目录

[第一章 需求分析 1](#_Toc74835142)

[1.1目标 1](#_Toc74835143)

[1.2对功能的规定 1](#_Toc74835144)

[1.3对性能的规定 2](#_Toc74835145)

[第二章 系统程序结构设计 3](#_Toc74835146)

[2.1系统部署 3](#_Toc74835147)

[2.2程序流程图 3](#_Toc74835148)

[2.3时序图 4](#_Toc74835149)

[2.4状态机 5](#_Toc74835150)

[第三章 报文设计 6](#_Toc74835151)

[3.1AS收到的包 6](#_Toc74835152)

[3.1.1 C🡪AS登录包（PackageCtoAsLogin） 6](#_Toc74835153)

[3.1.2 C🡪AS注册包（PackageCtoAsRegister） 6](#_Toc74835154)

[3.1.3 C🡪AS修改密码包（PackageCtoAsModify） 7](#_Toc74835155)

[3.1.4 C🡪AS查询收件人是否存在（PackageCtoAsCheck） 7](#_Toc74835156)

[3.2 客户端C收到的包 8](#_Toc74835157)

[3.2.1 AS🡪C认证包（PackageAstoCAuth） 8](#_Toc74835158)

[3.2.2 AS🡪C注册返回状态包（PackageAstoCRegister） 9](#_Toc74835159)

[3.2.3 AS🡪C 登录失败返回状态包（PackageAstoCLogin） 10](#_Toc74835160)

[3.2.4 AS🡪C查询收件人返回包（PackageAstoCCheck） 10](#_Toc74835161)

[3.2.5 TGS🡪C认证包（PackageTgstoC） 10](#_Toc74835162)

[3.2.6 V🡪C收邮件包（PackageVtoCPost） 11](#_Toc74835163)

[3.2.7 V🡪C认证包（PackageVtoCAuth） 13](#_Toc74835164)

[3.2.8 AS🡪C修改密码状态返回包（PackageAstoCModify） 13](#_Toc74835165)

[3.2.9 V🡪C空邮件状态返回包（PackageVtoCPostStatus） 14](#_Toc74835166)

[3.3 TGS收到的包 14](#_Toc74835167)

[3.3.1 C🡪TGS认证包（PackageCtoTgs） 14](#_Toc74835168)

[3.4V收到的包 15](#_Toc74835169)

[3.4.1C🡪V认证包（PackageCtoVAuth） 15](#_Toc74835170)

[3.4.2C🡪V发邮件包（PackageCtoVPost） 16](#_Toc74835171)

[3.4.3C🡪V请求邮件包（PackageCtoVRequestMail） 18](#_Toc74835172)

[第四章 模块设计 19](#_Toc74835173)

[4.1客户端模块 19](#_Toc74835174)

[4.1.1客户端模块设计 19](#_Toc74835175)

[4.1.2 LogRegWindow类设计 19](#_Toc74835176)

[4.1.3 MainWindow类设计 26](#_Toc74835177)

[4.1.4 ModifyPasswordWindow设计 31](#_Toc74835178)

[4.2 AS服务器模块 33](#_Toc74835179)

[4.2.1 AS服务器模块设计 33](#_Toc74835180)

[4.2.2 AS类设计 34](#_Toc74835181)

[4.3 TGS服务器模块 39](#_Toc74835182)

[4.3.1 TGS服务器模块设计 39](#_Toc74835183)

[4.3.2 TGS类设计 39](#_Toc74835184)

[4.4邮件服务器模块 41](#_Toc74835185)

[4.4.1邮件服务器模块设计 41](#_Toc74835186)

[4.4.2 EmailServer类设计 41](#_Toc74835187)

[4.5 Package模块 45](#_Toc74835188)

[4.5.1 Package类设计 45](#_Toc74835189)

[4.5.2 Ticket类设计 50](#_Toc74835190)

[4.5.3 Authenticator类设计 51](#_Toc74835191)

[4.6 DES模块 51](#_Toc74835192)

[4.7日志模块 51](#_Toc74835193)

[4.8数据库模块 52](#_Toc74835194)

# 第一章 需求分析

## 1.1目标

利用kerberos网络认证协议实现网络环境下客户及、服务器的安全认证以及邮件的收发。

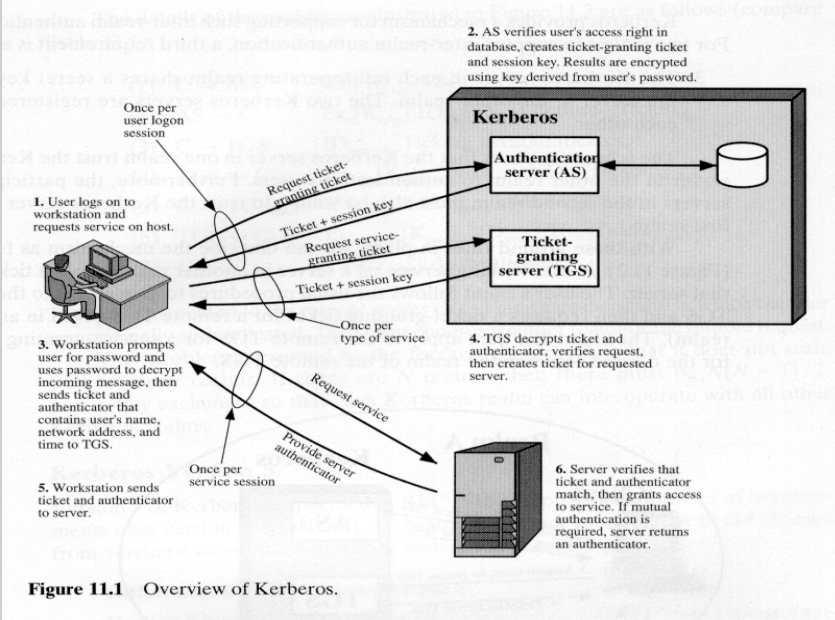


Figure 1

## 1.2对功能的规定

A．用户可以实现登录注册功能。在这里要声明的登录和注册时分开的两个操作，登录是输入我的账号以及密码即可享受服务，注册时要先输入账号以及密码之后重返页面再次进行登录操作才可以享受服务。

B．支持多个用户同时收发邮件。因为现实生活中不可能说有两个人同时给一个人发邮件，另一个人要等这一个人发完之后才能接着发送，所以在本次的程序中，邮件的收发一定是要支持多用户同时收发的，这样才能做到完整的模仿这样的一个系统。

C．发邮件是可以在文本框输入正文的内容，也可以添加附件。这一点是通过QQ邮箱的启发，通过QQ邮箱我们就可以很明确的指出在本次的邮箱收发我们所需要的界面设计，添加的附件可以兼容多种格式，不用担心发不出去。

D．用户可以查看已经发送的邮件和收到的邮件。这一点的功能所属于附加功能，每个用户发出的邮件以及收到的邮件全部都会通过程序保存在本地的一个文件夹里面，以供所需要时进行查看。

## 1.3对性能的规定

A．服务器是并发的，支持四台以及以上的机器同时登录并进行邮件的收发。这一个性能与上面所说的同时首发邮件有异曲同工之妙，在这里只是说的更加的具体，我们要实现对邮件收发的并发就一定要去实现服务器的并发，因为我们的邮件都是发送到服务器上之后再传输到收件人的手里，恰好小组里有四个人，所以说并发的最低要求就是四台电脑可以同时进行邮件的发送以及收取。

B.服务器对客户端的响应时间应该在用户可以接受的范围之内。这句话通俗的来讲就是我们不能让用户觉得这样的一个邮件收发系统不好用，太慢了，也就是说邮件的发送以及接收是必须在一段时间内完成的，这一段是不能大于用户的等待耐心时间的，假如大于了用户等待耐心时间，是会影响用户体验从而导致整个邮件收发系统的一个不成功，所以说邮件收发一定要快，在力所能及的时间内完成即可。

