

## User Manual/Readme

This section details the process required to run the chat application as well as the steps needed to be taken to fully evaluate/use each feature of the chat application.

### Known Issues

We have found a nasty bug that occurs specifically when *more than one chat "room" has been created*. Due to time constraints, we unfortunately could not fix this issue in time.

The bug creates an error (could be due to a variety of problems, but we suspect a key exchange issue) which breaks the message decryption part of the application, and users receive still-encrypted messages. This behavior can be circumvented by taking the following steps:

It is recommended that during marking, only one chat is tested at a time, be it a group chat or an individual chat. So, for instance, after an individual chat has been created and tested, please close all instances of the client and server, and restart them to create a new group chat for testing.

If this is not followed, please be aware of this bug that might occur.

We apologize for this inconvenience! (and please have mercy)

### Running the application

- The environment

It is suggested to run this application on **Ubuntu Linux**, as all development and testing was done on this operating system. It is also suggested tests are run on a **1080x1920 resolution** screen, due to the graphical user interface (4k screens can distort the GUI).

The directory structure for this application is pictured below:

```
.Root (WD)
├── Resources/
│   └── ...
├── src/
│   ├── Objects/
│   │   ├── NetworkMessages/
│   │   │   └── ...
│   │   ├── Chat.java
│   │   ├── GUIChat.java
│   │   ├── Group.java
│   │   ├── Message.java
│   │   └── User.java
│   └── GUI.java
```

```
|  |
|  | └─ GUIClient.java
|  | └─ Server.java
|  └─ makefile
```

This application allows for easy compilation and execution using a makefile. When running makefile commands, please have your working directory in the **same directory as the makefile**.

- Compilation and Setting-up

When you first open the folder, you will be greeted to the directory structure pictured above. Navigate to the same directory as the makefile in your terminal and run the command **“make”** to compile all the Java source files.

After this step, both the server and client(s) need to be set up. For this, another terminal window/tab needs to be open in the same working directory as before. Now that you have two terminal windows open in the same directory, first run **“make server”** in the one terminal. Once the server has started, you have the option to either (1) populate the server with some data for easier marking, or (2) run a client.

- (1) Populating the server with dummy data can help with marking and is done by typing **“make populate”** in the terminal separate from the server, while the server is running. This will spit out a lot of terminal output which is the byproduct of sending requests to the server.

The populated data contains three users: Alice, Bob and Eve as well as a group chat created between them with some messages. If you would like to sign-in as any of these users, please use the credentials below:

Username	Password
Alice	Alice#1
Bob	Bob#2
Eve	Eve#3

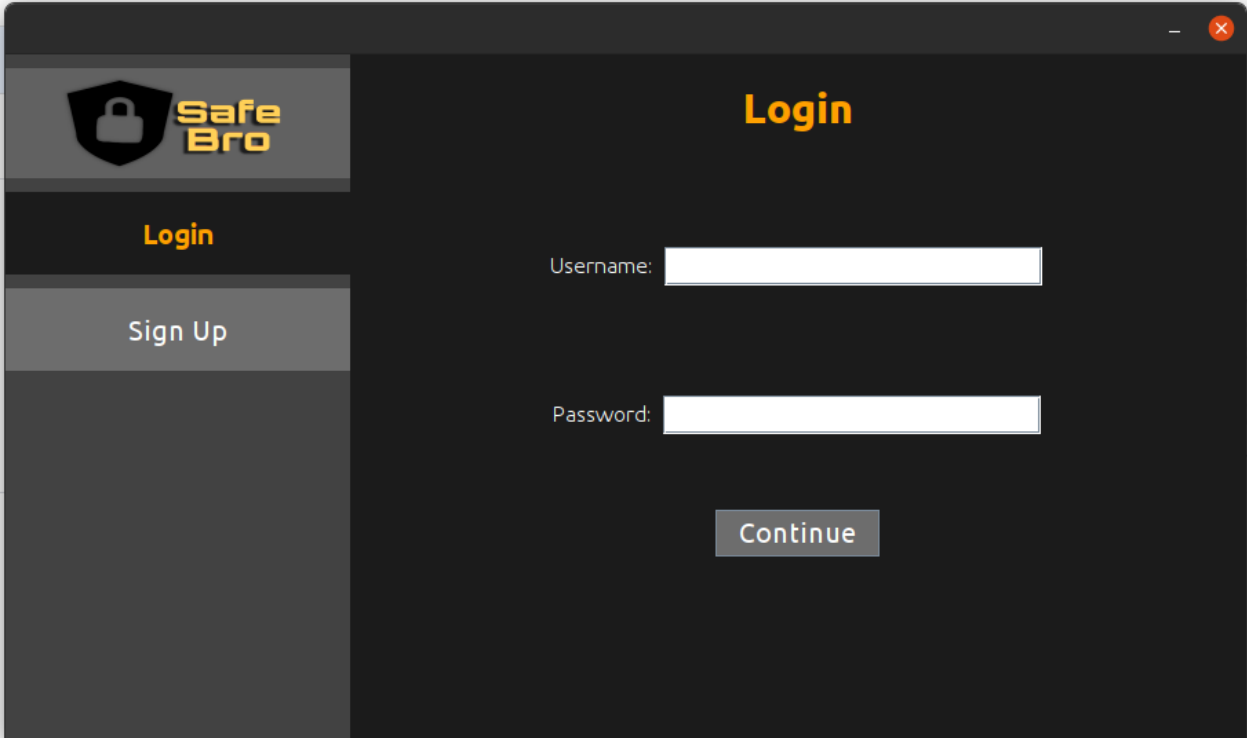
- (2) You can create a client instance in two separate ways using the makefile. The simplest method involves opening a new terminal in the same directory as the makefile and running the command **“make GUI”**. This will create a client instance. **Please ensure the server is running separately before creating any clients.**

