

《物联网安全》

设计文档

组 长： 魏来 班号： 193182

组 员： 郭鸿绅、 段依琳、 刘畅海

院（系）： 计算机学院 专业： 网络工程

指导教师： 姚 宏 职称： 教授

2021 年 5 月

目录

[《物联网安全》 1](#_Toc19892)

[设计文档 1](#_Toc19800)

[1. 序言 1](#_Toc13305)

[1.1. 程序目的 1](#_Toc8697)

[1.2. 定义..................................... 2](#_Toc12546)

[2. 需求分析 3](#_Toc23742)

[2.1. 任务概述 3](#_Toc25394)

[2.2. 数据流图 4](#_Toc13897)

[2.2.1. 顶层数据流图 4](#_Toc16224)

[2.2.2. 一层数据流图 4](#_Toc15353)

[2.2.3. P1二层数据流图 5](#_Toc25026)

[2.2.4. P2二层数据流图 5](#_Toc13343)

[2.2.5. P3二层数据流图 6](#_Toc28305)

[2.3. 状态机设计 6](#_Toc29836)

[2.3.1. AS状态机 6](#_Toc9428)

[2.3.2. TGS状态机 7](#_Toc26309)

[2.3.3. Client状态机 7](#_Toc7237)

[2.3.4. Server状态机 9](#_Toc28344)

[2.4. UI布局设计 9](#_Toc12217)

[3. 系统概要设计 14](#_Toc20507)

[3.1. 系统概述 14](#_Toc24435)

[3.1.1功能需求规定： 14](#_Toc20457)

[3.1.2运行环境： 14](#_Toc4924)

[3.2. 系统结构 14](#_Toc31380)

[3.3. 模块设计 15](#_Toc8778)

*[4. 数据](#_Toc31966)* [15](#_Toc31966)

[4.1. 时序图 15](#_Toc30511)

[4.1.1. Kerberos认证时序图 16](#_Toc11393)

[4.1.2. 应用时序图 17](#_Toc6249)

[4.2. 数据报文 18](#_Toc1802)

[4.2.1. Kerberos数据报文 18](#_Toc25162)

[4.2.2. 应用数据报文 18](#_Toc24722)

[5. 详细设计 19](#_Toc25938)

[5.1. 错误码 19](#_Toc2362)

[5.2. 模块详细设计 19](#_Toc13366)

[5.3. 函数设计 19](#_Toc10137)

[5.4. 函数流程图 19](#_Toc1820)

[5.5. 甘特图 19](#_Toc14201)

# 序言

## 1.1. 程序目的

在学习了对称加密—DES，和非对称加密—RSA之后，目前，课程设计需要将上述两种工具，以及Kerberos的认证体系结合，并应用于自定义的网络应用中。

在近期的讨论中，我们小组认为本次课程设计的重点并不在于Kerberos认证过程，而是实现一套逻辑严密，结构完整，内容丰富的网络应用，并在应用的使用过程中体现出Kerberos认证过程以及两套加解密内容；经过小组讨论并结合了现有水平能力，时间限制后，我们小组从简单的网络游戏，模拟美团点餐系统，IM（即时通讯）中敲定了本次课程设计的大致方向，即设计一套能够满足一定量并发需求的C/S模式的通讯应用，通过端到端的通讯过程，体现Kerberos认证流程与RSA加密过程，在通讯内容中体现DES加密过程。

本次应用设计中，我们将尽可能将程序模块拆开掰碎，使得模块分工细致完整，同时降低函数间，模块间的耦合度，提高程序的可拓展性；同时在实现基础通讯功能外，我还构想了邀约游戏等功能，可以理解为一套新的程序，作为一个黑盒子添加到通讯程序的拓展模块，使得程序内容更加丰富。

