**🙞🙞TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN🙜🙜**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙣🕮🙡**

**BỘ MÔN HỆ ĐIỀU HÀNH**

ĐỒ ÁN 1: QUẢN LÍ HỆ THỐNG TẬP TIN TRÊN WINDOWS

* Giảng viên: Lê Viết Long
* Sinh viên thực hiện: Võ Hoàng Bảo Duy – 19127027

Trần Ngọc Lam – 19127040

Lê Minh Sĩ – 19127064

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Khoa Công nghệ Thông tin

Đại học Khoa học Tự nhiên TP HCM

Tháng 12/ 2020

Mục lục

[1. Thông tin chung 3](#_Toc59141276)

[a. Thông tin nhóm 3](#_Toc59141277)

[b. Bảng phân công công việc 3](#_Toc59141278)

[2. Cách xây dựng chương trình 4](#_Toc59141279)

[a. Kịch bản chương trình 4](#_Toc59141280)

[b. Cấu trúc chương trình 7](#_Toc59141281)

* [Các kiểu dữ liệu cơ bản 7](#_Toc59141282)
* [Các hàm chính: 8](#_Toc59141283)

[3. Sử dụng Wireshark bắt gói tin 11](#_Toc59141284)

# Thông tin chung

## Thông tin nhóm

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Họ Tên** | **Email** |
| *19127027* | Võ Hoàng Bảo Duy | [19127027@student.hcmus.edu.vn](mailto:19127027@student.hcmus.edu.vn) |
| *19127040* | Trần Ngọc Lam | [19127040@student.hcmus.edu.vn](mailto:19127040@student.hcmus.edu.vn) |
| *19127064* | Lê Minh Sĩ | [19127064@student.hcmus.edu.vn](mailto:19127064@student.hcmus.edu.vn) |

## Bảng phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ Tên** | **Phân công công việc** | **Mức độ hoàn thành** |
| 19127027 | Võ Hoàng Bảo Duy | * Đọc các thông tin được mô tả trong Boot Sector (FAT32). * Đọc và phân tích bảng RDET và bảng FAT. * Xây dựng cây thư mục cho hệ thống FAT32. | 100% |
| 19127040 | Trần Ngọc Lam | * Viết báo cáo. * Xây dựng giao diện hiển thị thông tin cho chương trình. | 100% |
| 19127064 | Lê Minh Sĩ | * Đọc các thông tin được mô tả trong Partition Boot Sector (NTFS). * Đọc và phân tích Master file table. * Xây dựng cây thư mục cho hệ thống NTFS. | 100% |
| Toàn bộ project | | | 100% |

