《网络安全》
报告

组　　员：孔浩 胡洋 张毛毛　班号：193152
学　　号：20151001606　组长：许晓晖
院（系）：计算机学院　专业：网络工程
指导教师：姚 宏　职称：副教授

2018 年 4 月

独立工作成果声明

本人声明所呈交的《网络安全》报告，是我个人在导师指导下进行的程序编制工作及取得的成果。

尽我所知，除文中已经标明的引用内容，和已经标明的他人工作外，本报告未包含任何抄袭自他人的工作成果。对本报告的工作做出贡献的个人，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

报告作者签名：

日期：　　年　　月

目录

第一章　系统及开发环境概述

1.1.1 系统

系统为基于 KERBEROS 的多人聊天室系统。

1.1.2 环境

（1）环境系统

Window 7 系统版本及以上

Java jdk 1.8.0 版本及以上

（2）运行平台

Eclipse Java EE IDE for Web Developers.

Version: Oxygen.2 Release (4.7.2)

Build id: 20171218-0600

（3）开发语言：

JAVA

（4）硬件配置

个人笔记本电脑 4 台

（5）源代码管理工具

GitHub for Windows 1.0

（6）数据库

MYSQL 数据库。

第二章　UI 设计

§2.1 引言

本模块属于客户端模块，因比较重要，单独叙述。

本小组应用程序目前设定为多人聊天室，拟定主要功能如下：用户可以注册，用户登陆后可以实时与聊天室内所有成员聊天。UI 分为 3 个部分，分别为用户登陆界面、注册界面、用户聊天界面。具体加解密流程在控制台显示，后续功能在做完基本的实现后进行实现。

2.1.1 注册界面

因 Kerberos 系统的前提是，需要客户端的秘钥以一种安全的方式交给 AS 认证服务器，所以，我们需要注册界面，注册后台使用 RSA 算法。

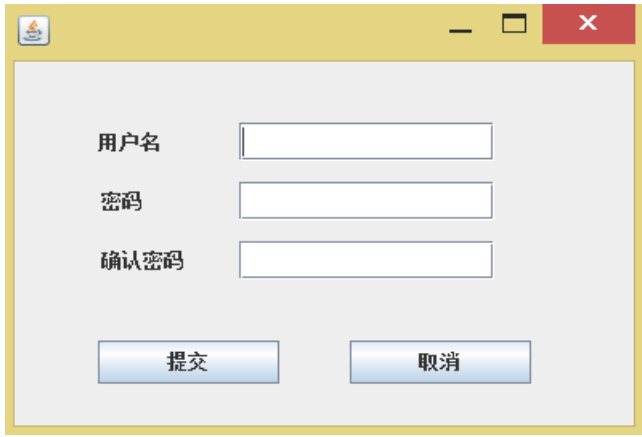


图 二-1 注册界面

2.1.2 登陆界面

用户需要被验证，所以设立登陆界面。需要做如下事情：

1、 输入用户名
2、 输入密码
3、 输入验证码

设计界面如下：

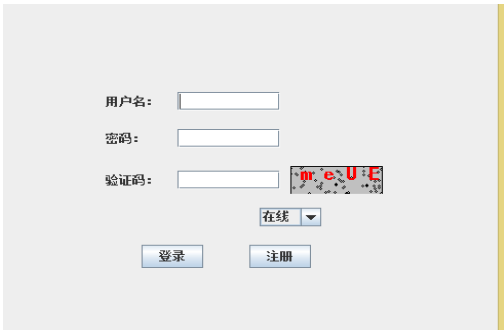


图 二-2 登陆界面

按钮解释——注册

AS 期待接收 1 号包。C 期待返回 2 号包。（包的格式已在下面章节定义好）。

当用户输入用户名、密码以及重复密码后，点击注册按钮。将用户名和密码从 UI 框中提取为 String 类型，使用 RSA 加密算法，用 AS 公钥加密用户名和密码，包头标记为 1 号（包头不加密），发送给 AS。

AS 获取 1 号包后，按照 1 号数据包数据格式，使用自己私钥获取明文，然后将用户名在数据库中进行比对，如果已经存在此用户，返回 2 号数据包，包内信息为 NO1，表示已存在用户注册失败，提示用户重新注册。

如果 AS 存入数据库或者其他问题，返回保内信息 NO2，表示注册失败，提示重新注册。

如果数据库中存入成功，则返回 YES。

所有的返回 2 号数据包均用客户端的公钥加密，包头标记 2 号不加密。

客户端接收 2 号包，使用自己私钥解密，比对信息，查看是否注册成功，否则提示用户重新点击登录按钮聊天室界面。

按钮解释——登陆

用户点击登录按钮后，根据 KERBEROS 处理信息流程，依次发出请求信息。

第一步，按钮点击后，客户端发出 3 号包，发往 AS 请求访问 TGS，如果通过，则 AS 发回 4 号包。客户端发出 5 号包，请求访问服务器 V，TGS 如果通过，则 TGS 发回 6 号包。客户端发出 7 号包，请求服务，V 如果通过，在 V 发回 8 号包，保内标记信息，如果请求成功，则返回信息为 YES，代表登陆成功。否则返回信息 NO，代表登陆失败。登陆成功后弹出聊天室界面。

选择框解释——在线

如果用户在选择框选择在线，则用户登陆成功后，发送 11 号数据包，提示 V 客户端在线状态。数据包广播至所有在线客户端，更改在线状态。

直接弹出聊天室界面，直接参与聊天室的聊天。后期添加的私聊功能，则用户登陆后提示其他在线的客户端此用户已登陆。

选择框解释——隐身

如果用户在选择框选择隐身，则用户在登陆成功后，发送 11 号数据包，提示 V 客户

端在线状态。

不会进入聊天室和收到聊天室信息，只在 UI 中显示当前在线的用户名称。后期加入私聊功能，登陆后，不提示其他人新的用户登陆。

选择框解释——离开

如果用户在选择框选择离开，则用户在登陆成功后，发送 11 号数据包，提示 V 客户端在线状态。

不会进入聊天室和收到聊天信息，不会显示当前在线的用户名称。不能进行所有聊天功能。

2.1.3 聊天室

提供多人实时聊天功能。需要做如下事情：

1、 用户输入内容区域。

2、 聊天信息显示。

图 二-3 聊天室界面

预定私聊功能如图，私聊功能后续实现，本文档暂不做设计。

按钮解释——发送

用户弹出本聊天室界面后，在输入框内输入要发送的信息，点击发送，则会将信息发送至聊天室。发送 9 号数据包至服务器，服务器广播至所有客户端。

其他客户端收到数据包 DES 解密，则聊天框内容显示刚才的聊天信息。

每个客户端重复以上操作。

按钮解释——确认

向服务器发送 11 号数据包，更改客户端在线状态。

服务器向所有用户广播此数据包，更改此客户端状态。

2.1.4 加解密显示

在注册和登陆时，在 eclipse 中的控制台输出加密和解密信息。

在聊天室中，在 UI 中设计一个文本框呈现对应收到或发送的数据包的加解密状态以及采用的加解密算法，另外还有对应的认证成功失败的标志或弹窗显示在 UI。

第三章　状态机

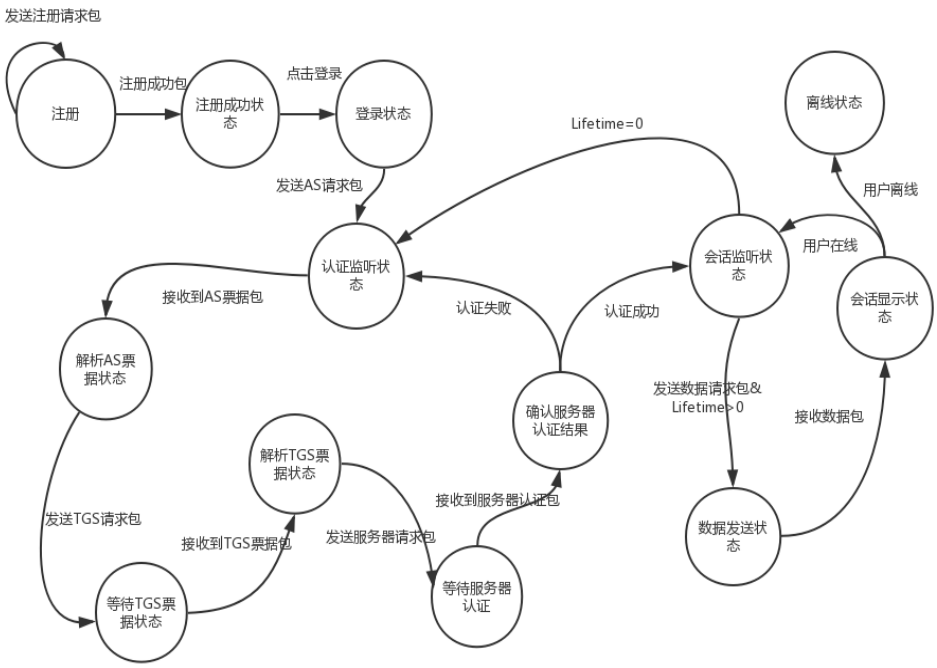


图 三-1 客户端状态机

对于多个 Client，每个 Client 都是上面的状态，在获得服务器 V 之后，进入聊天室，服务器将每个 Client 信息互相转发广播。当某一个用户离线，则 socket 断开连接，其也就收不到消息。但是要注意线程冲突。

首先对于单个 Client 来说，用户点击注册按钮进入注册状态，此时会 Client 会向服务器发送一个注册请求包，如果收到服务器的注册返回包成功，则说明 Client 注册成功。

如果用户点击登录，则 Client 进入登录状态，此时 Client 会向 AS 发送一个认证请求，Client 进入认证监听状态，如果接收到 AS 发来的票据，则 Client 进入 AS 票据解析状态，然后向 TGS 发送票据请求包，进入等待 TGS 票据状态，如果接收到 TGS 的票据包，则 Client 进入 TGS 票据解析状态，解析成功以后会发送一个服务器认证请求，进入等待服务器认证状态，然后如果接收到服务器认证包，则 Client 进入认证确认状态，如果认证成功，Client 进入会话监听状态，如果认证失败则返回认证监听状态。在会话监听状态如果用户发送数据则进入数据发送状态，如果用户收到服务器发来的数据包则进入会话显

示状态显示服务器消息，如果用户在线则继续进入会话监听状态，如果用户离线，则进入离线状态。在会话监听状态，如果 Client 的票据的生存周期为 0，则需要重新返回到认证监听状态，进行 AS 重放，获取新一轮的会话密钥。

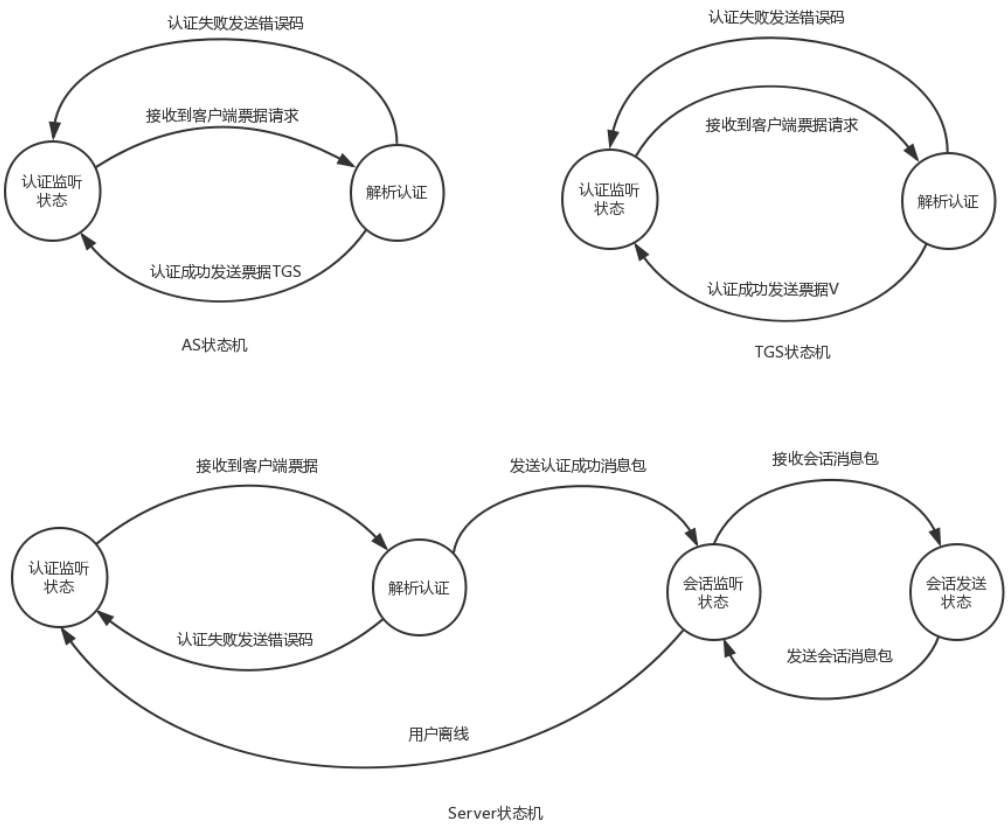


图 三-2 AS/TGS/V 状态机

对于 AS 来说，状态比较简单，一开始处于认证监听状态，等待客户端认证请求，当接收到客户端票据请求报文的时候进入解析认证状态，如果认证成功，则向客户端发送 TGS 票据，如果认证失败，则向客户端发送认证错误码。