# 第二章 系统程序结构设计

## 2.1系统部署

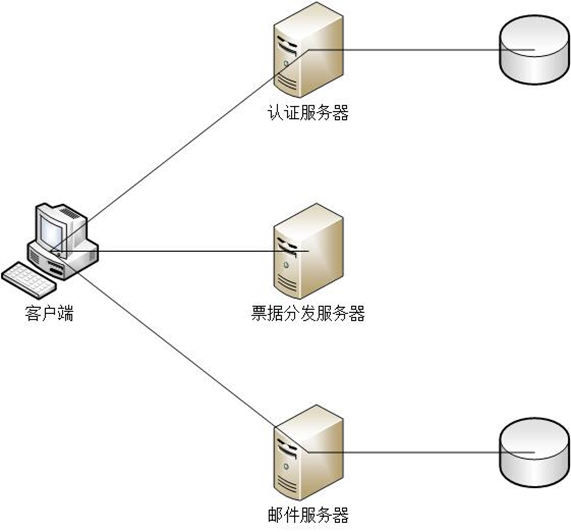


Figure 2

## 2.2程序流程图

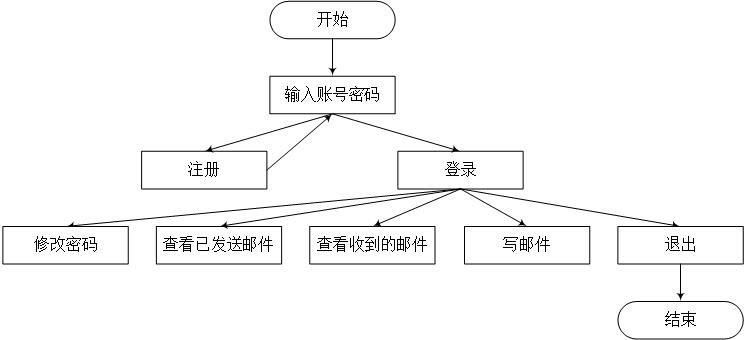


Figure 3

## 2.3时序图

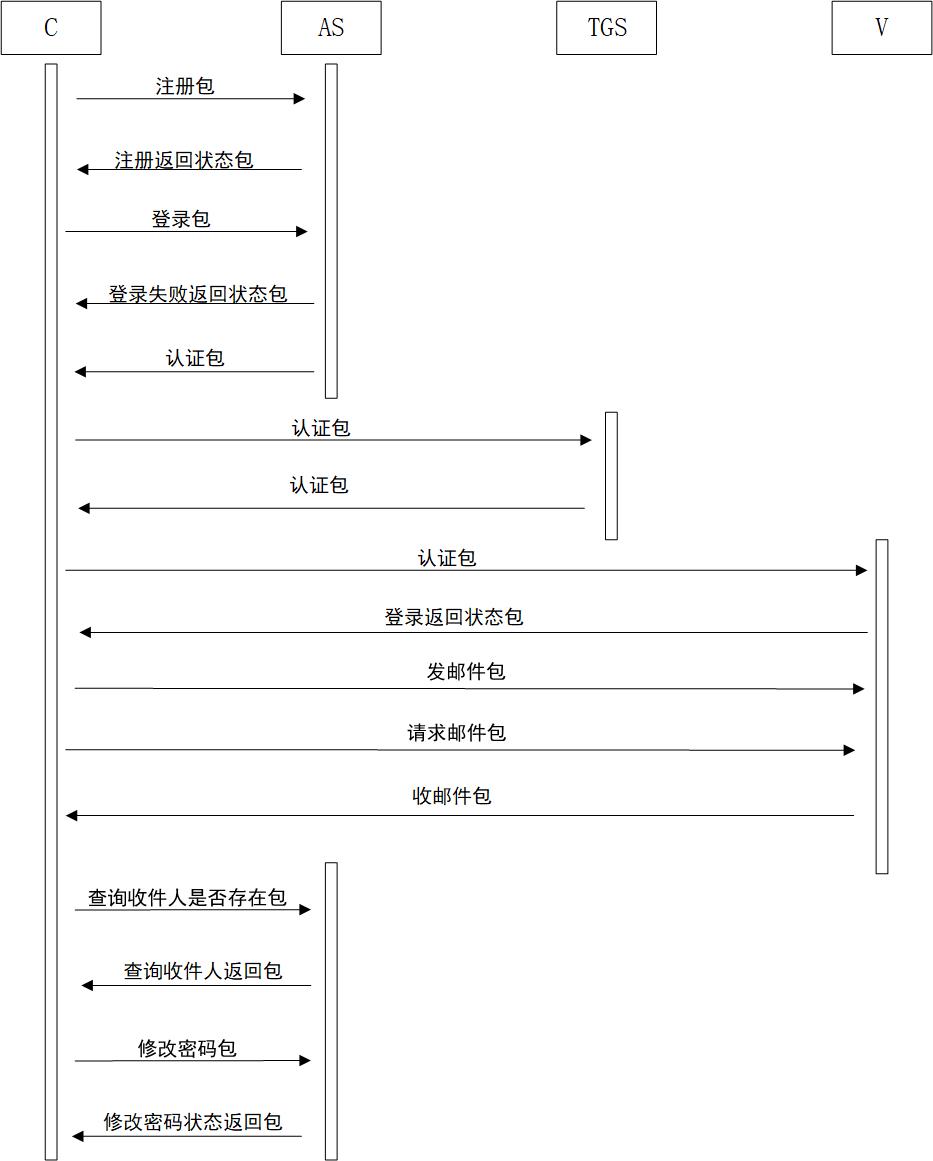


Figure 4

## 2.4状态机

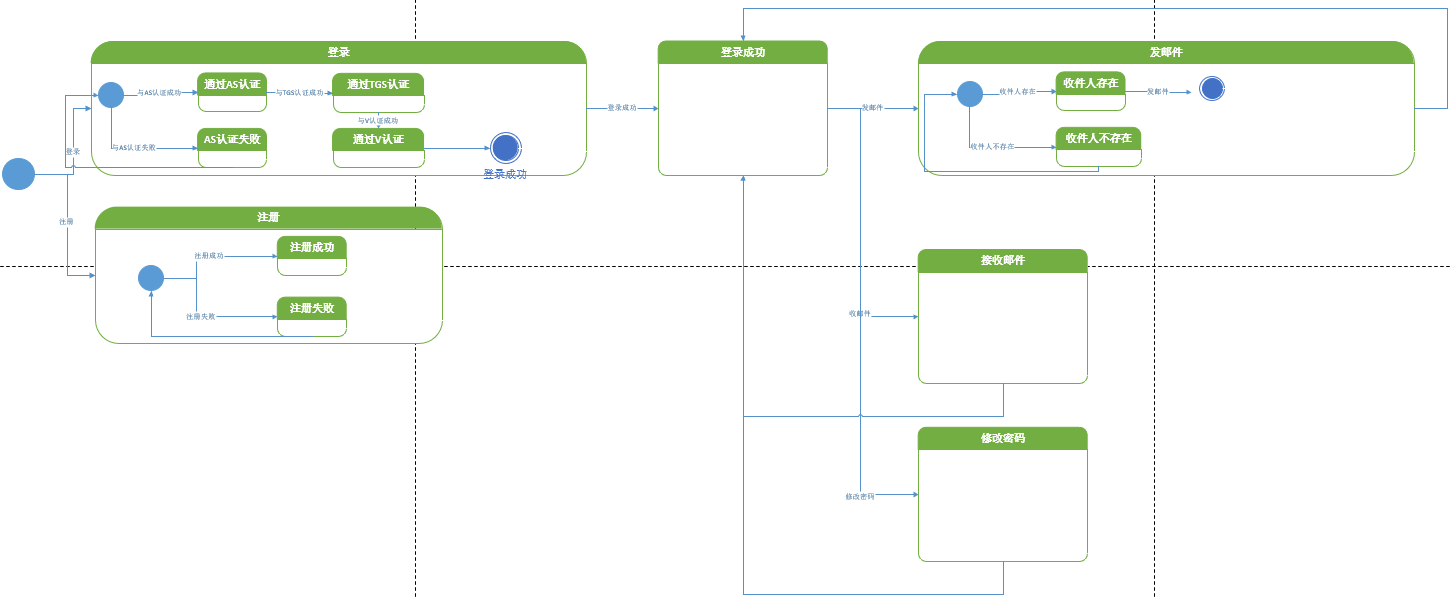


Figure 5

# 第三章 报文设计

## 3.1AS收到的包

3.1.1 C🡪AS登录包（PackageCtoAsLogin）

第一个部分状态位：长度 1个字节，里面内容是在这个包里状态位置为1，其意思是用来区分不同的包。

第二个部分IDc：长度为10个字节，里面内容是用户所拥有的账号。

第三个部分IDtgs：长度为4个字节，里面内容为TGS的IP地址。

第四个部分TS1：长度为6个字节，里面内容是时间戳，具体而言就是13位十进制数，用来表示时间。

对于登录和注册我们也有所区分：对于登录而言，我们采用一个字节数1来表示，对于注册来说，我们采用一个字节数0来表示。

冗余位：10字节，该部分存在的用处是为了方便扩展功能，以后有什么需要添加的就可以添加在其中。

内容：最大长度为16字节，相对应的也就是十位字符。

图表标识如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 状态位 | Ekc(报文)，Kc为C和AS之间的对称钥，默认已经提前分发 | | | | |
|  |  | IDc | IDtgs | TS1 | 冗余位 | 内容 |
| 长度 | 1字节 | 10字节 | 4字节 | 6字节 | 10字节 | 16字节 |
| 说明 | 在这个包里状态位置为1 | 账号十位（10字节） | TGS IP地址 | 时间戳（13位十进制） | 方便扩展功能 | Md5（密码）（16字节） |

3.1.2 C🡪AS注册包（PackageCtoAsRegister）

第一个部分状态位：长度为1个字节，里面内容是在这个包里状态位置1，用来表示不同的包，在这里表示注册。

第二个部分IDc：长度为10个字节，里面内容是用户所拥有的账号。

第三个部分TS1：长度为6个字节，里面内容是时间戳，具体而言就是13位十进制数，用来表示时间。

对于登录和注册我们也有所区分：对于登录而言，我们采用一个字节数1来表示，对于注册来说，我们采用一个字节数0来表示。

冗余位：10字节，该部分存在的用处是为了方便扩展功能，以后有什么需要添加的就可以添加在其中。

内容：最大长度为10字节，相对应的也就是十位字符。

图表所示如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 状态位 | Ekc(报文)，Kc为C和AS之间的对称钥，默认已经提前分发 | | | |
|  |  | IDc | TS1 | 冗余位 | 内容 |
| 长度 | 1字节 | 10字节 | 6字节 | 10字节 | 10字节 |
| 说明 | 在这个包里状态位置为0 | 账号十位（10字节） | 时间戳（13位十进制） | 方便扩展功能 | 密码十位字符（10字节） |

3.1.3 C🡪AS修改密码包（PackageCtoAsModify）

第一个部分状态位：长度为1个字节，里面内容是在这个包里状态位置2，用来表示不同的包，在这里表示修改密码。

第二个部分IDc：长度为10个字节，里面内容是用户所拥有的账号。

第三个部分TS1：长度为6个字节，里面内容是时间戳，具体而言就是13位十进制数，用来表示时间。

冗余位：10字节，该部分存在的用处是为了方便扩展功能，以后有什么需要添加的就可以添加在其中。

内容：最大长度为32字节，相对应的也就是二十位字符。

图表所示如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 状态位 | Ekc(报文)，Kc为C和AS之间的对称钥，默认已经提前分发 | | | |
|  |  | IDc | TS1 | 冗余位 | 内容 |
| 长度 | 1字节 | 10字节 | 6字节 | 10字节 | 32字节 |
| 说明 | 在这个包里状态位置为2 | 账号十位（10字节） | 时间戳（13位十进制） | 方便扩展功能 | 旧密码十位字符（16字节），新密码十位字符（16字节） |

3.1.4 C🡪AS查询收件人是否存在（PackageCtoAsCheck）

第一个部分状态位：长度为1个字节，里面内容是在这个包里状态位置3，用来表示不同的包，在这里是查询收件人是否存在。

第二个部分IDc：长度为10个字节，里面内容是用户所拥有的账号。

第三个部分TS1：长度为6个字节，里面内容是时间戳，具体而言就是13位十进制数，用来表示时间。

冗余位：10字节，该部分存在的用处是为了方便扩展功能，以后有什么需要添加的就可以添加在其中。

内容：最大长度为10字节，相对应的也就是十位字符。

图表所示如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 状态位 | Ekc(报文)，Kc为C和AS之间的对称钥，默认已经提前分发 | | | |
|  |  | IDc | TS1 | 冗余位 | 内容 |
| 长度 | 1字节 | 10字节 | 6字节 | 10字节 | 10字节 |
| 说明 | 在这个包里状态位置为3 | 账号十位（20字节） | 时间戳（13位十进制） | 方便扩展功能 | 收件人ID |

## 3.2 客户端C收到的包

3.2.1 AS🡪C认证包（PackageAstoCAuth）

第一个部分状态位：长度为一个字节，里面内容是在这个包里状态位置为0，用来表示认证包。

第一个部分Kc,tgs：长度为8个字节，里面内容是AS生成的C与TGS共享的会话密钥，也就是说，这两个相互传递信息的时候，加密通过这样的一个密钥来加密，同样内容也可以用这个密钥来解开。

第二个部分IDtgs：长度为4个字节，，里面内容是TGS的标识，此处选用TGS的IP地址作为其标识。也就是说，这个地方是一个定的不变的一长串内容，里面就是TGS服务器的IP地址。

第三个部分TS2：长度为6个字节，里面内容是时间戳，告诉C票据签发的时间。简单来说就是AS什么时候将这样的一个票据传递给C的。

第四个部分Lifetime2：长度为6个字节，里面内容是票据的有效期，也就是在多长时间以内，我可以直接和TGS进行通信。

最后一个部分Tickettgs= Ektgs[ Kc,tgs|| IDC|| ADC|| IDtgs|| TS2|| Lifetime2 ]

1. Kc,tgs：长度为8个字节，里面内容是AS生成的C与TGS共享的会话密钥，其意思已经在上文进行了解释。
2. IDc：长度为20个字节，里面内容是C的标识，选用C的账号作为其标识。
3. ADc：长度为4个字节，里面内容是C的网络地址，防止非法冒用，此处所说的网络地址是该用户现在所处所使用的机器的IP地址，是可以变换的，不是一串固定的内容。
4. IDtgs：长度为4个字节，里面内容是TGS的标识，此处选用TGS的IP地址作为其标识，其意思已经在上文进行了解释。
5. TS2：长度为6个字节，里面内容是时间戳，告诉C票据签发的时间，其意思已经在上文进行了解释。
6. Lifetime2：长度为6个字节，里面内容是票据的有效期，其意思已经在上文进行了解释。

为了能够更直观的认清每一个部分的作用，我们绘制了一份图表可供参考：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 状态位 | Ekc(报文)，Kc为C和AS之间的对称钥，默认已经提前分发 | | | | | | | | | | |
|  |  | Kc,tgs | IDtgs | TS2 | Lifetime2 | Tickettgs(Ektgs加密报文，ktgs8字节) | | | | | | 冗余位 |
|  |  |  |  |  |  | Kc,tgs | IDc | ADc | IDtgs | TS2 | Lifetime2 |  |
| 长度 | 1字节 | 8字节 | 4字节 | 6字节 | 6字节 | 8字节 | 10字节 | 4字节 | 4字节 | 6字节 | 6字节 | 10字节 |
| 说明 | 在这个包里状态位置为0 | AS生成的C与TGS共享的会话密钥 | TGS的标识，此处选用TGS的IP地址作为其标识 | 时间戳，告诉C票据签发的时间 | 票据的有效期 | AS生成的C与TGS共享的会话密钥 | C的标识，选用C的账号作为其标识 | C的网络地址，防止非法冒用 | TGS的标识，此处选用TGS的IP地址作为其标识 | 时间戳，告诉C票据签发的时间 | 票据的有效期 | 方便扩展功能 |

3.2.2 AS🡪C注册返回状态包（PackageAstoCRegister）

第一个部分状态位，长度为1个字节，里面内容是这个包里状态置为1，来表示这是一个注册状态返回包。

第二个部分注册状态：长度为一个字节，里面内容是0标识注册失败，1标识注册成功。

图表所示如下：

|  |  |  |
| --- | --- | --- |
|  | 状态位 | 注册状态 |
| 长度 | 1字节 | 1字节 |
| 说明 | 这个包里状态位置为1 | 0表示注册失败，1表示注册成功 |

3.2.3 AS🡪C 登录失败返回状态包（PackageAstoCLogin）

第一个部分状态位，长度为1个字节，里面内容是这个包里状态置为2，来表示这是一个登录失败状态返回包。

第二个部分登录状态：长度为一个字节，里面内容是0标识登录失败，成功则不会返回这种类型的包。

|  |  |  |
| --- | --- | --- |
|  | 状态位 | 登录状态 |
|  |  |  |
| 长度 | 1字节 | 1字节 |
| 说明 | 这个包里状态位置为2 | 0表示登录失败 |

3.2.4 AS🡪C查询收件人返回包（PackageAstoCCheck）

第一个部分状态位，长度为1个字节，里面内容是这个包里状态置为3，来表示这是一个查询收件人返回包。

第二个部分是注册状态：长度为一个字节，里面内容是0标识收件人不存在，1标识收件人存在。

图表所示如下：

|  |  |  |
| --- | --- | --- |
|  | 状态位 | 查询状态 |
|  |  |  |
| 长度 | 1字节 | 1字节 |
| 说明 | 这个包里状态位置为3 | 0表示收件人不存在，1表示收件人存在 |