魏来

2021年4月28日

## 1.2. 定义

|  |  |
| --- | --- |
| **专业术语** | **定义** |
| **Kerberos** | 计算机认证协议 |
| **数据库** | 用于存放数据，租借的阿里云MySQL数据库  rm-uf6t4cbyfz681x569.mysql.rds.aliyuncs.com 端口：3306 |
| **终端** | 4台设备模拟启动6个终端，即6个设备，分为KDC和User两种类别 |
| **KDC** | 密钥分配中心，由两个终端扮演，在本次程序中分为AS终端和TGS终端 |
| **AS** | 身份验证服务，为client生成TGT的服务 |
| **TGS** | 票务发放服务，为client生成某个服务的ticket |
| **TGT** | 用于获取ticket的票据 |
| **ticket** | 用于发送信息的凭证 |
| **User** | 实际运行应用程序的设备，又分为Client，Server |
| **Server** | 通讯服务的服务器设备 |
| **Client** | 需要通讯的用户的设备 |
| **K（）** | 密钥，用于加解密工具 |
| **聊天用户** | 使用本程序的用户 |
| **Session key** | 发送应用信息时，DES加密时用到的秘钥 |

表 1 名词定义

|  |  |
| --- | --- |
| **密钥表** | **意义** |
|  | User证书中的的公钥，用于发送对称钥和数字签名 |
|  | 对称钥，由AS生成，通过加密发送给User |
|  | TGS和另一个User之间的对称钥，与等价 |
|  | c和TGS之间的对称钥，用于发送session key |
|  | c和v之间的session key |
|  | AS和TGS之间的对称钥，写入文本。 |

表 2 密钥定义

# 需求分析

## 任务概述

本程序为基于 Kerberos 认证（DES加密）和 RSA加密的分布式通讯工具，需要实现下述功能：

工具：

1. 加密工具：使用DES与RSA两种加密方式，根据报文的不同需求，分别使用不同的加密方式。
2. RSA加密中的证书：约定好证书格式后，每个设备启动时，都会生成自己的证书，同时我们认为这个证书在系统内，独一无二，绝对真实，绝对安全，因此，先通过明文通讯，使得所有User设备与KDC设备间得到对方的公钥，即证书互信。同时可以认为系统内的Client与Server间都有对方的公钥，传输后续DES加密用到的session key。

Kerberos认证：

1. Kerberos认证体系：当所有设备启动后，其中两台设备扮演TGS和AS的角色，一台设备扮演通讯工具中的Server，其他设备扮演Client；所有Client和Server根据自己在AS中的账号登录验证身份，AS和TGS为每一个Client与Server间提供认证服务（由Client发起），Client与Client间没有必要提供认证服务。
2. 在Kerberos认证之前：由于Kerberos本身无RSA加解密需求，但为了将该功能引入Kerberos中，我们将最开始认证中出现的作为明文，根据最初已经获得的证书信息，使用RSA加密，通过通信功能传输。
3. 通过上述顺序为4）---> 3）--->2）的逻辑，实现了Kerberos认证功能后，认为除了出现新的User设备，整个系统内无需再使用Kerberos认证，AS和TGS设备监听新设备，不执行后续逻辑功能。

通讯工具：

1. 加好友：Client A向Server发送添加Client B为好友的请求，Server将此请求转发给Client B，Client B再将回复信息发送给Server，Server继续转发。
2. 加群聊：邀请加群，一个Client向多个Client发送邀请，方式同6）；申请入群，一个Client向拥有群的Client发送申请，方式同6）。
3. 删除好友、退出群聊：Client A向Server发送删除好友Client B的信息，Server更改二者数据库信息。
4. 聊天：Client A与Client B或多个Client的即时通讯，通过Server转发。
5. 搜索功能：提供Search界面，某个Client可以输入某些关键词，Server会根据内容在系统数据库中查找并将一定信息返回给Client。
6. 登录界面：每个用户输入账号密码，登录聊天程序
7. 注册，修改密码+拓展（忘记密码）：通过网页修改，Server修改Server数据库。
8. 聊天记录存储在本地log文件中。