# Cách xây dựng chương trình

## Kịch bản chương trình

|  |  |
| --- | --- |
| **Client** | **Server** |
| **Chức năng kết nối từ client đến server** | |
|  | Nhập IP của máy và Port rồi chọn “Run Server” để khởi tạo server. |
| **Connect :** Nhập IP server và Port server, client kết nối với server với IP server và Port server đang mở | Chấp nhận kết nối và tạo thread mới để gửi nhận dữ liệu với client |
| **Close**: Client kết thúc phiên kết nối với server. | Đóng kết nối và hủy thread với client (Ghi chú: Server vẫn hoạt động bình thường) |
| **Chức năng đăng nhập và quản lý thông tin người dùng** | |
| **Login**: login server với username và password của người dung (password không hiện ra khi người dung nhập). Khi đăng nhập, sẽ hiện ra yêu cầu có bảo mật tài khoản người dùng khi gửi dữ liệu cho Server không?   * Nếu có: thực hiện mã hóa username và password của người dùng, sau đó gửi chuỗi đã mã hóa cho server. * Nếu không: gửi username và password dưới dạng chuỗi (không sử dụng thuật toán mã hóa). | Nhận username và password từ phía Client. Thực hiện kiểm tra thông tin sau:   * Username không tồn tại trong Database → thông báo lỗi và gửi message phản hồi cho client. * Username tồn tại và password không đúng → thông báo lỗi và gửi message phản hồi cho client. * Username tồn tại và password đúng → thông báo đăng nhập thành công. |
| **Sign up**: đăng ký tài khoản mới với username và password muốn tạo (password không hiện ra khi người dùng nhập). Khi đăng ký tài khoản mới, sẽ hiện ra yêu cầu có bảo mật tài khoản người dùng khi gửi dữ liệu cho Server không?   * Nếu có: thực hiện mã hóa username và password của người dùng, sau đó gửi chuỗi đã mã hóa cho server. * Nếu không: gửi username và password dưới dạng chuỗi (không sử dụng thuật toán mã hóa). | Nhận username và password từ phía Client. Thực hiện kiểm tra thông tin sau:   * Username đã tồn tại trong database → thông báo lỗi và gửi message phản hồi cho client. * Username không tồn tại trong database → thông báo đăng ký thành công, lưu tài khoản mới vào database và gửi yêu cầu client đăng nhập. |
| **Change Password:** (Lưu ý: cần phải đăng nhập để thực hiện yêu cầu này) yêu cầu đổi password mới của user, yêu cầu người dùng nhập username và password cũ, password mới và xác nhận lại password mới (password được hiện ra khi người dùng nhập). Khi thay đổi password, sẽ hiện ra yêu cầu có bảo mật tài khoản người dùng khi gửi dữ liệu cho Server không?   * Nếu có: thực hiện mã hóa password của người dùng, sau đó gửi chuỗi đã mã hóa cho server. * Nếu không: gửi password dưới dạng chuỗi (không sử dụng thuật toán mã hóa). | Nhận username và password từ phía Client. Thực hiện kiểm tra thông tin sau:   * Username không tồn tại trong Database → thông báo lỗi và gửi message phản hồi cho client. * Username tồn tại và password cũ không đúng → thông báo lỗi và gửi message phản hồi cho client. * Username tồn tại và password cũ đúng → nhận password mới từ phía Client → cập nhật password mới trong database. |
| **Check user:** kiểm tra thông tin của một người dung khác. Tất cả người dung đang online (ngoại trừ người dung hiện hành) sẽ được hiển thị tại khung “Online friends” ở tại giao diện của người dung. |  |
| **Set Information:** thiết lập hoặc thay đổi thông tin cá nhân của người dùng hiện hành (đang đăng nhập) chỉ bao gồm ngày tháng năm sinh. | Nhận lệnh từ phía Client và thực hiện các lệnh cập nhật trong database. |
| **Show Information:** hiện thông tin cá nhân của người dung hiện hành (đang đăng nhập) chỉ bao gồm ngày tháng năm sinh. | Nhận lệnh từ phía Client và tìm kiếm thông tin trong database. |
| **Chức năng upload, download dữ liệu** | |
| **Upload:** thực hiện upload dữ liệu lên server.   * Change Name: thực hiện đổi tên file thành tên mới. | Nhận yêu cầu upload file từ Client. Nhận nội dung và lưu vào thư mục hiện hành trên Server. |
| **Download:** thực hiện download dữ liệu lên server. | Nhận yêu cầu download file từ phía Client. Server kiểm tra file có tồn tại hay không?   * Nếu có: cho phép download tại thư mục hiện hành trên server. * Nếu không có: thông báo lỗi và gửi message cho client. |

**Tạo kênh chat riêng cho cặp người dung đang online (1:1)**

|  |  |  |
| --- | --- | --- |
| **Client 1** | **Client 2** | **Server** |
| **Chat:** thực hiện yêu cầu chat với user online được hiện trong khung “Online friends” của giao diện | Tạo cửa sổ chat cho tác vụ chat | Nhận thông điệp từ Client 1 gửi thông điệp báo cho Client 2 có yêu cầu chat. Thực hiện nhận và gửi dữ liệu qua lại cho Client 1 và Client 2 |
| **Send File:** thực hiện yêu cầu gửi một file qua cho Client 2. | Hiện khung thông báo có đồng ý nhận file được gửi từ Client 1 hay không?   * Nếu có: cho phép download tại thư mục hiện hành trên server. * Nếu không: không thể thực hiện việc gửi file | Nhận yêu cầu từ client 1 và gửi thông báo cho Client 2:   * Thực hiện việc gửi file và download về thư mục hiện hành của Client 2 nếu được đồng ý. |

