Programming Assignment

Course: Computer Networks
Professor Yeali S. Sun

November 18, 2021

1. The Problem of "A Secure Person2Person (P2P) Micropayment System"（安全的第三方支付使用者對使用者小額付款系統）

一套具安全傳輸的簡單網際網路第三方支付使用者對使用者小額付款（Micropayment）系統。此系統包含三大功能：

一、第三方支付 Server 端對 Client 端（使用者）的統一管理，包含帳號管理、好友名單管理、認證以及 Client 帳戶管理等。

二、Client 間即時通訊。

三、Client 與 Server 以及 Client 間的通訊，都可以各自加密，加密的鑰匙（encryption key，又稱 secret key）由當下通訊的雙方議定。

本作業的目標是設計與實作一套簡單的好友間轉帳功能。同學將設計、實作一套安全傳輸的簡單「安全的第三方支付使用者對使用者小額付款系統」包含 Client 與 multithreaded Server 端的軟體，以及安全傳輸的軟體撰寫。

Client 端的兩個主要功能：

- 安全的與第三方支付 Server 的通訊
- 一對一安全的 Client 間對談

Multi-threaded Server 端的主要功能：

- 接受 Client 的安全連結，並根據要求（request）回覆訊息（reply）

安全通訊的主要功能：

- 每一個 Client 與 Server 間，以及 Client 間的通訊，都必須加密，加密的鑰匙（encryption key，又稱 secret key）由當下通訊的雙方議定。

2. Requirements

- ✧ 本作業所有的通訊皆須採用 TCP protocol 以達到可靠傳輸，分三階段繳交及驗收：
  - ■ 第一階段，同學要先完成 Client 端的程式；
  - ■ 第二階段完成 Multi-threaded Server 端程式；
  - ■ 第三階段完成安全通訊的功能。
- ✧ 可使用的語言及 Library：Unix/Linux Socket Programming(in C/C++)、Win Socket；不可使用Java、C#、Python、……等高階語言。

A. Client

Client 端程式必須能夠：

a. 向 Server 註冊（註冊時輸入使用者名稱，每人預設帳戶餘額一萬）。
b. 登入助教所提供 Server 端程式（填入使用者名稱、Port Number）。
c. 向 Server 要最新的帳戶餘額、上線清單、Public key 並接收 Server 端的回覆。
d. 和其它 Client 執行轉帳功能（不可透過 Server 端轉送）。
e. 進行離線動作前需主動告知 Server 端程式。

上線清單包含的訊息為：

- 上線的總人數
- 線上的使用者名稱、其 IP address 以及可用來通訊的 port number。

B. Multi-Threaded Server

A multithreaded Server is capable of serving multiple requests in parallel. The Server will create a separate thread to handle each of the connections for accepted requests. There will also be a main thread, in which the Server listens for Clients that want to establish connections.

第二階段 Server 端程式的開發要能提供 Client 端的註冊與登入，發送 Client 目前的帳戶餘額與上線清單的回覆訊息給 Client，以及接收處理 Client 端離線前的通知，Server 提供的功能**請使用 thread 及 worker pool 的方式進行程式的開發，不要使用 fork。**

C. Client 與 Server 溝通

Client 與 Server 間的溝通訊息主要有四種：

(1) Client 端向 Server 註冊：

Client 端傳給 Server 端的訊息為：

REGISTER#<*UserAccountName*>

Server 端會回給 Client 端註冊成功或不成功的訊息分別為：

100<*space*>OK<*CRLF*>
210<*space*>FAIL<*CRLF*>

(2) Client 端登入 Server：

Client 端傳給 Server 端的訊息為：

<*UserAccountName*>#<*portNum*>

若使用者有註冊過，Server 端會回給 Client 端上線清單，清單格式為：

<accountBalance><CRLF>
<serverPublicKey><CRLF>
<number of accounts online><CRLF>
<userAccount1>#<userAccount1_IPaddr>#<userAccount1_portNum><CRLF>
<userAccount2>#<userAccount2_ IPaddr>#<userAccount2_portNum><CRLF>
…