## 数据流图

### 顶层数据流图



图 1 顶层数据流图

### 一层数据流图



图 2 一层数据流图

### P1二层数据流图



图 3 P1二层数据流图

### P2二层数据流图



图 4 P2二层数据流图

### P3二层数据流图



图 5 P3二层数据流图

## 状态机设计

### AS状态机



图 6 AS状态机

### TGS状态机



图 7 TGS状态机

### Client状态机



图 8 Client Kerberos认证过程状态机



图 9 Client 程序状态机

### Server状态机



图 10 Server认证状态机

## UI布局设计

登录界面

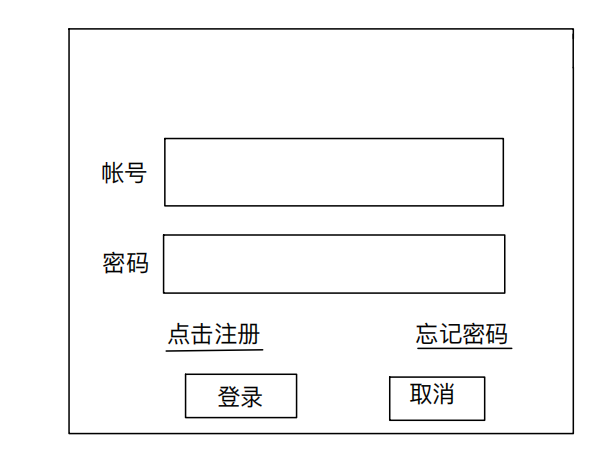


图 11 登录界面

1. 消息列表

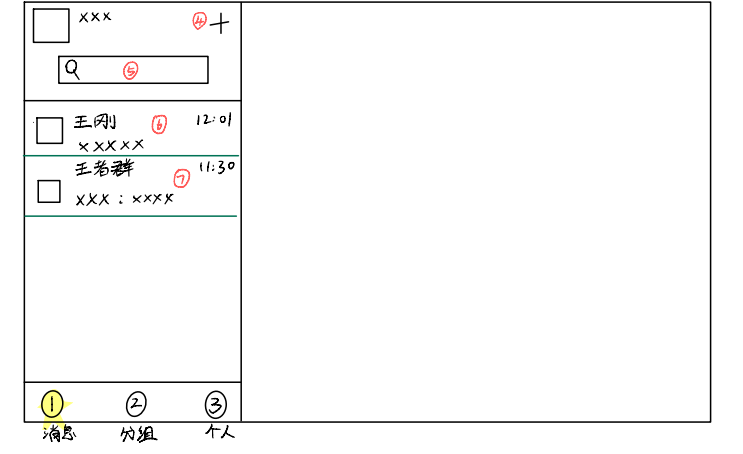