### 3.2.5 TGS🡪C认证包（PackageTgstoC）

第一个部分状态位：长度为1个字节，里面内容是这个包里状态位置为4，标识从TGS传递给C的包。

第二个部分KC,V：长度为8个字节，里面内容是由TGS生成的供C和V之间消息交换的密钥，其意思是C和V的通行所采用的加密解密密钥都是它，是一个对称加密。

第三个部分IDV：长度为4个字节，里面内容是服务器V的标识，选用服务器V的IP地址作为其标识，其意思是因为服务器的IP地址是个定下来不动的一串数字，所以说用其来表示或者是声明我就是服务器。

第四个部分TS4：长度为6个字节，里面内容是时间戳，告诉C票据签发的时间，简单来讲就是什么时候给C的一个票据。

最后一个部分TicketV=EKV[ Kc,tgs || IDC || ADC || IDtgs || TS2 || Lifetime2 ]

1. KC,V：长度为八个字节，里面内容是由TGS生成的供C和V之间消息交换的密钥，其意思已经在上文进行了解释。
2. IDC：长度为10个字节，选用C的账号作为其标识。
3. ADC：长度为4个字节，里面内容是C的网络地址，防止非法冒用。
4. IDV：长度为4个字节，里面内容是服务器V的标识，选用服务器V的IP地址作为其标识，其意思已经在上文进行了解释。
5. TS4：长度为6个字节，里面内容是时间戳，告诉C票据签发的时间，简单来讲就是什么时候给C的一个票据。
6. Lifetime4：长度为6个字节，里面内容是票据的有效期。

图表表示如下：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 状态位 | Ekc,tgs加密报文,Kc,tgs8字节 | | | | | | | | | |
|  |  | Kc,v | IDv | TS4 | Ticketv(Ekv加密报文，Kv8字节) | | | | | | 冗余位 |
|  |  |  |  |  | Kc,v | IDc | ADc | IDv | TS4 | Lifetime4 |  |
| 长度 | 1字节 | 8字节 | 4字节 | 6字节 | 8字节 | 10字节 | 4字节 | 4字节 | 6字节 | 6字节 | 10字节 |
| 说明 | 这个包里状态位置为4 | 由TGS生成的供C和V之间消息交换的密钥 | 服务器V的标识，选用服务器V的IP地址作为其标识 | 时间戳，告诉C票据签发的时间 | 由TGS生成的供C和V之间消息交换的密钥 | C的标识，选用C的账号作为其标识 | C的网络地址，防止非法冒用 | 服务器V的标识，选用服务器V的IP地址作为其标识 | 时间戳，告诉C票据签发的时间 | 票据的有效期 | 方便扩展功能 |

3.2.6 V🡪C收邮件包（PackageVtoCPost）

不难看出和发出邮件包是一样的

第一个部分状态位：长度为1个字节，里面内容是这个包里的状态位置为6，表示是一个发送邮件的包。

第二个部分报文总长度：长度为4个字节，里面内容是整个包的总长度，这个字段主要是为了客户端在接收到服务器端发送的连续的邮件包时能够从连续的包里分割出一个个邮件。

第三个部分Receiver\_ID：长度为10个字节，里面内容是接收者的账号，长度是10位10进制。

第四个部分Sender\_ID：长度为10个字节，里面内容是发送者账号，长度是10位10进制数，跟接收者账号的长度是一样的。

第五个部分TS：长度为6个字节，里面内容是时间戳，里面就是邮件发送的时间，简单来说是什么时候发送的邮件。

第六个部分Subject\_Len：长度为1个字节，里面内容是主题的长度，长度最大为80个中文字符。

第七个部分Content\_Len：长度为2个字节，里面内容是正文的长度最大为21KB个字符。

第八个部分Attachment\_Len：长度为3个字节，里面内容是附件长度，大小可为5MB以内，也就是说5MB以内的文件都可以进行发送。

第九个部分Content：长度不定，里面内容是邮件的内容，里面包含了邮件的所有内容，包括主题，正文以及附加文件。

最后一个部分为Esc（Md5（报文））：长度不定，里面内容是将前面报文经过MD5哈希然后用C的私有密钥生成签名。

图表所示如下：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 状态位 | 报文总长度 | Ekc,v（报文），Kc,v8字节 | | | | | | | | | |
|  |  |  | Receiver\_ID | Sender\_ID | TS | Subject\_Len | Text\_Len | Attachment\_Name | Attachment\_Len | Content | Esc（Md5（报文）） | 冗余位 |
| 长度 | 1字节 | 4字节 | 10字节 | 10字节 | 6字节 | 1字节 | 2字节 |  | 3字节 | 不定 | 不定 | 10字节 |
| 说明 | 这个包里状态位置为6 | 整个包的总长度，方便客户端收到连续的包的报文时分割包 | 接收者的账号，10位十进制 | 发送者的账号，10位十进制 | 时间戳，邮件发送的时间 | 主题长度，最大是2的八次方256字节即256/3≈80个中文字符 | 正文长度，最大是2的16次方64KB即64/3≈21KB字符 |  | 附件长度，最大是2的24次方16M即16/3≈5M字符 | 邮件内容，包括主题、正文、附件 | 将前面报文经过MD5哈希然后用C的私钥生成签名 | 方便扩展功能 |

3.2.7 V🡪C认证包（PackageVtoCAuth）

第一个部分状态位：长度为一个字节，里面内容是在这个包里状态位置为7，代表了C和V认证成功，即登陆成功。

第二个部分TS5+1：长度为6个字节，里面内容是时间戳，是C确定这不是报文的重放，用来确保没有被 攻击。

图表所示如下：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 状态位 |  | |
|  |  | TS5+1 | 冗余位 |
| 长度 | 1字节 | 6字节 | 10字节 |
| 说明 | 在这个包里状态位置为7 | 时间戳，使C确认这不是报文重放 | 方便扩展功能 |

3.2.8 AS🡪C修改密码状态返回包（PackageAstoCModify）

第一个部分状态位：长度为1个字节，里面内容是在这个包里状态位置为8，用来表示不同的包，在这里表示修改密码状态返回包。

第二个部分IDc：长度为10个字节，里面内容是用户所拥有的账号。

第三个部分TS1+1：长度为6个字节，里面内容是TS1+1，具体而言就是13位十进制数，用来验证不是消息重放。

冗余位：10字节，该部分存在的用处是为了方便扩展功能，以后有什么需要添加的就可以添加在其中。

内容：最大长度为10字节，相对应的也就是十位字符。

图表所示如下：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 状态位 | IDc | 修改状态 | TS1+1 | | 冗余位 | | |
|  |  |  |  | |  | |  |
| 长度 | 1字节 |  | 1字节 | | 6字节 | | 10字节 |
| 说明 | 在这个包里状态位置为8 |  | 0代表失败，1代表成功 | | 时间戳（13位十进制） | | 方便扩展功能 |

3.2.9 V🡪C空邮件状态返回包（PackageVtoCPostStatus）

这个包里状态位置为9，表明暂时没有收件人为该客户端的邮件

|  |  |  |  |
| --- | --- | --- | --- |
|  | 状态位 |  | |
|  |  | TS5+1 | 冗余位 |
| 长度 | 1字节 | 6字节 | 10字节 |
| 说明 | 在这个包里状态位置为9 | 时间戳，使C确认这不是报文重放 | 方便扩展功能 |

## 3.3 TGS收到的包

### 3.3.1 C🡪TGS认证包（PackageCtoTgs）

第一个部分IDV：长度为四个字节，里面内容是服务器V的标识。选用服务器V的IP地址作为其标识。

第二个部分Tickettgs=EKtgs[ Kc,tgs || IDC || ADC || IDtgs || TS2 || Lifetime2 ]

1. Kc,tgs：长度为8个字节，里面内容是AS生成的C与TGS共享的会话密钥，其意思在上文已经进行了解释。
2. IDc：长度为10个字节，里面内容是C的标识，选用C的账号作为其标识，是一个固定的数据，为用户一开始创建的账号。
3. ADc：长度为4个字节，里面内容是C的网络地址，防止非法冒用，如果这个地方的内容在一段时间的内的会话中改变了，我们就可以推断是否有攻击者的出现。
4. IDtgs：长度为4个字节，里面内容是TGS的标识，此处选用TGS的IP地址作为其标识，其意思在上文已经进行了解释。
5. TS2：长度为6个字节，里面内容是告诉C票据签发的时间。
6. Lifetime2：长度为6个字节里面内容是票据的有效期。

第三个部分是Authenticator：Ekc,tgs加密报文，Ekc,tgs8字节。

1. IDc：长度为10个字节，里面内容是C的标识，选用C的账号作为其标识。
2. ADc：长度为四个字节，里面内容是C的网络地址，防止非法冒用，其意思已经在上文进行了解释。
3. TS3：长度为6个字节，里面内容是C生成的时间戳，其意思就是C发送这个包的时候的具体时间。

为了更方便的理解，我们同样对此进行了图表的绘制：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | IDv | Tickettgs（Ektgs加密报文，Ektgs8字节） | | | | | | Authenticatorc（Ekc,tgs加密报文，Ekc,tgs8字节） | | | 冗余位 |
|  |  | Kc,tgs | IDc | ADc | IDtgs | TS2 | Lifetime2 | IDc | ADc | TS3 |  |
| 长度 | 4字节 | 8字节 | 10字节 | 4字节 | 4字节 | 6字节 | 6字节 | 10字节 | 4字节 | 6字节 | 10字节 |
| 说明 | 服务器V的标识，选用服务器V的IP地址作为其标识 | AS生成的C与TGS共享的会话密钥 | C的标识，选用C的账号作为其标识 | C的网络地址，防止非法冒用 | TGS的标识，此处选用TGS的IP地址作为其标识 | 时间戳，告诉C票据签发的时间 | 票据的有效期 | C的标识，选用C的账号作为其标识 | C的网络地址，防止非法冒用 | C生成的时间戳 | 方便扩展功能 |

## 3.4V收到的包

3.4.1C🡪V认证包（PackageCtoVAuth）

第一个部分状态位：长度为一个字节，里面内容是这个包里状态位置为0，表示的就是一个认证包。

第二个部分TicketV=EKV[ KC,V || IDC || ADC || IDV || TS4 || Lifetime4 ]

1. KC,V：长度为八个字节，里面内容是由TGS生成的供C和V之间消息交换的密钥。
2. IDC：长度为10个字节，里面内容是C的网络地址，选用C的账号作为其标识。
3. ADC：长度为4个字节，里面内容是C的网络地址，防止非法冒用。
4. IDV：长度为4个字节，里面内容是服务器V的标识，选用服务器V的IP地址作为其标识。
5. TS4：长度为6个字节，里面内容是时间戳，告诉C票据签发的时间，简单来讲就是什么时候给C的一个票据。

第三个部分Authenticatorc：Ekc,tgs加密报文，Ekc,tgs8字节。

1. IDc：长度为10个字节，里面内容是C的标识，选用C的账号作为其标识。
2. ADc：长度为四个字节，里面内容是C的网络地址，防止非法冒用，其意思已经在上文进行了解释。
3. TS3：长度为6个字节，里面内容是C生成的时间戳，其意思就是C发送这个包的时候的具体时间。

图表所示如下：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 状态位 | Ticketv(Ekv加密报文，Kv8字节) | | | | | | Authenticatorc（Ekc,v加密报文，kc,v8字节） | | | 冗余位 |
|  |  | Kc,v | IDc | ADc | IDv | TS4 | Lifetime4 | IDc | ADc | TS5 |  |
| 长度 | 1字节 | 8字节 | 10字节 | 4字节 | 4字节 | 6字节 | 6字节 | 10字节 | 4字节 | 6字节 | 10字节 |
| 说明 | 这个包里状态位置为0 | 由TGS生成的供C和V之间消息交换的密钥 | C的标识，选用C的账号作为其标识 | C的网络地址，防止非法冒用 | 服务器V的标识，选用服务器V的IP地址作为其标识 | 时间戳，告诉C票据签发的时间 | 票据的有效期 | C的标识，选用C的账号作为其标识 | C的网络地址，防止非法冒用 | C生成的时间戳 | 方便扩展功能 |

3.4.2C🡪V发邮件包（PackageCtoVPost）

1.第一个部分状态位：长度为1个字节，里面内容是这个包里的状态位置为1，表示是一个发送邮件的包。

2.第二个部分报文总长度：长度为4个字节，里面内容是整个包的总长度，这个字段主要是为了客户端在接收到服务器端发送的连续的邮件包时能够从连续的包里分割出一个个邮件。