The second method is tailored to hopefully make marking easier by creating three separate instances of a client at once. This saves you the hassle of opening three new terminal windows for three separate clients. Run **“make mark”** for this functionality in a

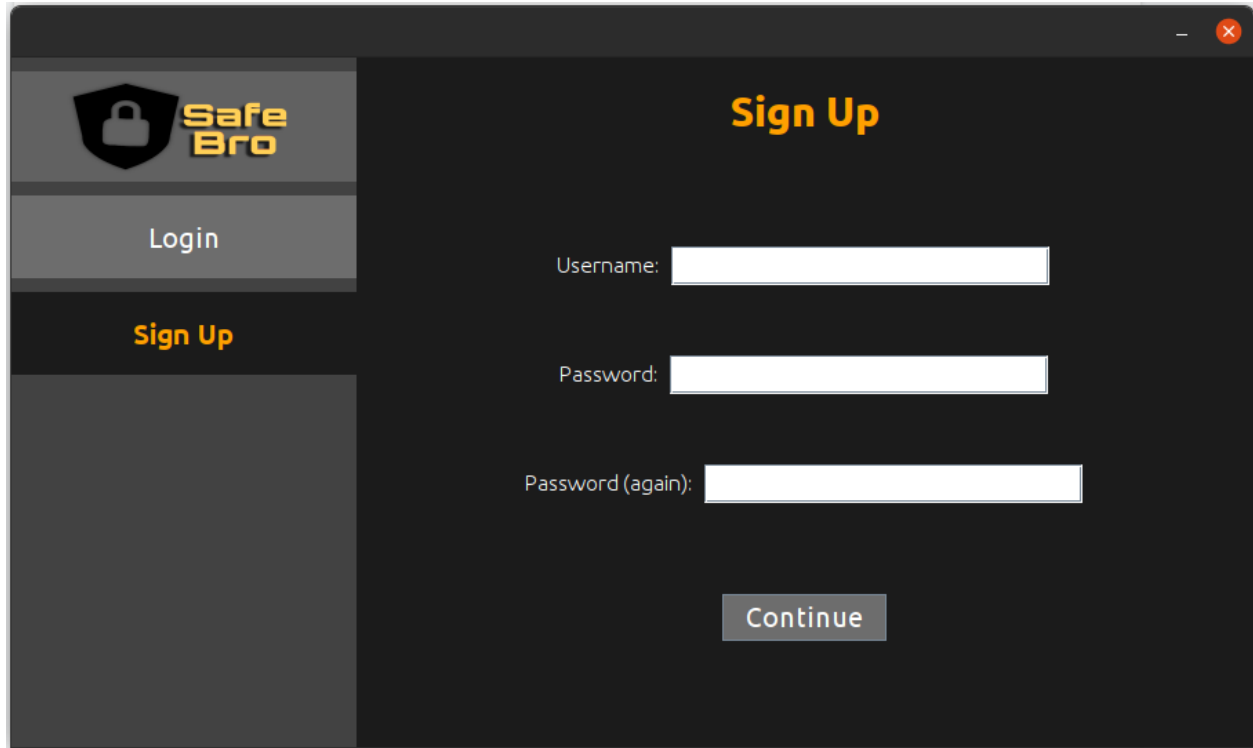
separate terminal window from the server, in the same directory as the makefile. It is noted however that all client output from each of the clients' instances will be output *to the same terminal* when using this. It may be favorable to run "make GUI" in separate terminals for better text output.