图 12 消息列表界面

1. 查看联系人界面

8与王刚聊天，右侧跳转至6，9跳转至7

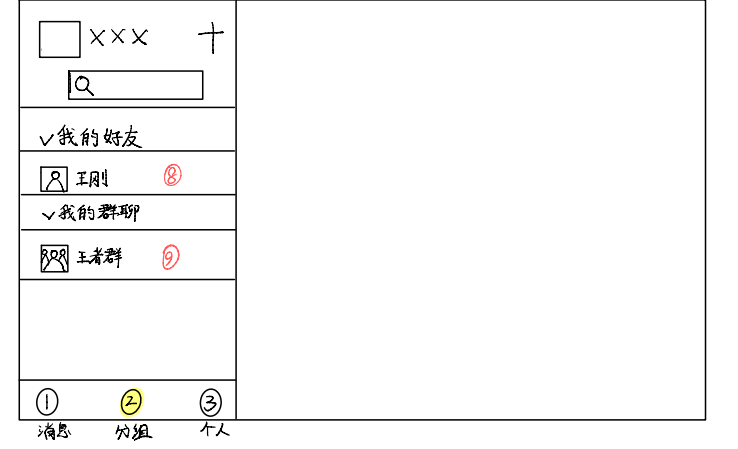


图 13 查看联系人界面

1. 查看个人信息界面

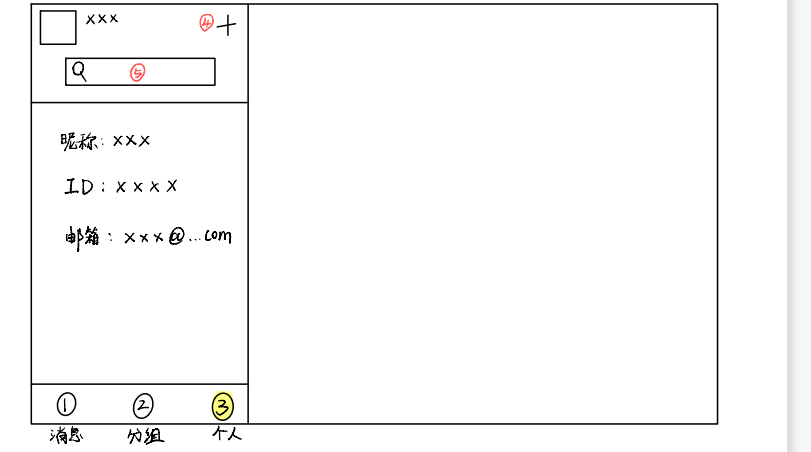


图 14 查看个人信息界面

1. 添加界面

点击+号，弹出功能

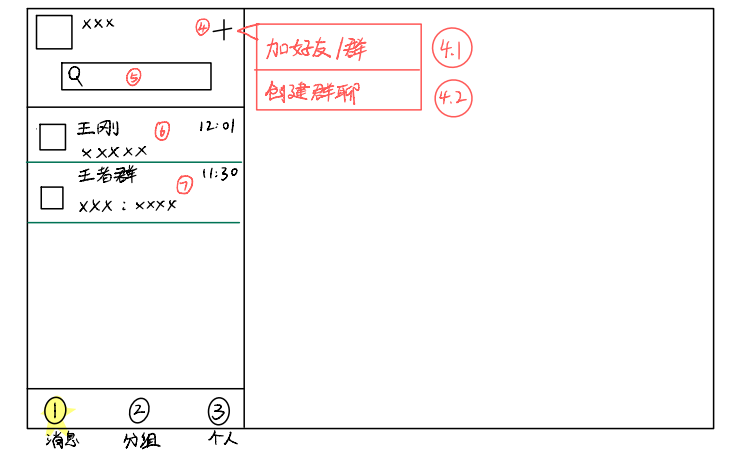


图 15 添加界面

* 1. 添加好友、群界面

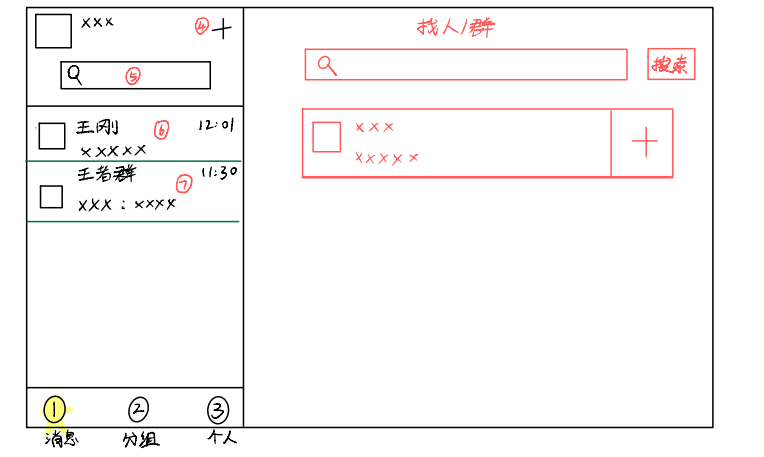


图 16 添加好友、群聊界面

* 1. 创建群聊界面

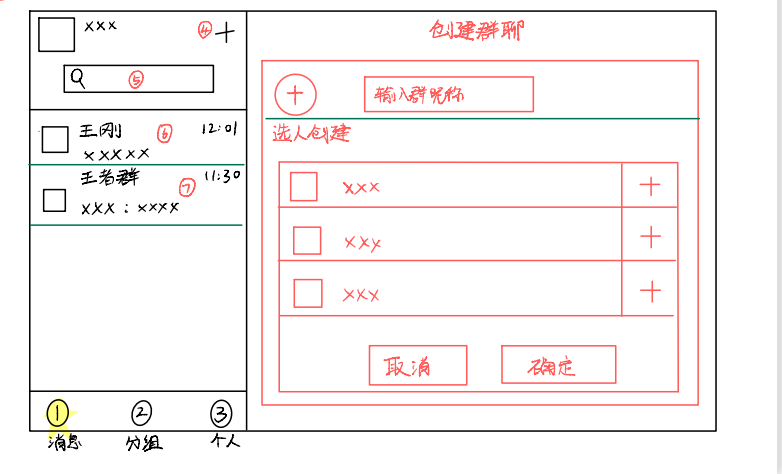
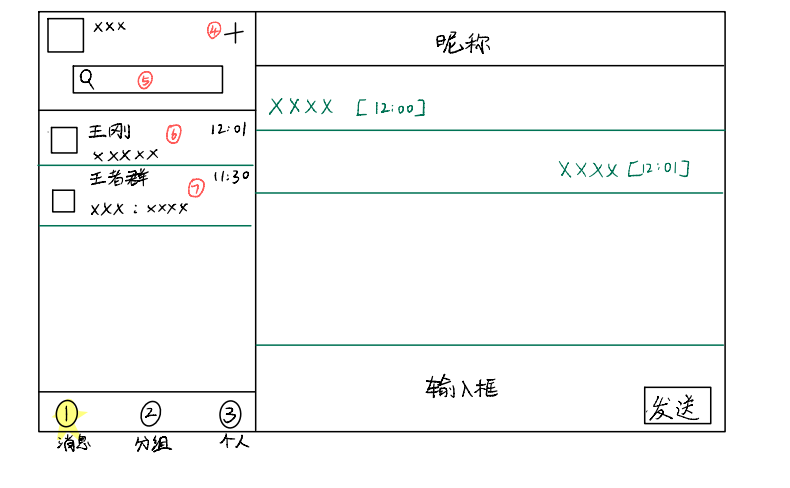