若使用者尚未註冊過，Server 會回傳給 Client 端驗證失敗的訊息：

220<*space*>AUTH_FAIL<*CRLF*>

(3) Client 端向 Server 要最新的帳戶餘額與上線清單：

Client 端傳給 Server 端的訊息為：

List

Server 端會回給 Client 端上線清單，清單格式為：

<accountBalance><CRLF>
<serverPublicKey><CRLF>
<number of accounts online><CRLF>
<userAccount1>#<userAccount1_IPaddr>#<userAccount1_portNum><CRLF>
<userAccount2>#<userAccount2_ IPaddr>#<userAccount2_portNum><CRLF>
…

(4) Client 端結束程式：

Client 端傳給 Server 端的訊息為：

Exit

Server 端會回給 Client 端上線清單，清單格式為：

Bye<*CRLF*>

(5) Client 端送 micropayment transaction 訊息給 Server：

Client 端之間的訊息傳送格式為：

<MyUserAccountName>#<payAmount>#<PayeeUserAccountName>

本訊息 Client A 要用好友 B 的 public key 加密。好友 B 收到後用自己的 private key 解密取得轉帳額度。確認金額後再用 Server 的 public key 將訊息加密，送給 Server（此處假設 B 不會竄改金額)。Server 拿到後用自己的 private key 解密並更新雙方帳戶餘額。訊息內容假設都是 ASCII 字元文字（text）內容。

注意事項：**Server 端不替 Client 端做任何訊息的 relay。**

D. 安全傳輸

Socket 安全傳輸部分，請使用 openssl 這個 open source toolkits https://www.openssl.org/，並且你的 source code 中使用 openssl toolkits 進行加密的安全傳輸。

【套件安裝】

- 在 Unix/Linux 上（ex. Ubuntu)可直接用下列指令進行安裝。
  $ apt-get install openssl
- 在 Windows 上請自行上網下載 Openssl for winsocket 版本。

3. 作業繳交

本次作業分三階段繳交：

A. 第一階段：Client 端程式

助教將在本階段提供 Server 端程式執行檔。該 Server 端程式需在 Linux kernel 2.6.x 環境上執行。同學可以利用該程式來測試自己的 Client 端程式功能是否正常。執行 Server 端程式 command 的格式為：

$ ./*<Server_name><space><portNum><space><Option>*

Option 的選項有 -d, -s, -a，說明如下：

(每次執行只能輸入一個 Option 參數)

-d：Server 只會顯示簡單的訊息表示 Client 註冊、登入或離開。

-s：除了以上，Server 在每次有 Client 登入或離開時都會顯示現在上線的清單。

-a：除了以上，Server 還會顯示每一次 Client 與 Server 之間的訊息傳送。

此 Option 是為了方便同學除錯，顯示的訊息可以當作參考。

Server name 為程式執行檔名稱，port 必須在 1024 到 65535 之間。

本階段為 Client 端程式的開發，同學所撰寫的 Client 端程式必須要能：

a. 向 Server 註冊（註冊時輸入名稱）。
b. 登入助教所提供 Server 端程式（填入使用者名稱、Port Number）。
c. 向 Server 要最新的帳戶餘額、上線清單、Public key 並接收 Server 端的回覆。
d. 和其它 Client 執行轉帳功能（不可透過 Server 端轉送）。
e. 進行離線動作前需主動告知 Server 端程式。

