Socket Programming #3 說明文件

B10703132 謝佳妤

環境：Ubuntu (使用 UTM 虛擬機)

執行方式

- 編譯

```
g++ client.cpp -pthread -lssl -lcrypto -o client
g++ server.cpp -pthread -lssl -lcrypto -o server
```

- 執行

```
// server 端
./server 8888

// client 端
./client 127.0.0.1 8888
```

技術細節

key 產生過程

- 使用 `init_client_ctx()` 或 `init_server_ctx()` 初始化 ctx
- 使用 `EVP_RSA_gen()` 生成 private key
- 使用 `generate_certificate(pkey)` 生成自簽名憑證，其中包含了 public key
- 使用 `load_certificate(ctx, cert, pkey)` 將憑證及 private key 讀進 ctx
- 流程如下：

```
SSL_CTX* ctx = init_client_ctx();
pkey = EVP_RSA_gen(2048);
X509* cert = generate_certificate(pkey);
load_certificate(ctx, cert, pkey);
```

SSL 通道

- 在 socket 成功連線後
- 使用 `SSL_new(ctx)` 基於 ctx 產生一個新的 SSL
- 使用 `SSL_set_fd(ssl, server_socket_fd)` 將 SSL 通道與 socket fd 連結
- server 使用 `SSL_accept(ssl)`，client 使用 `SSL_connect(ssl)` 建立 SSL 連線

- 成功連線後
  - 使用 `show_certificate(ssl)` 印出對方的憑證
  - 使用 `get_peer_public_key(ssl)` 拿到對方的 public key，作為後續加密使用
- 以 client 為例：

```
// client
// 假設已經成功透過 socket API 與 server 連線

ssl = SSL_new(ctx);
SSL_set_fd(ssl, server_socket_fd);

// 建立 SSL 連線
if (SSL_connect(ssl) == -1)
{
    ERR_print_errors_fp(stderr);
    SSL_shutdown(ssl);
    SSL_free(ssl);
    return -1;
}
else
{
    printf("Connected with %s encryption\n", SSL_get_cipher(ssl));
    show_certificate(ssl);
    server_public_key = get_peer_public_key(ssl);
}
```

安全的訊息傳輸

- 加密
  - 使用 `encrypt(message, public_key)` 進行加密，其中 public_key 為接收方的公鑰
  - 使用 `SSL_write()` 取代原先的 `send()` ，將密文傳送給對方

```
string ciphertext = encrypt(string(command), server_public_key);

int bytes_sent = SSL_write(ssl, ciphertext.c_str(), ciphertext.size())
if (bytes_sent == -1) {
    cerr << ("Fail to send message to server") << endl;
    continue;
}
```

- 解密
  - 使用 `SSL_read()` 取代原先的 `recv()` ，接收密文
  - 使用 `decrypt(message, private_key)` 進行解密，其中 private_key 為自己的私鑰

```
char buffer[BUFFER_SIZE];

int bytesReceived = SSL_read(ssl, buffer, BUFFER_SIZE);
if (bytesReceived <= 0) {
    cerr << "Client disconnected or error occurred." << endl;
    break;
}

string plaintext = decrypt(string(buffer, bytesReceived), pkey);
```

p2p transaction

- sender 用 receiver 的 public key 加密原始訊息，傳給 receiver
- receiver 用自己的 private key 解密
- receiver 用 server 的 public key 加密原始訊息，傳給 server
- server 用自己的 private key 解密
- server 用 sender 的 public key 加密確認訊息，傳給 sender
- sender 用自己的 private key 解密

參考資料

- https://hackmd.io/@G9IwPB5oTmOK_qFXzKABGg/rJkvqdgJ_#%E7%B7%A8%E8%AD%AF
- https://hackmd.io/@J-How/B1vC_LmAD#FAQ