## Cấu trúc chương trình

*Giao thức thực hiện:* TCP/IP

*Công cụ lập trình:* Microsoft Visual Studio 2019

*Môi trường lập trình:*Winform .Net Framwork

*Môi trường hoạt động:*Yêu cầu máy tính cài đặt .net framework 4.0 trở lên.

### Các kiểu dữ liệu cơ bản

|  |  |  |
| --- | --- | --- |
| *LoginStruct* | strUsername: String | Tên đăng nhập của tài khoản |
| strPassword: String | Mật khẩu đăng nhập của tài khoản |
| isEncrypted: bool | True: mã hóa thông tin đăng nhập  False: không mã hóa thông tin |
| *SignupStruct* | strUsername: String | Tên đăng nhập cần đăng ký |
| strPassword: String | Mật khẩu cần đăng ký |
| isEncrypted: bool | True: mã hóa thông tin đăng nhập  False: không mã hóa thông tin |
| *PrivateMessageStruct* | strToUsername: String | Tên người muốn nhận tin nhắn |
| strMessage: String | Nội dung tin nhắn |
| *PrivateFileStruct* | strUsername: String | Tên người dung cần gửi tới |
| strFilename: String | Tên file cần gửi |
| iPackageNumber: int | Xác định đây là gói tin thứ iPackageNumber của file |
| iTotalPackage: int | Tổng số package của file cần gửi |
| bData: array<Byte> | Data một phần của file |
| *SetInformationStruct* | username: String | Tên đăng nhập của người dùng |
| birthdate: String | Ngày sinh của người dùng |
| *UserStatusStruct* | lstOnlineUsers: array<String> | Tên người dung đang online |
| *UploadPublicFileStruct* | filename: String | Tên file cần upload |
| iPackageNumber: int | Xác định đây là gói tin thứ iPackageNumber của file |
| iTotalPackage: int | Tổng số package của file cần gửi |
| bData: array<Byte> | Data một phần của file |
| iFileSize: int | Kích thước của file |

### Các hàm chính:

* Các thông điệp được thiết kế sao cho thể hiện được nội dung của thông điệp cần truyền tải và nhận, cấu trúc thông điệp sẽ được dùng chung bởi Client và Server.
* Thông điệp sẽ được *pack()* thành *array<Byte>^* để truyền đi và sẽ được *unpack()* khi nhận.
* 4 bytes đầu tiên kiểu int32 của mỗi gói tin thông diệp dùng để xác định loại thông điệp. Loại thông điệp kiểu **enum** sẽ được ép kiểu thành **int32** để gửi đi.

enum class MessageType {

Login, ResponseLogin, Signup, ResponseSignup, UserStatus, PrivateMessage, PrivateFile, RequestSendFile, ChangePassword, ResponseChangePassword, LoginNotification, LogoutNotification, ResponseSendFile, SetInfor, RequestInfor, ResponseInfor