对于 TGS 来说，与 AS 的状态类似，一开始处于认证监听状态，等当接收到客户端票据请求报文的时候进入解析认证状态，如果认证成功，则向客户端发送 V 票据，如果认证失败，则向客户端发送认证错误码。

对于服务器 V 来说，首先处于认证监听状态，如果收到客户端的票据则进入解析认证状态。如果认证失败，则向客户端发送认证错误码。如果认证成功，则向客户端发送认证成功报文进入会话监听状态，如果收到客户端数据包，则进入会话发送状态将数据广播给其他在线客户端，然后再次进入会话监听状态。

第四章　模块设计

§4.1 引言

各个模块输出与输入必须按照模块要求，只要输出输入正确，系统集成才会没有问题。

4.1.1 本小组模块分为：

1、 RSA、DES 加密算法。
2、 登陆与注册 UI 设计。
3、 多人聊天室 UI 设计。
4、 Kerberos Cllient 端设计。
5、 Kerberos AS 端设计。
6、 Kerberos TGS 设计。
7、 Kerberos V 设计。
8、 聊天室服务器端设计。
9、 聊天室客户端设计。

4.1.2 人员分工情况如下：

多人聊天室服务器许晓晖。
多人聊天室客户端孔浩。
Client 端——胡洋
AS 端——孔浩
TGS——张毛毛
V——许晓晖
RSA 孔浩
DES 许晓晖
UI 登陆与注册 UI 许晓晖
多人聊天室界面 UI 孔浩

§4.2 子模块——Client 端设计

CLient 要实现功能为：请求访问 as、tgs、v，实现聊天功能。

4.2.1 Client 之间的关系

当用户采用私聊的方式时，多个 Client 之间是点对点的关系，即 client 端不通过服务器进行数据传递，两个 Client 之间进行点对点之间的数据传输。两个 Client 之间进行证书的交换之后，确定采用 RSA 加密算法获取会话秘钥，然后根据会话秘钥采用 DES 对称加密进行数据传输。

当用户采用群聊的方式，多个 Client 之间没有关系，统一由服务器来进行数据包的中转与控制，每个 Client 与服务器构成 C/S 架构，服务器将用户发来的包进行广播，用户接受服务器发来的数据包并显示在本地聊天室中。

4.2.2 实现功能

1、 正确发送 C 请求 AS 的控制包。
2、 正确解开 AS 的控制包。
3、 正确解开 TGS 控制包。
4、 正确发送 C 请求 TGS 控制包。
5、 正确发送 C 请求 v 的控制包。
6、 正确解开 v 的控制包。
7、正确发送 c 发往 v 的数据包。
8、正确解开 v 的数据包。
9、比较是否 C 信息是否正确。
10、根据约定格式封装和加密数据包。
11、Socket 发出至客户端。

4.2.3 模块输入

长度为 9 的 String 的 8 号控制包。
长度为 112 的 String 的 4 号控制包。
长度为 108 的 String 的 6 号控制包。
长度为 200 的 String 的 9 号数据包。
长度为 9 的 String 的 2 号控制包。

4.2.4 模块输出

长度为 45 的 String 的 1 号控制包。
长度为 109 的 String 的 5 号控制包。

长度为 107 的 String 的 7 号控制包。

长度为 31 的 String 的 3 号控制包。

长度为 200 的 String 的 9 号数据包。

4.2.5 代码命名规范

1、Client 主体函数。

Client_(功能对应英文)；

2、socket 发送接收。

Client_socket_(功能对应英文);

4.2.6 发送 1 号控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 1。

2、要求接受到的字符串 DES 解密。

3、前 6 位 char 经转换为 int 类型 1，代表控制包种类，否则结束本模块。

4、7-26 位 char 经转换为 String 类型，长度必须小于等于 20，代表用户名，否则结束本模块。

5、27-29 位 char 经转换为 int 类型，且大于等于 0 小于 8，标记客户端代号，否则结束本模块。

6、30-45 位 char 经转换为 String 类型，代表用户秘钥，必须小于 16 位，且字符串内无空格，否则结束本模块。

7、必须共 45 位，否则结束本模块。

4.2.7 接收 2 号控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 2。

2、要求接收到的字符串 DES 解密。

3、前 6 位 char 经转换为 int 类型 2，代表控制包种类，查看是否是其他控制包，否则结束本模块。

4、7-9 位 char 经转换为 string 类型，且为 YES 和 NO，标记是否注册成功，否则结束本模块。

5、必须共 9 位，否则结束本模块。

4.2.8 发送 3 号控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 3。

2、要求接受到的字符串 DES 解密。

3、前 6 位 char 经转换为 int 类型 3，代表控制包种类，否则结束本模块。

4、7-9 位 char 经转换为 int 类型，且大于等于 0 小于 8，标记客户端代号，否则结束本模块。

5、10-11 位 char 经转换为 int 类型，且大于等于 0 小于 4，标记访问 tgs 标号，否则结束本模块。

6、11-31 位 char 经转换为 String 类型，代表系统时间，格式必须为年月日时分秒，且字符串内无空格，样例：2018 年 4 月 28 日 8:35:42，否则结束本模块。

7、必须共 31 位，否则结束本模块。

4.2.9 接收 4 号控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 4。

2、要求接收到的字符串 DES 解密。

3、前 6 位 char 经转换为 int 类型 4，代表控制包种类，查看是否是其他控制包，否则结束本模块。

4、7-9 位 char 经转换为 int 类型，且大于 0 小于等于 8，标记是哪个客户端，如果不是本机，否则结束本模块。

5、10-25 位 char 经转换为 String 类型，且必须 16 位，代表 K（c，tgs）秘钥。

6、26-27 位 char 经转换为 int 类型，且大于 0 小于等于 4，否则结束本模块。

7、28-47 位 char 经转换为 Stirng 类型，代表系统时间。

8、48-53 位 char 经转换为 String 类型，代表存活时间。

9、54-112 位 char 经转换，为 Ticket（tgs），参考其格式转换。

10、必须 112 位，否则结束本模块。

4.2.10 发送 5 号控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 5。

2、要求接收到的字符串 DES 解密。

3、前 6 位 char 经转换为 int 类型 5，代表控制包种类，否则结束本模块。

4、7-9 位 char 经转换为 int 类型，且大于等于 0 小于 8，标记服务器 v 代号，否则结束本模块。

5、10-71 位 char 经转换，格式参照 Ticket（tgs）格式，代表 Ticket（tgs），否则结束本模块。

6、72-109 位 char 经转换为 int 类型，格式参照 Authenticator(c)，否则结束本模块。

7、必须共 109 位，否则结束本模块。

4.2.11 接收 6 号控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 6。

2、要求接受到的字符串 DES 解密。

3、前 6 位 char 经转换为 int 类型 6，代表控制包种类，查看是否是其他控制包，否则结束本模块。

4、7-22 位 char 经转换为 String 类型，长度必须小于等于 16，代表 k（cv）客户端与服务器 v 的秘钥，否则结束本模块。

5、23-25 位 char 经转换为 int 类型，且大于等于 0 小于 8，标记服务器 v 代号，否则结束本模块。

6、26-45 位 char 经转换为 String 类型，代表系统时间，格式必须为年月日时分秒，且字符串内无空格，样例：2018 年 4 月 28 日 8:35:42，否则结束本模块。

