# Secure Person2Person Micropayment with Text-based Client-Server Program

## Language

C++ (coded in Visual Studio Code)

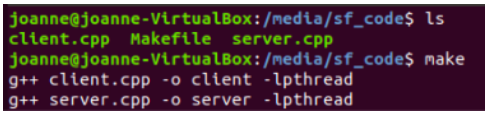
## Environment

VirtualBox 6.1 / Ubuntu 18.04

## Compilation

1. Under Linux environment, please make sure you have installed the Linux version of openssl before doing any of the following actions.
2. Next, in order to compile the source code, open the terminal in Linux.
3. Change the working directory to the folder containing the files including the source codes and the certificates. (In my case: `/media/sf_code` as seen in the screenshot)
4. Use the command `make` to run the Makefile that compiles the client and server programs.
   `$ make`

Example:



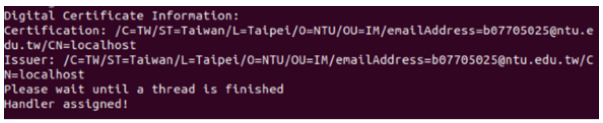

## Execution as an User

First, we need to execute the server program. Here I choose `16000` as the port to which the server should listen.

`$ ./server`


Next, execute the client program by passing 2 additional arguments, which are the IP address of the server and the port number to which it listens. Now you will be prompted to enter your own desired port number, which will be assigned to you to listen to transaction messages that other clients may send to you. In the example, the server's IP address is `127.0.0.1` and the port is `16000`; and `2000` as the client's port number.

`$ ./client 127.0.0.1 16000`

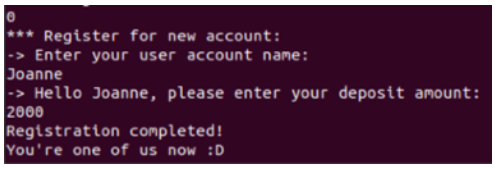



server

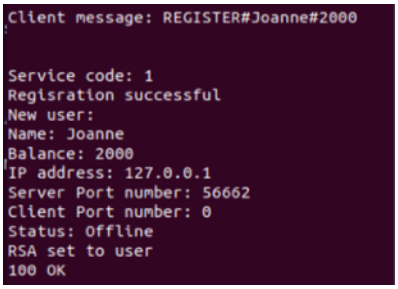

client

— Notice —
For a worker pool with size *n*, if *n* + 1 clients want to connect to server, the (*n* + 1)-th client will be blocked until one of the *n* clients exits.

1. REGISTER:
   After successfully connecting to the server, the program will ask you what service that you wish to use. First, I register an account with command `0`, named “Joanne” and deposit $2000.

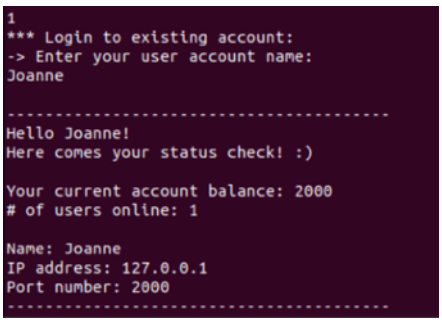



   Upon receipt of the registration request, the server shows: (`100 OK` indicates success)

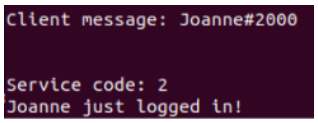



   — Notice —
   Server only accepts registrations with unique names. That is, if a client registers a name that is previously registered, this registration request will fail and return `210 FAIL`.
   e.g. I have registered an account named “Joanne”, then any other users registering later will not be able to register with the name “Joanne”.

2. LOGIN:
   Next, I will log in to my account with command `1`, and then the program will print out the list of the online users.



Upon receipt of the login request, the server shows:

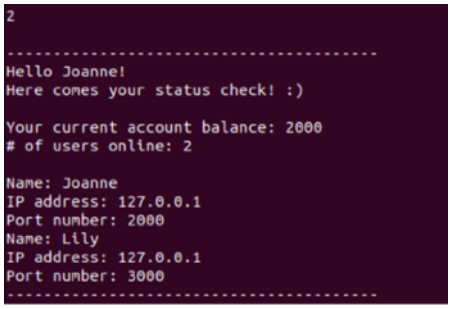



— Notice —
If the client logins with a name that has not been registered in the server's database, server will return `220 AUTH_FAIL`.

If the account has been logged in by another client, server will return `User already logged in!`
If the client has already logged in and tries to login again, server will return

`Only one account can log in once!`

3. LIST:
   Having opened another client program that are logged in, use command `2` to check the online list in Joanne's window:

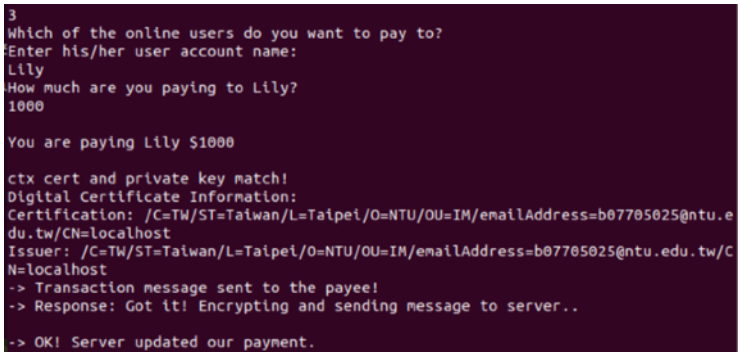



Upon receipt of the list request, the server shows:


— Notice —
If the client accesses the list without logging in first, server will return

`Please login to access the list!`

4. MICROPAYMENT:
   After checking the online list, Joanne's going to pay to Lily $1000. Joanne's program:



On the other hand, Lily's program looks like this:

```
Digital Certificate Information:
Certification: /C=TW/ST=Taiwan/L=Taipei/O=NTU/OU=IM/emailAddress=b07705025@ntu.e
du.tw/CN=localhost
Issuer: /C=TW/ST=Taiwan/L=Taipei/O=NTU/OU=IM/emailAddress=b07705025@ntu.edu.tw/C
N=localhost
Transaction message:
Joanne#1000#Lily
Message sent to server..
Server response:
TRANS OK
```

Upon receipt of the payment request, the server shows: (`TRANS OK` indicates success)

```
transMsg:
TRANS#Joanne#1000#Lily
original message by payer:
Joanne#1000#Lily
decrypted payee's message
decrypting payer's message
will be decrypting payer's message...
decrypted payer's message:
Joanne#1000#Lily
Transaction identities verified!
user balance sufficient, transacting
payer: Joanne
payee: Lily
amt: 1000
getting payer and payee indices
Joanne just paid $1000 to Lily
transaction complete
TRANS OK
```

— Notice —

If the payer does not have enough money to pay to the payee, server will return `TRANS FAIL`.

5. EXIT:
   Lastly, Joanne terminates her program.

```
4
Sad to see you leave :(
Server confirmed your departure.


Connection terminatd.
Thank you! :) See you again soon
```

Upon receipt of the payment request, the server shows:

```
Client message: EXIT

Service code: 5
Bye

Joanne just logged out.
Client Joanne is leaving!
```

## References


http://www.linuxhowtos.org/C_C++/socket.htm
https://github.com/hassanyousufx/Simple-Server-and-Chat-Program/blob/master/server.cpp
https://www.openssl.org/docs/man1.1.1/man3/RSA_private_encrypt.html
https://www.ibm.com/support/knowledgecenter/SSB23S_1.1.0.2020/gtps7/s5sple1.html