};

|  |  |
| --- | --- |
| *int AppSocket::connectToServer(String^ \_ServerIPAddr, int \_ServerPort)*   * Kết nối tới server | |
| \_ServerIPAddr: String | Địa chỉ IP server |
| \_ServerPort: int | Port server |
| *int AppController::createThreadListenMessageFromServer()*   * Tạo một thread mới để lắng nghe các thông điệp từ Server gửi về. | |
| *System::Void AppController::ListenMessage*()   * Lắng nghe thông điệp được truyền về từ server và hiển thị ra màn hình. | |
| *int AppController::login(String^ \_Username, String^ \_Password)*   * Nhận vào username và password, mã hóa thông tin (nếu có) và gửi về server để thực hiện kiểm tra đăng nhập có thành công hay không. | |
| \_Username: String | Tên đăng nhập tài khoản |
| \_Password: String | Mật khẩu của tài khoản |
| *int AppController::signup(String^ \_Username, String^ \_Password)*   * Nhận vào username và password, mã hóa thông tin (nếu có) và gửi về server để thực hiện đăng ký tài khoản mới. | |
| \_Username: String | Tên đăng nhập cần đăng ký |
| \_Password: String | Mật khẩu của tài khoản cần đăng ký |
| *int AppController::changePassword(String^ Username, String^ oldPassword, String^ newPassword, String^ confirmnewPassword)*   * Nhận vào username, password cũ và password mới, mã hóa thông tin (nếu có) và gửi về server để thực hiện đổi password mới cho tài khoản. | |
| Username: String | Tên đăng nhập |
| oldPassword: String | Mật khẩu của tài khoản |
| newPassword: String | Mật khẩu mới |
| confirmedPassword: String | Xác nhận lại mật khẩu mới |
| int AppController::logout()   * Thông báo user đã đăng xuất. | |
| *String^ AppController::convertStringToHex(String^ input)*   * Hàm mã hóa thông tin | |
| input: String | Thông tin cần mã hóa |
| *int AppController::sendPrivateMessage(String^ \_ToUsername, String^ \_Message)*   * Gửi tin nhắn riêng tư với một user đang online | |
| \_ToUsername: String | User mà người dung muốn gửi tin nhắn tới |
| \_Message: String | Nội dung tin nhắn |
| *int AppController::requestSendFile(String^ \_ToUsername, String^ \_FileName, int \_iFileSize)*   * Gửi yêu cầu muốn gửi file trong tin nhắn | |
| \_ToUsername: String | User mà người dùng muốn gửi file tới |
| \_FileName: String | Tên của file muốn gửi đi |
| \_iFileSize: int | Kích thước của file muốn gửi |
| *int AppController::sendPrivateFile(String^ \_ToUsername, String^ \_FileName, String^ \_FilePath)*   * Thực hiện việc gửi file qua tin nhắn. | |
| \_ToUsername: String | User mà người dùng muốn gửi file tới |
| \_FileName: String | Tên của file muốn gửi đi |
| \_FilePath: String | Đường dẫn tới file muốn gửi |
| *int AppController::setInfor(String^ \_Username, String^ \_Birthdate)*   * Thiết lập thông tin người dung. | |
| \_Username: String | Tên đăng nhập của người dùng |
| \_Birthdate: String | Ngày tháng năm sinh của người dùng |
| *int ServerController::createSocket*   * Server tạo socket mới. | |
| *bool ServerController::checkLogin(String^ Username, String^ Password, String^& errorMsg)*   * Kiểm tra thông tin đăng nhập | |
| Username: String | Tên đăng nhập |
| Password: String | Mật khẩu đăng nhập |
| errorMsg: String | Lỗi (nếu có) trong quá trình kiểm tra |
| *bool ServerController::checkSignup(String^ Username, String^ Password, String^& errorMsg)*   * Kiểm tra các thông tin đăng ký có thỏa yêu cầu hay không. | |
| Username: String | Tên đăng nhập cần đăng ký |
| Password: String | Mật khẩu đăng nhập cần đăng ký |
| errorMsg: String | Lỗi (nếu có) trong quá trình kiểm tra đăng ký |
| *bool ServerController::login(String^ Username, String^ Password, Socket^ \_ClientSocket)*   * Thực hiện việc đăng nhập của client | |
| Username: String | Tên đăng nhập |
| Password: String | Mật khẩu đăng nhập |
| \_ClientSocket: Socket | Socket của client |
| *String^ ServerController::getUsernameBySocket(Socket^ \_socket)*   * Trả về username của client thông qua socket | |
| \_socket: Socket | Socket của client |
| *Socket^ ServerController::getSocketByUsername(String^ Username)*   * Trả về socket của username cần tìm | |
| Username: String | Tên người dùng cần tìm |
| *String^ Client::PublicFile::splitPathFileToReceiver(String^ pathFile)*   * Tách lấy loại tập tin được upload | |
| pathFile: String | Đường dẫn file |
| *String^ Client::PublicFile::splitNameFile(String^ fileName)*   * Tách lấy tên file được upload | |
| filename: String | Tên file |

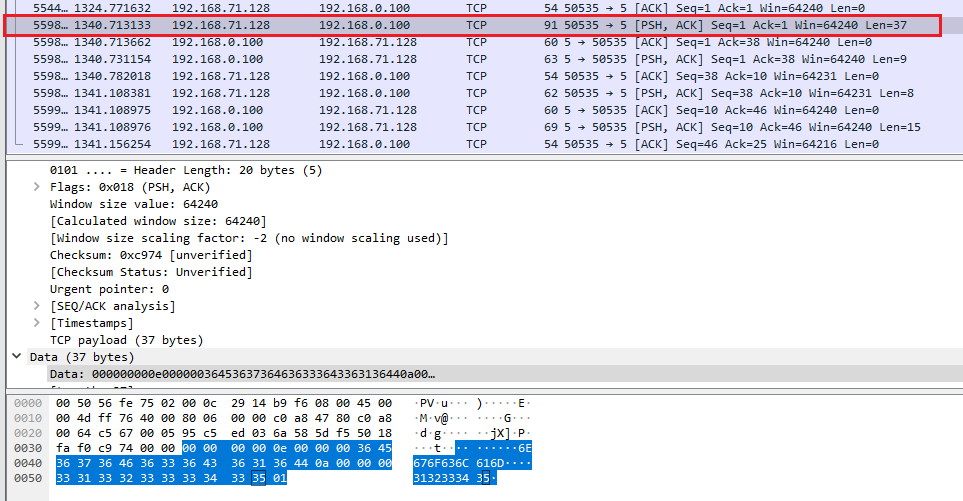
# Sử dụng Wireshark bắt gói tin

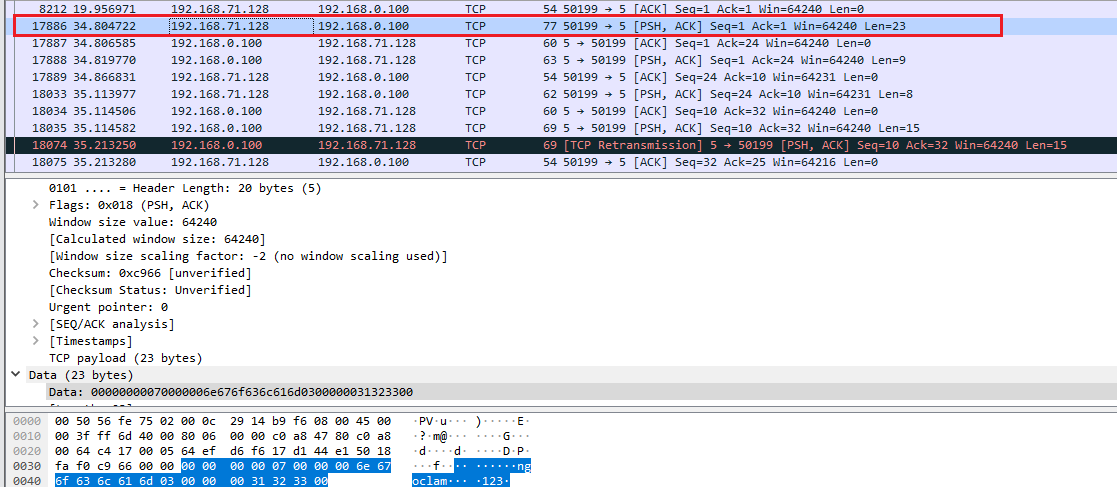
Server IP: 192.168.0.100

Client IP: 192.168.71.128

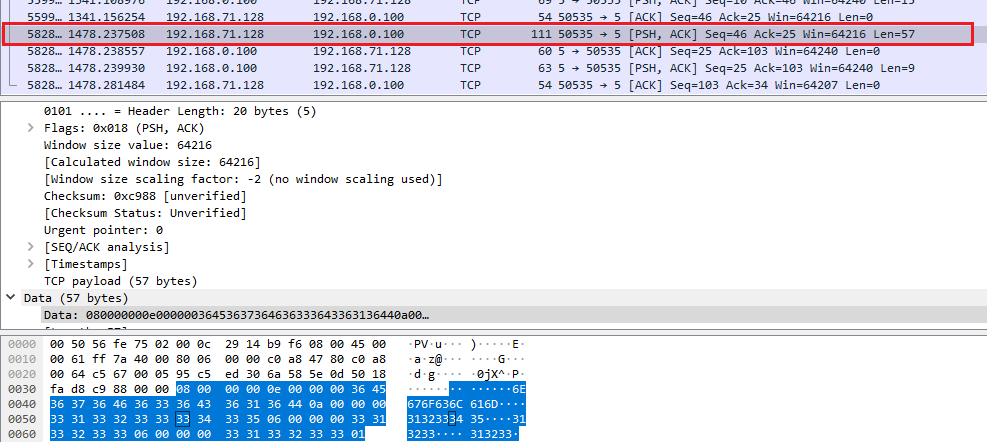
Port number: 5

1. **Chức năng login**

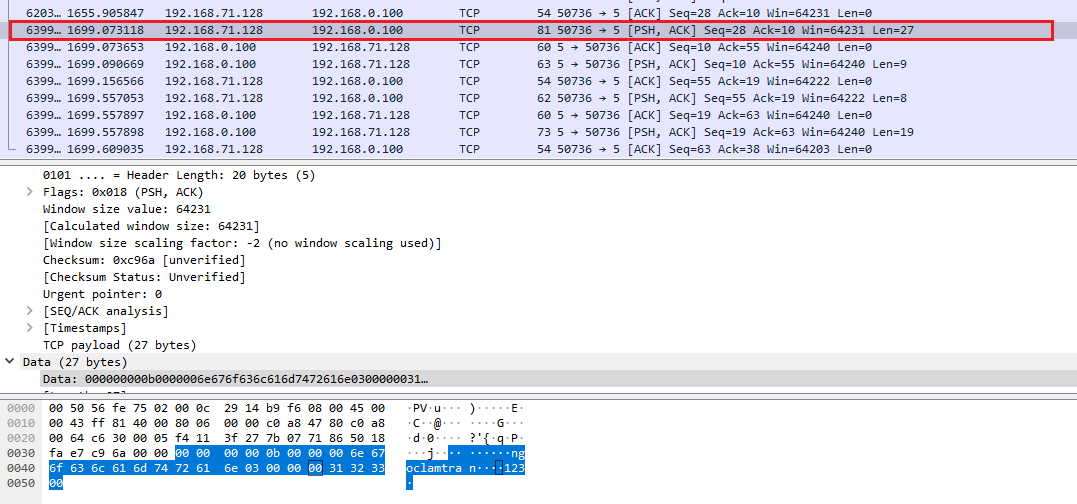
*(Mã hóa)*

*(Không mã hóa)*

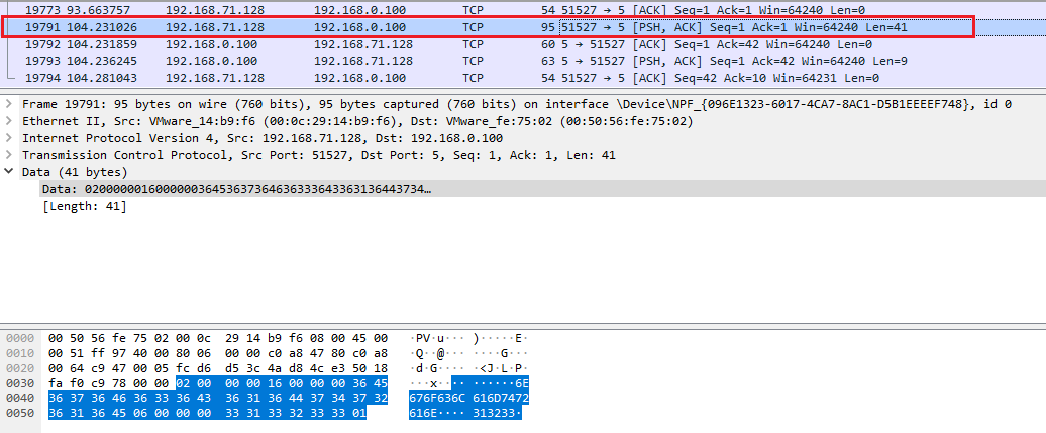