### Navigating the GUI

The first GUI window you will see is the log-in/sign-up window pictured below:



The image shows a GUI window titled "Safe Bro" with a dark theme. On the left side, there is a vertical sidebar with a "Login" button highlighted in orange and a "Sign Up" button in white. The main area of the window is titled "Login" in orange. It contains two input fields: "Username:" and "Password:", both with white text and white input boxes. Below the password field is a "Continue" button with white text on a dark background. The window has a standard macOS-style title bar with a red close button in the top right corner.



The image shows a web application window titled "Safe Bro" with a dark theme. The window has a sidebar on the left with a "Login" button and a "Sign Up" button. The main content area is titled "Sign Up" and contains three input fields for "Username:", "Password:", and "Password (again):". A "Continue" button is located at the bottom right of the main content area.

**Safe Bro**

Login

Sign Up

Sign Up

Username:

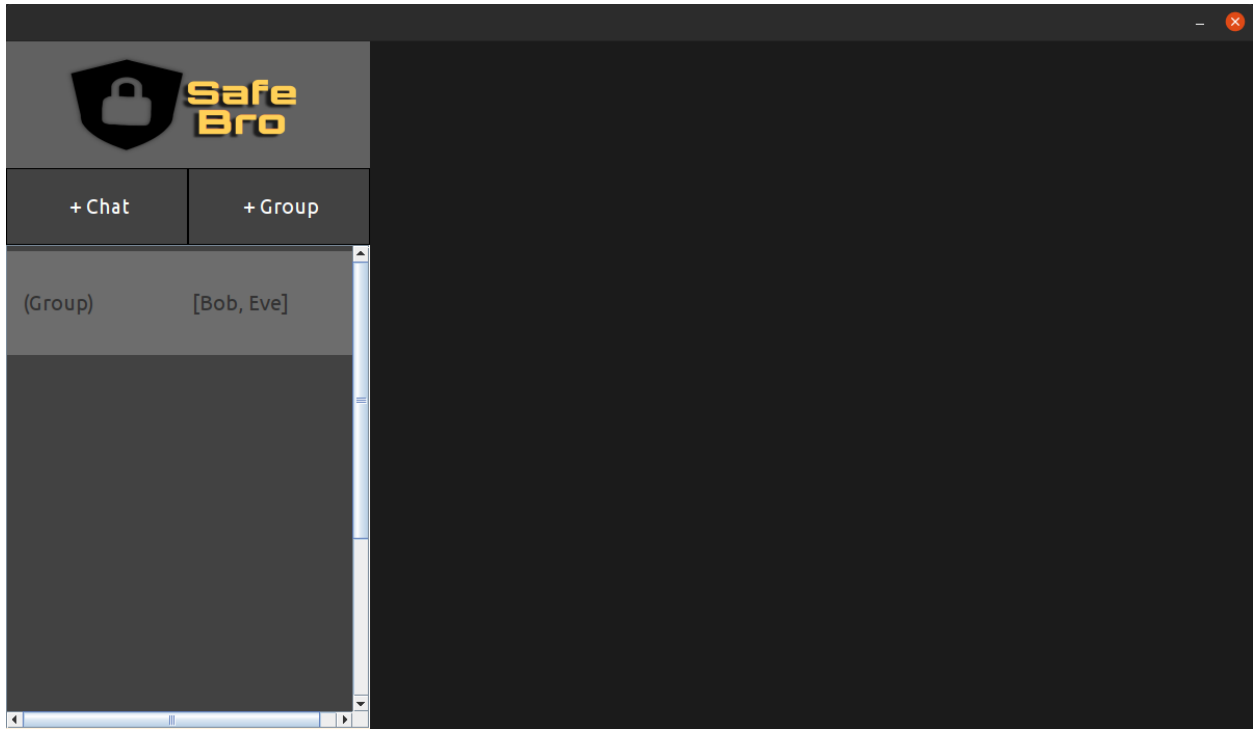
Password:

Password (again):

Continue

You can create a new user in the sign up window by clicking the “sign up” button and filling in the required data. On success, you will be presented by a success popup and redirected to the sign-up page, where you may then sign up using your credentials (just remember your username and password!).

