網路**socket programming**—第三階段（安全通訊）

B07705047 資管三 林昱辰

程式語言：C++

程式執行環境：WSL Ubuntu 18.04

執行說明：在WSL環境下cd to程式所在位置（desktop），先下make指令compile檔案（g++ -pthread -o client_s1 client_ssl1.cpp -lcrypto –lssl、g++ -pthread -o client_s1 client_ssl1.cpp -lcrypto –lssl、g++ -pthread -o server_s server_ssl.cpp -lcrypto -lssl），之後下./client_s1、./client_s2 或 ./server_s 執行程式。

安全傳輸實作的方法及流程說明：

I. SSL部分

- Client端—
  在跟 server 進行通訊前，首先 initialize ctx, ssl library，接著將自己的 certificate 和 key load 下來，先用 socket connect 之後，再開啟 ssl 通道並 connect，成功後便可利用 SSL_write 和 SSL_read 和對方進行 message 的傳遞和接收，ssl 內部會自己實作以下加密過程：利用接收者的 public key 對 message 進行加密，接收者收到後再用自己的 private key 解密。
- Server端—
  流程大致和 client 端相同，不同的部分在於 server 端建立 ssl 通道時是使用 SSL_accept，而 client 端是使用 SSL_connect（在 initialize ctx 時分別使用 server_method、client_method）。

II. 加密部分

client 間互相傳輸的部分，payer 像是 client，而 payee 像是 server，像上述步驟建立 ssl 通道後，payer 用自己的私鑰將訊息加密，並傳給 payee，payee 會取得對方的憑證裡公鑰，將加密訊息轉為明文，同時也會用自己的私鑰加密密文，將明文和經過兩次加密的密文連接，傳給 server，server 便會根據明文中的 payee 和 payer 名字，找到對應的公鑰（在 user login 時即存取），將密文解密後與明文比對，正確即更新雙方帳戶金額。

參考資料來源：

https://hackmd.io/@G9IwPB5oTmOK_qFXzKABGg/rJkvqdgJ
https://hackmd.io/@J-How/B1vC_LmAD

https://aticleworld.com/ssl-server-client-using-openssl-in-c/