图 17 创建群聊界面

1. 查看信息界面



18 查看信息界面

1. 查看群信息界面

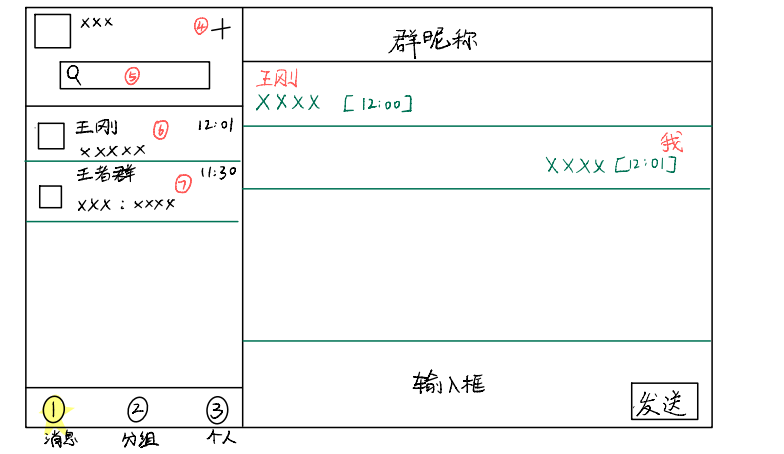


图 19 查看群信息界面

# 系统概要设计

## 系统概述

### 3.1.1功能需求规定：

AS系统负责向User端发放TGT；

TGS系统验证User的TGT，生成并发放密钥。

Client系统完成Kerberos系统认证：向AS请求TGT，凭TGT向TGS请求密钥，与Server系统完成双向验证；完成与Server端的交互，包括，聊天，添加好友、群，删除好友，退出群聊，登录，注册，查看个人信息。

Server系统完成Client系统的交互请求，转发信息。

### 3.1.2运行环境：

运行平台：Windows10（AS、TGS、Client），Linux（Server）

数据库：MySQL

Web：Node.js平台下的JavaScript

开发语言：C#、Java、JavaScript

## 系统结构



## 模块设计

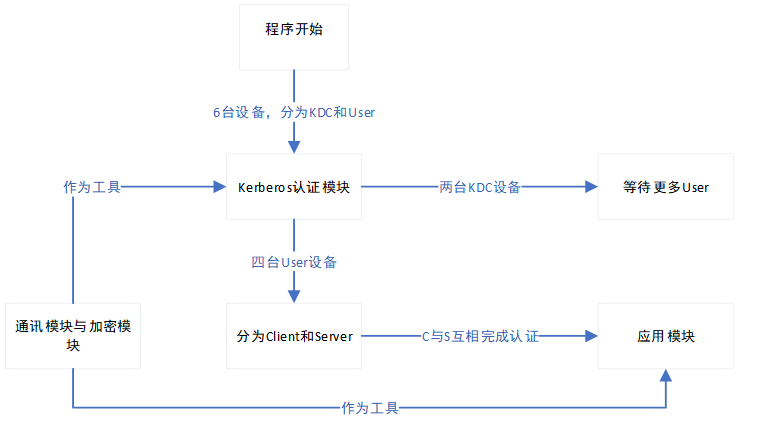
程序将从整体分为四大模块，如下所示：

1. 加解密模块：分别实现DES、RSA加解密功能，为其他模块调用提供接口
2. Kerberos认证模块：为User设备提供认证服务，使得User间都有各自间通信的session key。
3. 网络通讯模块：提供设备间通讯的框架
4. 应用模块：实现通讯协议，将User分为Client和Server设备，实现Client间的即时通讯功能。