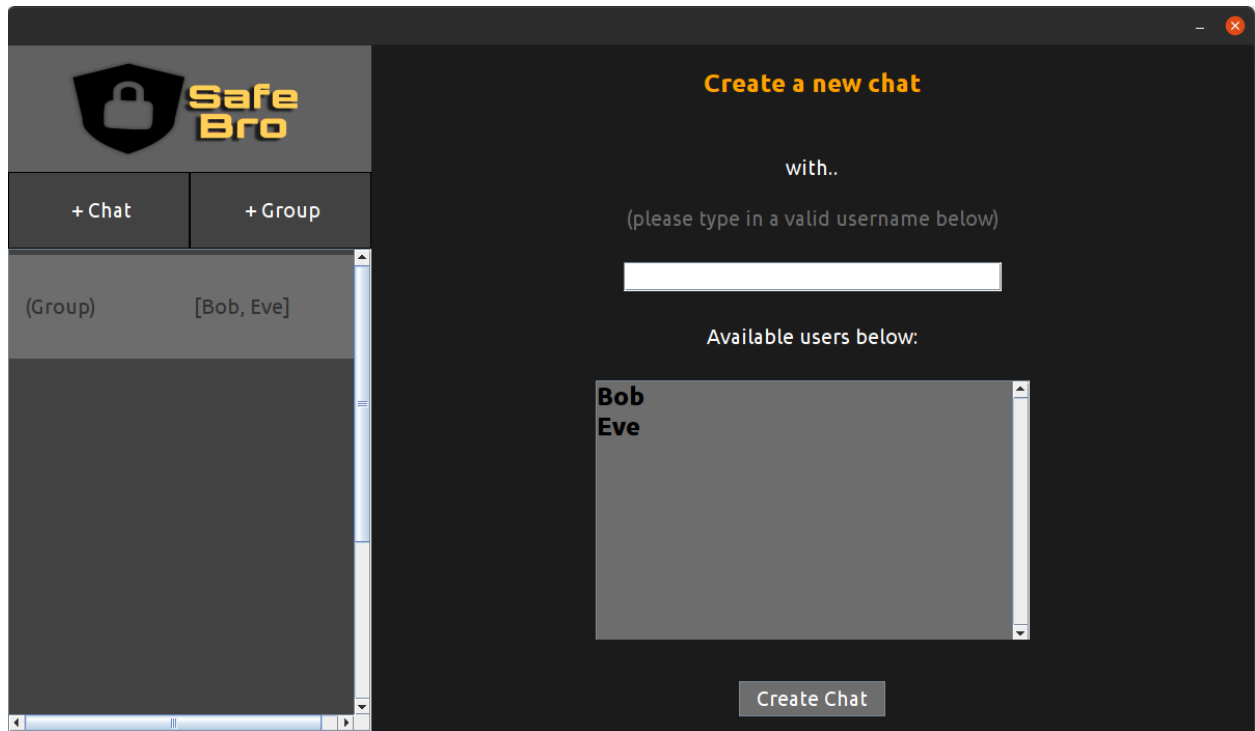
The main “chat window” is pictured below and contains the main workflow of the entire application:



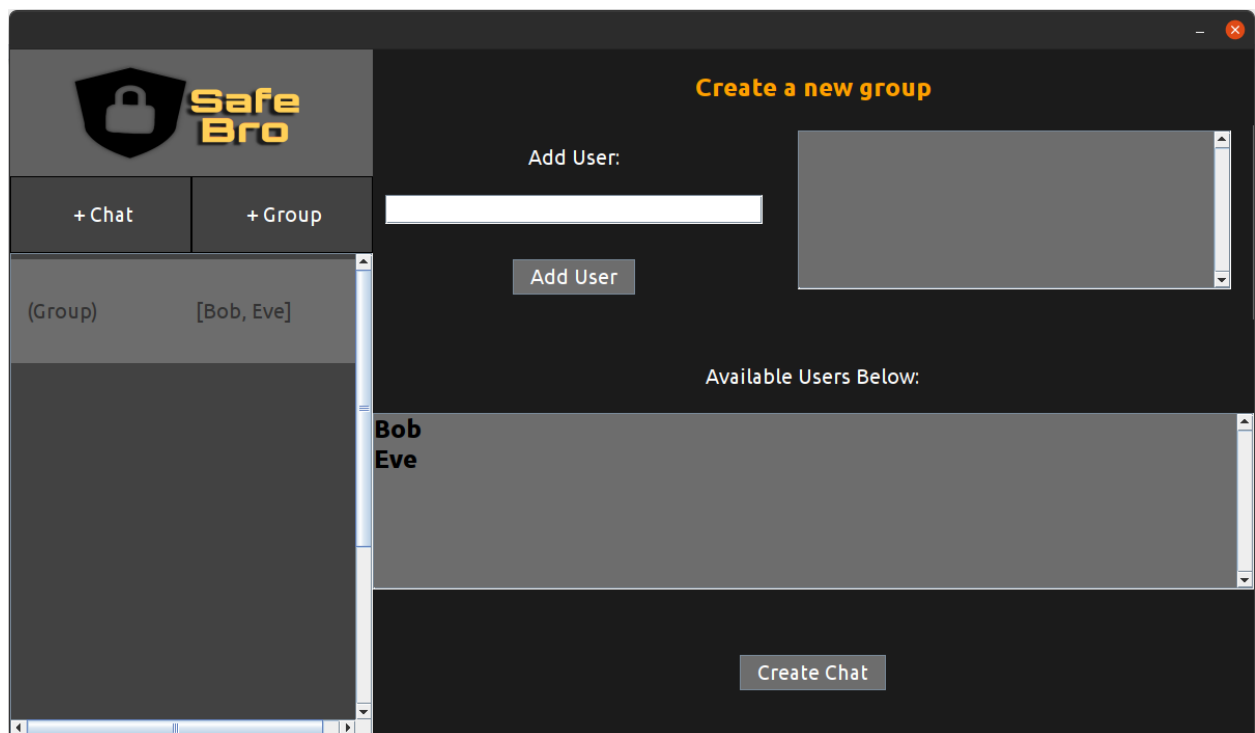
There are three permanent clickable buttons on the left: The “Safe Bro™” Logo, “+Chat” and “+Group”.

Clicking the Safe Bro logo **refreshes** the page, retrieving all new chats from the server and all new messages. This button is important since no new messages sent to the logged in user will show unless refresh has been clicked or that chat has been clicked again. When in doubt, click this button please! Also please note that the refresh button will close any current chat windows, so you will have to click the chat you were focused on again, apologies for the inconvenience here.

The other two buttons are quite self-explanatory and redirect you to a window where you can create a new chat or group chat if you so desire. These windows are shown below for reference:

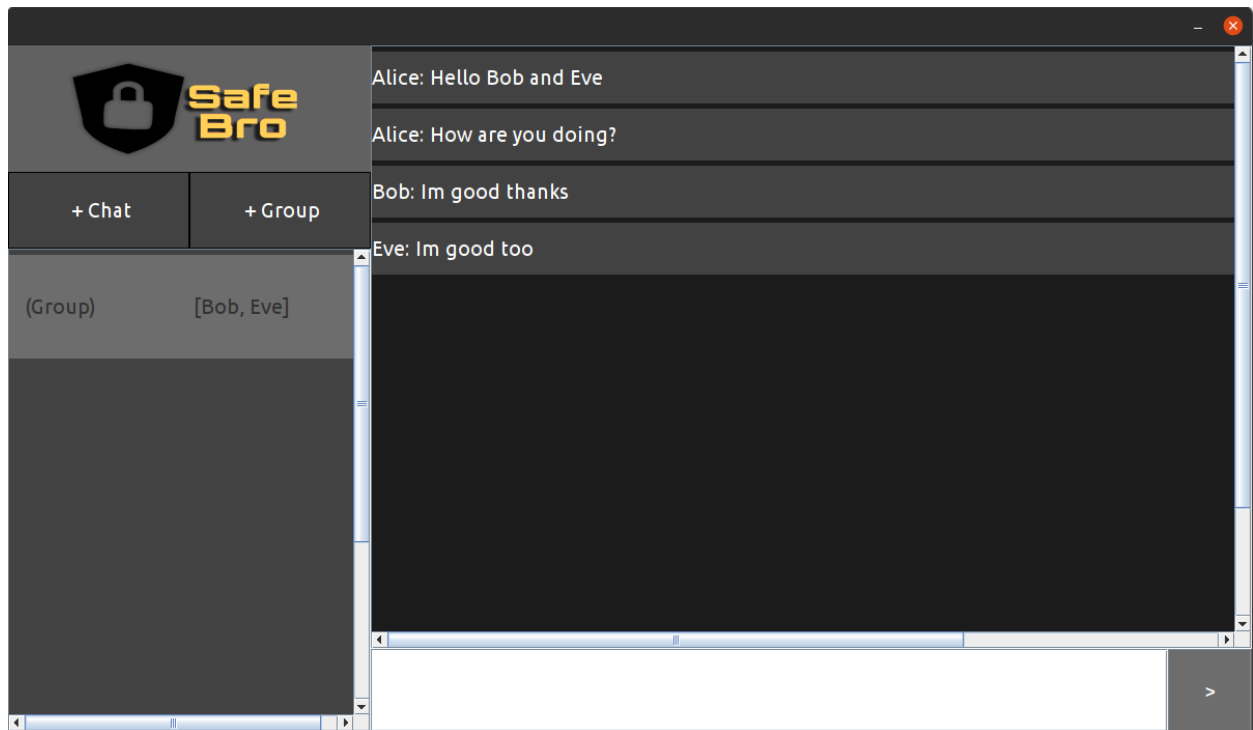


(\*available users is every user besides currently logged-in user)