7、46-108 位经转换，参考 Ticket（v）。

8、必须共 108 位，否则结束本模块。

4.2.12 发送 7 号控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 7。

2、要求接受到的字符串 DES 解密。

3、7-69 位 char 经转换，格式参照 Ticket（v）格式，代表 Ticket（v），否则结束本模块。

4、69-107 位 char 经转换，格式参照 Authenticator(c)，否则结束本模块。

5、必须共 107 位，否则结束本模块。

4.2.13 接收 8 号控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 8。

2、前 6 位 char 经转换为 int 类型 8，代表控制包种类，查看是否是其他控制包，否则结束本模块。

3、后 7-9 位，char 内容为 YES 或者 NO ，不够的补空格（末尾补空格）。

4.2.14 发送 9 号数据包

注意：必须转为 Sting 类型。

1、属于数据包范围，格式标记为 9-31。

2、前 6 位 char 经转换为 int 类型 9-31，代表数据包种类。

3、后 7-200 位，char 内容为聊天内容数据，不够的补空格（末尾补空格）。

4.2.15 Socket 接发要求

发送：

建立 socket 之后，使用监听方法 Socket cs = s.accept();监听客户端请求。

定义输出流 OutputStream，数据流 DataOutputStream，使用数据流方法 writeUTF 将

字符串发送至客户端。

接收：

接收方式定义输入流 OutputStream，数据流 DataOutputStream，使用数据流方法 readUTF 将字符串接收客户端信息。

注意

必须使用统一接收格式，按上述约定写代码。

4.2.16 程序流程图

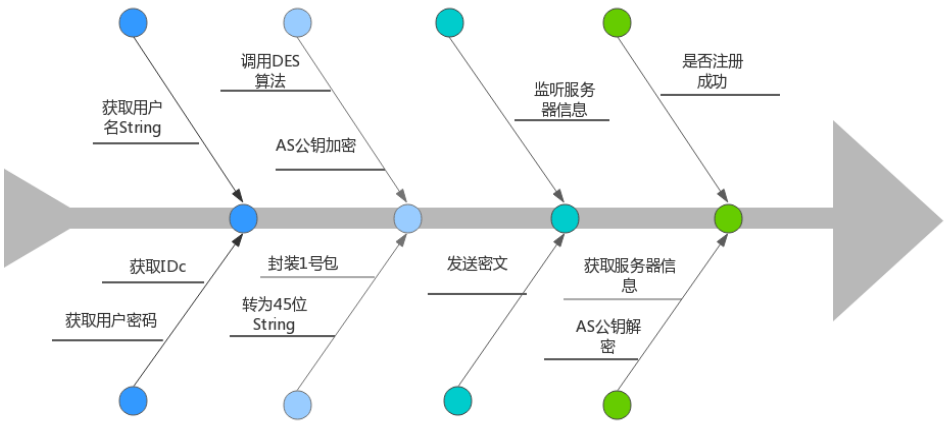


图 四-2 注册 client

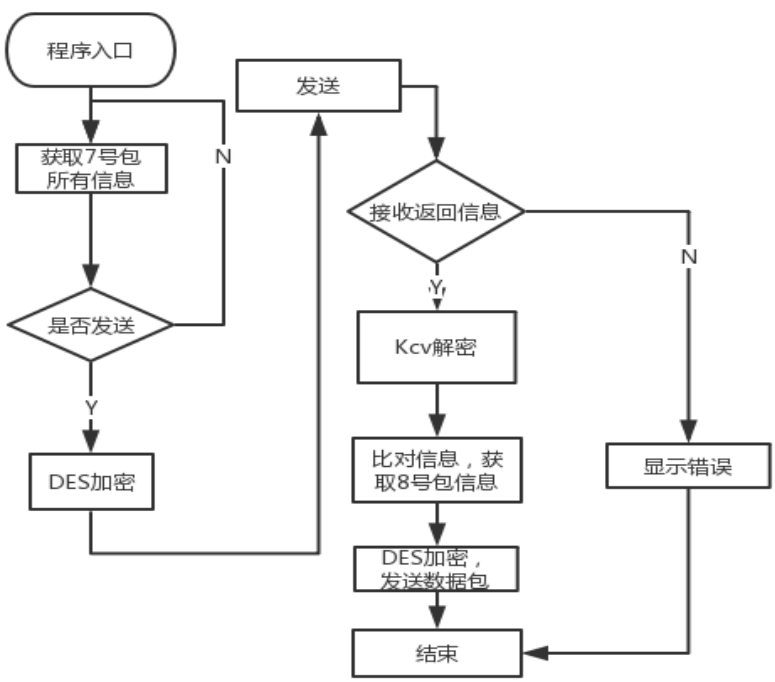


图 四-1 C 与 V 流程

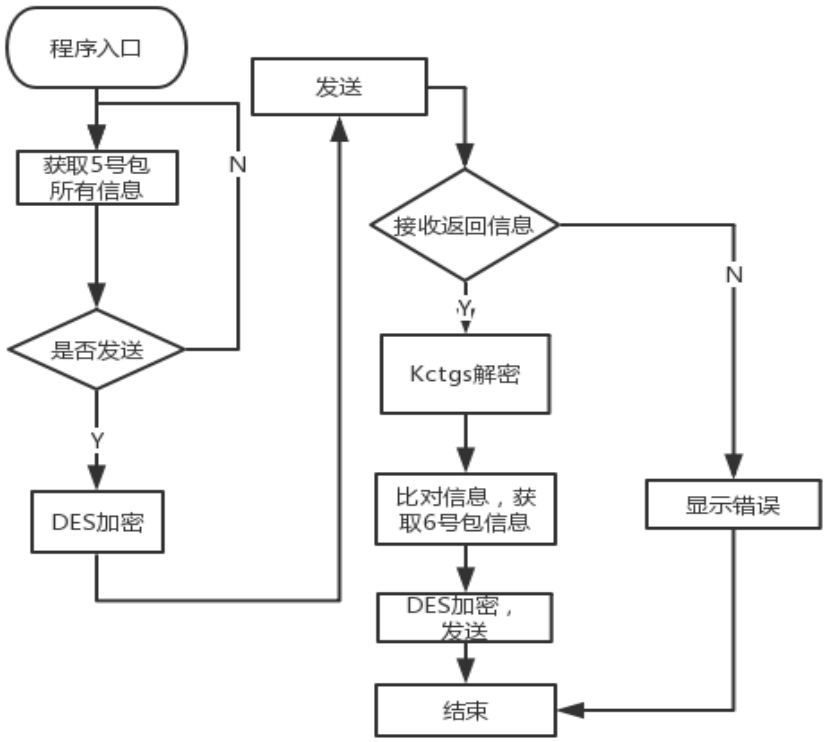


图 四-2 C 与 TGS 流程

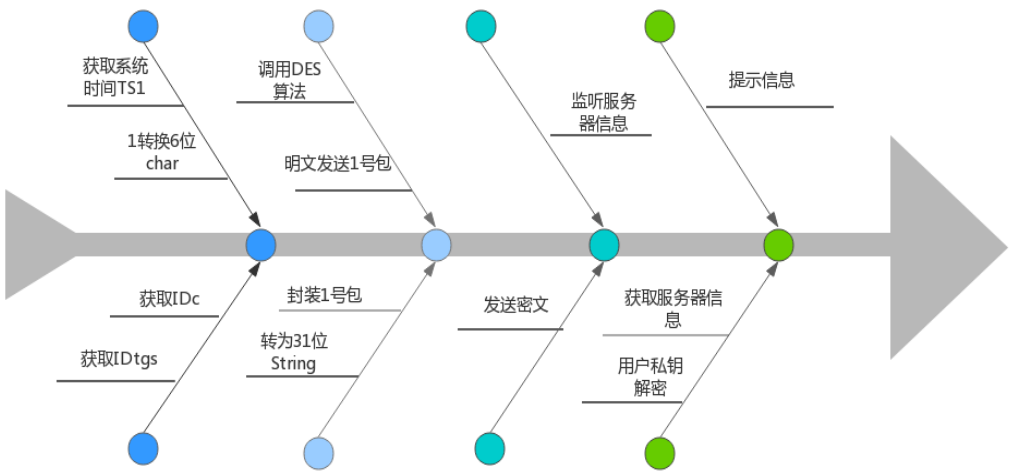


图 四-3 C-AS

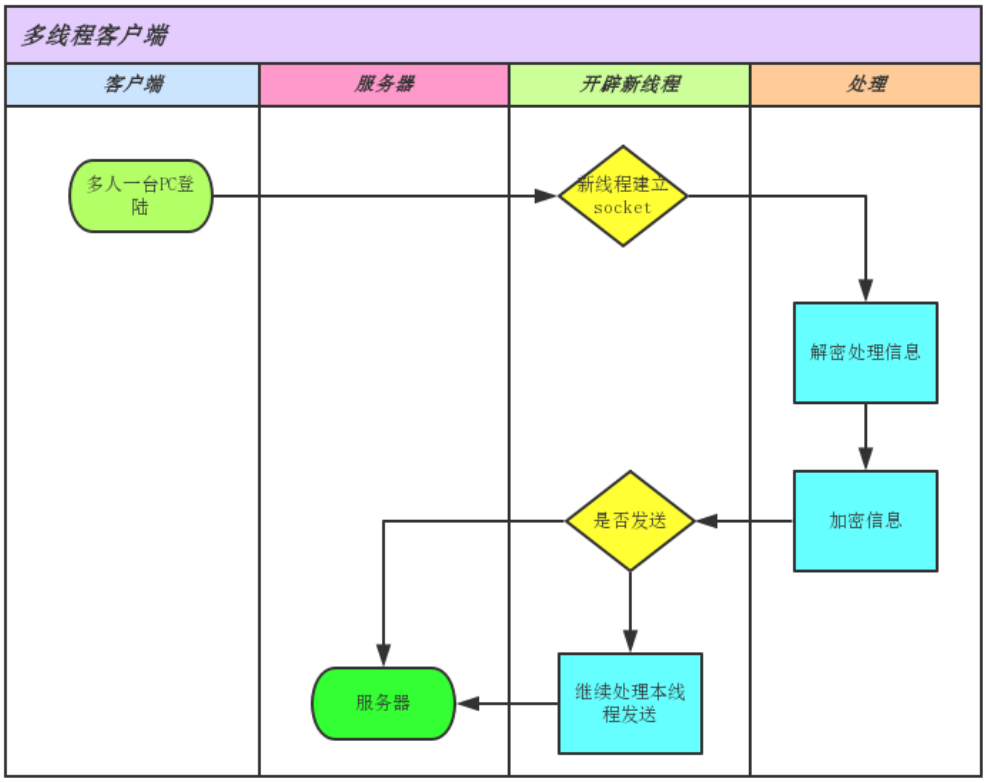


图 四-3 同 PC 多人登陆流程

§4.3 子模块——AS 设计

V 要实现功能为：提供聊天服务。

4.3.1 实现功能

1、 正确接收 C 请求的控制包。
2、 正确解开 C 请求的控制包。
3、 比较是否 C 控制信息是否正确。
4、根据约定格式封装和加密数据包。
5、Socket 发出至客户端。

4.3.2 模块输入

长度为 45 的 Stirng 控制包。
长度为的 31 的 String 数据包。

4.3.3 模块输出

长度为的 9 的 String 控制包。
长度为的 112 的 String 数据包。

4.3.4 代码命名规范

1、AS 主体函数。
AS_(功能对应英文)；
2、socket 发送接收。
AS_socket_(功能对应英文);

4.3.5 接收 1 号控制包格式要求

注意：必须转为 Sting 类型。
1、属于控制包范围，格式标记为 1。
2、要求接受到的字符串 DES 解密。
3、前 6 位 char 经转换为 int 类型 1，代表控制包种类，否则结束本模块。
4、7-26 位 char 经转换为 String 类型，长度必须小于等于 20，代表用户名，否则结束本模块。
5、27-29 位 char 经转换为 int 类型，且大于等于 0 小于 8，标记客户端代号，否则结束本模块。
6、30-45 位 char 经转换为 String 类型，代表用户秘钥，必须小于 16 位，且字符串内无空格，否则结束本模块。
7、必须共 45 位，否则结束本模块。

4.3.6 发送 2 号控制包格式要求

注意：必须转为 Sting 类型。
1、属于控制包范围，格式标记为 2。
2、要求接收到的字符串 DES 解密。
3、前 6 位 char 经转换为 int 类型 2，代表控制包种类，查看是否是其他控制包，否则结束本模块。
4、7-9 位 char 经转换为 string 类型，且为 YES 和 NO，标记是否注册成功，否则结束本模块。
5、必须共 9 位，否则结束本模块。

4.3.7 接收 3 号控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 3。

2、要求接受到的字符串 DES 解密。

3、前 6 位 char 经转换为 int 类型 3，代表控制包种类，否则结束本模块。

4、7-9 位 char 经转换为 int 类型，且大于等于 0 小于 8，标记客户端代号，否则结束本模块。

5、10-11 位 char 经转换为 int 类型，且大于等于 0 小于 4，标记访问 tgs 标号，否则结束本模块。

6、11-31 位 char 经转换为 String 类型，代表系统时间，格式必须为年月日时分秒，且字符串内无空格，样例：2018 年 4 月 28 日 8:35:42，否则结束本模块。

7、必须共 31 位，否则结束本模块。

4.3.8 发送 4 号控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 4。

2、要求接收到的字符串 DES 解密。

3、前 6 位 char 经转换为 int 类型 4，代表控制包种类，查看是否是其他控制包，否则结束本模块。

4、7-9 位 char 经转换为 int 类型，且大于 0 小于等于 8，标记是哪个客户端，如果不是本机，否则结束本模块。

5、10-25 位 char 经转换为 String 类型，且必须 16 位，代表 K（c，tgs）秘钥。

6、26-27 位 char 经转换为 int 类型，且大于 0 小于等于 4，否则结束本模块。

7、28-47 位 char 经转换为 Stirng 类型，代表系统时间。

8、48-53 位 char 经转换为 String 类型，代表存活时间。

9、54-112 位 char 经转换，为 Ticket（tgs），参考其格式转换。

10、必须 112 位，否则结束本模块。

4.3.9 Socket 接发要求

发送：

建立 socket 之后，使用监听方法 Socket cs = s.accept();监听客户端请求。

定义输出流 OutputStream，数据流 DataOutputStream，使用数据流方法 writeUTF 将字符串发送至客户端。

接收：

接收方式定义输入流 OutputStream，数据流 DataOutputStream，使用数据流方法 readUTF 将字符串接收客户端信息。

注意

必须使用统一接收格式，按上述约定写代码。

4.3.10 程序流程与要求

1、文字描述：

始终监听客户端请求。

始终监听客户端提交的信息。

如果是 1 号控制包，则使用 RSA 自己的私钥解开密文，提取信息。

发送 2 号控制包，格式参照其格式，转为 String，socket 发送。

如果是 3 号控制包，则使用 K（c）DES 解密密文。如果解开，则继续。否则，结束本模块。

提取前 6 位 char 经转换为 int 类型 3，代表控制包种类，否则结束本模块。

提取 7-9 位 char 经转换为 int 类型，且大于等于 0 小于 8，标记客户端代号，否则结束本模块。

提取 10-11 位 char 经转换为 int 类型，且大于等于 0 小于 4，标记访问 tgs 标号，否则结束本模块。

提取 11-31 位 char 经转换为 String 类型，代表系统时间，格式必须为年月日时分秒，且字符串内无空格，样例：2018 年 4 月 28 日 8:35:42，否则结束本模块。