1. **Chức năng Signup**

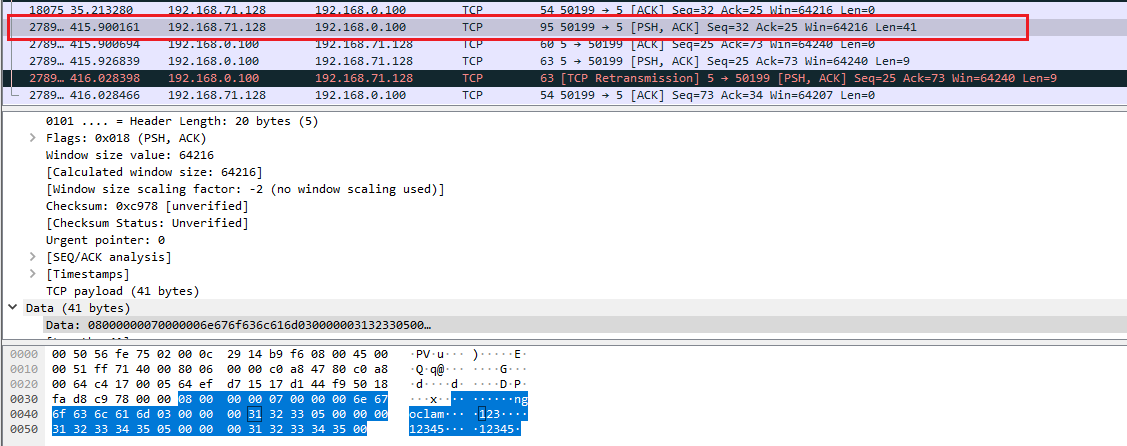
*(Mã hóa)*

*(Không mã hóa)*



1. **Chức năng Change Password**

*(Mã hóa)*

*(Không mã hóa)*

* **Kết luận:** Đối với dữ liệu đã qua mã hóa thì khi thực hiện bắt gói tin chúng ta không thể đọc được nội dung được gửi đi nhưng đối với dữ liệu chưa qua mã hóa thì gói tin được bắt có thể dễ dàng đọc được nội dung được gửi đi.

# Tài liệu tham khảo

https://www.codeproject.com/Questions/759680/How-do-I-serialize-a-bytearray-written-as-a-string

http://www.codeproject.com/Articles/12893/TCP-IP-Chat-Application-Using-C

https://www.youtube.com/watch?v=XC4OzdeB8gQ&ab\_channel=Nguy%E1%BB%85nV%C5%A9Ki%E1%BB%87t

https://www.youtube.com/watch?v=WrVavRc5wls&t=25s&ab\_channel=HiepMaiThanh