指挥与控制实验报告

|  |  |
| --- | --- |
| **组别** | **第六组** |
| **组名** | **BigChallenge** |
| **组员** | **刘凤生 4012018013** |
|  | **杨启升4012018001** |
|  | **孟乾威4012018015** |
|  | **刘国志4012018021** |
|  | **徐王忠4012018033** |
|  |  |
| **报告日期** | **2021年11月21日** |

# 实验要求

综合运用所学的网络、安全协议、密码等方面的知识，设计一个安全的网络通信系统:

* 1. 系统能够完成基本的字符型数据传输功能
  2. 设计私有的应用层通信协议用以进行信息交互
  3. 选取或设计合适的密码算法，用于保证数据 通信安全
  4. 设计合理的应用层安全通信协议，达到保护数据通信的效果
  5. 撰写实验报告，对所设计的系统进行详细描述

# 实验分工及完成任务量

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | | | | 实验分工 | | | |
|  | 组织讨论 | 参与讨论 | 代码搜集 | | 代码实现 | 程序检测 | 搭档组程序分析 |
| 刘凤生(组长) |  |  |  | |  |  |  |
| 杨启升 |  |  |  | |  |  |  |
| 孟乾威 |  |  |  | |  |  |  |
| 刘国志 |  |  |  | |  |  |  |
| 徐王忠 |  |  |  | |  |  |  |

# 实验准备

## 实验分工及讨论

### 实验分工

实验的具体分工在实验分工及完成任务量中已经展示。

### 实验讨论

实验分组确定后，小组内首先对要实现的网络通信系统的基本功能开展了广泛而热烈的讨论。基本确定了通信系统的模型，以及一些具体的通信数据类型。

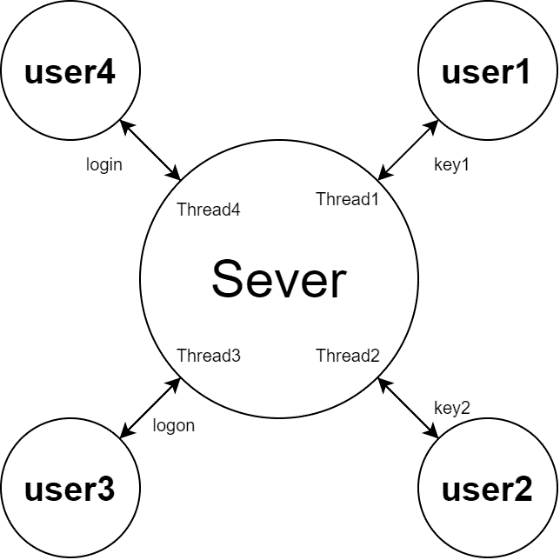
讨论确定了一些大家一致认为应该具有的功能：

1. 用户的登录
2. 用户的注册
3. 用户间的消息互相共享

针对这些功能，再结合我们所学的网络、安全协议、密码等方面的知识。大家对这三个功能的具体实现又有了进一步的讨论。最后形成了最终一些基本的收发传送数据思路：

1. 用户选择登录或是注册，并向服务器发送用户名，服务器进行相应的检测，并且反馈相应的检测结果。
2. 用户输入密码，进行登录操作，或者是先进性注册再进行登录操作。
3. 进入聊天室内，用户之间相互进行通信。

具体讨论的代码思路在附表1：



# 实验环境

|  |  |
| --- | --- |
| 操作系统 | Windows10 20H2 |
| 设备名称 | FengSHaG |
| 处理器 | Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 2.00 GHz |
| RAM | 8.00 GB (7.88 GB 可用) |
| 编译环境 | 9.2.0 (MinGW.org GCC Build-2) |

# 实验结果

|  |  |
| --- | --- |
| 服务器初态 |  |
| 进入客户端程序 |  |
| 用户名检测 |  |
| 注册 |  |
| 登录 |  |
| 消息发送 |  |
|  |  |
| 退出 |  |