发送 4 号控制包，格式参照其格式，转为 String，socket 发送。

2、流程图

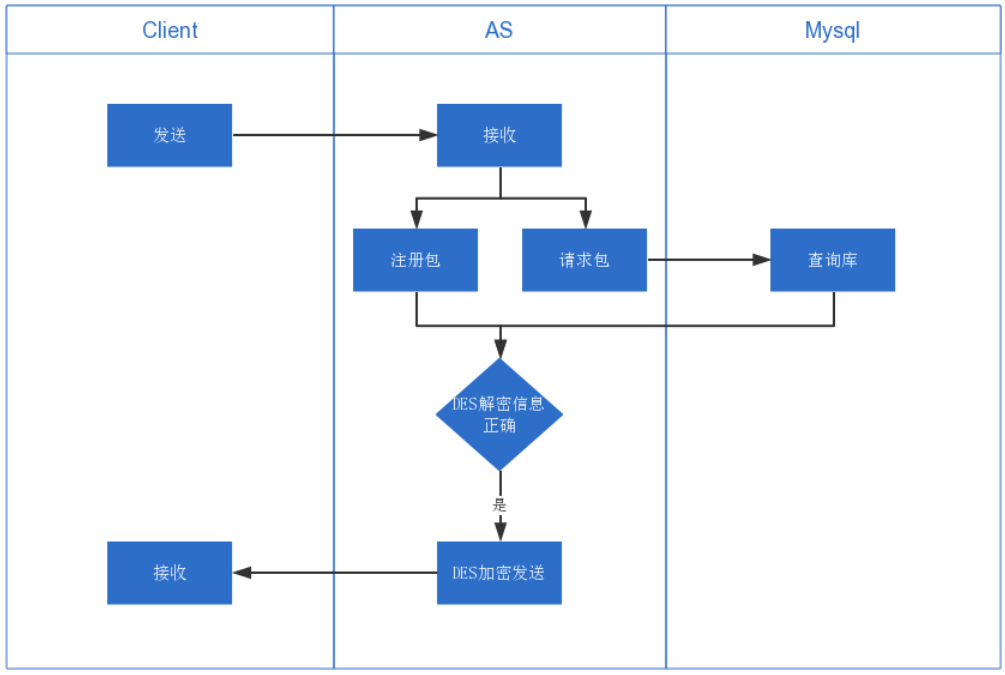


图 四-4 AS 单独线程流程

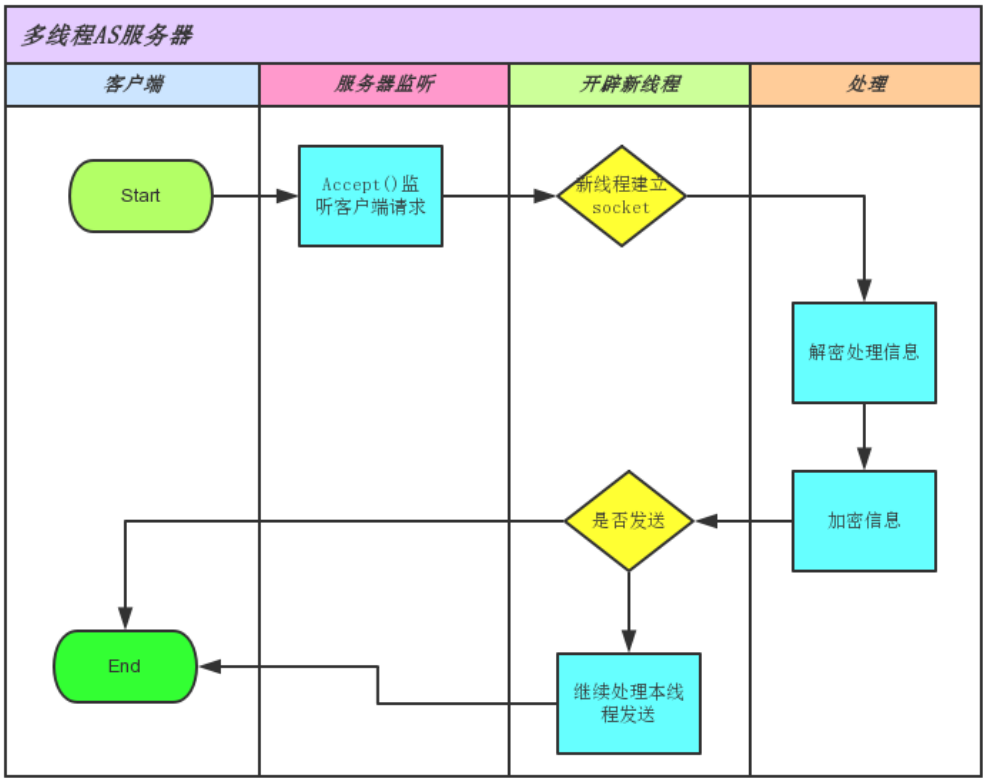


图 四-5 多客户端 AS 流程

§4.4 子模块——TGS 设计

TGS 要实现功能为：发放访问服务器 v 的票据 ticket。

4.4.1 实现功能

1、 正确接收 C 请求的控制包。
2、 正确解开 C 请求的控制包。
3、 比较是否 C 信息是否正确。
4、 根据约定格式封装和加密数据包。
5、 Socket 发出至客户端。

4.4.2 模块输入

长度为 109 的 Stirng 控制包。

4.4.3 模块输出

长度为 108 的 String 控制包。

4.4.4 代码命名规范

1、TGS 主体函数。

TGS_(功能对应英文)；

2、socket 发送接收。

TGS_socket_(功能对应英文);

4.4.5 接收控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 5。

2、要求接收到的字符串 DES 解密。

3、前 6 位 char 经转换为 int 类型 5，代表控制包种类，否则结束本模块。

4、7-9 位 char 经转换为 int 类型，且大于等于 0 小于 8，标记服务器 v 代号，否则结束本模块。

5、10-71 位 char 经转换，格式参照 Ticket（tgs）格式，代表 Ticket（tgs），否则结束本模块。

6、72-109 位 char 经转换为 int 类型，格式参照 Authenticator(c)，否则结束本模块。

7、必须共 109 位，否则结束本模块。

4.4.6 发送控制包格式要求

注意：必须转为 Sting 类型。

1、属于控制包范围，格式标记为 6。

2、要求接受到的字符串 DES 解密。

3、前 6 位 char 经转换为 int 类型 6，代表控制包种类，否则结束本模块。

4、7-22 位 char 经转换为 String 类型，长度必须小于等于 16，代表 k（cv）客户端与服务器 v 的秘钥，否则结束本模块。

5、23-25 位 char 经转换为 int 类型，且大于等于 0 小于 8，标记服务器 v 代号，否则结束本模块。

6、26-45 位 char 经转换为 String 类型，代表系统时间，格式必须为年月日时分秒，且字符串内无空格，样例：2018 年 4 月 28 日 8:35:42，否则结束本模块。

7、46-108 位经转换，参考 Ticket（v）。

8、必须共 108 位，否则结束本模块。

4.4.7 Socket 接发要求

发送：

建立 socket 之后，使用监听方法 Socket cs = s.accept();监听客户端请求。

定义输出流 OutputStream，数据流 DataOutputStream，使用数据流方法 writeUTF 将字符串发送至客户端。

接收：

接收方式定义输入流 OutputStream，数据流 DataOutputStream，使用数据流方法 readUTF 将字符串接收客户端信息。

注意

必须使用统一接收格式，按上述约定写代码。

4.4.8 程序流程与要求

1、文字描述

始终监听客户端提交的信息。

使用 K（tsg）DES 解密密文。如果解开，则继续。否则，结束本模块。

提取前 6 位，比对是否是 C 发来的请求 v 的控制报文，否则，结束本模块。

提取 7-9 位，比对访问的 V 的服务器 ID。

提取 10-71 位，比对 Tickettgs 内容。提取 K（c，tgs），IDc 和 ADc，IDtgs，TS2，和 LIFETIME2。分别存入对应字段。

提取 72-109 位，提取 Authenticatorc 内容，内容前 3 位为 IDc，4-18 位为 ADc，最后 20 位为 TS3。

比对 C 的 IDc 和 ADc 是否相同，相同则继续，否则结束模块。

比对 TS3 和 LIFETIME2，算出时间差，计算剩余时间，如果大于 0，则继续，否则，结束本模块。

使用 Kc，tgs 加密发送的报文，报文格式参照 2.2.4 发送报文格式要求，转换为字符串 String，使用 DES 加密。

Socket 发送给 C。

2、流程图

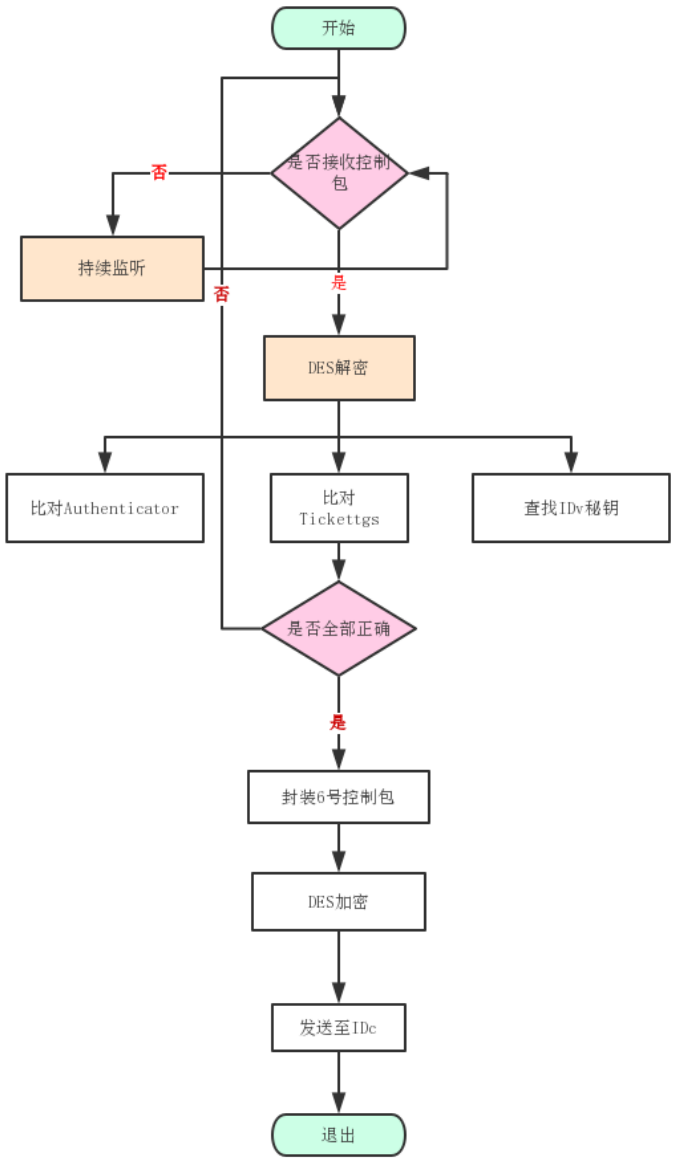


图 四-6 TGS 单独线程流程图

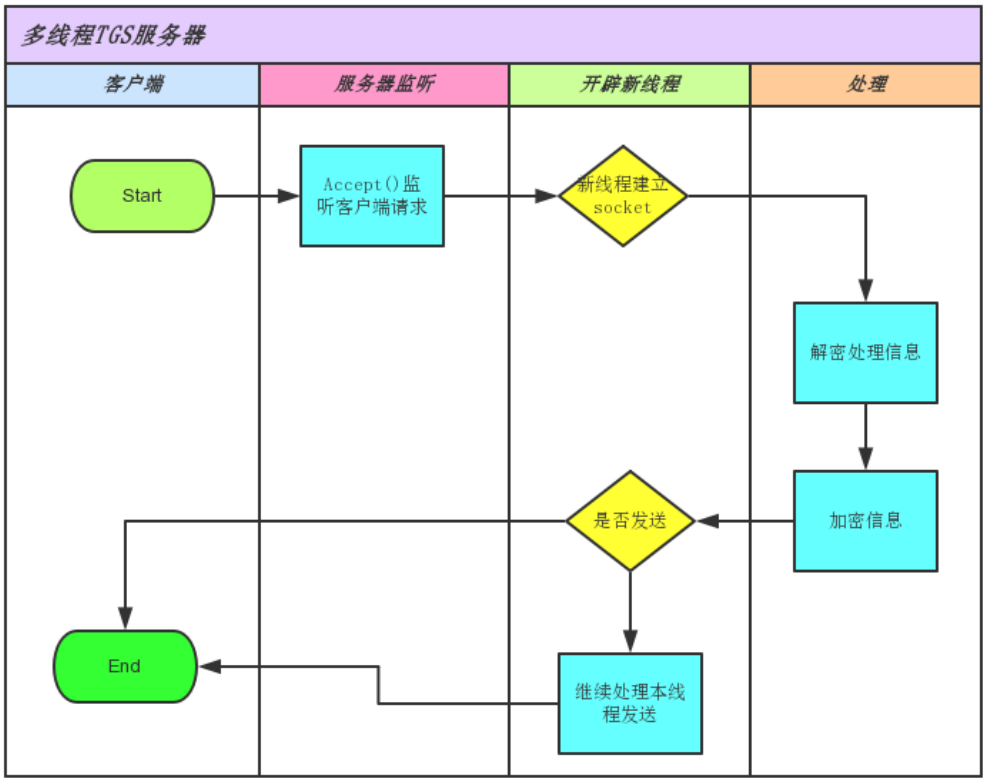


图 四-6 TGS 多客户端流程

§4.5 子模块——服务器 V 设计

V 要实现功能为：提供聊天服务。

4.5.1 实现功能

4、 正确接收 C 请求的控制包和数据包。
5、 正确解开 C 请求的控制包和数据包。
6、 比较是否 C 控制信息是否正确。
7、 处理 C 的数据包，正确交互聊天代码。
5、根据约定格式封装和加密数据包。
6、Socket 发出至客户端。

4.5.2 模块输入

长度为 107 的 Stirng 控制包。
长度为的 200 的 String 数据包。

4.5.3 模块输出

长度为的 9 的 String 控制包。
长度为的 200 的 String 数据包。

4.5.4 代码命名规范

1、V 主体函数。
V_(功能对应英文)；
2、socket 发送接收。
V_socket_(功能对应英文);

4.5.5 接收控制包格式要求

注意：必须转为 Sting 类型。
1、属于控制包范围，格式标记为 7。
2、要求接受到的字符串 DES 解密。
3、7-69 位 char 经转换，格式参照 Ticket（v）格式，代表 Ticket（v），否则结束本模块。
4、69-107 位 char 经转换，格式参照 Authenticator(c)，否则结束本模块。
5、必须共 107 位，否则结束本模块。