以六台设备为例，当六台设备分别作为KDC和User启动时，KDC将会作为AS和TGS为User提供Kerberos的认证服务，认证完成后，作为KDC的两台设备不参与应用模块；认证后，每个独立的Client与Server间都存在一个Session key，为应用模块服务，应用模块实现即时通讯的逻辑功能。

同时通讯模块与加密模块作为工具模块，为Kerberos认证和应用提供接口，实现数据的加解密，以及用户间数据的发送。

模块功能图如下所示：



# *数据*

## 时序图

### 证书时序图



图 20 证书时序图

### Kerberos认证时序图



图 21 Kerberos认证时序图

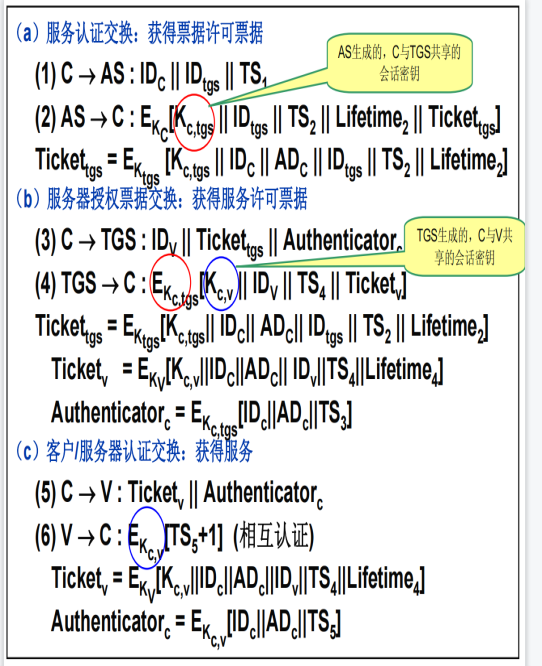


图 22 Kerberos时序图报文

### 

### 应用时序图



图 23 用户应用时序图

## 数据报文

Json是Java和C#之间通信的利器，Java端将Java对象转变为Json串后发出，C#端接收到Json串后转换为C#对象；C#发出转变为Json串的对象，Java收到后解析成Java对象，Json串在不同语言之间起到一个桥梁的作用。对定义的Java或C#对象生成Json字串，以及从Json字串生成Java或C#对象，有很方便的方法。

程序通过socket使用Json发送数据报文。

**4.2.1. K协议**

K协议应用于Client、AS、TGS和Server之间的通讯，只要是为了完成Kerberos认证和证书的互传。工作在应用层，传输层使用TCP协议。

一、K协议的报文固定格式及相关说明：

|  |  |  |
| --- | --- | --- |
|  | 报文类型 | 报文内容 |
| 简称 | type(T) | content(C) |
| 长度 | 2 Byte | … |
| 内容 | 0x00xx | … |

二、报文介绍

1、Server和Client依次启动并向AS发送证书，AS将证书添加到DB。

|  |  |
| --- | --- |
| 0x0001 | 证书 |

Server和Client各自生成自己的证书，证书格式一致。证书样例如下：

|  |  |
| --- | --- |
| **证书** | |
| 账号 | 10位数字 |
| 公钥 | …… |
| 有效期 | 时间戳 |

2、Client向AS请求票据

|  |  |
| --- | --- |
| 0x0002 |  |

3、AS用Client的公钥实现加密回复

|  |  |
| --- | --- |
| 0x0003 |  |

4、Client向TGS出示票据

|  |  |
| --- | --- |
| 0x0004 |  |

5、TGS发送会话秘钥

|  |  |
| --- | --- |
| 0x0005 |  |

6、Client向Server发送认证

|  |  |
| --- | --- |
| 0x0006 |  |

7、Server回复认证

|  |  |
| --- | --- |
| 0x0007 |  |

8、Client向Server发送证书

|  |  |
| --- | --- |
| 0x0008 | 证书 |

9、Server向Client发送证书

|  |  |
| --- | --- |
| 0x0009 | 证书 |

**4.2.2. EC协议**

EC协议应用于EasyChat的Server和Client之间的通讯，工作在应用层，传输层使用TCP协议。Web-Server和Browser之间的通讯使用HTTPS（一种应用层传输协议），并借用EC协议中的注册功能并额外实现了找回密码的功能，这一部分的协议在WEC中介绍。