B. 第二階段：Server 端程式

完成 Server 端的程式。可以用你的 Client 端的程式與 Server 端的程式一起執行。

C. 第三階段：Client 端以及 Server 端安全通訊程式

完成 Client 端以及 Server 端的安全通訊程式。

4. Demo

驗收階段請同學自備筆電，並確保筆電電量充足以及能夠連上網。驗收時提早 5 分鐘到場，以提前準備好執行環境。

A. **第一階段：**助教會在自己的機器上執行 Server 端程式（Server IP 及 port number 將當場告知）。驗收目標是你的 Client 程式能夠連上助教提供的 Server 程式並完成上述功能。

Demo 時間：December 8th & 9th, 2021 （將提供時間表供同學填選）

Demo 地點：教研管 317

B. 第二階段：

助教會測試你的 Client and Server 程式是否可以正常執行。

Demo 時間：December 22th & 23th, 2021 （將提供時間表供同學填選）

Demo 地點：教研管 317

C. 第三階段：

助教會測試你的 Client and Server 程式是否能進行安全傳輸。

Demo 時間：January 18th & 19th, 2022（將提供時間表供同學填選）

Demo 地點：教研管 317

5. Submission

A. 第一階段：

- **需繳交 Source code 以及說明文件**
  - 上傳繳交的部份包含以下四項；
    1. Source Code（Client 端程式的原始碼）。
    2. 操作說明文件 PDF 檔（包含如何編譯、執行 Client 端程式，程式執行環境說明，參考資料、來源等）。
    3. Binary 執行檔（已 Compile 及 Linking 完成並可執行的 Client 端程式）。
    4. 用以編譯程式之 Makefile。
  - 請將上述四項檔案壓縮成：學號_part1.tar.gz (e.g. b027050xx_part1.tar.gz)，上傳至 NTU COOL 平台課程作業區。
- **Deadline: Sunday, 12 December 2021, 23:59:59**

B. 第二階段：

- **需繳交 Source code 以及說明文件**
  - 上傳繳交的部份包含以下四項；
    1. Source Code（Server 端程式的原始碼）。
    2. 操作說明文件電子檔（包含如何編譯、執行 multi-threaded Server 端程式，程式執行環境說明，參考資料、來源等）.

3. Binary 執行檔 (已 Compile 及 Linking 完成並可執行的 Server 端程式)。
4. 用以編譯程式之 Makefile。

請將上述四項檔案壓縮成：學號_part2.tar.gz (e.g. b027050xx_part2.tar.gz)，上傳至 NTU COOL 平台課程作業區。

✧ **Deadline: Sunday, 26 December 2021, 23:59:59**

C. 第三階段:

✧ **需繳交 Source code 以及說明文件**

■ 上傳繳交的部份包含以下四項;

1. Source Code（Server 端 與 Client 端程式的原始碼）。
2. 操作說明文件電子檔（包含如何編譯、執行 Client 端及 multi-threaded Server 端程式，程式執行環境說明，**安全傳輸實作的方法及流程說明**，參考資料、來源等）。
3. Binary 執行檔（已 Compile 及 Linking 完成並可執行的 Client 端及 Server 端程式)。
4. 用以編譯程式之 Makefile。

■ 請將上述四項檔案壓縮成：學號_part3.tar.gz (e.g. b027050xx_part3.tar.gz)，上傳至 NTU COOL 平台課程作業區。

✧ **Deadline: Sunday, 23 January 2022, 23:59:59**

6. Grading

(a) (35%) Client 端程式的評分方式如下:

■ 說明文件：20%

■ 基本要求（Client 端可以註冊（填入使用者名稱）、登入 Server、向 Server 請求查看所有線上使用者的資訊、可接收 Server 所回傳的訊息、Client 間可以彼此傳送訊息、主動向 Server 發送離線訊息）：80%

■ Bonus（介面、GUI、Exception handling）：15 %

(b) (35%) Server 端程式的評分方式如下:

- 說明文件：20%
- 基本要求（Server 端可以接收多個 Client 的連線並各用一個 thread 處理 Client 端的連線、提供 Client 端的註冊（填入使用者名稱）、登入（填入使用者名稱、port number）、發送最新帳戶餘額、上線清單、Public key 給 Client，以及接收處理 Client 端離線前的通知）：80%
- Bonus（介面、GUI、Exception handling）：15 %

(c) (30%) 具安全通訊的 Client 與 Server 程式，評分方式如下：

- 說明文件：20%
- 具安全通訊的 Client 與 Server 程式：80%
- Bonus（介面、GUI、Exception handling）：15 %