4.5.6 发送控制包格式要求

注意：必须转为 Sting 类型。
1、属于控制包范围，格式标记为 8。
2、前 6 位 char 经转换为 int 类型 8，代表控制包种类，否则结束本模块。
3、后 7-9 位，char 内容为 YES 或者 NO ，不够的补空格（末尾补空格）。

4.5.7 发送数据包格式要求

注意：必须转为 Sting 类型。
1、属于数据包范围，格式标记为 9-31。
2、前 6 位 char 经转换为 int 类型 9-31，代表数据包种类。
3、后 7-200 位，char 内容为聊天内容数据，不够的补空格（末尾补空格）。

4.5.8 发送数据包格式要求

注意：必须转为 Sting 类型。

1、属于数据包范围，格式标记为 9-31。

2、前 6 位 char 经转换为 int 类型 9-31，代表数据包种类。

3、后 7-200 位，char 内容为聊天内容数据，不够的补空格（末尾补空格）。

4.5.9 Socket 接发要求

发送：

建立 socket 之后，使用监听方法 Socket cs = s.accept();监听客户端请求。

定义输出流 OutputStream，数据流 DataOutputStream，使用数据流方法 writeUTF 将字符串发送至客户端。

接收：

接收方式定义输入流 OutputStream，数据流 DataOutputStream，使用数据流方法 readUTF 将字符串接收客户端信息。

注意

必须使用统一接收格式，按上述约定写代码。

4.5.10 程序流程与要求

1、文字描述：

始终监听客户端请求。

始终监听客户端提交的信息。

使用 K（v）DES 解密密文。如果解开，则继续。否则，结束本模块。

提取前 6 位，比对是否是 C 发来的请求服务的控制报文，否则，结束本模块。

提取 7-69 位，比对 Ticketv 内容。提取 K（c，v），IDc 和 ADc，IDtgs，TS4，和 LIFETIME4。分别存入对应字段。

提取 72-109 位，提取 Authenticatorc 内容，内容前 3 位为 IDc，4-18 位为 ADc，最后 20 位为 TS3。

比对 C 的 IDc 和 ADc 是否相同，相同则继续，否则结束模块。

比对 TS3 和 LIFETIME2，算出时间差，计算剩余时间，如果大于 0，则继续，否则

在剩余时间内开放聊天室服务。转发数据包。

根据数据包包头 6 位比对数据所对应的信息，执行相应的函数操作。比如 9 代表多人聊天室，数据内容转发至多人聊天室。

2、流程图

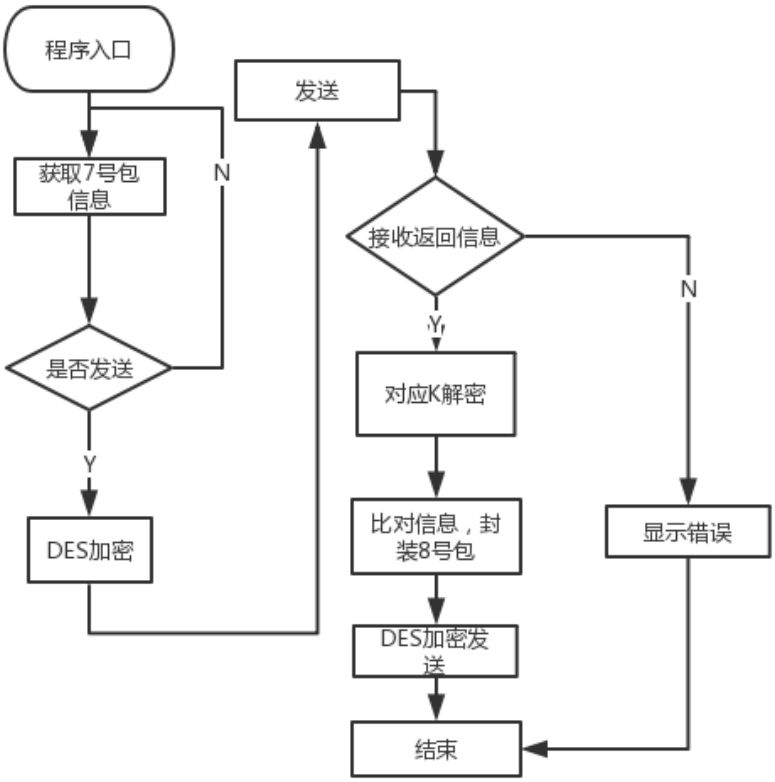

程序入口
获取7号包信息
是否发送
N
Y
DES加密
发送
接收返回信息
N
Y
对应K解密
比对信息，封装8号包
DES加密发送
显示错误
结束

图 四-7 V 控制包信息

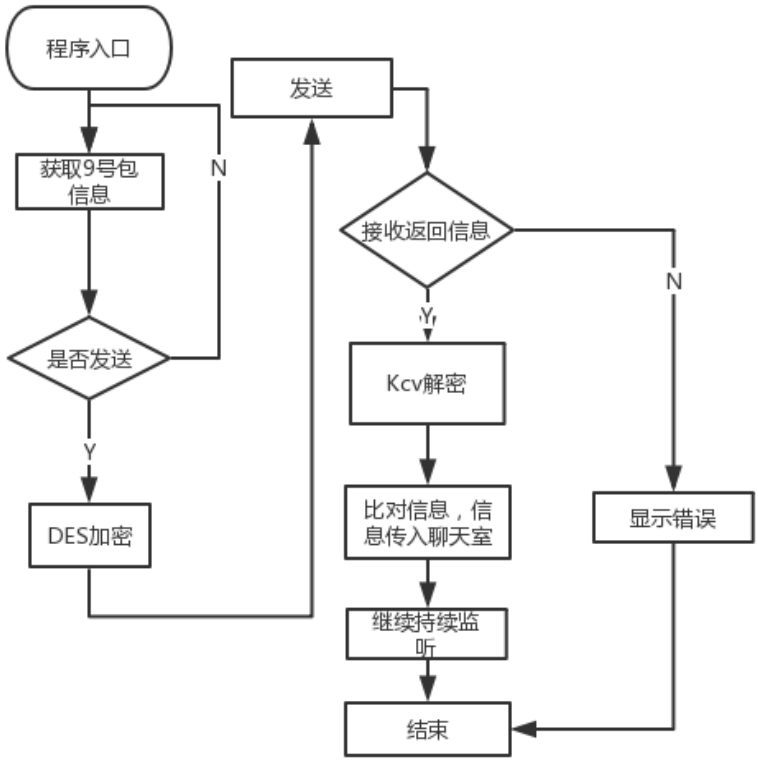


图 四-7 服务器处理数据包

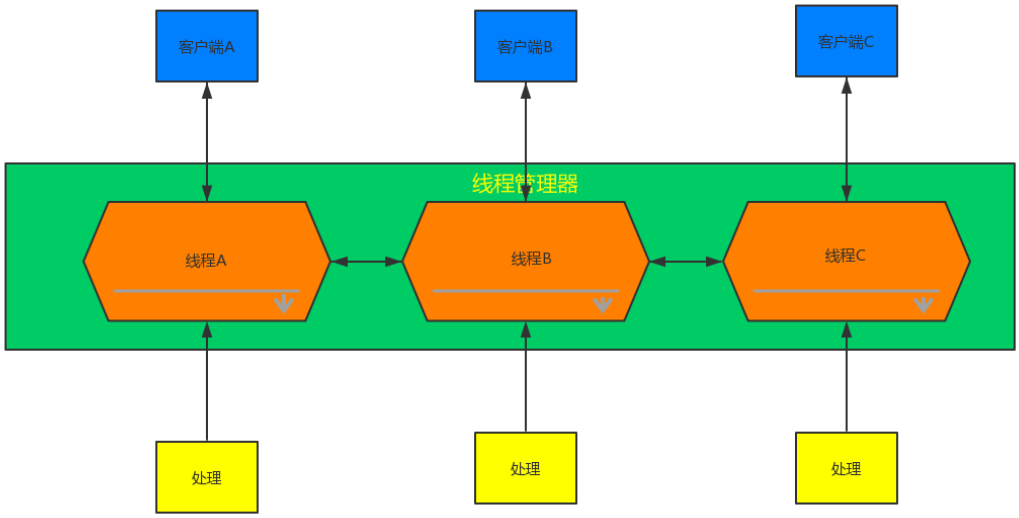


图 四-8 客户端在服务器中关系

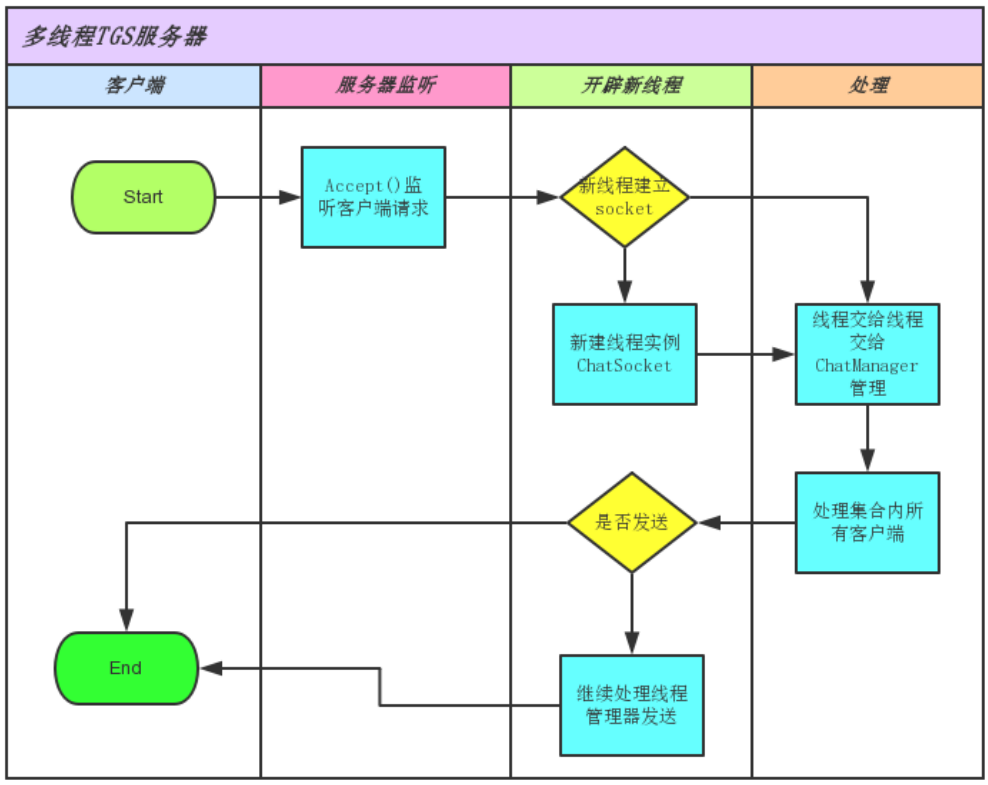


图 四-9 服务器对应多个 client 流程

§4.6 子模块——聊天室设计

4.6.1 实现功能

1、显示用户是否在线。
2、显示用户历史消息。
3、用户之间在聊天室相互聊天。
4、后期其他功能。

4.6.2 模块输入

长度不限的 String。

4.6.3 模块输出

长度不限的 String。