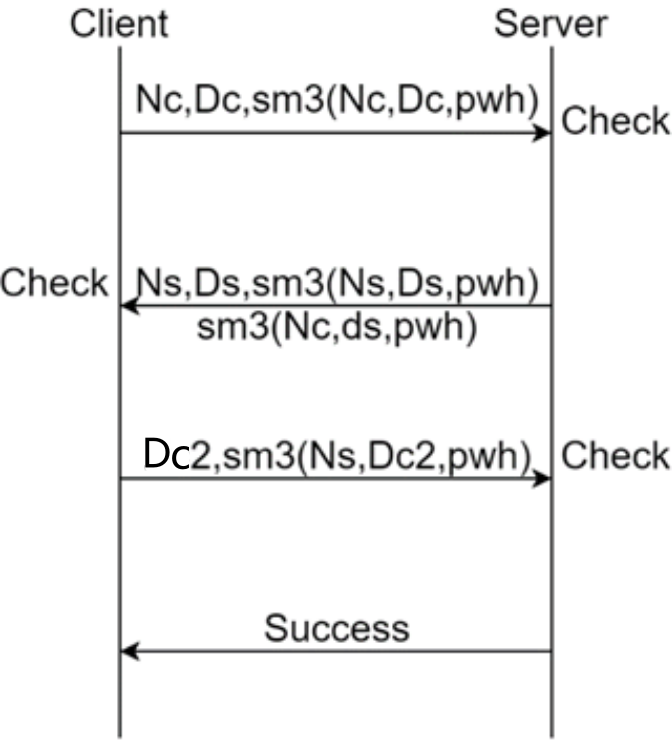
# 实验代码



# 安全性分析

## 身份认证

身份认证总结一下相当于是客户端与服务器的四次握手：



其中Nc，Dc，Dc2为客户端生成的随机数和两个时间戳，Ns，Ds为服务器生成的随机数和时间戳。

四次握手不仅保证了用户身份的正确，还保证了服务器的正确，中间也相当于双发互相发起了挑战，同时加入时间戳用来防止重放。

## 杂凑函数和加密函数的选择

### 杂凑函数

在这里杂凑函数选择的是SM3算法，同时四次握手的安全性也全基于SM3算法的安全性

### 加密函数

加密函数选择的时SM4算法，密钥是双方进行四次握手中sm3(nc,ns,dc,d s,pwh)输出的32长度字符串的奇数位。

## 用户注册的安全性

用户注册时，服务器向用户发送RSA的公钥部分，客户端用公钥加密。这里选择的N的比特数位2048位，安全性应该还可以。

## 安全性问题

忽略了这里是一个聊天室的情境。任意一个获得客户端软件的人，都可以通过注册方式注册到服务器，这里没有对注册用户的甄别。所以任何一个中间人，都可以通过精心构造的数据报实现任意用户的注册。可以通过在服务器加入一个管理员或者设置一个邀请码，对每个用户注册申请进行判断。

# 存在的问题

1. 在使用SM4的过程中，在对数据的加解密过程中存在明密文长度控制的问题，导致发送的消息是一些奇奇怪怪的字符串。这里在客户端对加密后密文的长度进行了判断，对不是16倍数长度密文的加密，视为时错误加密，直接选择不发送。但这并不能从根本上解决加解密存在的问题。
2. 同样，在用户登录的阶段。在偶然的情况下同样也会出现用户输入正确的口令，但是却返回登录错误，重新登录还是如此。但是，重新登录客户端可能就不再存在这个问题，但是这个时候可能就存在其他问题了。
3. 由于对编译的环境不太熟悉，在程序中的输出大多使用ascii码，没有输出中文，导致过程中的交互性较差。
4. 最后，就是在小组合作中存在着许多问题。在形成通信系统的结构的过程中，讨论的时间过久，导致后续推动整体代码形成的速度比较慢。

# 未实现的预计功能

1. 留言
2. 单独私聊

5 1 1 别人的留言 等用户登录时 发送

5 1 2 自己在线，服务器直接转发消息

6 1 0 state=0 用户名 进入下一级 留言

6 1 1 state=1 用户名 两人单独聊天（实际上时服务器转发消息）

# 收获与总结

1. 通过对简易的通信系统的设计与实现，提高了我们对网络中密码应用与实现的认识。让我们认识到，一个安全的通信系统不仅需要安全的密码算法，更需要安全的密码协议。
2. 同时，通过对通信系统的整体设计，也锻炼了我们的编程实现能力。主要是解决自己简陋程序的BUG的能力。

附表1：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 用户名的检测 0 | | | |
| 用户名的检测 0 | C:发送用户名0 | 发送登录用户名1 |  | 0 1 username |
|  | 发送注册用户名2 |  | 0 2 username |
| S:收到用户名0 | 登录check | 存在用户名 | 0 1 |
|  |  | 不存在 | 0 3 |
|  | 注册check | 已存在 | 0 4 |
|  |  | 不存在 | 1 RSA参数 |
|  | 双方进行身份登录认证（4轮） 1 1/(0验证是否要中断） | | | |
|  | 1 C生成随机数Nc | C读取当前时间dc | 生成字符串 | 1 1 namelen name Nc,dc,str(Nc,dc) |
| 2 S收到 C的1消息 | 对其进行判断 | Failure  成功 | Return 1 0 |
|  |  | S生成随机数Ns | 1 2 Ns,ds,str(Ns,ds) |
|  | S读取当前时ds | 生成字符串 | + str(Nc,ds) |
| 3 C收到 S的2消息 | 对两个str进行判断 | Failure  成功 | Return 1 0 |
|  |  | Dc2 | 1 3 dc2 str(Ns,dc2) |
| 4S收到C的3消息 | 字符串进行检测 | Failure  成功🡪 | 1 0  1 4 usernam state |
| 成功后服务器内用户state=1,双方密钥为str(nc,ns,dc,ds)中奇数位 | | | |
|  | 用户注册 2 1/0 | | | |
| 用户注册 2 1/0 | C | 请求注册 |  | 2 1 name |
| S | 返回RSA参数 |  | 2 2 1 N |
|  |  |  | 2 2 1 e |
| C | 用公钥加密passwh |  | 2 3 C(ph) |
| S | recive | 解密hash 判断长度 | 成功 2 2  失败 2 0 |
| C | 2 1 | State | 标志本人已注册1  注册完毕就到step2 |
|  | 通信(进入聊天室后,客户端与服务器之间的消息都是先加密再发送) | | | |
|  | C | 4 [name]:message | 加密 | 发送C\_message |
|  | S | 收到C\_message | 解密P\_message | 用对应线程用户的key  加密发送 C\_message |
|  |  |  | 每个在线的线程 |
| 客户端断开连接 | | | | |
|  | C |  |  | 7 |