



《物联网安全》 课程设计报告

姓 名	:	李全可	班号:	193171
学 号	:	20171000160	组长:	李全可
院 (系)	:	计算机学院	专业:	网络工程
指导教师	:	姚 宏	职称:	教授

2020 年 9 月

独立工作成果声明

本人声明所呈交的《物联网安全课程设计》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 日

目录

第一章 引言.....	1
§1.1 项目描述	1
§1.2 开发环境	1
第二章 概要设计	2
§2.1 设备部署图	2
§2.2 时序图	2
§2.3 任务划分和分工	3
2.3.1 任务划分	3
2.3.2 任务分工	3
第三章 kerberos 认证设计	5
§3.1 C->TGS IDv Tickettgs Authenticator	5
3.1.1 流程	5
3.1.2 报文格式	5
3.1.3 函数接口	5
§3.2 TGS->C EKc,tgs[Kc,v IDv TS4 Ticketv]	7
3.2.1 流程	7
3.2.2 报文格式	8
3.2.3 函数接口	8
第四章 模块设计	9
§4.1 kerberos Client 端设计	9
4.1.1 功能描述	9
4.1.2 全局变量	9
4.1.3 输入输出	9
4.1.4 程序流程	11
4.1.5 函数接口	12
§4.2 文件下载	13
4.2.1 功能描述	13
4.2.2 功能描述	13
4.2.3 函数接口	14
4.2.4 数据包格式	15
§4.3 加解密设计	17
4.3.1 DES 加密算法	17
4.3.2 RSA 加密算法	17
4.3.3 hash 函数	17
4.3.4 函数接口	17
第五章 运行展示	19
§5.1 AS 端	19
§5.2 TGS 端	20
§5.3 C 端	21
§5.4 V 端	24
第六章 感想	27

第一章 引言

§1.1 项目描述

基于 kerberos 的文件传输系统，利用 DES 和 RSA 对数据进行加密，使用 kerberos 认证体系，实现数据传输的安全性和可靠性。

主要功能：实现一个文件安全传输系统，单个用户可实现文件的上传和下载，用户间可实现文件的共享，文件传输过程中经发送端加密和接受端解密确保文件的保密性和安全性。每一个用户（client）注册账号信息后登陆时进行 kerberos 认证，client 访问 AS 服务器获取身份许可，成功后得到许可票据 ticket1，利用该票据 ticket1 访问 TGS 服务器获取服务许可票据 ticket2，利用 ticket2 访问文件应用服务器获取文件传输服务并得到用于文件传输的 session key，利用 session key 实现文件的安全传输。

§1.2 开发环境

1. 环境系统

Window 10 系统

2. 运行平台

Eclipse

3. 开发语言

Java

4. 硬件配置

个人笔记本电脑 4 台

5. 数据库

MYSQL 数据库

第二章 概要设计

§2.1 设备部署图

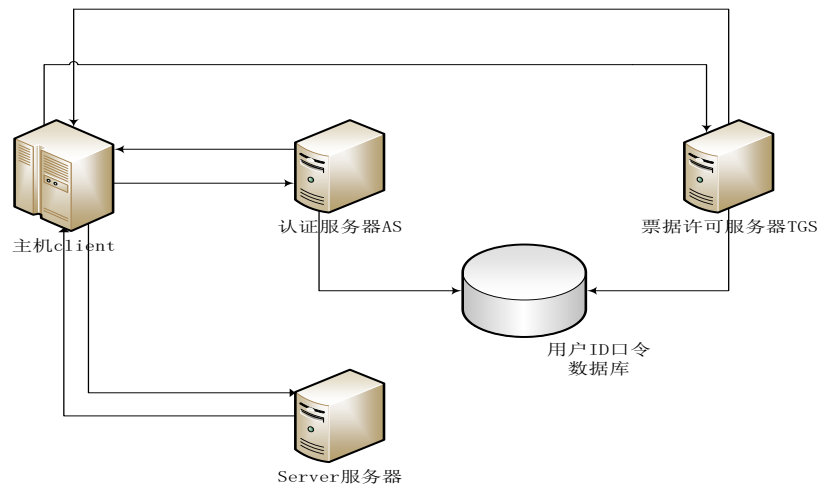
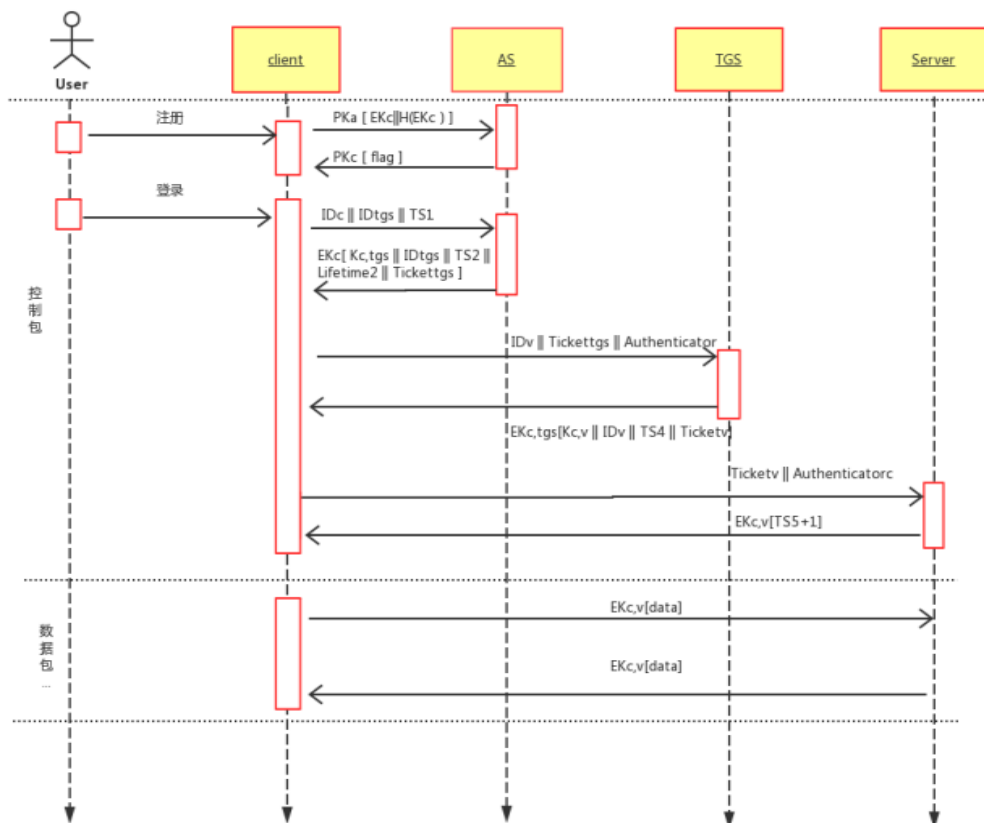


图 2-1 设备部署图

§2.2 时序图



§2.3 任务划分和分工

2.3.1 任务划分

Kerberos 认证任务划分

1. C -> AS IDc|IDtgs|TS1 Client 端数据包生成函数，AS 端解包函数；
AS-> C EKc[Kc,tgs|IDtgs|TS2|lifetime2|Tickettgs] AS 端数据包生成函数，Client 端解包函数。
2. C->TGS IDv|Tickettgs|Authenticator Client 端数据包生成函数，TGS 端解包函数；
TGS->C EKc,tgs[Kc,v|IDv|TS4|Ticketv] TGS 端数据包生成函数，Client 端解包函数。
3. C->V Ticketv|Authenticator Client 端数据包生成函数，V 端解包函数；
V->C EKc,v[TS5+1] V 端数据包生成函数，Client 解包函数
4. