

Project Phase 3

|  |  |
| --- | --- |
| **Student Name** | Abdullah Alnasser |
| **Student ID** | 201535050 |
| **Instructor** | Dr. Marwan Abu-Amara |
| **Term** | 201 |
| **Course** | COE451 |



December 11, 2020

Table of Contents

[I. Program 2](#_Toc58600984)

[A. Screenshots of how to use the program 3](#_Toc58600985)

[B. Screenshots of Cypher stuff 4](#_Toc58600986)

[C. Readme 7](#_Toc58600987)

# Program

**Name**: P2P Chat Application

**Link**: <https://github.com/sdht0/P2P-chat-application>

**Description**: A python program which acts as a P2P chat application, created using Sockets and Tkinter GUI. The only requirement is to know the peer's network address in the form of IP and Port number.

**Architecture**: Peer-two-Peer.

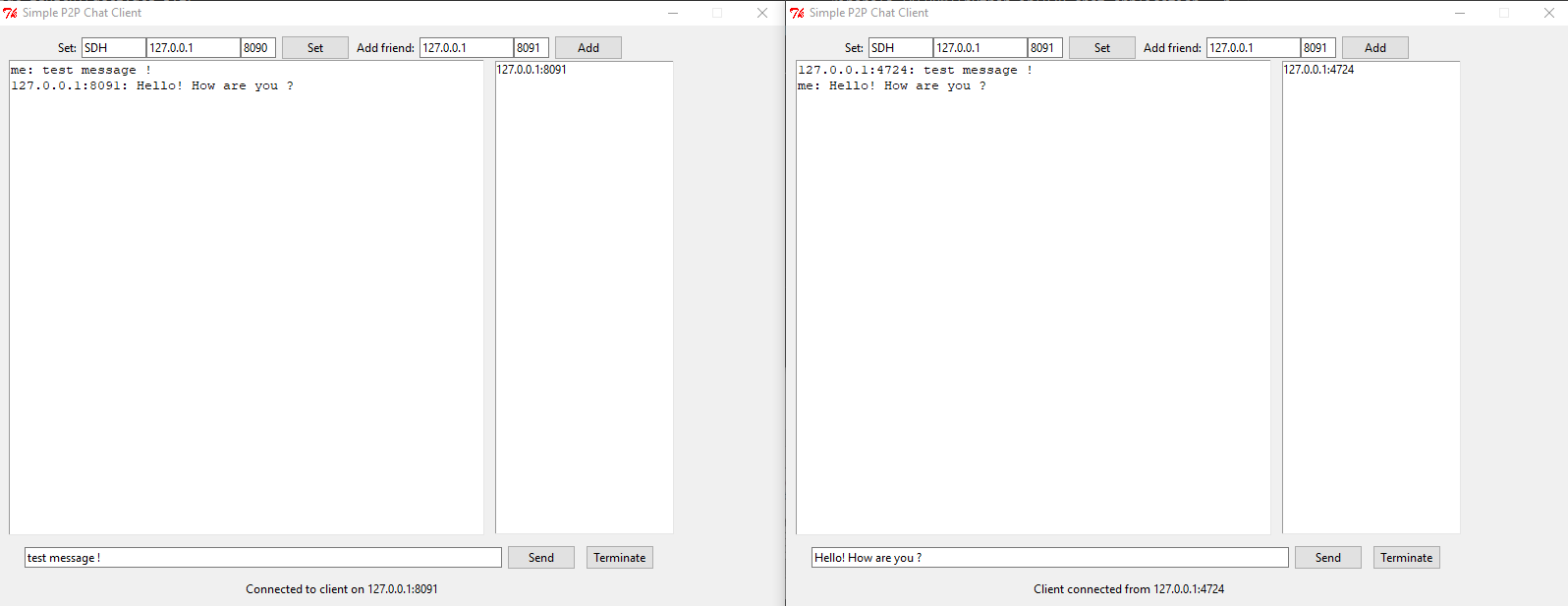
**Programming Language**: Python 2

**How I tested it**: I have tested it on two different computers in the same LAN and I have tested it on one of the computers with two different ports. It worked nicely using both methods.

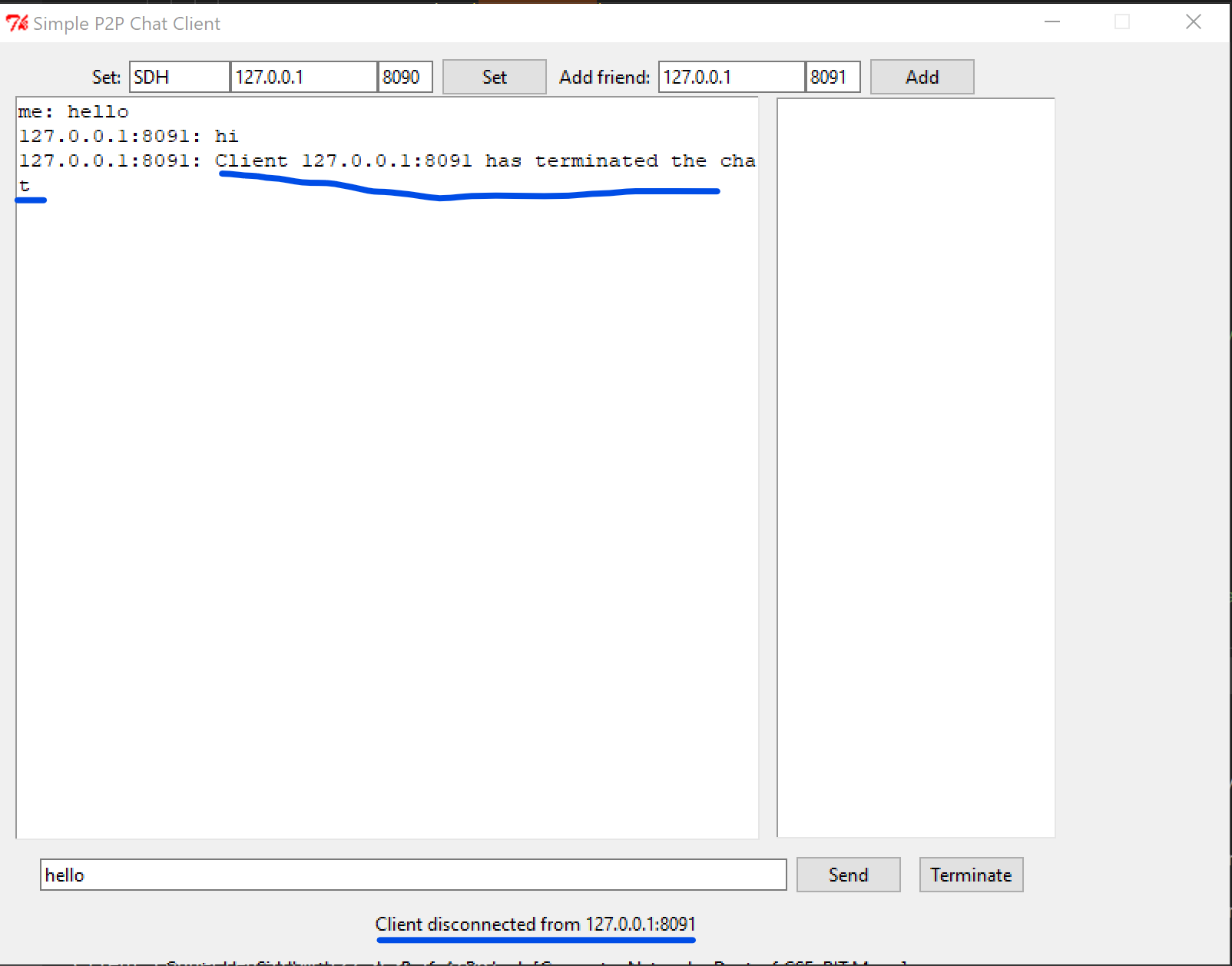
**Extra Features**: I have added to the code a “Button” to terminate the program that sends a message to the second user when other user terminates the chat. Furthermore, I have edited the code to support writing “Blank Space” such as “ “ between text. + I have added Cypher SSH with AES256-CBC to it !

## Screenshots of how to use the program

First, Set the server ip and port on both windows from top left, add client ip and port on one of the two windows as a friend from top right, then send a message from either window.