4.6.4 代码命名规范

1、 Client 主题函数

C_chat_(功能对应英文)；

2、 server 主体函数

S_chat_(功能对应英文)；

4.6.5 程序流程与要求

1、 文字描述

客户端设计为多线程模式。可以在一台机子上运行多个用户。

客户端一直处于监听状态，如果收到服务器发来的数据包，则进行拆分，然后利用自己的 DES 进行解密，将解密后的数据显示在客户端界面上。

客户端监听用户输入，如果有数据发送，则加密封装成数据包发送给服务器。

服务器设计为多线程模式，可以接入多个客户端。

服务器开启后，时刻监听客户端交互信息。

服务器如果收到客户端的消息，进行解密

将客户端消息封装成数据包转发给所有已连接客户端信息。

2、 流程图

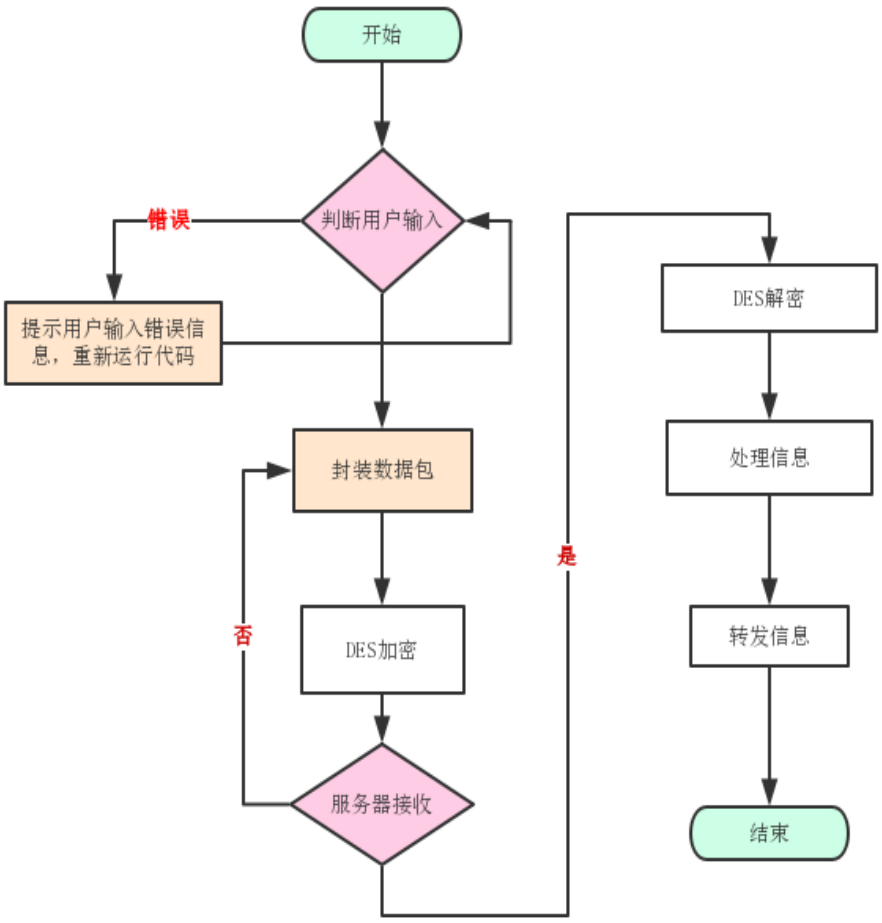

开始
判断用户输入
错误
提示用户输入错误信息，重新运行代码
封装数据包
DES加密
服务器接收
否
是
DES解密
处理信息
转发信息
结束

图 四-10 聊天室

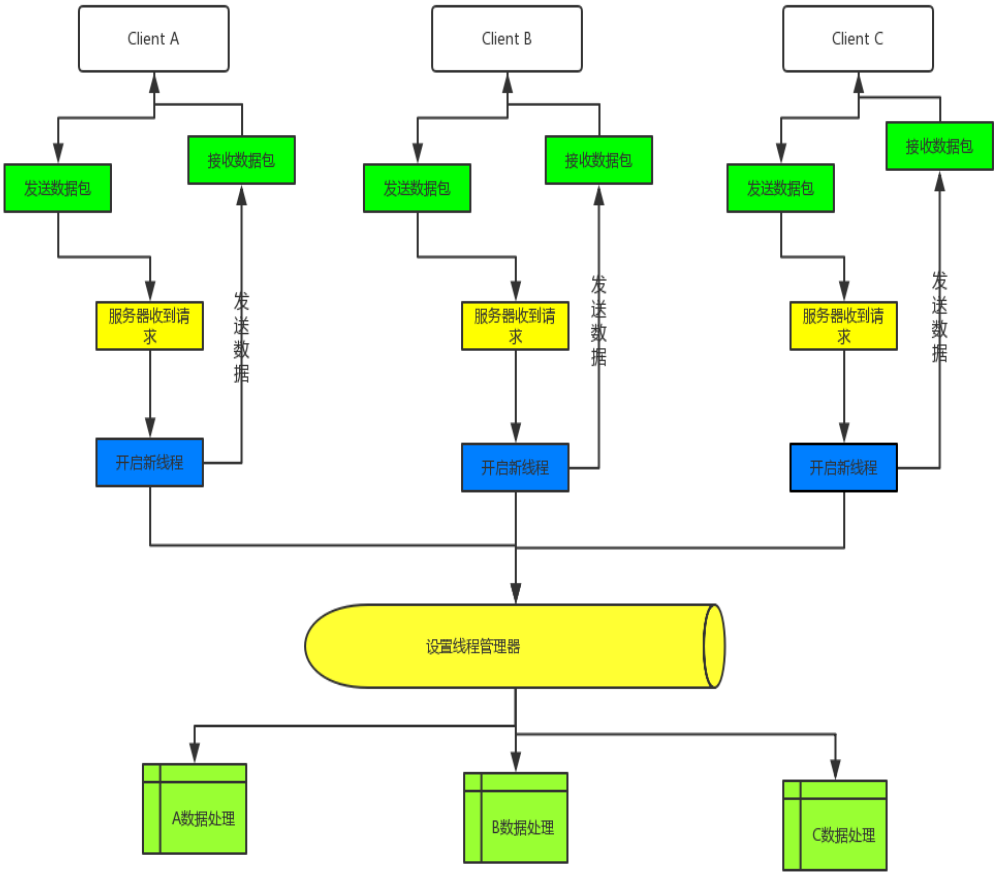


图 四-11 多客户端流程图

§4.7 加密解密

加密解密不需要加解密包头前 6 位。

加解密信息输出在控制台。

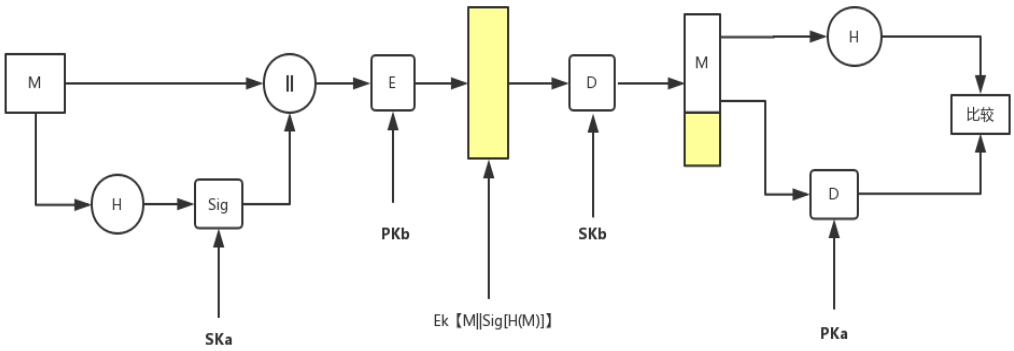
4.7.1 RSA

RSA 在注册时使用，使用 AS 的公钥加密，AS 使用私钥解密，这是 1 号控制包。

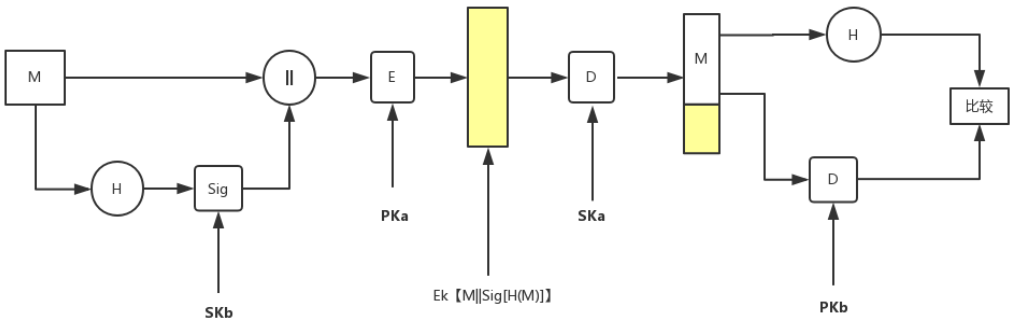
C 公钥加密，C 使用私钥解密，这是 2 号控制包。

证书交换采用提前物理分发的方式使 Client 与 AS 分别获得对方的秘钥。根据 RSA 消息认证方案，将用户口令首先采用 md5 进行 hash 变换生成 H（M），然后将 H（M）用 client 的私钥进行签名得到 Sig[H(M)],然后与 M 进行拼接，然后采用 PKb 即 AS 的公钥进行

加密，生成 EK[M||Sig[H(M)]]，然后发送给 AS，AS 收到数据以后，首先利用自己的私钥 SKb 进行解密得到 M||Sig[H(M)，然后将 M 与 Sig[H(M)]分离，Sig[H(M)]利用 client 的公钥 PKa 进行签名认证，解密得到 H(M)，然后将它之前分离的 M 采用 md5 进行 Hash，得到的结果与 H(M)进行对比，如果一致，则说明用户口令传输是完整的，AS 接收到的 M 对的。



同理 AS 将认证成功或失败的结果采用 RSA 加密发送给 Client，具体流程与上面类似，只是采用的秘钥不同。



4.7.2 多个 client 之间 RSA 认证

当程序中增加私聊功能后，通过如上 RSA 认证算法，在双方发送信息前，通过在信息后拼接认证信息，双方验证后，开始通信，私聊中服务器将信息的转发直接移交至客户端，这在最后功能完善中实现。

4.7.3 DES

根据 Kerboros 要求，除了包头 6 位，其他部分按照模块设计中的要求加解密即可。

DES 主要用在大批量数据传输中，在本系统中主要用在登录流程的验证以及用户之间数据包的传递。

第五章　时序图

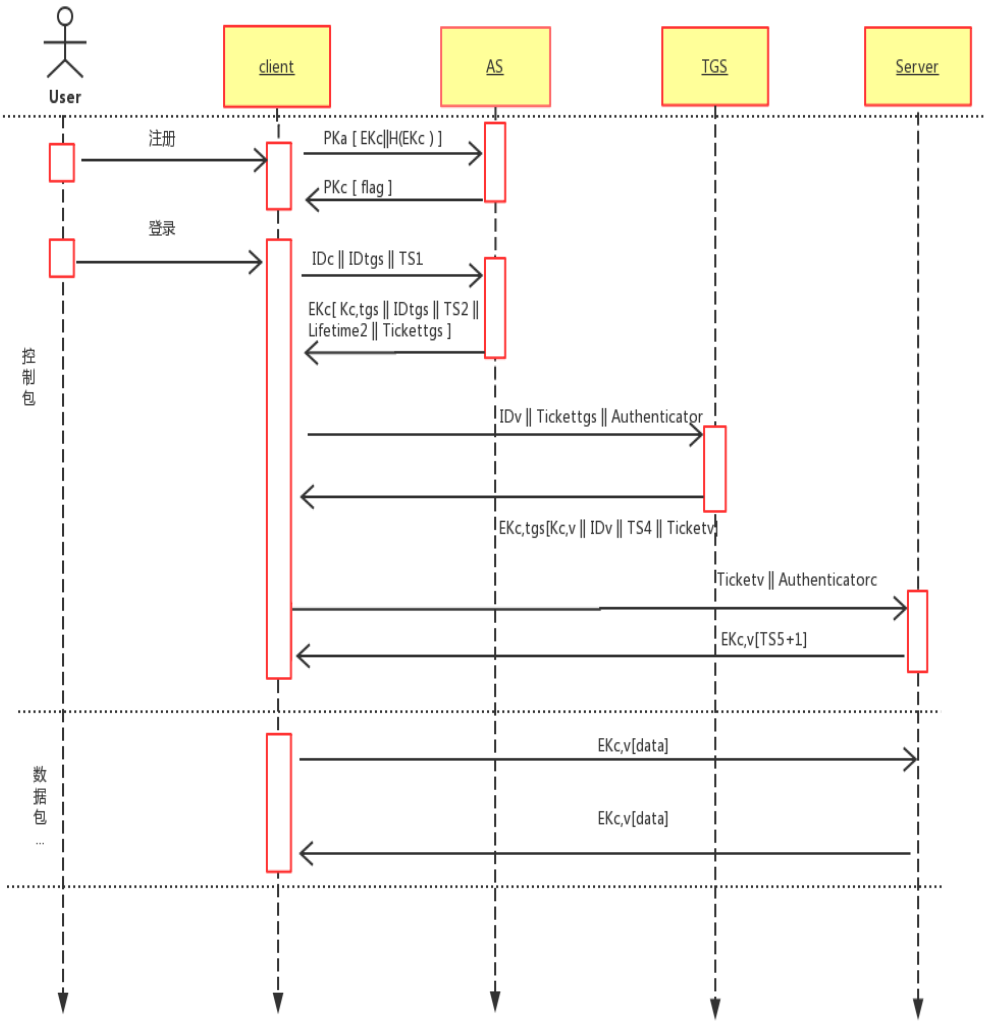


图 五-1 总时序图

1.Client 端注册

首先用户在登陆之前需要注册，注册的主要目的是将自己的口令即 Kc 告诉 AS，这样当客户端请求票据许可服务的时候，AS 就可以用 Kc 将票据加密发送给 client，client 就可以用 Kc 解密获得票据。