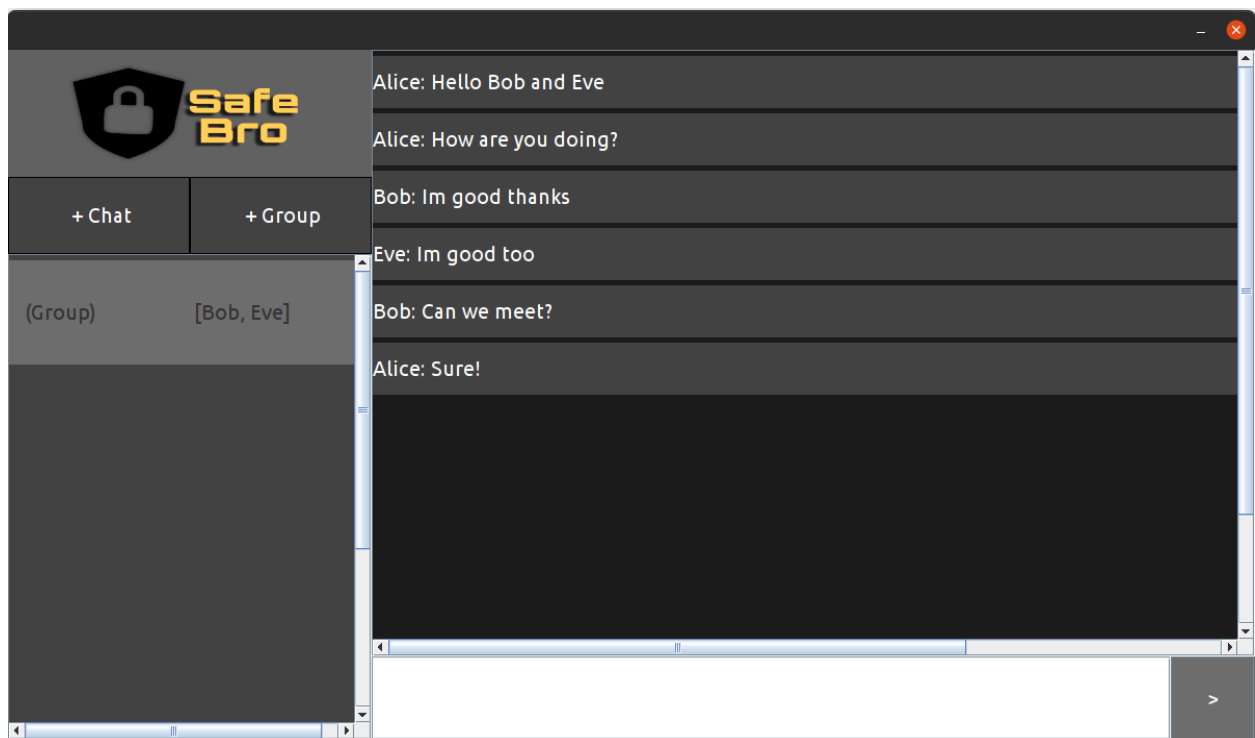
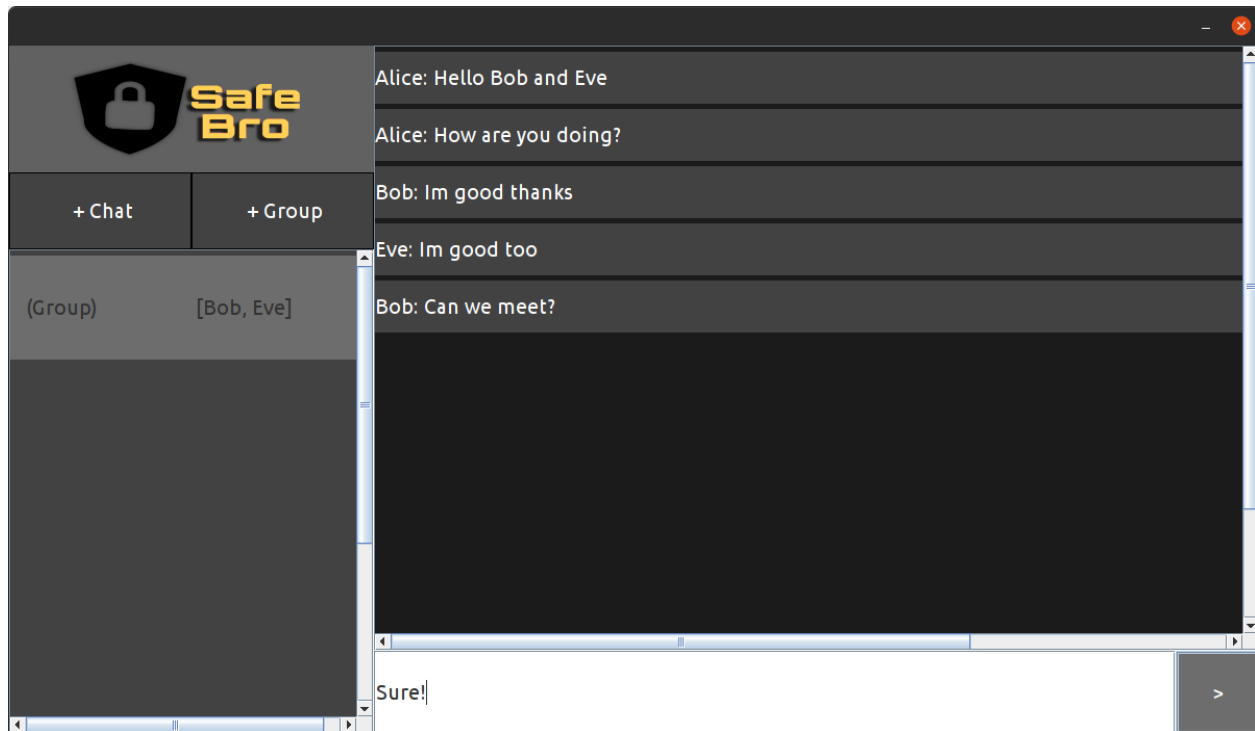


(\*add each user you want in your group chat separately with "Add User", and then confirm when all are added with "Create Chat")

Chats can be accessed by simply clicking them on the side-panel, which takes you to the messaging window:



From this window you are free to send messages by typing in the text field and clicking the send button:



Finally, you can check how the application is working (private key, public key, before decryption, after decryption etc.) by observing the terminal output through these messaging stages.

Thank you for marking! If anything isn't working or you have any questions, please direct them to: [juliandavidlord@gmail.com](mailto:juliandavidlord@gmail.com)



