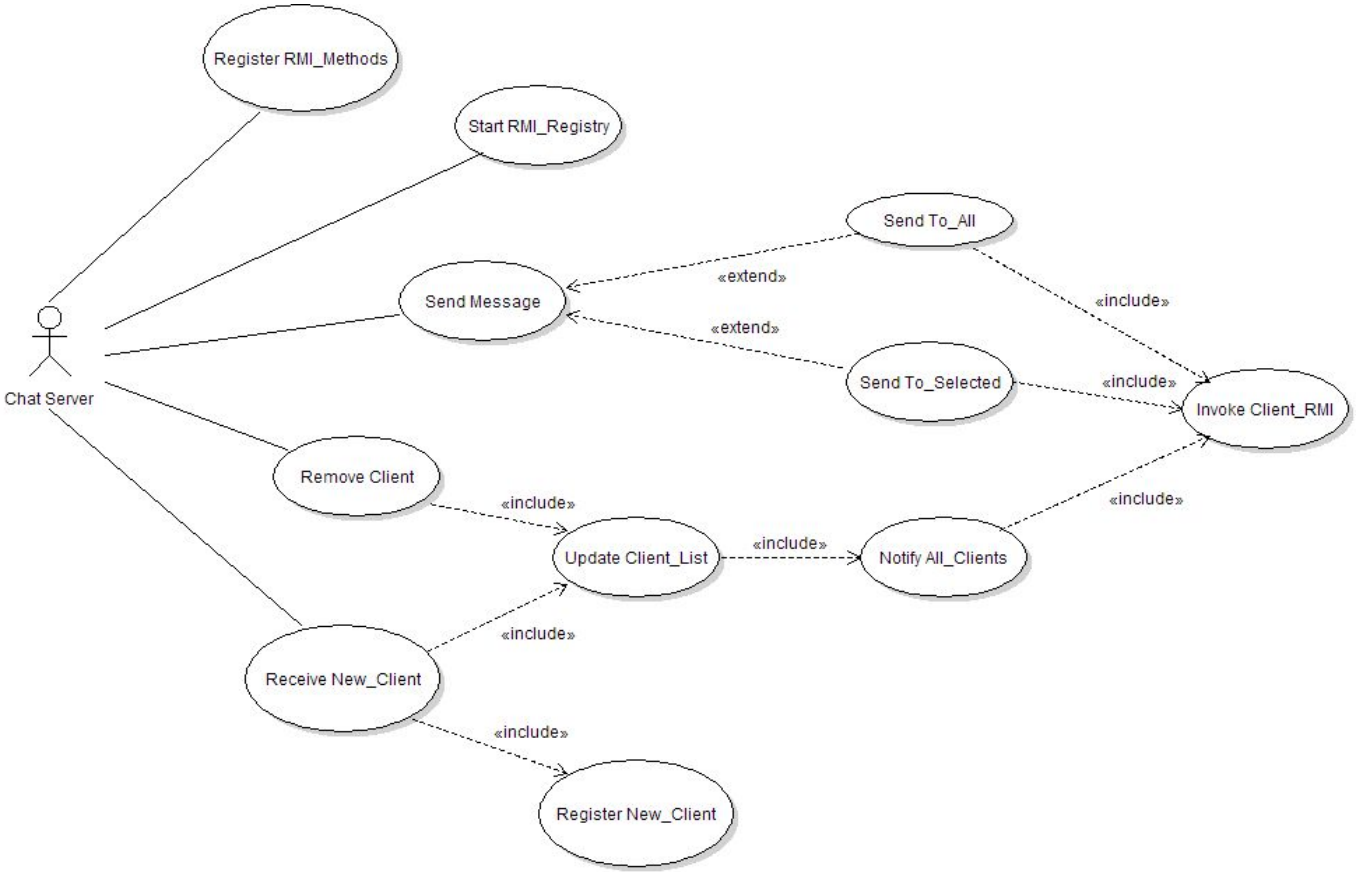
How the messages are encrypted and decrypted