3. 第三个部分Ticketv。

4.第四个部分Receiver\_ID：长度为10个字节，里面内容是接收者的账号，长度是10位10进制。

5.第五个部分Sender\_ID：长度为10个字节，里面内容是发送者账号，长度是10位10进制数，跟接收者账号的长度是一样的。

6.第六个部分TS：长度为6个字节，里面内容是时间戳，里面就是邮件发送的时间，简单来说是什么时候发送的邮件。

7.第七个部分Subject\_Len：长度为1个字节，里面内容是主题的长度，长度最大为80个中文字符。

8.第八个部分Content\_Len：长度为2个字节，里面内容是正文的长度最大为21KB个字符。

9.第九个部分Attachment\_Len：长度为3个字节，里面内容是附件长度，大小可为5MB以内，也就是说5MB以内的文件都可以进行发送。

10.第十个部分Content：长度不定，里面内容是邮件的内容，里面包含了邮件的所有内容，包括主题，正文以及附加文件。

11.最后一个部分为Esc（Md5（报文））：长度不定，里面内容是将前面报文经过MD5哈希然后用C的私有密钥生成签名。

表格 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 状态位 | 报文总长度 | Ticketv(Ekv加密报文，Kv8字节) | Ekc,v（报文），Kc,v8字节 | | | | | | | | | | |
|  |  |  |  |  | Receiver\_ID | Sender\_ID | TS | Subject\_Len | Text\_Len | Attachment\_Name | Attachment\_Len | Content | Esc（Md5（报文）） | 冗余位 |
| 长度 | 1字节 | 4字节 |  |  | 10字节 | 10字节 | 6字节 | 1字节 | 2字节 |  | 3字节 | 不定 | 不定 | 10字节 |
| 说明 | 这个包里状态位置为6 | 整个包的总长度，方便客户端收到连续的包的报文时分割包 |  |  | 接收者的账号，10位十进制 | 发送者的账号，10位十进制 | 时间戳，邮件发送的时间 | 主题长度，最大是2的八次方256字节即256/3≈80个中文字符 | 正文长度，最大是2的16次方64KB即64/3≈21KB字符 |  | 附件长度，最大是2的24次方16M即16/3≈5M字符 | 邮件内容，包括主题、正文、附件 | 将前面报文经过MD5哈希然后用C的私钥生成签名 | 方便扩展功能 |

### 3.4.3C🡪V请求邮件包（PackageCtoVRequestMail）

1..第一个部分状态位：长度为1个字节，里面内容是在这个包里状态位置为2，用来表示不同的包，在这里是客户端向服务器请求接收邮件。

2.第二个部分IDc：长度为10个字节，里面内容是用户所拥有的账号。

3.第三个部分Ticketv。

4.第四个部分TS1：长度为6个字节，里面内容是时间戳，具体而言就是13位十进制数，用来表示时间。

5．冗余位：10字节，该部分存在的用处是为了方便扩展功能，以后有什么需要添加的就可以添加在其中。

6．内容：最大长度为20字节，相对应的也就是十位字符。

图表所示如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 状态位 | Ticketv(Ekv加密报文，Kv8字节) | Ekc，v(报文)，Kc，v为C和V之间的对称钥，由TGS分发 | | | |
|  |  |  |  | IDc | TS1 | 冗余位 |
| 长度 | 1字节 |  |  | 10字节 | 6字节 | 10字节 |
| 说明 | 在这个包里状态位置为2 |  |  | 账号十位（20字节） | 时间戳（13位十进制） | 方便扩展功能 |

# 第四章 模块设计

## 4.1客户端模块

4.1.1客户端模块设计

客户端模块设计时以UI为主导，根据UI设计，客户端会有三个不同的界面，每个界面都需要单独的类来实现UI的生成和事件的响应。

这三个界面类分别为登录注册界面类LogRegWindow、主界面类MainWindow以及修改个人信息界面（目前仅修改密码）类ModifyWindow。

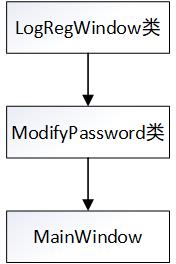


Figure 6

4.1.2 LogRegWindow类设计

**登录注册类LogRegWindow成员变量**：

private TextField account：登录注册界面输入账号的控件

private PasswordField password：登录注册界面输入密码的控件

private Button loginButton：登录注册界面的登录按钮

private Button registerButton：登陆注册界面的注册按钮

private String IP\_AS：AS服务器的IP地址

private String IDtgs：TGS服务器的IP地址

private String IDv：邮件服务器的IP地址

private String KeyCAS：客户端和AS服务器之间的对称密钥

private String KeyCTgs：AS生成的客户端和TGS服务器之间的会话密钥

private String KeyCV：TGS生成的客户端和邮件服务器之间的会话密钥

private TicketTgs ticketTgs：客户端用来访问TGS服务器的票据

private TicketV ticketV：客户端用来访问邮件服务器的票据

private String TS5：客户端和邮件服务器相互认证的时间戳

**登录注册类LogRegWindow成员函数**：

public void start(Stage primaryStage)：生成登陆注册界面的函数

public void register（）：注册按钮事件的响应函数

public void login（）：登录按钮事件的响应函数

public byte[] cConnectAS（byte[] packageToAS）：客户端和AS服务器通信的函数

public byte[] cConnectTgs（byte[] packageToTgs）：客户端和TGS服务器通信的函数

public byte[] cConnectV（byte[] packageToV）：客户端和邮件服务器V通信的函数

public byte[] packCtoASRegister（String IDc，String password）：把客户端发给AS服务器的注册包封装并加密的函数

public byte[] packCtoASLogin（String IDc，String password）：把客户端发给AS服务器的登录包封装并加密的函数

public byte[] packCtoTgs（）：把客户端发给TGS服务器的包封装的函数

public byte[] packCtoV（）：把客户端发给邮件服务器V的包封装的函数

public void unpackAStoCAuth（byte[] packageToC）：把AS服务器发给客户端的认证包解密并解包的函数

public void unpackTgstoC（byte[] packageToC）：把TGS服务器发给客户端的包解密并解包的函数

public void unpackVtoC（byte[] packageToC）：把邮件服务器V发给客户端的包解密并解包的函数

public byte[] authentic（int choice）：把Authenticator封装并加密的函数（choice为1代表用KeyCTgs加密，choice为2代表用KeyCV加密）

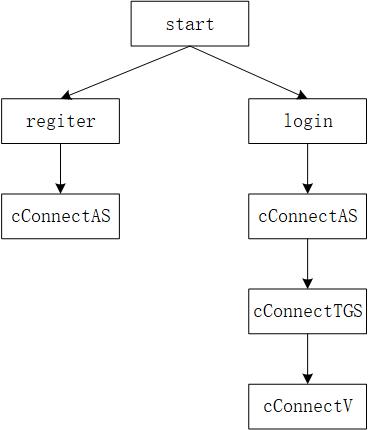


Figure 7

#### 4.1.2.1 start函数

**函数声明**：public void start(Stage primaryStage)

**函数简介**：该函数为生成登陆注册界面的函数，也是客户端程序启动的首界面。函数的返回值为void，传入参数为Stage类型的参数primaryStage（来自javafx包内的数据类型，表示界面）。

**函数实现**：本次实验使用javafx实现UI，所以生成界面时从编写的fxml文件读取并加载界面，通过调用FXMLLoader.load(getClass().getResource("xxx.fxml"))实现。

然后通过primaryStage的setTitle方法设置登录注册界面的title，通过primaryStage的setScene方法设置界面的长宽等参数。

最后调用primaryStage的show方法展示界面，即生成了登录注册界面。

#### 4.1.2.2 register函数

**函数声明**：public void register（）

**函数简介**：该函数是登录注册界面注册按钮的事件响应函数，在登录注册界面点击注册会调用该函数。函数的返回值为void，无传入参数。

**函数实现**：首先从UI控件中获取用户输入的账号和密码，由于成员变量TextField account、PasswordField password已分别和输入账号的文本框、输入密码的密码框控件绑定，所以通过成员变量account和password的getText方法即可得到用户输入的账号和密码。

第二步判断用户输入的账号密码是否符合规范，即账号是否为10位数的字母和数字的组合，密码长度是否大于6位，如果不符合规范弹窗提示错误，如果符合规范继续下面的步骤。

第三步调用函数packCtoASRegister（String IDc，String password）封装客户端发给AS服务器的包，并将封装之后的包加密（除第一位状态位之外均加密）。

第四步调用cConnectAS（byte[] packageCtoAS）函数，和AS服务器建立连接，把加密之后的包的byte数组发给AS，并接收AS的返回包，接收之后断开和AS的连接。

第五步查看AS返回包的注册状态位，判断是否注册成功，注册状态位为0代表注册失败，注册状态位为1代表注册成功，并弹窗提示成功/失败。

#### 4.1.2.3 login函数

**函数声明**：public void login（）；

**函数简介**：该函数是登录注册界面登录按钮的事件响应函数，在登录注册界面点击登录会调用该函数。函数的返回值为void，无传入参数；

**函数实现**：首先从UI控件中获取用户输入的账号和密码，由于成员变量TextField account、PasswordField password已分别和输入账号的文本框、输入密码的密码框控件绑定，所以通过成员变量account和password的getText方法即可得到用户输入的账号和密码。

第二步判断用户输入的账号密码是否符合规范，即账号是否为10位数的字母和数字的组合，密码长度是否大于6位，如果不符合规范弹窗提示错误，如果符合规范继续下面的步骤。

第三步调用函数packCtoASLogin（String IDc，String password）封装客户端发给AS服务器的包，并将封装之后的包加密（除第一位状态位之外均加密）

第四步调用cConnectAS（byte[] packageToAS）函数，和AS服务器建立连接，把加密之后的包的byte数组发给AS，并接收AS的返回包，接收之后断开和AS的连接。

第五步查看AS返回包的状态位，判断是否登录认证成功，状态位为0代表认证成功，接收到的是AS发给C的认证包，并继续执行第六步；状态位为2代表认证失败，并弹窗提示失败。

第六步调用函数unpackAStoCAuth（byte[] packageToC）将收到的包解密（除状态位）、解包，得到PackageAStoCAuth结构体类的一个对象，并获取下一步需要的字段：客户端和TGS服务器的会话密钥KeyCTgs、TGS服务器的IP地址IDtgs、TicketTgs结构体类型的ticketTgs，以上字段均为类的成员变量。

第七步调用函数authentic（int choice）生成Authenticator结构体类型的对象，使用该类的pack方法转为byte数组，再使用密钥KeyCTgs加密。

第八步调用函数packCtoTgs（）封装客户端发给TGS服务器的包并转为byte数组。

第九步调用函数cConnectTgs（byte[] packageToTgs）和TGS服务器建立连接，把上一步生成的byte数组发给TGS，并接收TGS的返回包，接收之后断开和AS的连接。

第十步调用函数unpackTgstoC（byte[] packageToC）将收到的包解密（除状态位）、解包，得到PackageTgstoC结构体类的一个对象，并获取下一步需要的字段：客户端和邮件服务器的会话密钥KeyCV、邮件服务器的IP地址IDv、TicketV结构体类型的ticketV，以上字段均为类的成员变量。

第十一步调用函数authentic（int choice）生成Authenticator结构体类型的对象，使用该类的pack方法转为byte数组，再使用密钥KeyCV加密。

第十二步调用函数packCtoV（）封装客户端发给邮件服务器V的包并转为byte数组。

第十三步函数cConnectTgs（byte[] packageToV）和邮件服务器建立连接，把上一步生成的byte数组发给邮件服务器V，并接收邮件服务器V的返回包，接收之后断开和V的连接。

第十四步调用函数unpackVtoC（byte[] packageToC）将收到的包解密、解包，并获取字段字段TS5+1。

第十五步验证TS5+1的值，认证成功则调用主界面类的构造函数进入主界面。

#### 4.1.2.4 cConnectAS函数

**函数声明**：public byte[] cConnectAS（byte[] packageToAS）

**函数简介**：该函数是负责客户端和AS服务器通信的函数，在登录和注册时都会调用该函数。函数的返回值为byte[]，即AS发回的数据包，函数的参数为byte[] packageToAS，即加密之后的需要发给AS的包。

**函数实现**：首先向AS服务器发起Socket连接请求，等待AS服务器接收请求。

AS服务器接收请求之后，将传入的参数，即加密之后的需要发给AS的包发给AS。

保持连接，等待AS的回应包。

接收到AS的回应包之后断开连接，将返回的包转为byte数组返回。

#### 4.1.2.5 cConnectTgs函数

**函数声明**：public byte[] cConnectTgs（byte[] packageToTgs）

**函数简介**：该函数是负责客户端和TGS服务器通信的函数，在登录时会调用该函数。函数的返回值为byte[]，即TGS发回的数据包，函数的参数为byte[] packageToTGS，即需要发给TGS的包。

**函数实现**：先向TGS服务器发起Socket连接请求，等待TGS服务器接收请求。

TGS服务器接收请求之后，将传入的参数，即加密之后的需要发给TGS的包发给TGS。

保持连接，等待TGS的回应包。

接收到TGS的回应包之后断开连接，将返回的包转为byte数组返回。

#### 4.1.2.6 cConnectV函数

**函数声明**：public byte[] cConnectV（byte[] packageToV）

**函数简介**：该函数是负责客户端和邮件服务器V通信的函数，在登录时会调用该函数。函数的返回值为byte[]，即邮件服务器V发回的数据包，函数的参数为byte[] packageToV，即需要发给V的包。

**函数实现**：先向邮件服务器V发起Socket连接请求，等待邮件服务器V接收请求。

邮件服务器V接收请求之后，将传入的参数，即需要发给邮件服务器V的包发给邮件服务器V。

保持连接，等待邮件服务器V的回应包。

接收到邮件服务器V的回应包之后断开连接，将返回的包转为byte数组返回。

#### 4.1.2.7 packCtoASRegister函数

**函数声明**：public byte[] packCtoASRegister（String IDc，String password）

**函数简介**：该函数是把客户端发给AS服务器的注册包封装并加密的函数，在注册时会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数的传入参数为String IDc，String password，即从控件获取的，需要在包里封装的数据。