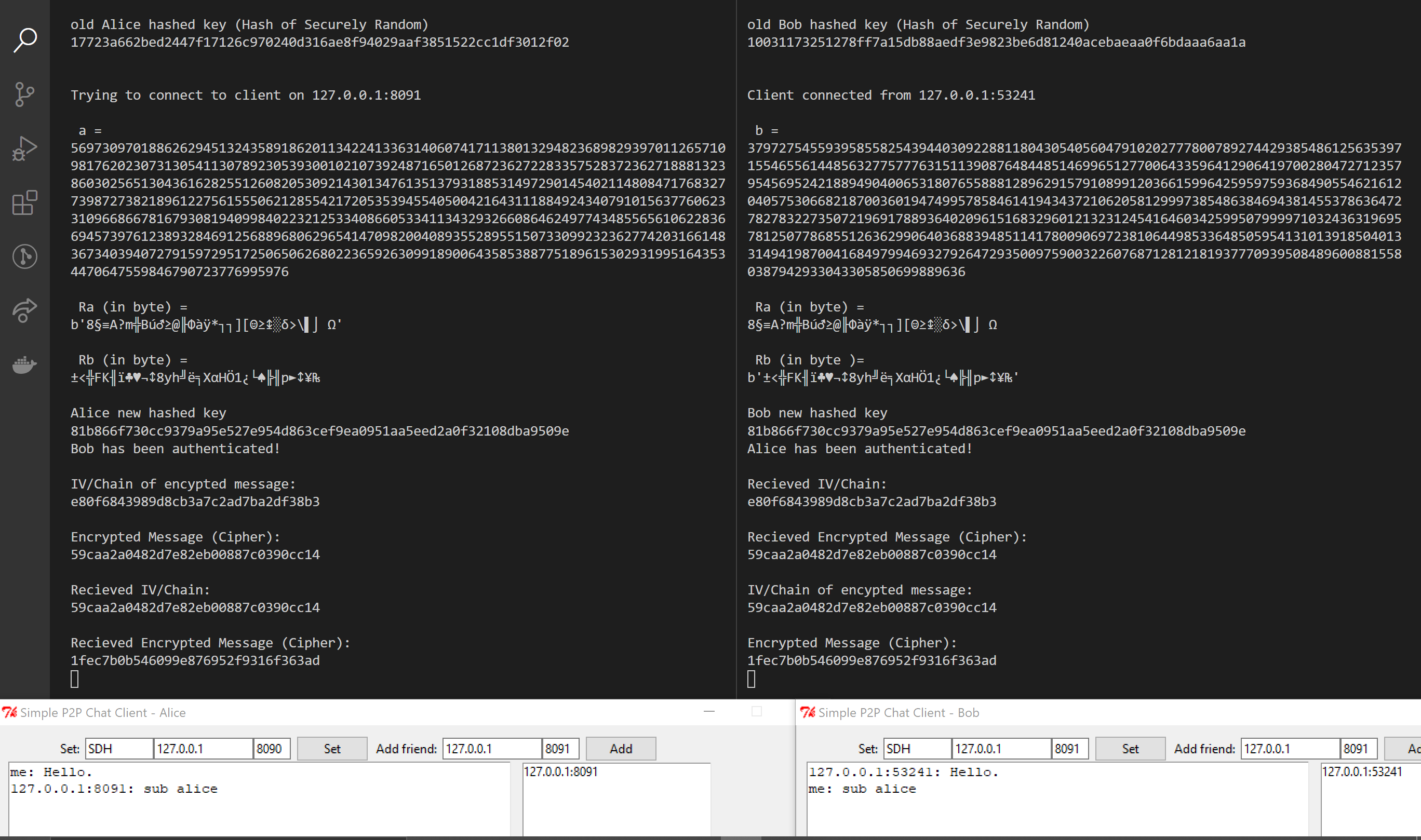
To terminate the program, press “Terminate” button, which will send a message to the other client and close the window.