一、Server-Client的应用层通信协议（EC协议）的报文固定格式及相关说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 报文类型 | 源地址 | 目的地址 | 报文内容 |
| 简称 | type(T) | src(S) | dst(D) | content(C) |
| 长度 | 2 Byte | 10 Byte | 10 Byte | … |
| 内容 | 0x01xx | 服务号 | 服务号 | 内容 |

说明：

1. 报文类型：标识这个报文的属于哪个业务流程（如：登录、注册或请求加好友等等），详见下文。
2. 源地址和目的地址都是服务号（也即：ID）。用户的ID（uid）为非零数字开头的10位数字（如：1084421957）；Server的ID（sid）为00000 00000；Web-Server的ID（wid）为00000 00001。
3. 报文内容：即正文。

二、报文介绍

下文将Client成为C；Server称为S；C—>S：客户端发送至服务器；S—>C：服务器发送至客户端。

1. 注册功能

（1）C发送注册消息给S

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C—>S | 0x0101 | S | D | 昵称;头像;psswd;个签;邮箱 |

说明： S：任意；D：sid。

（2）S处理注册信息后将结果回复给C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S—>C | 0x0101 | S | D | 结果位;新uid |

说明：S：sid；D：任意。

1. 登录功能

（1）C发送登录消息给S

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C—>S | 0x0102 | S | D | psswd |

说明： S：uid；D：sid。

（2）S处理登录消息后将结果发回C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S—>C | 0x0102 | S | D | 结果;好友个数;若干个{uid;头像;昵称;在线位}; |
| 群聊个数;若干个{gid;头像;群聊名称;群人数} | | | | |

说明：S：sid；D：uid。

1. 查看某人的个人信息功能

（1）C0发送查看C1信息请求给S

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C0—>S | 0x0103 | S | D | uid1 |

说明： S：uid0；D：sid。

（2）S处理C0的请求并回复结果给C0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S—>C0 | 0x0103 | S | D | 结果;{uid1;昵称;头像;个签;邮箱} |

说明：S：sid；D：uid0；结果：成功为1，失败为0。

4、请求加好友功能

（1）C0发送请求加好友消息给S；S收到后直接转发至C1

|  |  |  |  |
| --- | --- | --- | --- |
| C0—>S—>C1 | 0x0104 | S | D |

说明：C0—>S—>C1：S收到后直接转发；S：uid0；D：uid1。

（2）C1处理请求并将结果返回给S；S根据结果更新DB；S将此报文回复给C0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C1—>S—>C0 | 0x0104 | S | D | 结果 |

说明：S：uid1；D：uid0；结果：成功为1，失败为0。

5、删除好友功能

（1）C0向C1发送删除好友通知

|  |  |  |  |
| --- | --- | --- | --- |
| C0—>S | 0x0105 | S | D |

约定： S：uid0；D：uid1。

（2）S处理删除请求；并将结果发送至C0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S—>C0 | 0x0105 | S | D | 结果+ uid1 |

约定：S：sid；D：uid0；结果：成功为1，失败为0。

（3）若（2）处理成功后，S向C1发送删除通知

|  |  |  |  |
| --- | --- | --- | --- |
| S—>C1 | 0x0105 | S | D |

约定： S：uid0；D：uid1。

6、创建群聊

（1）C向S发送创建群聊请求

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C—>S | 0x0106 | S | D | {群聊名+群头像} |

过程：C发送报文给S；S收到。

约定：S：uid；D：sid。

（2）S向C回复创建结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C—>S | 0x0106 | S | D | 结果;{gid+群聊名+群头像+群主} |

过程：S处理请求；将处理结果返回给C。

约定：S：sid；D：uid。

7、邀请好友加入群聊

（1）C0向C1发送邀请信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C0—>S—>C1 | 0x0107 | S | D | gid |