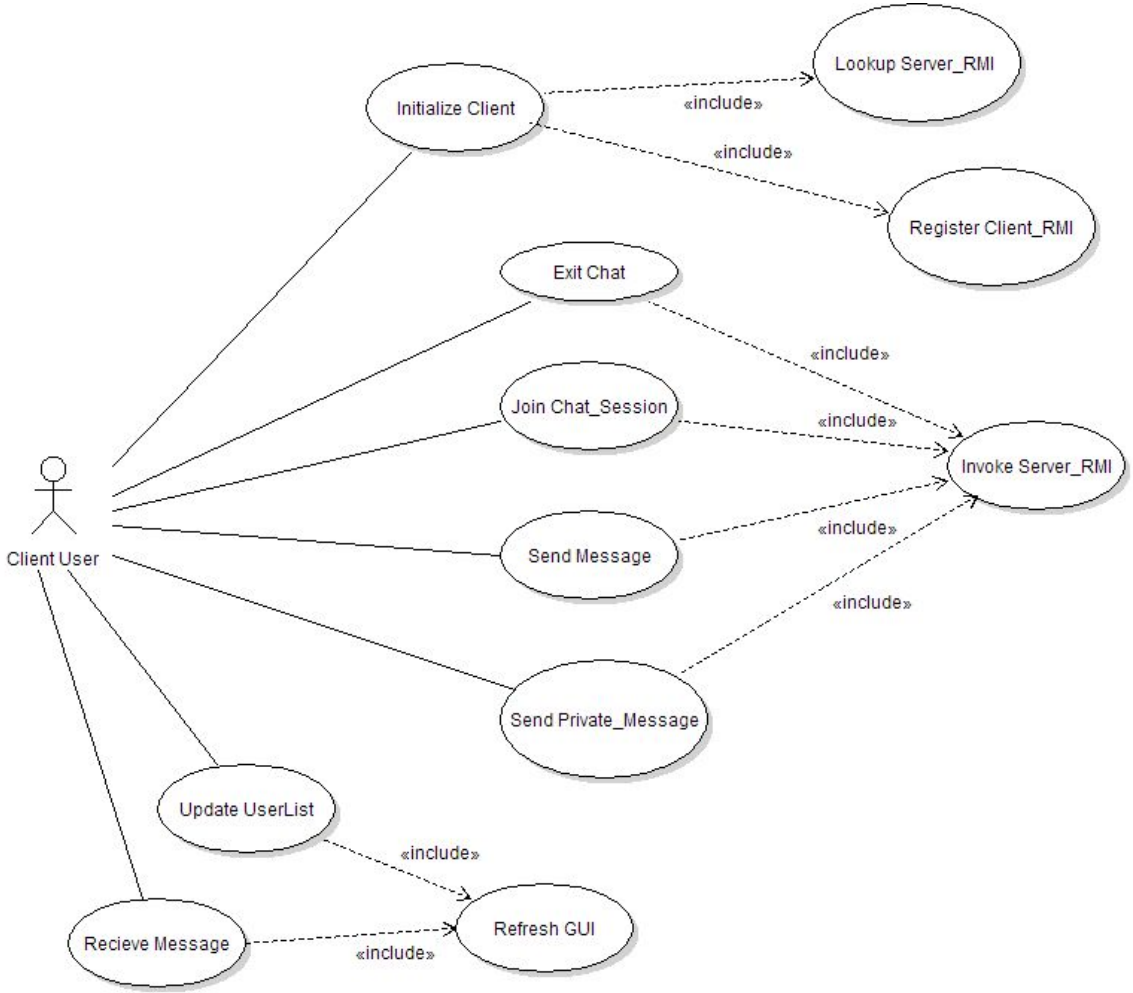


How the messages are encrypted and decrypted as shown in code



Usecase Diagrams




Lookup Server_RMI
Initialize Client
«include»
«include»
Register Client_RMI
Exit Chat
«include»
Join Chat_Session
«include»
Invoke Server_RMI
Client User
Send Message
«include»
«include»
Send Private_Message
Update UserList
«include»
Refresh GUI
Recieve Message
«include»