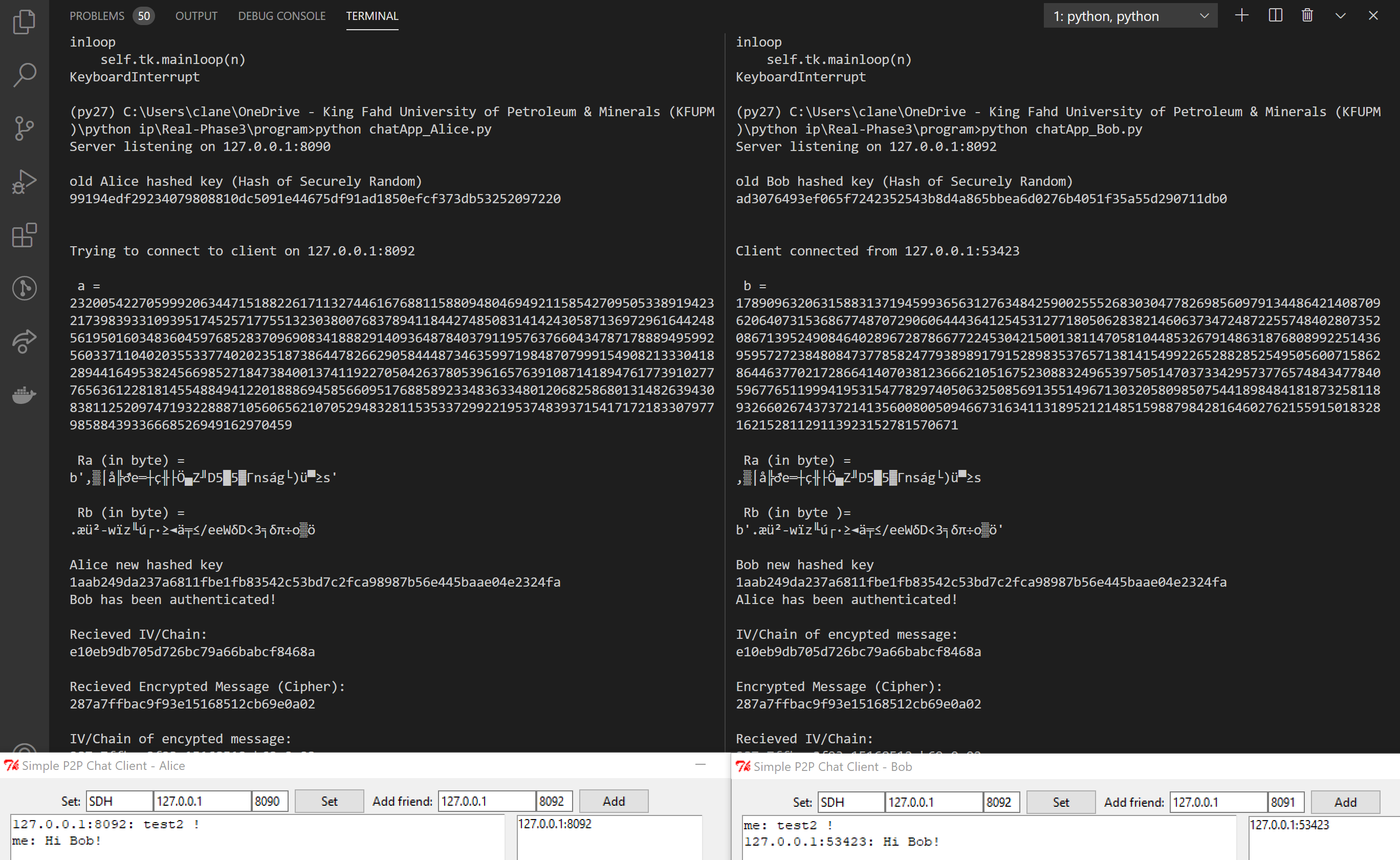
## Screenshots of Cypher stuff

1. Screenshot of 3 different matches showing old key, a, b, Ra, Rb, new key, authentication message, and encrypted messages with IV.