过程：C0向S发送邀请消息；S转发至C1。

约定：S：uid0；D：uid1。

（2）C1答复C0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C1—>S—>C0 | 0x0107 | S | D | 结果;gid |

过程：C1处理邀请信息后将回复发送至S；S根据结果更新DB；S将本条消息转发至C0。

约定：S：uid1；D：uid0；结果位：成功=1，失败=0。

8、获取某个群的信息

（1）C向S发送获取群聊信息的请求

|  |  |  |  |
| --- | --- | --- | --- |
| C—>S | 0x0108 | S | D |

过程：C向S发送请求群聊消息的报文；S收到。

说明：S：uid；D：gid。

（2）S答复C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S—>C | 0x0108 | S | D | 结果;{gid+群聊名+群头像+群主} |

过程：S处理请求；将群聊信息返回给C。

约定：S：gid；D：uid；结果：成功=1，失败=0。

9、退出群聊

（1）C向S发送邀请信息

|  |  |  |  |
| --- | --- | --- | --- |
| C—>S | 0x0109 | S | D |

过程：C向S发送退出群聊消息；S收到C的退群请求。

说明：S：uid；D：gid。

（2）S回复处理结果给所有成员或者C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S—>C | 0x0109 | S | D | 结果 |

过程：S处理退群请求；根据处理结果更新DB；若C退群成功则将结果返回给所有群成员，否则只返回给C。

约定：S：gid；D：uid（接收方的uid）；结果：成功=1，失败=0。

10、移出群聊

（1）C0向S发送消息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C0—>S | 0x010A | S | D | uid1 |

过程：C0向S发送将C1移出群聊的消息；S收到。

说明：S：uid0；D：gid。

（2）S回复处理结果给所有成员或者C0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S—>C0/C | 0x010A | S | D | 结果;uid1 |

过程：S处理移出群聊请求；根据处理结果更新DB；若移出成功则将结果返回给所有群成员，否则只返回给C0。

约定：S：gid；D：uid（接收方的uid）；结果：成功=1，失败=0。

11、发送消息给某人

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C0—>S—>C1 | 0x010B | S | D | 消息 |

过程：C0发送报文给S；S收到；S转发至C1。

约定：S：uid0；D：uid1。

12、发送消息给群聊

（1）C向S发送消息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C—>S | 0x010C | S | D | 消息 |

过程：C向S发送群消息；S收到。

说明：S：uid；D：gid。

（2）S将消息转发给所有的成员（包括C，这作为消息发送成功的确认）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S—>C | 0x010C | S | D | uid;消息 |

过程：S将消息转发至所有群聊成员。

约定：S：gid；D：uid。

**4.2.3. WEC协议**

WEC协议应用于EasyChat的Web-Server和Browser之间的通讯，消息使用HTTPS协议进行负载。报文介绍（下文将Browser称为B；Web-Server称为W）：

1、注册

（1）B发送POST（/getvcode）请求：

|  |
| --- |
| {邮箱} |

（2）W发送验证码（6位随机数）到指定的邮箱，并响应B：

|  |
| --- |
| {已发送or发送失败} |

（3）B发送POST（/register）请求：

|  |
| --- |
| {昵称;头像;psswd;个签;邮箱;验证码} |

（4）W处理并响应B：

|  |
| --- |
| {结果;账号;昵称;头像;psswd;个签;邮箱} |

2、重置密码

（1）B发送POST（/getvcode）请求：

|  |
| --- |
| {账号} |

（2）W向B的邮箱发送验证码（6位随机数）并响应B：

|  |
| --- |
| {邮箱（部分遮盖）;发送} |

（3）B发送POST（/resetpsswd）请求

|  |
| --- |
| {账号;邮箱;密码;验证码} |

（4）W响应B

|  |
| --- |
| {结果} |

# 详细设计

## 详细流程图



图 24 AS流程图



图 25 TGS流程图



图 26 Server流程图



图 27 Client流程图

## 数据结构

报文数据结构：

程序内部数据结构：

## 模块详细设计

1. Client模块

2）AS、TGS模块

3）Server模块

4）工具模块（RSA、DES、数字签名）

5）通讯模块

接口：My\_send(string a,stirng b,string c)

A,为ip，b为port，c为报文

## 函数设计

## 函数流程图

## 甘特图

Ex：具体函数代码，传入传出，有待商榷