**函数实现**：首先实例化PackageCtoASRegister结构体类，即PackageCtoASRegister类的IDc字段为用户输入的账号、TS1为当前的时间戳、报文内容为用户输入的密码的MD5 Hash值。

调用PackageCtoASRegister对象的pack方法将报文封装转为byte数组。

然后调用DES加密模块的加密函数将报文加密，加密密钥为客户端和AS服务器拥有的对称密钥（即成员变量KeyCAS），最后得到加密以后的byte数组。

#### 4.1.2.8 packCtoASLogin函数

**函数声明**：public byte[] packCtoASLogin（String IDc，String password）

**函数简介**：该函数是把客户端发给AS服务器的登录包封装并加密的函数，在登录时会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数的传入参数为String IDc，String password，即从控件获取的，需要在包里封装的数据。

**函数实现**：首先实例化PackageCtoASLogin结构体类，即PackageCtoASLogin类的IDc字段为用户输入的账号、IDtgs为本类的成员变量、TS1为当前的时间戳、报文内容为用户输入的密码的MD5 Hash值。

调用PackageCtoASLoginr对象的pack方法将报文封装转为byte数组。

然后调用DES加密模块的加密函数将报文加密，加密密钥为客户端和AS服务器拥有的对称密钥（即成员变量KeyCAS），最后得到加密以后的byte数组。

#### 4.1.2.9 packCtoTgs函数

**函数声明**：public byte[] packCtoTgs（）

**函数简介**：该函数是把客户端发给TGS服务器的包封装的函数，在登录时会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数无传入参数

**函数实现**：首先实例化PackageCtoTgs结构体类，即给PackageCtoTgs结构体类的每个字段赋值，PackageCtoTgs类的字段ticketTgs、authticator均为本类的成员变量。

然后调用结构体类自带的pack方法，将所有字段转为一个byte数组。

调用DES加密函数，使用KeyCTgs加密上述byte数组，得到加密之后的byte数组，即为返回值。

#### 4.1.2.10 packCtoV函数

**函数声明**：public byte[] packCtoV（）

**函数简介**：该函数是把客户端发给邮件服务器V的包封装的函数，在登录时会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数无传入参数

**函数实现**：首先实例化PackageCtoV结构体类，即给PackageCtoV结构体类的每个字段赋值，PackageCtoV类的字段ticketV、authenticator均为本类的成员变量。

然后调用结构体类自带的pack方法，将所有字段转为一个byte数组。

调用DES加密函数，使用KeyCV加密上述byte数组，得到加密之后的byte数组，即为返回值.

#### 4.1.2.11 unpackAStoCAuth函数

**函数声明**：public void unpackAStoCAuth（byte[] packageToC）

**函数简介**：该函数是把AS服务器发给客户端的认证包解密并解包的函数，在登录时会调用该函数。函数无返回值，函数的传入参数为byte[] packageToC，即需要解密和解包的byte数组。

**函数实现**：首先调用DES解密函数，使用KeyCAS解密传入参数packageToC。

然后调用PackageAStoCAuth结构体类自带的unpack方法，将上述byte数组转为PackageAStoCAuth结构体类各个字段的值，即PackageAStoCAuth类的字段KeyCTgs、IDtgs和ticketTgs，均为本类的成员变量。

最后用PackageAStoCAuth结构体类各个字段的值为本类的成员变量赋值。

#### 4.1.2.12 unpackTgstoC函数

**函数声明**：public void unpackTgstoC（byte[] packageToC）

**函数简介**：该函数是把TGS服务器发给客户端的包解密并解包的函数，在登录时会调用该函数。函数无返回值，函数的传入参数为byte[] packageToC，即需要解密和解包的byte数组。

**函数实现**：首先调用DES解密函数，使用KeyCTgs解密传入参数packageToC。

然后调用PackageTgstoC结构体类自带的unpack方法，将上述byte数组转为PackageTgStoC结构体类各个字段的值，即PackageTgStoC类的字段KeyCV、ticketV均为本类的成员变量。

最后用PackageAStoCAuth结构体类各个字段的值为本类的成员变量赋值。

#### 4.1.2.13 unpackVtoC函数

**函数声明**：public void unpackVtoC（byte[] packageToC）

**函数简介**：该函数是把邮件服务器V发给客户端的包解密并解包的函数，在登录时会调用该函数。函数无返回值，函数的传入参数为byte[] packageToC，即需要解密和解包的byte数组。

**函数实现**：首先调用DES解密函数，使用KeyCV解密传入参数packageToC。

然后调用PackageVtoCAuth结构体类自带的unpack方法，将上述byte数组转为PackageVtoCAuth结构体类字段的值，即PackageTgStoC类的字段TS5+1为本类的成员变量。

最后用PackageAStoCAuth结构体类各个字段的值为本类的成员变量赋值。

#### 4.1.2.14 authentic的函数

**函数声明**：public byte[] authentic（int choice）：

**函数简介**：该函数是把Authenticator封装并加密的函数，在登录时会调用该函数。函数返回值为byte数组，即authenticator的加密之后的值，函数的传入参数为int choice，choice为1代表用KeyCTgs加密，choice为2代表用KeyCV加密。

**函数实现**：首先调用Ticket结构体类自带的pack方法，将TicketTgs/TicketV转为byte数组。判断choice的值，choice为1代表用KeyCTgs加密，choice为2代表用KeyCV加密

然后调用DES加密函数，使用对应的密钥加密上述byte数组，即为函数的返回值。

4.1.3 MainWindow类设计

**主界面类MainWindow的成员变量**：

private String IDc：用户的账号

private String passwd：用户的密码

private Button inboxButton：主界面的查看收件箱按钮

private Button outboxButton：主界面的查看发件箱按钮

private Button sendMailButton：主界面的发送邮件按钮

private Button modifyPasswordButton：主界面的修改密码按钮

private Button logoutButton：主界面的退出登录按钮

private HtmlEditor writeMailEditor：主界面写邮件的控件

private ScrollBar showSubjectScroller：主界面可滑动的显示收件箱和发件箱的控件

private String sender：邮件的发件人

private String receiver：邮件的收件人

private String subject：邮件的主题

private String sendTime：发送邮件的时间

private String content：邮件的内容

private String IDv：邮件服务器V的IP地址

private lifetime：ticketV的有效时间

private String KeyCAS：客户端和AS服务器之间的对称密钥

private String KeyCV：客户端和邮件服务器V之间的对称密钥

**主界面类MainWindow的成员函数**：

public MainWindow（String ID，String password）：MainWindow类的构造函数

public void start(Stage primaryStage)：生成邮箱主界面的函数

public void checkInbox（）：查看收件箱按钮事件的响应函数

public void checkOutbox（）：查看发件箱按钮事件的响应函数

public void sendMail（）：发送邮件按钮事件的响应函数

public void modifyPassword（）：修改按钮事件的响应函数

public void logout（）：退出登录按钮事件的响应函数

public byte[] cConnectV（byte[] packageToV）：客户端和邮件服务器V通信的函数

public byte[] packCtoVRequest（）：把客户端发给邮件服务器V的包封装的函数

public void unpackVtoCPost（）：把邮件服务器V发给客户端的包解密并解包的函数

public void saveToLocal（）：将邮件保存到客户端本地的函数

public void readSenderSubject（int choice）：从本地读取收件箱和发件箱的函数

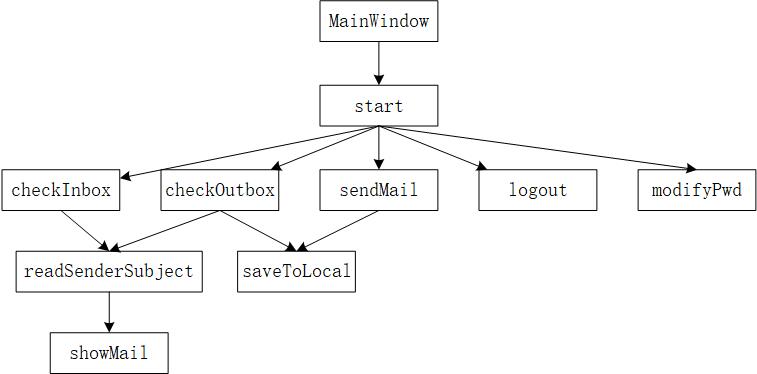


Figure 8

#### 4.1.3.1 MainWindow构造函数

**函数声明**：public MainWindow（String ID，String password）

**函数简介**：该函数由LogRegWindow类的login函数调用，在登录成功之后，调用该函数进入邮箱主界面。函数的传入参数为String ID，String password，即登录成功的用户的账号和密码，并赋值给本类的成员变量，以便于后续修改密码的操作。

**函数实现**：将传入的参数ID、password分别赋值给本类的成员变量IDc、password。

然后调用start函数生成主界面。

#### 4.1.3.2 start函数

**函数声明**：public void start(Stage primaryStage)

**函数简介**：该函数为生成邮箱主界面的函数。函数的返回值为void，传入参数为Stage类型的参数primaryStage（来自javafx包内的数据类型，表示界面）。

**函数实现**：本次实验使用javafx实现UI，所以生成界面时从编写的fxml文件读取并加载界面，通过调用FXMLLoader.load(getClass().getResource("xxx.fxml"))实现。

然后通过primaryStage的setTitle方法设置登录注册界面的title，通过primaryStage的setScene方法设置界面的长宽等参数。

最后调用primaryStage的show方法展示界面，即生成了邮箱主界面。

#### 4.1.3.3 checkInbox函数

**函数声明**：public void checkInbox（）

**函数简介**：该函数是查看收件箱按钮事件的响应函数，在邮箱主界面点击查看收件箱按钮会调用该函数。函数的返回值为void，无传入参数。

**函数实现**：首先调用函数packCtoVRequest（），把客户端发给V的请求获取邮件包封装并加密，得到一个byte数组。

第二步调用函数cConnectV（），和V建立连接，并把上述生成的byte数组发给V，等待V的回应包，接收到V的回应包后关闭连接。

第三步调用unpackVtoCPost（）函数，把邮件服务器V发给客户端的包解密并解包得到一个PackageVtoCPost类的对象。

第四步调用saveToLocal（PackageVtoCPost pvc）函数，先将接收到的邮件保存到本地，再调用函数readSenderSubject（int choice）（choice为1代表读取收件箱2代表读取发件箱，并用scoller控件展示）将简略内容读取。