注册口令传输的具体方法是采用 RSA 加密，即将用户 ID，用户输入的口令用 AS 的公钥加密然后再拼接上口令的哈希值生成注册报文，发送给 AS，AS 接收到报文以后，需要先用自己的私钥进行解密，获取用户口令，然后将口令进行同样的 hash 函数变换与收到报文中的 hash 值比对，如果一致则说明口令是完整的。然后 AS 将用户 ID 以及用户口令存放入数据库。

用户 ID	用户口令 Kc
000	Aaaaaa
001	Bbbbbb
010	Ccccccc

表 3-1 用户口令

AS 最后发送一个注册成功报文给客户端，通知客户端注册成功。

2.Client 端登录

用户输入用户名和密码，然后点击登录

对于客户端来说，则是通过 EKc 解密来获取 EKc,tgs，通过 EKc,tgs 解密来获取 Kc,v

对于 TGS 来说，是通过 EKtgs 获取 Ticketgs 中的 Kc,tgs。

对于 V 来说，是通过 EKv 获取 Ticketv 中的 Kc,v。

因此 Ticket 的一个很重要的作用就是把 Kc,tgs 告诉 TGS，把 Kc,v 告诉 V。

EKtgs:ticket 用只有 AS 和 TGS 才知道的密钥加密，以预防篡改；

EKv:用只有 AS 和 TGS 才知道的密钥加密的票据，以预防篡改；

这样应该提前生成,然后存起来

Kc,tgs: session key 的副本，由 AS 产生，client 可用于在 AS 与 client 之间信息的安全交换，而不必共用一个永久的 key。

Kc,v: session key 的副本，由 TGS 生成，供 client 和 server 之间信息的安全交换，而无须共用一个永久密钥。这两个应该是临时生成的，不用存。

于是，首先 AS 上存有 Ktgs

IDtgs	Ktgs
xxx	xxxxxxx

表 3-2 AS—Ktgs 秘钥

TGS 上存有

IDtgs	Ktgs
xxx	xxxxxxx

表 3-3 TGS—Ktgs 秘钥

TGS 上存有

IDv	Kv
xxx	xxxxxxx

表 3-4 TGS— Kv 秘钥信息

V 上存有

IDv	Kv
xxx	xxxxxxx

表 3-5 V—Kv 秘钥信息

3.C -> AS　IDc|IDtgs|TS1

客户端向 AS 请求获取 Tickettgs，发送的报文中包含客户端的 ID，TGS 的 ID，当前系统时间 TS1。

客户端操作流程如下：

（1） 获取 IDc
（2） 获取 IDtgs
（3） 获取 TS1
（4） 拼接成数据包
（5） 发送给 AS

4. AS-> C　EKc[Kc,tgs|IDtgs|TS2|lifetime2|Tickettgs]

当 AS 收到客户端的报文后，AS 会随机生成一个 C 和 tgs 之间的对称加密的秘钥 Kc,tgs，这个秘钥的作用是 C 与 TGS 之间的会话秘钥，在下一步的访问 TGS 的时候使用。IDtgs 表示访问的 TGS 的 ID，TS2 表示当前系统时间，lifetime2 表示 Tickettgs 的生命周期，进行消息重放的时候使用，太短则用户需要重复输入口 令，太长则为攻击者提供了大量机会。Tickettgs 是客户端访问 TGS 是的票据，里面包含了 Kc,tgs|IDc|ADc|IDtgs|TS2||lifetime2 然后采用 EKtgs 加密封装，EKtgs 是只有 AS 与 TGS 之间才知道的秘钥，以预防 Tickettgs 被篡改。

TGS 端在收到票据以后可以利用 EKtgs 解密获取 Kc,tgs 与 Client 会话。

AS 操作流程如下：

（1） 解析 Client 发来的包
（2） 随机生成 Kc,tgs
（3） 获取 IDtgs
（4） 获取 TS2
（5） 获取 lifetime2
（6） 生成 Tickettgs
（7） 拼装各字段
（8） 发送给 Client

生成 Tickettgs 的流程如下：

（1） 获取 Kc,tgs

（2） 获取 IDc
（3） 获取 ADc
（4） 获取 IDtgs
（5） 获取 lifetime2
（6） 获取 EKtgs（AS 与 TGS 之间的秘钥）
（7） 拼装各字段
（8） 采用 EKtgs 加密
（9） 生成 Tickettgs EKtgs[Kc,tgs|IDc|ADc|IDtgs|TS2||lifetime2]

5. C->TGS IDv|Tickettgs|Authenticator

Tickettgs EKtgs[Kc,tgs|IDc|ADc|IDtgs|TS2||lifetime2]

Authenticator：Kc,tgs [IDc|ADc|TS3]

客户端收到 AS 发来的包后对包进行解析，首先输入口令 EKc 进行解密，解密的结果是可以获取 Kc,tgs（AS 生成的 client 与 TGS 会话的秘钥）、IDc、ADc 和 Tickettgs 等，然后就可以利用 Kc,tgs 对 IDc|ADc|TS3 加密生成 Authenticator，然后 IDc|Tickettgs|Authenticator 发送给 TGS。这样我们就明白了 AS 随机生成 Kc,tgs,以后是如何把 Kc,tgs 告诉 Client 和 TGS 的。

Client 操作流程如下：

（1） 输入 EKc 解析 AS 发来的包，获取 Kc,tgs 以及 Tickettgs
（2） 获取 IDv
（3） 获取 ADc
（4） 获取 TS3
（5） 生成 Authenticator
（6） 拼接 IDv|Tickettgs|Authenticator
（7） 发送给 TGS

6.TGS->C EKc,tgs[Kc,v|IDv|TS4|Ticketv]

TGS 收到 Client 发来的包后，首先分割出来 IDv，Tickettgs，Authenticator。然后通过自己存的 Ektgs（AS 与 TGS 之间的会话秘钥）解开 Tickettgs，获取 Kc,tgs,IDc,ADc,TS2。然后利用 Kc,tgs 解开 Authenticator，获取 IDc 和 ADc 和 TS3 与 Tickettgs 中的数据进行校验，如果不符合要求，则认证失败。否则，TGS 随机生成一个 KCv（Client 与 Server 之间的秘钥），然后生成一个 Ticketv，采用 EKc,tgs 加密发送给客户端。

TGS 操作流程如下：

（1） 解析 Client 发来的包，获取 IDv、Tickettgs、Authenticator
（2） 用 EKtgs 解析 Tickettgs 获取 Kc,tgs,IDc,ADc 等
（3） 用 Kc,tgs 解析 Authenticator 获取 IDc,ADc 等
（4） 校验数据信息
（5） 随机生成一个 Kc,v

（6） 获取 IDv
（7） 获取 TS4
（8） 生成 Ticketv
（9） 拼接各字段，采用 EKc,tgs 加密
（10） 发送给 Client

生成 Ticketv 的流程如下：

（1） 获取 Kc,v
（2） 获取 IDc
（3） 获取 ADc
（4） 获取 IDv
（5） 获取 TS4
（6） 获取 LifeTime4
（7） 采用 EKv（TGS 与 V 之间的秘钥）加密
（8） 生成 Ticketv EKv[Kc,v|IDc|ADc|IDv|TS4|LifeTime4]

7. C->V Ticketv|Authenticator

Ticketv EKv[Kc,v|IDc|ADc|IDv|TS4|LifeTime4]
Authenticator EKc,v[IDc|ADc|TS5]

客户端收到 TGS 发来的包后，先用 Ec,tgs 解密，获取到 Ticketv，Kc,v 等字段。

然后利用 Kc,v 加密 IDc、ADc、TS5 等字段生成 Authenticator，然后与 Ticketv 一起发送给服务器 V。

Client 操作流程如下：

（1） 采用 EKc,tgs 解密 TGS 发来的包，获取 Ticketv 和 Kc,v
（2） 获取 IDc
（3） 获取 ADc
（4） 获取 TS5
（5） 用 Kc,v 生成 Authenticator
（6） Authenticator 与 Tickev 拼接发送给 V

8 . V->C EKc,v[TS5+1]

服务器收到 Client 端的包之后，首先分割出 Ticketv，然后利用自己保持的 Kv 对 Ticketv 进行解密，获取 Kc,v、IDc、ADc、IDv 等字段，然后分割出 Authenticator，利用 Kc,v 进行解密，解密之后获取 IDc、ADc 等字段进行校验。如果校验成功将 TS5 的值加 1 然后用 Kc,v 加密之后传给客户端，到此认证成功。

服务器操作流程如下：

（1）解析 clien 发来的包，获取 Ticketv 和 Authenticator
（2）用 Kv 解密 Ticketv，获取 Kc,v 等字段
（3）用 Kc,v 解密 Authenticator，获取 IDc、ADc，TS5

（4）进行校验。

（5）Kc,v 加密[TS5+1]发送给客户端，表示认证成功。

9. C->V EKc,v[data]

认证成功之后，客户端与服务器进行数据通信，因为数据量比较大采用 DES 加密算法，采用的秘钥就是 C 与 V 通过认证流程获取的 EKc,v,为了防止秘钥被破解，C 与 V 之间的秘钥当生存时间到了之后就需要进行重新生成，进行 AS 重放，重放时间不宜过短也不宜过长，过短会导致重放次数太多，过长会导致用户秘钥被破解的时间增加。

客户端主要流程如下：

（1） 监听服务器数据报文

（2） 如果接收到服务器数据包，则进行解密将数据显示在本地

（3） 如果有消息发送则输入聊天数据进行加密

（4） 加密完之后封装成数据包

（5） 发送给 V

10. V->C EKc,v[data]

认证成功之后，当服务器收到各客户端的数据包之后，将数据包进行解析，分解出数据项之后进行解密，将解密之后的会话内容广播到所有在线 client 端，服务器端主要流程如下：

（1） 监听客户端发来的数据报文

（2） 如果收到则解析数据包对数据部分解密

（3） 解密完之后将数据部分加密封装成包

（4） 广播给所有在线客户端

第六章　甘特图

6.1.1 概述

代码进度，规定每天每人在比较难的模块上 150 行，简单时 200 以上，得出如下工作安排。其中，测试阶段在代码开始一定程度后就需要不断测试与集成系统 ，防止后续重大错误。代码升级阶段花费 6 天，增加小组设计系统的功能。