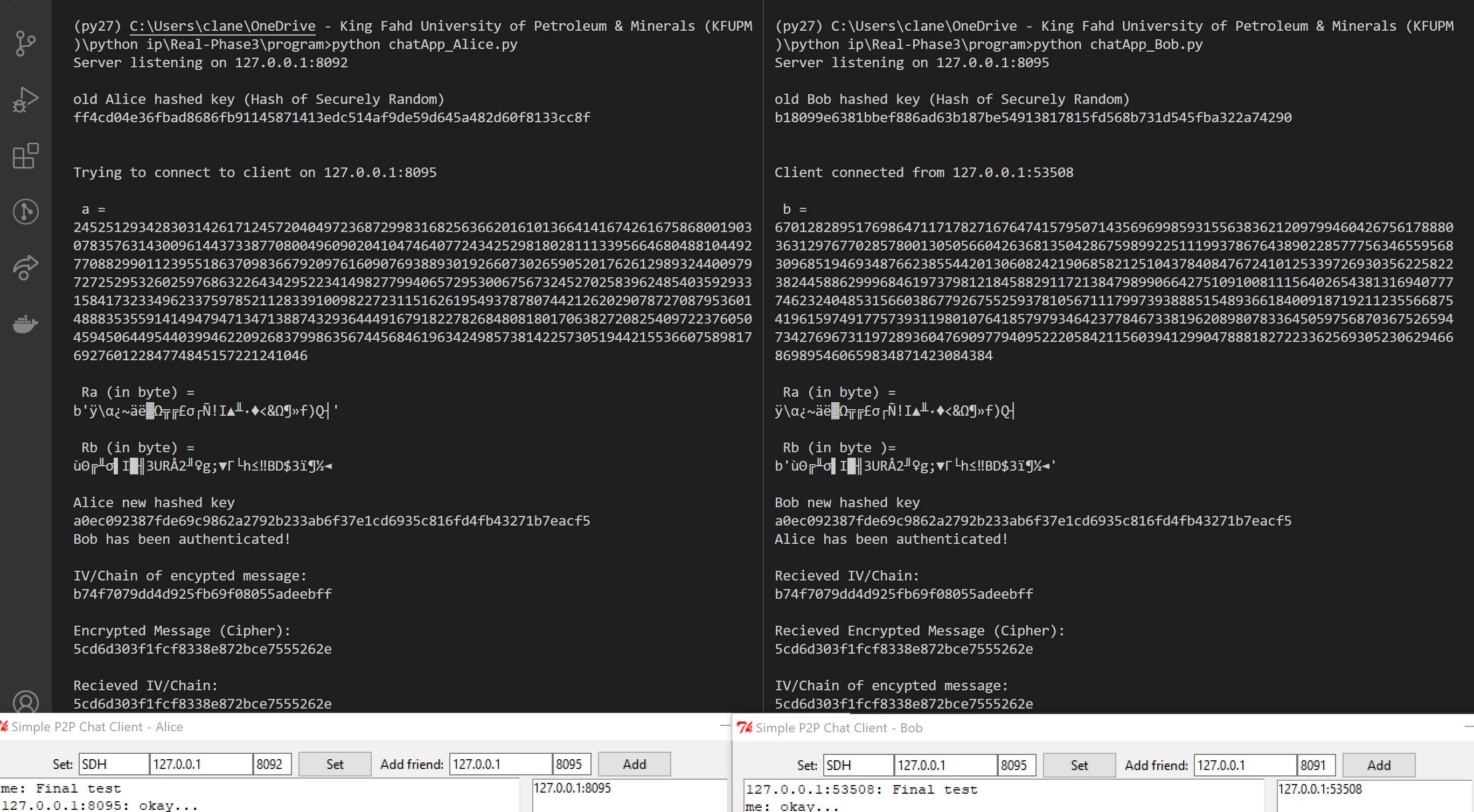
Screenshot 1



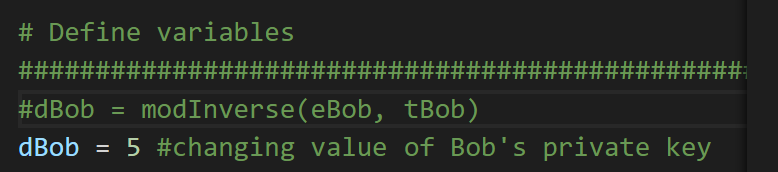
Screenshot 2

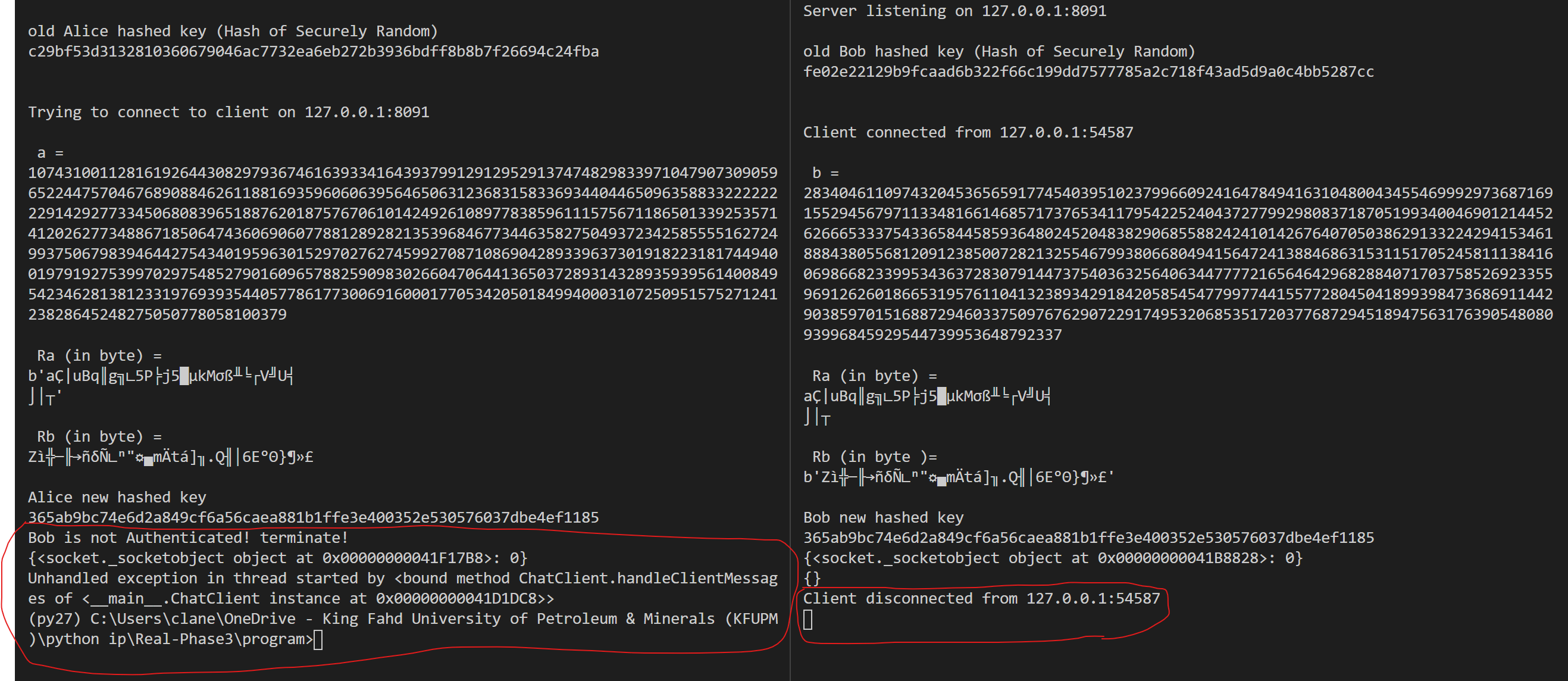


Screenshot 3

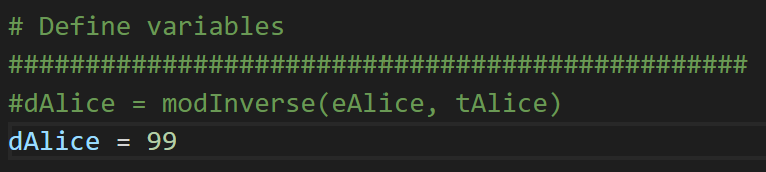


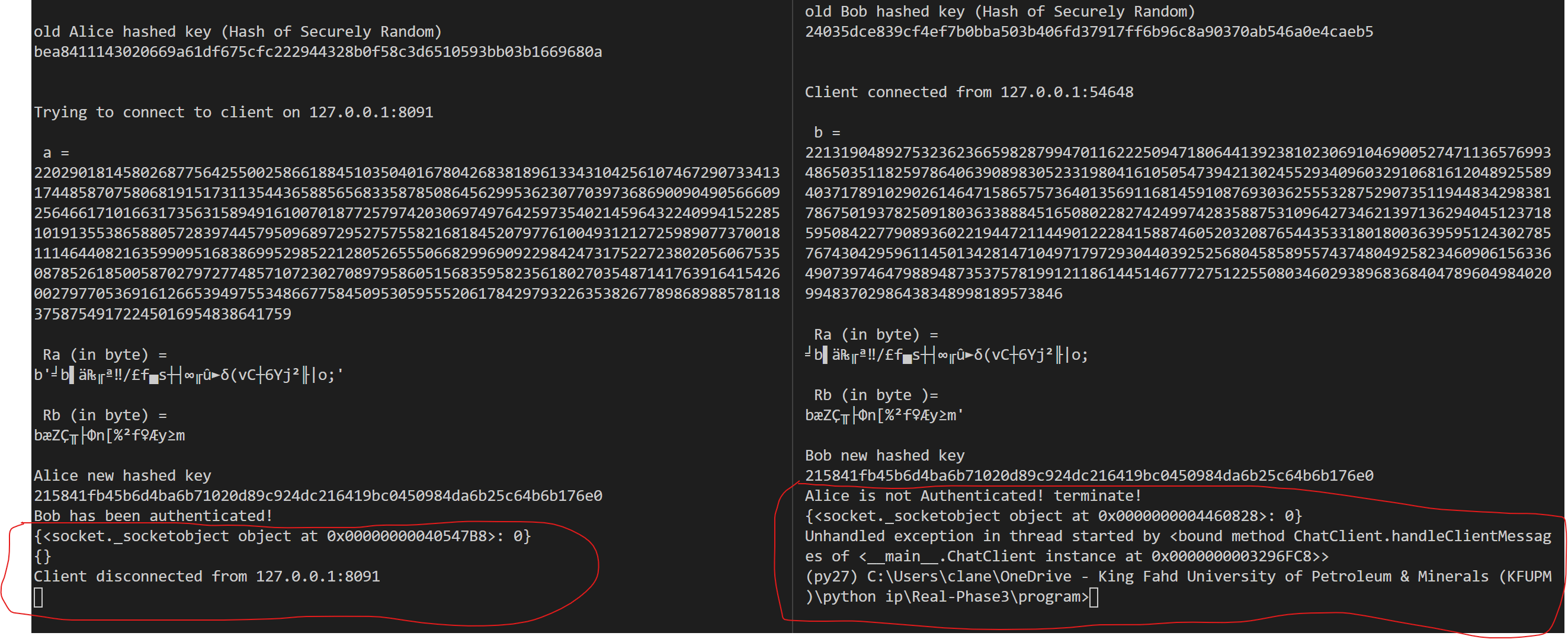
1. Trudy posing as Bob.





1. Trudy posing as Alice





## Readme

1. Download Anaconda 3/2, OR mini-anaconda 3/2, OR download python 2 ( At your own risk ). Install what you have downloaded. I recommend Anaconda because I have included an anaconda environment that can be installed using the included file, environment.yml. For any reason, if environment.yml did not work, you need to install the packages in environment.yml manually using pip after creating new environment in Anaconda. Will discuss it more next steps.
2. To import the environment in Anaconda, open Anaconda Prompt, go to directory of environment.yml file using command cd “directory”, write “conda env create --file environment.yml”, or, you can only use “conda env create” since the name of the yml file is the default. Note that all previous commands are to be written in Anaconda Prompt. You can then activate the environment using “activate py27” and use it straight away in Anaconda Prompt to run codes.
3. To create an environment IF and only IF the previous import step did not work, write “conda create –name py27 python=2.7.\*\*” in Anaconda Prompt. You can then use the environment by running it using “activate py27” command and the command “pip install -r requirements.txt” to install dependencies if needed. The previous command only works from HW directory.
4. Use cd “directory” in Anaconda Prompt to go to program file (chatApp.py) directory to run python file.
5. Write “python [file name]” in Anaconda Prompt, where file name is chatApp.py.
6. Follow program usage on the screenshots section (Same as phase 1). For completion, note that the above steps have been verified to work on two different machines.