第五步将邮件的主题、发件人、事件等概括内容显示在ScrollBar showSubjectScroller控件中。

#### 4.1.3.4 checkOutbox函数

**函数声明**：public void checkInbox（）

**函数简介**：该函数是查看发件箱按钮事件的响应函数，在邮箱主界面点击查看发件箱按钮会调用该函数。函数的返回值为void，无传入参数。

**函数实现**：调用函数readSenderSubject（int choice）（choice为1代表读取收件箱2代表读取发件箱，并用scoller控件展示）读取所有收件的简略内容。

然后将邮件的主题、发件人、事件等概括内容显示在ScrollBar showSubjectScroller控件中。

#### 4.1.3.5 sendMail函数

**函数声明**：public void sendMail（）

**函数简介**：该函数是发送邮件按钮事件的响应函数，在邮箱主界面点击发送邮件按钮会调用该函数。函数的返回值为void，无传入参数。

**函数实现**：首先从 HtmlEditor控件中读取用户输入的文件内容。

第二步调用函数packCtoVPost（）函数将上述内容封装并加密，得到一个byte数组。

第三步调用函数cConnectV（byte[] packageToV）将上述byte数组发给邮件服务器V，等待V的回应包，接收到V的回应包后关闭连接。

第四步调用函数saveToLocal（PackageVtoCPost pvc）将发送的邮件保存在本地。

#### 4.1.3.6 modifyPassword函数

**函数声明**：public void modifyPassword（）

**函数简介**：该函数是修改密码按钮事件的响应函数，在邮箱主界面点击修改密码按钮会调用该函数。函数的返回值为void，无传入参数。

**函数实现**：在该函数内调用ModifyPaaword类的构造方法，进入修改密码界面。

#### 4.1.3.7 logout函数

**函数声明**：public void logout（）

**函数简介**：该函数是退出登录按钮事件的响应函数，在邮箱主界面点击退出登录按钮会调用该函数。函数的返回值为void，无传入参数。

**函数实现**：在该函数内关闭邮件主界面界面并调用LogRegWindow类的构造方法，返回到登录注册界面。

#### 4.1.3.8 cConnectV函数

**函数声明**：public byte[] cConnectV（byte[] packageToV）

**函数简介**：该函数是负责客户端和邮件服务器V通信的函数，在查看收件箱和发邮件时会调用该函数。函数的返回值为byte[]，即邮件服务器V发回的数据包，函数的参数为byte[] packageToV，即需要发给V的包。

**函数实现**：先向邮件服务器V发起Socket连接请求，等待邮件服务器V接收请求。

邮件服务器V接收请求之后，将传入的参数，即需要发给邮件服务器V的包发给邮件服务器V。

保持连接，等待邮件服务器V的回应包。

接收到邮件服务器V的回应包之后断开连接，将返回的包转为byte数组返回。

#### 4.1.3.9 packCtoVRequest函数

**函数声明**：public byte[] packCtoVRequest（）

**函数简介**：该函数是把客户端发给邮件服务器V的请求获取邮件的包封装并加密的函数，在查看收件箱时会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数无传入参数。

**函数实现**：首先实例化PackageCtoVRequest结构体类。

调用PackageCtoASRegister对象的pack方法将报文封装转为byte数组。

然后调用DES加密模块的加密函数将报文加密，加密密钥为客户端和V服务器拥有的对称密钥（即成员变量KeyCV），最后得到加密以后的byte数组。

#### 4.1.3.10 unpackVtoCPost函数

**函数声明**：public void unpackVtoCPost（byte[] packageToC）

**函数简介**：该函数是把邮件服务器V发给客户端的包解密并解包的函数，在查看收件箱时会调用该函数。函数无返回值，函数的传入参数为byte[] packageToC，即需要解密和解包的byte数组。

**函数实现**：首先调用DES解密函数，使用KeyCV解密传入参数packageToC。

然后调用PackageVtoCPost结构体类自带的unpack方法，将上述byte数组转为PackageTgStoC结构体类的字段发件人sender、发送时间sendTime、邮件主题subject、邮件内容content、以及邮件附件。

最后用PackageAStoCAuth结构体类各个字段的值为本类的成员变量赋值。

#### 4.1.3.11 saveToLocal函数

**函数声明**：public void saveToLocal（）

**函数简介**：该函数是把收到的邮件保存到本地函数，在查看收件箱checkInbox函数中

调用该函数。函数无返回值，函数无传入参数。

**函数实现**：首先判断本地有无对应账号IDc的文件夹，没有则创建以IDc为文件名的文件夹，有则继续下一步。

第二步根据本类的成员变量sendTime和sender为文件名创建文件夹。

第三步分别以文件的形式在上述创建的文件夹内保存主题、内容、附件。

#### 4.1.3.12 readSenderSubject函数

**函数声明**：public void readSenderSubject（int choice）

**函数简介**：该函数是从本地文件中读取收件箱/发件箱并在控件中展示的函数。函数无返回值，函数的传入参数为int choice，choice为1代表读取收件箱，2代表读取发件箱。

**函数实现**：首先判断读取收件箱还是发件箱。

第二步从对应的文件夹读取发件人sender、发时间sendTime、主题subject。

第三步将上述内容展示在控件ScrollBar showSubjectScroller中。

4.1.4 ModifyPasswordWindow设计

**ModifyPasswordWindow类的成员变量**：

private String IDc：用户的账号

private String pwdOrigin：用户的原始密码

private Button modifyButton：修改密码界面的修改按钮

private PasswdField passwdField1：修改密码界面的新密码输入框

Private PasswdField passwdField2：修改密码界面的新密码确认输入框

**ModifyPasswordWindow类的成员函数**：

public ModifyPasswd（String ID,String passwd）：ModifyPasswordWindow类的构造函数

public void start（Stage primaryStage）：生成修改密码界面的函数

public void modifyPassword（）：修改按钮事件的响应函数

public byte[] cConnectAS（byte[] packageToAS）：客户端和AS服务器通信的函数

public packCtoASModify（）：把客户端发给AS服务器的包封装并加密的函数

public unpackAStoCModify（byte[] packageToC）：把AS服务器返回包解密并解包的函数

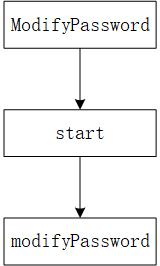


Figure 9

#### 4.1.4.1 ModifyPasswd构造函数

**函数声明**：public ModifyPasswd（String ID,String passwd）

**函数简介**：该函数是ModifyPasswordWindow类的构造函数，在主界面的modifyPassword函数中调用。函数的传入参数为String ID,String passwd，分别为用户的账号和密码。

**函数实现**：将传入的参数ID、passwd分别赋值给本类的成员变量IDc、pwdOrigin。

然后调用start函数生成主界面。

#### 4.1.4.2 start函数

**函数声明**：public void start(Stage primaryStage)

**函数简介**：该函数为生成修改密码界面的函数。函数的返回值为void，传入参数为Stage类型的参数primaryStage（来自javafx包内的数据类型，表示界面）。

**函数实现**：本次实验使用javafx实现UI，所以生成界面时从编写的fxml文件读取并加载界面，通过调用FXMLLoader.load(getClass().getResource("xxx.fxml"))实现。

然后通过primaryStage的setTitle方法设置登录注册界面的title，通过primaryStage的setScene方法设置界面的长宽等参数。

最后调用primaryStage的show方法展示界面，即生成了修改密码界面。

#### 4.1.4.3 modifyPassword函数

**函数声明**：public void modifyPassword（）

**函数简介**：该函数为实现修改密码操作的函数。函数无返回值，无传入参数。

**函数实现**：首先从两个控件passwdField1和passwdField2读取输入的两次新密码，并判断两个密码是否一样，两个密码不一样则弹窗提示重新输入，一样则继续下一步。

第二步调用函数packCtoASModify（）封装客户端发给AS服务器的修改密码的包并加密，加密的密钥为KeyCAS，最后得到一个byte数组。

第三步调用函数cConnectAS（byte[] packageToAS），和AS服务器建立连接，发送上述byte数组，并保持连接直到收到AS的回应包再断开连接。

第四步调用函数unpackAStoCModify（byte[] packageToC）将上述收到的包解密、解包。

第五步验证是否修改成功，如果成功弹窗提示成功，如果失败弹窗提示失败。

#### 4.1.4.4 cConnectAS函数

**函数声明**：public byte[] cConnectAS（byte[] packageToAS）

**函数简介**：该函数是负责客户端和AS服务器通信的函数，在修改密码时会调用该函数。函数的返回值为byte[]，即AS发回的数据包，函数的参数为byte[] packageToAS，即加密之后的需要发给AS的包。

**函数实现**：首先向AS服务器发起Socket连接请求，等待AS服务器接收请求。

AS服务器接收请求之后，将传入的参数，即加密之后的需要发给AS的包发给AS。

保持连接，等待AS的回应包。

接收到AS的回应包之后断开连接，将返回的包转为byte数组返回。

#### 4.1.4.5 packCtoASModify函数

**函数声明**：public byte[] packCtoASModify（）

**函数简介**：该函数是把客户端发给AS服务器的修改密码包封装并加密的函数，在修改密码时会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数无传入参数。

**函数实现**：首先实例化PackageCtoASModify结构体类，即PackageCtoASModify类的IDc字段为用户输入的账号、TS1为当前的时间戳、报文内容为用户输入的密码的MD5 Hash值。

调用PackageCtoASLoginr对象的pack方法将报文封装转为byte数组。

然后调用DES加密模块的加密函数将报文加密，加密密钥为客户端和AS服务器拥有的对称密钥（即成员变量KeyCAS），最后得到加密以后的byte数组。

#### 4.1.4.6 unpackAStoCModify函数

**函数声明**：public void unpackAStoCModify（byte[] packageToC）

**函数简介**：该函数是把AS服务器发给客户端的修改密码包解密并解包的函数，在修改密码时会调用该函数。函数无返回值，函数的传入参数为byte[] packageToC，即需要解密和解包的byte数组。

**函数实现**：首先调用DES解密函数，使用KeyCAS解密传入参数packageToC。

然后调用PackageAStoCModify结构体类自带的unpack方法，将上述byte数组转为PackageAStoCModify结构体类各个字段的值。

## 4.2 AS服务器模块

4.2.1 AS服务器模块设计

AS服务器模块以响应请求为主导，以多线程的形式来响应不同的连接请求，所以会引用接口Runnable。

其次是需要判断收到的包是哪种类型的包，需要回什么类型的包，可根据状态位来分辨并回复。

同时AS服务器还有一个数据库，该数据库存有所有用户的账号和密码、AS服务器与每个客户端的密钥、AS与TGS服务器的密钥以及每次生成的会话密钥。

4.2.2 AS类设计

**AS类的成员变量**：

private Socket socket：AS服务器用来接收连接请求的socket

private byte[] receivePackage：AS服务器用来接收数据包的数组

**AS类的成员函数**：

public AS（Socket socket）：AS类的构造函数

public static void main（）：AS类的主函数

public void run（）：每个线程均会调用的Runnable接口run函数的重写

public void handleRegister（）：处理注册包的函数

public void handleLogin（）：处理登录包的函数

public void handleModifyPwd（）：处理修改密码包的函数

public void handleCheck（）：处理查询收件人包的函数

public void unpackCtoASRegister（byte[] packageToAS）：将客户端发给AS的注册包解密并解包

public void unpackCtoASLogin（byte[] packageToAS）：将客户端发给AS的登录包解密并解包

public void unpackCtoASModify（byte[] packageToAS）：将客户端发给AS的修改密码包解密并解包

public void unpackCtoASCheck（byte[] packageToAS）：将客户端发给AS的查询收件人包解密并解包

public byte[] packAStoCRegister（）：将AS发给客户端的注册状态返回包封包并加密

public byte[] PackageAstoCAuth（）：将AS发给客户端的认证包封包并加密

public byte[] PackageAstoCLogin（）：将AS发给客户端的登录状态返回包封包并加密

public byte[] packAStoCCheck（）：将AS发给客户端的查询收件人返回包封包并加密

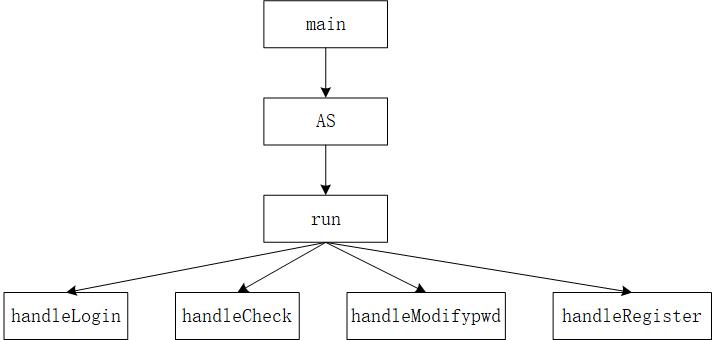


Figure 10

#### 4.2.2.1 AS构造函数

**函数声明**：public AS（Socket socket）

**函数简介**：该函数是AS类的构造函数， 在AS类的主函数中调用，每来一个连接请求便会在一个新线程中调用构造函数。函数无返回值，函数的传入参数为Socket socket，即接收连接请求的sicket。

**函数实现**：将传入参数socket赋值给本类的成员变量socket。

#### 4.2.2.2 main函数

**函数声明**：public static void main（）

**函数简介**：该函数是AS类的主函数，负责接收所有的socket连接请求。函数无返回值，函数无参数。

**函数实现**：在while（true）的循环监听客户端的请求。

每来一个新的请求便开一个新的线程处理该请求。

#### 4.2.2.3 run函数

**函数声明**：public void run（）

**函数简介**：该函数是在多线程内调用的函数，接收客户端发来的包，并根据状态位做判断，调用不同的处理函数。函数无返回值，函数无传入参数。

**函数实现**：首先用本类的成员变量byte数组接收客户端发来的包。

第二步根据包的状态位判断是哪种类型的包。

第三步对不同的包分别调用handleRegister（）、handleLogin（）、handleModifyPwd（）、 handleCheck（）函数来处理。

