

Programming Assignment 第三階段

B11303043 施漢樺

編譯說明

本專案已提供 Makefile，可使用指令 **make** 進行編譯(若需清理編譯過的執行檔，可以先用 **make clean** 指令)。編譯成功後，將會生成 **client** 和 **server** 執行檔，也可以直接用 **g++** 指令來執行，如下兩張圖所示。

```
toby@toby:~/b11303043_part3$ make
g++ -Wall -std=c++17 -O2 -Wno-unused-but-set-variable client.cpp -o client -lssl -lcrypto -lpthread
g++ -Wall -std=c++17 -O2 -Wno-unused-but-set-variable server.cpp -o server -lssl -lcrypto -lpthread

toby@toby:~/b11303043_part3$ g++ -o client client.cpp -lssl -lcrypto
toby@toby:~/b11303043_part3$ g++ -o server server.cpp -lssl -lcrypto
```

執行說明

1. 啟動伺服器與客戶端程式碼

首先在終端機輸入 **./server <IP 地址> <埠號>** 來執行 server 端程式，如下圖 1 所示。接下來在終端機輸入 **./client** 來執行 client 端程式，系統會讓使用者輸入欲連接之伺服器的 ip 位置及 port，在連接後系統會顯示 **Connected to server** 來告知使用者成功連上伺服器，並顯示一個主選單，可以選擇註冊、登入、查詢資訊、轉帳、登出等操作。執行結果如下圖 2 所示。

圖 1:

```
toby@toby:~/b11303043_part3$ ./server 127.0.0.1 8888
Server listening on 127.0.0.1:8888
```

圖 2:

```
toby@toby:~/b11303043_part3$ ./client
Initializing SSL...
Enter server IP: 127.0.0.1
Enter server port: 8888
Connected to server.

1. Register
2. Login
3. Request Info
4. Transfer
5. Logout
Enter your choice:
```

2.註冊功能

選擇 1 進行註冊。在使用者輸入使用者名稱之後，程式會自動向 Server 發送註冊請求，預設帳戶餘額為 10,000。執行結果如下圖 3 所示。

圖 3:

```
1. Register
2. Login
3. Request Info
4. Transfer
5. Logout
Enter your choice: 1
Enter username: A
Server:
100 OK
```

3.登入功能

選擇 2 進行登入。在使用者輸入輸入使用者名稱與 Port Number 之後，程式會向 Server 發送登入請求，並啟動接收訊息的執行緒，另外，客戶端會顯示餘額、用來解密的 Public Key 與上線清單。執行結果如下圖 4 所示。

圖 4:

```
1. Register
2. Login
3. Request Info
4. Transfer
5. Logout
Enter your choice: 2
Enter username to login: A
Enter port number for login: 1111
Server:
10000
78c45535cb9297c4e868167f49b43006b4f9d7d35d0848e8e471a4bb1f557d869687fa9b9195a9c3
2893bc44716e31c4
1
A#127.0.0.1#1111
```

4.查詢功能

選擇 3 查詢最新的帳戶餘額、用來解密的 Public Key 與上線清單。執行結果如下圖 5 所示。

圖 5:

```
1. Register
2. Login
3. Request Info
4. Transfer
5. Logout
Enter your choice: 3
Server:
10000
78c45535cb9297c4e868167f49b43006b4f9d7d35d0848e8e471a4bb1f557d869687fa9b9195a9c3
2893bc44716e31c4
1
A#127.0.0.1#1111
```

5.轉帳功能

選擇 4 進行轉帳。輸入轉帳金額與接收方使用者名稱，Client 會自動查詢接收方的 IP 與 Port 並發送交易請求。若交易成功，轉帳者的使用者介面會顯示 Transfer OK!，執行結果如下圖 6 所示；另一端的接收者的介面會顯示交易通知，執行結果如下圖 7 所示。

圖 6(轉帳者 A):

```
1. Register
2. Login
3. Request Info
4. Transfer
5. Logout
Enter your choice: 4
Enter amount to transfer: 4000
Enter receiver username: B
Server:
Transfer OK!
```

圖 7(收款者 B):

```
1. Register
2. Login
3. Request Info
4. Transfer
5. Logout
Enter your choice: [Notification] A sent you $4000!
```

6.登出功能

選擇 5 登出，Client 會通知 Server 並斷開連線，如下圖 8 所示

圖 8:

```
1. Register
2. Login
3. Request Info
4. Transfer
5. Logout
Enter your choice: 5
Bye
toby@toby:~/b11393943_part3$
```

安全傳輸實作方法及流程說明

本系統的安全傳輸實作主要基於 **TLS (Transport Layer Security)** 協議，並結合 **RSA 非對稱加密技術**，確保通訊過程中的數據機密性、完整性和身份驗證。以下是詳細的安全傳輸設計和流程：

1. TLS 加密通訊:

- 使用 OpenSSL 提供的 TLS 加密協定，確保通訊數據在傳輸過程中受到保護，防止竊聽和篡改。
- 伺服器端使用 SSL_CTX 初始化安全上下文，並載入 .crt 和 .key 檔案以啟用 TLS。
- 客戶端也同樣載入其憑證以建立安全的 TLS 通道。
- 使用的是基於公鑰 (Public Key) 和私鑰 (Private Key) 的非對稱加密 RSA 演算法。

2. 安全傳輸流程:

- 客戶端與伺服器(包括 client 與 server 連線與 client 與 client 之連線)通過握手協商會話密鑰。其中握手過程包括:
 - 1. **身份驗證**：伺服器證書由客戶端驗證。
 - 2. **密鑰交換**：通過非對稱加密 (RSA) 生成密鑰。
 - 3. **安全參數協商**：協商使用的對稱加密算法和散列算法。
- 握手完成後，雙方使用會話密鑰對數據進行對稱加密傳輸。便能使用 OpenSSL 的 SSL_write() 和 SSL_read() 函數處理加密後的數據讀寫。
- 每個加密數據塊附加消息認證碼 (MAC)，確保數據未被篡改。如果消息被修改，接收方會在驗證 MAC 時發現不匹配，丟棄數據。
- 會話結束後客戶端或伺服器可通過 SSL_shutdown() 終止會話。

3. 資料安全保障:

- 每次傳輸請求，數據均通過 TLS 加密通道傳輸。
- Session Key 用於進一步加密敏感訊息，防止竊聽。
- TLS 的協議保證即使會話密鑰被竊取，也無法解密過去的通信。
- 即使惡意用戶截獲通信，也無法重複使用該密鑰。

環境說明

開發環境：Ubuntu 64 的虛擬機

執行環境：Ubuntu 22.04 / macOS Ventura

編譯器：GCC 12.3.0

必需的套件: OpenSSL (libssl-dev) 與 Pthread (libpthread)

參考資料

C/C++ Linux TCP Socket Server/Client 網路通訊教學 <https://shengyu7697.github.io/cpp-linux-tcp-socket/>

TCP Socket Programming 學習筆記 <https://zake7749.github.io/2015/03/17/SocketProgramming/>

TCP Server-Client implementation in C <https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/>

Using thread pools in C++ <https://ncona.com/2019/05/using-thread-pools-in-cpp/>

OpenSSL 官方文件 <https://www.openssl.org/docs/>