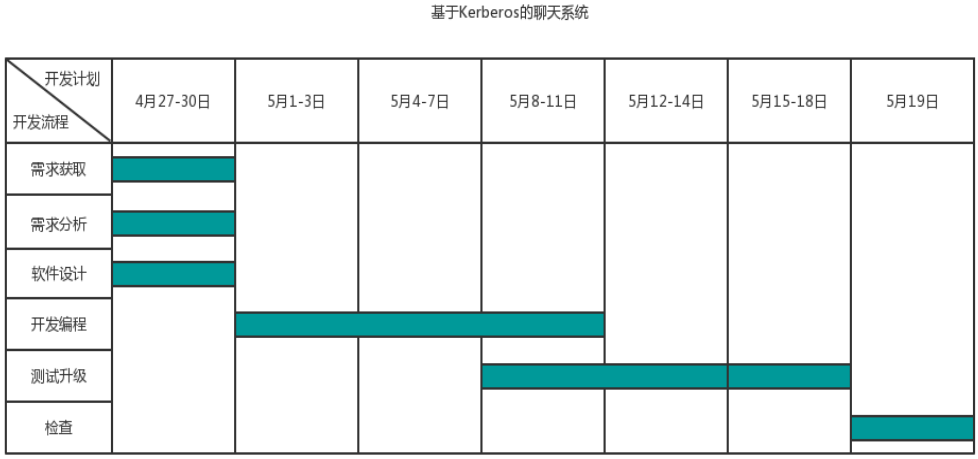


图 六-1 设计甘特图

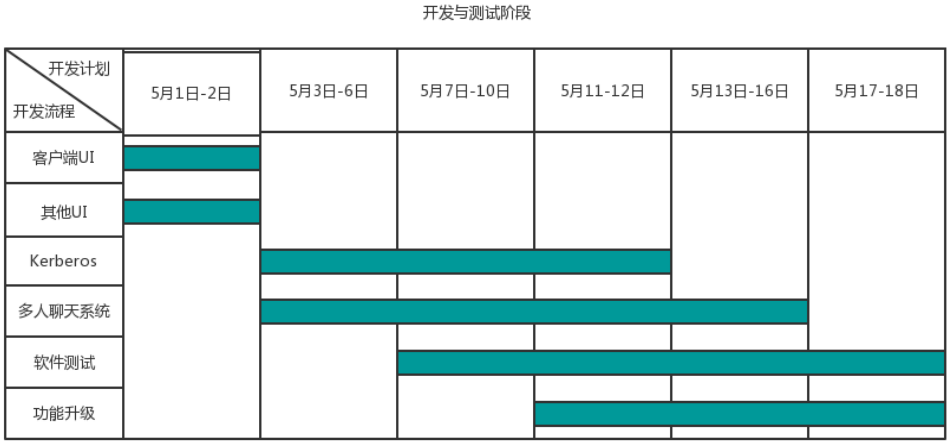


图 六-2 开发甘特图

组长将按照规定时间及时督促组员完成。当出现进度落后问题时，及时与所有人员沟通，安排新的进度安排与计划，争取提前完工。

第七章　部署图

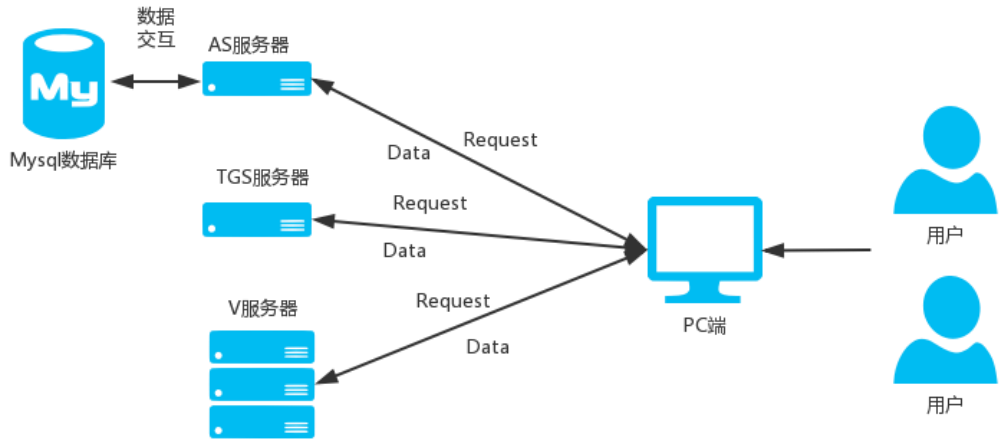


图 七-1 系统部署图

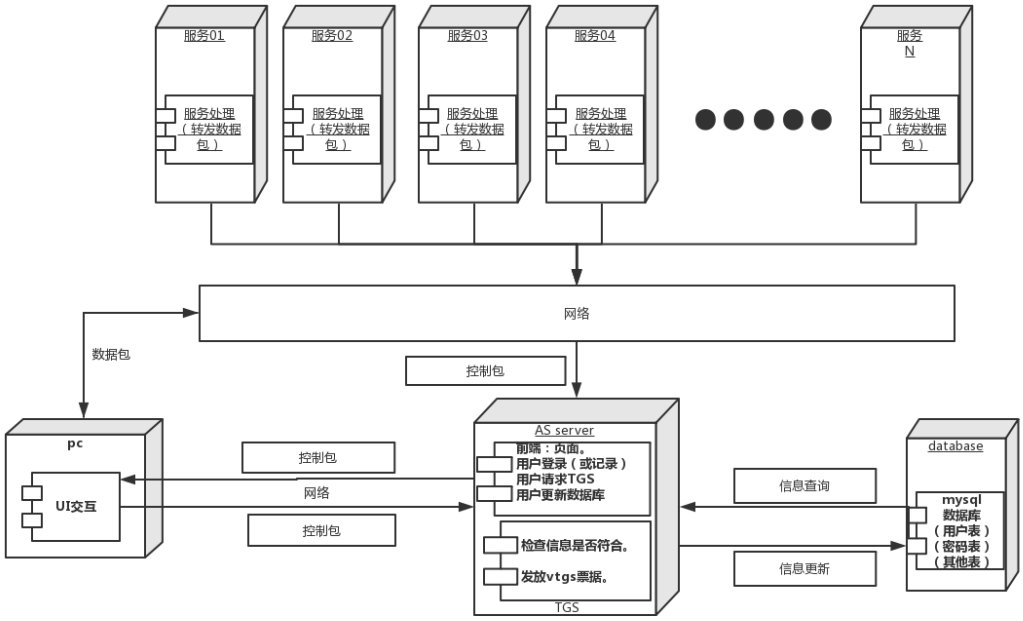


图 七-2 部署图

第八章　数据包规定

8.1.1 概述

长度可以供 64 个包使用，即 2 的 6 次方，6 位。

名称	代号
控制包	1-8
数据包	9-31
控制包	31-64

8.1.2 控制包

名称	信息
1 号	注册使用
代号	1-6 位
用户名	7-26 位
IDc	27-29 位
密码	30-45 位

名称	信息
2 号	AS 返回注册信息
代号	1-6 位
信息	6-9 位

名称	信息
3 号	C-AS
代号	1-6 位
IDv	7-9 位
IDtgs	10-11 位
TS	12-31 位

名称	信息
4 号	AS-C
代号	1-6 位

Kctgs	7-22 位
IDtgs	23-24 位
TS4	25-44 位
Lifetime	45-50 位
Tickettgs	51-112 位

名称	信息
5 号	C-TGS
代号	1-6 位
IDv	7-9 位
Tickettgs	10-71 位
Authenticator	72-109 位

名称	信息
6 号	TGS-C
代号	1-6 位
Kcv	7-22 位
IDv	23-25 位
TS4	26-45 位
Ticketv	46-108 位

名称	信息
7 号	C-V
代号	1-6 位
Ticketv	7-69 位
Authenticator	70-107 位

名称	信息
8 号	V-C
代号	1-6 位
信息	7-9 位

名称	信息
32 号	控制使用
代号	1-6 位
信息	7-9 位

名称	信息
33 号	控制使用
代号	1-6 位
信息	7-9 位

以及其他。

8.1.3 数据包

名称	信息
9 号	聊天
代号	1-6 位
信息	7-200 位

名称	信息
10 号	聊天
代号	1-6 位
信息	7-200 位

名称	信息
11 号	状态信息
代号	1-6 位
信息	7-200 位

以及其他。

8.1.4 数据包格式

源 IP 32 位	目的 IP 32 位	报文类型 （type） 6 位	数据长度 8 位	数据部分 200 位以内

第九章　多线程 socket 运转

§ 9.1 引言

分为 3 个服务器，分别为 AS、TGS、V。

分为 2 个客户端，分别为注册客户端 Register 和 Client。

名称	端口
AS	5432
TGS	5433
V	5434

9.1.1 运转代码结构

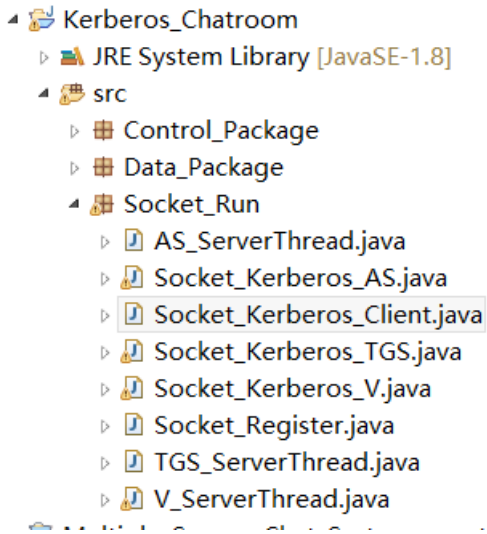


图 九-1 代码结构

各个客户端和各种服务器如图所示。命名方式统一按照模块设计中规定。

它们的建立 socket 方式如下，分别对应服务器和客户端。

9.1.2 代码详解——各个客户端

```
Socket s = new Socket("127.0.0.1", 5432);
OutputStream out = s.getOutputStream();
DataOutputStream dout = new DataOutputStream(out);
dout.writeUTF(str1);
InputStream in = s.getInputStream();
DataInputStream din = new DataInputStream(in);
st = din.readUTF();
```

【功能】：socket 连接服务器，传递加密密文。

【方式】：创建 Socket 对象，传入地址与端口。

```
Socket s = new Socket("127.0.0.1", 5432);
```

定义输出流 OutputStream，数据流 DataOutputStream，使用数据流方法 writeUTF 将字符串发送至服务器。

接收方式定义输入流 OutputStream，数据流 DataOutputStream，使用数据流方法 readUTF 将字符串接收服务器信息。

9.1.3 代码详解——各个服务器（多线程）

```
Socket cs = s.accept();
new ServerThread(cs).start();
System.out.println("接收了 第"+i+"个请求");
InputStream in = sock.getInputStream();
DataInputStream din = new DataInputStream(in);
String name = din.readUTF();
Decipher deph=new Decipher();
System.out.println(name+" 收到!!!!!!!!!!!!!");
name=deph.decipher(name);
OutputStream out = sock.getOutputStream();
DataOutputStream dos = new DataOutputStream(out);
```

【功能】：响应客户端请求，调用解密代码。

【方式】：建立 socket 之后，使用监听方法 Socket cs = s.accept();监听客户端请求。

定义输出流 OutputStream，数据流 DataOutputStream，使用数据流方法 writeUTF 将字符串发送至客户端。

接收方式定义输入流 OutputStream，数据流 DataOutputStream，使用数据流方法 readUTF 将字符串接收客户端信息。

9.1.4 Deg.Log

采用 Deg.Log 函数显示对应的认证流程，封装以及解析包的结果，如果发生异常，即使的能提示问题出在哪儿并报错，这样我们后期可以更加高效的进行调试。

9.1.5 运转结果截图

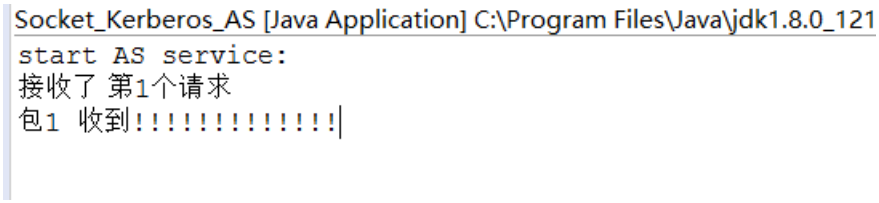


图 九-2 注册结果

图 九-3 client 结果

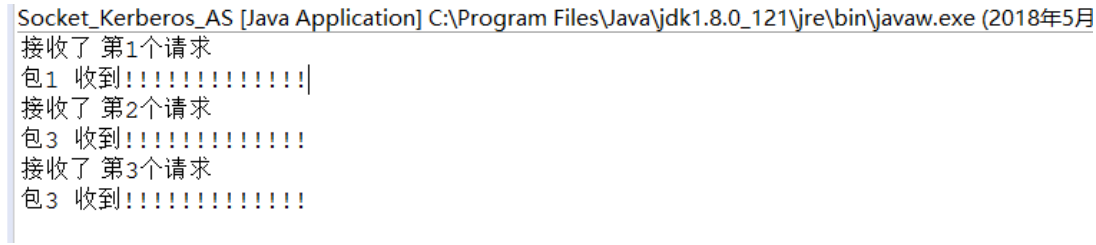


图 九-4 AS 结果

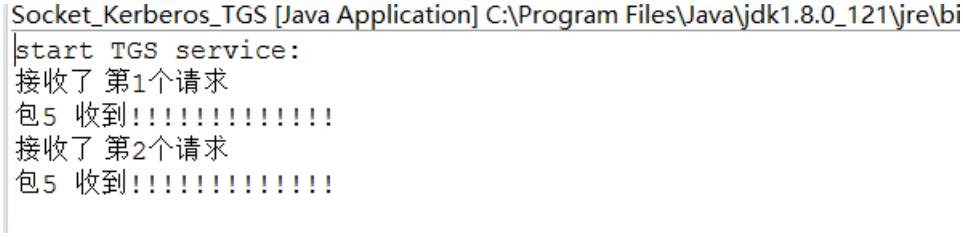


图 九-5 TGS 结果

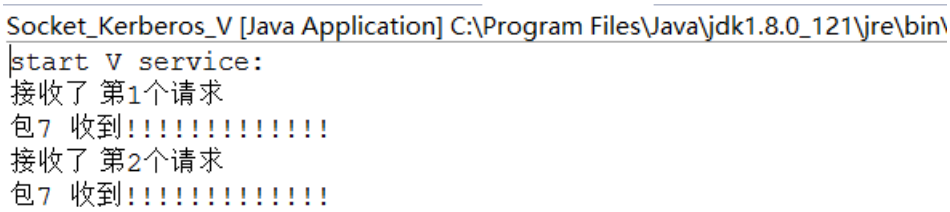

Socket_Kerberos_V [Java Application] C:\Program Files\Java\jdk1.8.0_121\jre\bin\
start V service:
接收了 第1个请求
包7 收到!!!!!!!!!!!!!
接收了 第2个请求
包7 收到!!!!!!!!!!!!!

图 九-6 V 结果