#### 4.2.2.4 handleRegister函数

**函数声明**：public void handleRegister（）

**函数简介**：该函数是处理注册包的函数，注册包主要用来判断要注册的账号是否已在数据库中存在。函数无返回值，函数无传入参数。

**函数实现**：首先调用函数unpackCtoASRegister（byte[] packageToAS）将接收到的包解密、解包。

第二步获取包内的字段账号ID和密码password，并去数据库中查找是否存在该账号。如果存在继续下一步，如果不存在，将账号和密码存入数据库，再继续下一步。

第三步调用函数packAStoCRegister（）发给客户端注册状态返回包，不合法的账号注册状态位为0，合法的账号注册状态位为1。

#### 4.2.2.5 handleLogin函数

**函数声明**：public void handleLogin（）

**函数简介**：该函数是处理登录包的函数，登录包主要用来判断要登录的账号对应的密码是否和数据库中的密码相同。函数无返回值，函数无传入参数。

**函数实现**：首先调用函数unpackCtoASLogin（byte[] packageToAS）将接收到的包解密、解包。

第二步获取包内的字段账号ID和密码password，并去数据库中查找是否该账号，并比较发来的密码和数据库中的密码Hash值是否一样。

第三步如果密码一样则调用函数packAStoCAuth（）发送一个PackageAstoCAuth包。如果密码不一样则调用函数packAStoCLogin（）发送一个PackageAstoCLogin包。

#### 4.2.2.6 handleModifyPwd函数

**函数声明**：public void handleModifyPwd（）

**函数简介**：该函数是处理修改密码包的函数，修改密码包主要用来修改数据库中对应账号的密码。函数无返回值，函数无传入参数。

**函数实现**：首先调用函数unpackCtoASModify（byte[] packageToAS）将接收到的包解密、解包。

第二步获取包内的字段账号ID和新密码password，去数据库中找到该账号，并将数据库中的密码改为新密码的Hash值。

#### 4.2.2.7 handleCheck函数

**函数声明**：public void handleCheck（）

**函数简介**：该函数是处理查询收件人的包，查询收件人包主要是用来确定收件人是否存在。函数无返回值，函数无传入参数。

**函数实现**：首先调用函数unpackCtoASCheck（byte[] packageToAS）将接收到的包解密、解包。

第二步获取包内的字段账号ID，去数据库中查找该账号。

第三步调用函数packAStoCCheck（）发给客户端查找的结果，查询状态位为0表示没有该账号，查询状态位为1代表数据库中存在该账号。

#### 4.2.2.8 unpackCtoASRegister函数

**函数声明**：public void unpackCtoASRegister（byte[] packageToAS）

**函数简介**：该函数是将客户端发给AS的注册包解密并解包的函数。函数无返回值，函数的传入参数为byte[] packageToAS，即客户端发给AS服务器的包。

**函数实现**：首先调用DES解密函数，使用密钥KeyCAS解密传入参数packageToAS。

然后调用PackageCtoASRegister结构体类自带的unpack方法，将上述byte数组转为PackageAStoCRegister结构体类各个字段的值。

#### 4.2.2.9 unpackCtoASLogin函数

**函数声明**：public void unpackCtoASLogin（byte[] packageToAS）

**函数简介**：该函数是将客户端发给AS的登录包解密并解包的函数。函数无返回值，函数的传入参数为byte[] packageToAS，即客户端发给AS服务器的包。

**函数实现**：首先调用DES解密函数，使用密钥KeyCAS解密传入参数packageToAS。

然后调用PackageCtoASLogin结构体类自带的unpack方法，将上述byte数组转为PackageAStoCLogin结构体类各个字段的值。

#### 4.2.2.10 unpackCtoASModify函数

**函数声明**：public void unpackCtoASModify（byte[] packageToAS）

**函数简介**：该函数是将客户端发给AS的修改密码包解密并解包的函数。函数无返回值，函数的传入参数为byte[] packageToAS，即客户端发给AS服务器的包。

**函数实现**：首先调用DES解密函数，使用密钥KeyCAS解密传入参数packageToAS。

然后调用PackageCtoASModify结构体类自带的unpack方法，将上述byte数组转为PackageAStoCModify结构体类各个字段的值。

#### 4.2.2.11 unpackCtoASCheck函数

**函数声明**：public void unpackCtoASCheck（byte[] packageToAS）

**函数简介**：该函数是将客户端发给AS的查询收件人包解密并解包的函数。函数无返回值，函数的传入参数为byte[] packageToAS，即客户端发给AS服务器的包。

**函数实现**：首先调用DES解密函数，使用密钥KeyCAS解密传入参数packageToAS。

然后调用PackageCtoASCheck结构体类自带的unpack方法，将上述byte数组转为PackageAStoCCheck结构体类各个字段的值。

#### 4.2.2.12 packAStoCRegister函数

**函数声明**：public byte[] packAStoCRegister（）

**函数简介**：该函数是把AS服务器发给客户端的注册状态返回包封装并加密的函数，在AS的handleRegister函数中会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数无传入参数。

**函数实现**：首先实例化PackageAStoCRegister结构体类，即PackageAStoCRegister类的各个字段。

调用PackageAStoCRegister对象的pack方法将报文封装转为byte数组。

然后调用DES加密模块的加密函数将报文加密，加密密钥为客户端和AS服务器拥有的对称密钥（即成员变量KeyCAS），最后得到加密以后的byte数组。

#### 4.2.2.13 PackageAstoCAuth函数

**函数声明**：public byte[] packAStoCAuth（）

**函数简介**：该函数是把AS服务器发给客户端的认证包封装并加密的函数，在AS的handleAuth函数中会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数无传入参数。

**函数实现**：首先实例化PackageAStoCAuth结构体类，即PackageAStoCAuth类的各个字段。

调用PackageAStoCAuth对象的pack方法将报文封装转为byte数组。

然后调用DES加密模块的加密函数将报文加密，加密密钥为客户端和AS服务器拥有的对称密钥（即成员变量KeyCAS），最后得到加密以后的byte数组。

#### 4.2.2.14 PackageAstoCLogin函数

**函数声明**：public byte[] packAStoCLogin（）

**函数简介**：该函数是把AS服务器发给客户端的登录状态返回包封装并加密的函数，在AS的handleLogin函数中会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数无传入参数。

**函数实现**：首先实例化PackageAStoCLogin结构体类，即PackageAStoCLogin类的各个字段。

调用PackageAStoCLogin对象的pack方法将报文封装转为byte数组。

然后调用DES加密模块的加密函数将报文加密，加密密钥为客户端和AS服务器拥有的对称密钥（即成员变量KeyCAS），最后得到加密以后的byte数组。

#### 4.2.2.15 packAStoCCheck函数

**函数声明**：public byte[] packAStoCCheck（）

**函数简介**：该函数是把AS服务器发给客户端的查询收件人状态返回包封装并加密的函数，在AS的handleCheck函数中会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数无传入参数。

**函数实现**：首先实例化PackageAStoCCheck结构体类，即PackageAStoCCheck类的各个字段。

调用PackageAStoCCheck对象的pack方法将报文封装转为byte数组。

然后调用DES加密模块的加密函数将报文加密，加密密钥为客户端和AS服务器拥有的对称密钥（即成员变量KeyCAS），最后得到加密以后的byte数组

## 4.3 TGS服务器模块

4.3.1 TGS服务器模块设计

TGS服务器模块以响应请求为主导，以多线程的形式来响应不同的连接请求，所以会引用接口Runnable。

4.3.2 TGS类设计

**TGS类的成员变量**：

private Socket socket：TGS服务器用来接收连接请求的socket

private String KeyAsTgs：AS服务器和TGS服务器之间的对称密钥

private byte[] packageReceived：TGS服务器用来接收数据包的数组

**TGS类的成员函数**：

public TGS(Socket socket)：TGS类的构造函数

public static void main（）：TGS类的主函数

public void run（）：每个线程均会调用的Runnable接口run函数的重写

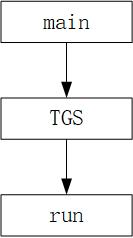


Figure 11

#### 4.3.2.1 TGS构造函数

**函数声明**：public TGS（Socket socket）

**函数简介**：该函数是TGS类的构造函数，在TGS类的主函数中调用，每来一个连接请求便会在一个新线程中调用构造函数。函数无返回值，函数的传入参数为Socket socket，即接收连接请求的sicket。

**函数实现**：将传入参数socket赋值给本类的成员变量socket。

#### 4.3.2.2 main函数

**函数声明**：public static void main（）

**函数简介**：该函数是TGS类的主函数，负责接收所有的socket连接请求。函数无返回值，函数无参数。

**函数实现**：在while（true）的循环监听客户端的请求。

每来一个新的请求便开一个新的线程处理该请求。

#### 4.3.2.3 run函数

**函数声明**：public void run（）

**函数简介**：该函数是在多线程内调用的函数，接收客户端发来的包，并根据状态位做判断，调用不同的处理函数。函数无返回值，函数无传入参数。

**函数实现**：首先将客户端发来的包解密、解包为PackageCtoTgs结构体类型的对象。

然后验证ticketTgs，如果ticketTgs无误，则发送给客户端一个PackageTgstoC包。

## 4.4邮件服务器模块

4.4.1邮件服务器模块设计

邮件服务器同样以响应请求为主导，以多线程的形式来响应不同的连接请求，所以会引用接口Runnable。

邮件服务器还需要能存储每个用户所收到的邮件，需要在服务器本地存储所有邮件。

4.4.2 EmailServer类设计

**EmailServer类的成员变量**：

private Socket socket：邮件服务器用来接收连接请求的socket

private String KeyVTgs：邮件服务器和TGS服务器之间的对称密钥

private byte[] packageReceived; 邮件服务器用来接收数据包的数组

**EmailServer类的成员函数**：

public static void main（）：EmailServer类的主函数

public void run（）：每个线程均会调用的Runnable接口run函数的重写

public void handleAuth（）：处理客户端认证包的函数

public void handlePost（）：处理客户端发过来的邮件包的函数

public void handleRequest（）：处理客户端请求获取邮件的包的函数

public void unpackCtoVAuth（byte[] packageToAS）：将客户端发给邮件服务器的认证包解密并解包

public void unpackCtoVPost（byte[] packageToAS）：将客户端发给邮件服务器的邮件包解密并解包

public void unpackCtoVRequest（byte[] packageToAS）：将客户端发给邮件服务器的请求邮件包解密并解包

public byte[] packVtoCAuth（）：将邮件服务器回给客户端的认证返回包封装并加密

public byte[] packVtoCPost（）：将邮件服务器发给客户端的邮件包封装并加密

public void saveToLocal（String ID，String sendTime）：将邮件保存到本地

public readMail（String ID，String sendTime）：从本地读取邮件

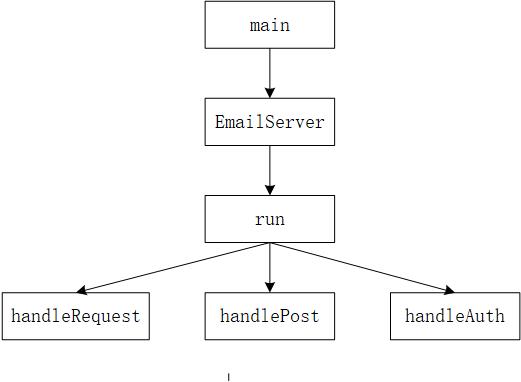


Figure 12

#### 4.4.2.1 main函数

**函数声明**：public static void main（）

**函数简介**：该函数是EmailServer类的主函数，负责接收所有的socket连接请求。函数无返回值，函数无参数。

**函数实现**：在while（true）的循环监听客户端的请求。

每来一个新的请求便开一个新的线程处理该请求。

#### 4.4.2.2 run函数

**函数声明**：public void run（）

**函数简介**：该函数是在多线程内调用的函数，接收客户端发来的包，并根据状态位做判断，调用不同的处理函数。函数无返回值，函数无传入参数。

**函数实现**：首先用本类的成员变量byte数组接收客户端发来的包。

第二步根据包的状态位判断是哪种类型的包。

第三步对不同的包分别调用handleAuth（）、handlePost（）、handleRequest（）函数来处理。

#### 4.4.2.3 handleAuth函数

**函数声明**：public void handleAuth（）

**函数简介**：该函数是处理客户端和邮件服务器认证包的函数，该包主要用来认证客户端的身份。函数无返回值，函数无传入参数。

**函数实现**：首先调用函数unpackCtoVAuth（byte[] packageToAS）将接收到的包解密、解包。

第二步获取包内的字段ticketV及KeyCV。

第三步验证ticketV后调用函数packVtoCAuth（）将邮件服务器发给客户端的包封装并加密，加密密钥为KeyCV。

#### 4.4.2.4 handlePost函数

**函数声明**：public void handlePost（）

**函数简介**：该函数是处理客户端给服务器发邮件的包的函数，该包主要用来发邮件。函数无返回值，函数无传入参数。

**函数实现**：首先调用函数unpackCtoVPost（byte[] packageToAS）将接收到的包解密、解包。

第二步获取包内的字段，即邮件的发件人、收件人、主题、时间、附件等。

第三步调用函数saveToLocal（String ID，String sendTime）将邮件保存到本地。

#### 4.4.2.5 handleRequest函数

**函数声明**：public void handleRequest（）

**函数简介**：该函数是处理客户端给服务器请求邮件的包的函数，该包主要用来更新收件箱内的邮件。函数无返回值，函数无传入参数。

**函数实现**：首先调用函数unpackCtoVRequest（byte[] packageToAS）将接收到的包解密、解包。

第二步获取包内的字段，即用户的ID。

第三步调用函数readMail（String ID，String sendTime），从服务器的文件中读取对应文件下的所有邮件。

第四步调用函数packVtoCPost（）将邮件封装为包，加密再发送给客户端。

#### 4.4.2.6 unpackCtoVAuth函数

**函数声明**：public void unpackCtoVAuth（byte[] packageToV）

**函数简介**：该函数是将客户端发给邮件服务器的认证包解密并解包的函数。函数无返回值，函数的传入参数为byte[] packageToAS，即客户端发给AS服务器的包。

**函数实现**：首先调用DES解密函数，使用密钥KeyCV解密传入参数packageToV。

然后调用PackageCtooVAuth结构体类自带的unpack方法，将上述byte数组转为PackageCtooVAuth结构体类各个字段的值。

#### 4.4.2.7 unpackCtoVPost函数

**函数声明**：public void unpackCtoVPost（byte[] packageToV）

**函数简介**：该函数是将客户端发给邮件服务器的邮件包解密并解包的函数。函数无返回值，函数的传入参数为byte[] packageToAS，即客户端发给AS服务器的包。

**函数实现**：首先调用DES解密函数，使用密钥KeyCV解密传入参数packageToV。

然后调用PackageCtooVPost结构体类自带的unpack方法，将上述byte数组转为PackageCtooVPost结构体类各个字段的值。

#### 4.4.2.8 unpackCtoVRequest函数

**函数声明**：public void unpackCtoVRequest（byte[] packageToV）

**函数简介**：该函数是将客户端发给邮件服务器的请求邮件包解密并解包的函数。函数无返回值，函数的传入参数为byte[] packageToAS，即客户端发给AS服务器的包。

**函数实现**：首先调用DES解密函数，使用密钥KeyCV解密传入参数packageToV。

然后调用PackageCtooVRequest结构体类自带的unpack方法，将上述byte数组转为PackageCtooVRequest结构体类各个字段的值。

#### 4.4.2.9 packVtoCAuth函数

**函数声明**：public byte[] packVtoCAuth（）

**函数简介**：该函数是把邮件服务器发给客户端的认证返回包封装并加密的函数，在邮件服务器的的handleAuth函数中会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数无传入参数。

**函数实现**：首先实例化PackageVtoCAuth结构体类，即PackageVtoCAuth类的各个字段。

调用PackageVtoCAuth对象的pack方法将报文封装转为byte数组。

然后调用DES加密模块的加密函数将报文加密，加密密钥为客户端和邮件服务器拥有的对称密钥（即成员变量KeyCV），最后得到加密以后的byte数组。

#### 4.4.2.10 packVtoCPost函数

**函数声明**：public byte[] packVtoCPost（）

**函数简介**：该函数是把邮件服务器发给客户端的邮件包封装并加密的函数，在邮件服务器的的handleRequest函数中会调用该函数。函数的返回值为byte数组，即封包并加密之后的结果，函数无传入参数。

**函数实现**：首先实例化PackageVtoCPost结构体类，即PackageVtoCPost类的各个字段。

调用PackageVtoCPost对象的pack方法将报文封装转为byte数组。

然后调用DES加密模块的加密函数将报文加密，加密密钥为客户端和邮件服务器拥有的对称密钥（即成员变量KeyCV），最后得到加密以后的byte数组。

#### 4.4.2.11 saveToLocal函数

**函数声明**：public void saveToLocal（String ID，String sendTime）

**函数简介**：该函数是把收到的邮件保存到本地函数，在处理邮件handlePost函数中

调用该函数。函数无返回值，函数无传入参数。

**函数实现**：首先判断本地有无对应账号IDc的文件夹，没有则创建以ID为文件名的文件夹，有则继续下一步。

第二步根据本类的成员变量sendTime和sender ID为文件名创建文件夹。

第三步分别以文件的形式在上述创建的文件夹内保存主题、内容、附件。

#### 4.4.2.12 readMail函数

**函数声明**：public void readMail（String ID，String sendTime）

**函数简介**：该函数是从本地文件中读取收件箱的函数。函数无返回值，函数无传入参数。

**函数实现**：从对应的文件夹读取发件人sender、发时间sendTime、主题subject。

第二步将上述内容封装在包PackageVtoCPost类中。

## 4.5 Package模块

4.5.1 Package类设计

public Class Package

{

@StructClass

Class PackageCtoAsRegister{}

@StructClass

Class PackageAstoCRegister {}

@StructClass

Class PackageCtoAsLogin{}

@StructClass

Class PackageAstoCLogin{}

@StructClass

Class PackageAstoCAuth{}

@StructClass

Class PackageCtoAsModify{}

@StructClass

Class PackageAstoCModify{}

@StructClass

Class PackageCtoAsCheck{}

@StructClass

Class PackageAstoCCheck{}

@StructClass

Class PackageCtoVRequestMail{}

@StructClass

Class PackageCtoTgs{}

@StructClass

Class PackageTgstoC{}

@StructClass

Class PackageCtoVAuth{}

@StructClass

Class PackageVtoCAuth{}

@StructClass

Class PackageCtoVPost{}

@StructClass

Class PackageVtoCPost{}

}

如上图，Package模块封装了此次程序中用到的所有的包。

**1.PackageCtoAsRegister**

PackageCtoAsRegister是C给AS发的注册包，定义如下：

@StructClass

Class PackageCtoAsRegister

{

byte status; //byte

String IDc; //char[10]

String TS1; //char[13]

byte[10] redundancy; //byte[10]

String content; //char[10]

}

**2.PackageAstoCRegiste**r是AS给C发的注册包，定义如下：

@StructClass

Class PackageAstoCRegister

{

byte status; //byte

byte registerStatus; //byte

}

**3.PackageCtoAsLogin**是C给AS发的登录包，定义如下：

@StructClass

Class PackageCtoAsLogin

{

byte status;

String IDc;

String IDtgs;

String TS1;

byte[10] redundancy;

String content;

}

**4.PackageAstoCLogin**

@StructClass

Class PackageAstoCLogin

{

byte status;

byte loginStatus;

}

**5 .PackageAstoCAuth**

@StructClass

Class PackageAstoCAuth

{

byte status;

String Kctgs;

String IDtgs;

String TS2;

String Lifetime

TicketTgs ticketTgs;

byte[10] redundancy;

}

**6. PackageCtoAsModify**

@StructClass

Class PackageCtoAsModify

{

byte status;

String IDc;

String TS1;

byte[10] redundancy;

String content;

}

**7. PackageAstoCModify**

@StructClass

Class PackageAstoCModify

{

byte status;

String IDc;

String TS1;

byte[10] redundancy;

String content;

}

**8. PackageCtoAsCheck**

@StructClass

Class PackageCtoAsCheck

{

byte status;

String IDc;

String TS1;

byte[10] redundancy;

String content;

}

**9. PackageAstoCCheck**

@StructClass

Class PackageAstoCCheck

{

byte status;

byte checkStatus;

}

**10. PackageCtoVRequestMail**

@StructClass

Class PackageCtoVRequestMail

{

byte status;

String IDc;

String TS1;

byte[10] redundancy;

}

**11. PackageCtoTgs**

@StructClass

Class PackageCtoTgs

{

String IDv;

TicketTgs ticketTgs;

byte[] authenticator1;

byte[10] redundancy;

}

**12. PackageTgstoC**

@StructClass

Class PackageTgstoC

{

byte status;

byte[8] Kcv;

String IDv;

String TS1;

TicketV ticketV;

byte[10] redundancy;

}

**13. PackageCtoVAuth**

@StructClass

Class PackageCtoVAuth

{

byte status;

TicketV ticketV;

byte[10] authenticator2;

byte[10] redundancy;

}

**14. PackageVtoCAuth**

@StructClass

Class PackageVtoCAuth

{

byte status;

String TS2;

byte[10] redundancy;

}

**15. PackageCtoVPost**

@StructClass

Class PackageCtoVPost

{

byte status;

byte[4] messageTotalLength;

String receiverID;

String senderID;

String TS;

byte subjectLen;

byte[2] contentLen;

byte[3] attachmentLen;

byte[] content;

byte[] sign;

byte[10] redundancy;

}

**16. PackageVtoCPost**

@StructClass

Class PackageVtoCPost

{

byte status;

byte[4] messageTotalLength;

String receiverID;

String senderID;

String TS;

byte subjectLen;

byte[2] contentLen;

byte[3] attachmentLen;

byte[] content;

byte[] sign;

byte[10] redundancy;

}

4.5.2 Ticket类设计

public Class Ticket

{

@StructClass

Class TicketTgs //Tickettgs

{

String KeyCTgs；

String IDc；

String ADc；

String IDtgs；

String TS2；

String lifetime2；

}

@StructClass

Class TicketV //Ticketv

{

String KeyCV；

String IDc；

String ADc；

String IDv；

String TS4；

String lifetime4；

}

4.5.3 Authenticator类设计

@StructClass

Class Authenticator

{

String IDc；

String ADc；

String TS；

}

## 4.6 DES模块

DES模块就是之前课程作业中实现的DES加密算法，我们将DES加密算法的所有的类都封在一个包里，其他模块调用DES模块只需要导入这个包。这个模块可以对外提供的方法一共有三个，生成密钥emergeKey、加密encrypt和解密decrypt。

emergeKey函数的原型是byte[] emergeKey(int length)，输入为一个int变量length，用来指定生成密钥的长度，函数功能是生成指定长度的密钥；

encrypt函数的原型是byte[] encrypt(byte[] Mdata,byte[] key)，输入为明文Mdata（byte数组）和密钥key（byte数组），输出为密文（byte数组），功能是对明文进行加密并返回密文；

decrypt函数的原型是byte[] decrypt(byte[] Edata,byte[] key)，输入为密文Edata（byte数组）和密钥key（byte数组），功能是对密文进行解密并返回明文。

## 4.7日志模块

日志模块我们采用的是Apache的开源项目log4j这个强大的日志组件，只需要把log4j的包导入并且配置好log4j.properties就可以使用了。

需要写入日志的消息和日志的类型：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日志类型 | 客户端 | AS | TGS | 邮件服务器 |
| Info | 发包的时间、所在模块、发的包的名称  收包的时间、所在模块、收的包的名称 | 发包的时间、所在模块、发的包的名称  收包的时间、所在模块、收的包的名称 | 发包的时间、所在模块、发的包的名称  收包的时间、所在模块、收的包的名称 | 发包的时间、所在模块、发的包的名称  收包的时间、所在模块、收的包的名称 |
| Error | 登录失败、注册失败、修改密码失败的信息 | 客户端与AS认证失败的信息 | 客户端与TGS认证失败的信息 | 客户端与V认证失败的信息 |

## 4.8数据库模块

账号数据库：

账户名（IDc） 密码（H（Password）） 密钥

|  |  |  |
| --- | --- | --- |
|  | 账户名 | 密码 |
| 数据类型 | varchar(10) | varbinary(128) |
| 数据说明 | 用户账号，10位字母数字组合 | 用户密码的md5Hash值，最多16字符 |

发件表：在该表中的均为收件人由于未在线所以未接受的邮件

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 收件人ID | 发件人ID | 发送时间 | 签名 |
| 数据类型 | varchar(10) | vacahr(10) | bigint | nvarcah(16) |
| 数据说明 | 收件人的账号，10位字母数字组合 | 发件人的账号，10位字母数字组合 | 发邮件的时间，13位整数的时间戳 |  |

邮件服务器存储方式：

以用户ID（账号）为文件夹，文件夹里的文件命名方式为发件人+时间，里面为主题、内容、附件

客户端邮件存储方式：

以用户ID（账号）为文件夹，子目录有收件箱、发件箱两个文件夹。收件箱中以发件人账号和时间戳的组合表示每条邮件，内有正文、主题、附件文件；发件箱以收件人+时间戳组合为文件名表示每一条已发送的文件，每条邮件内容也都有正文、主题、附件文件。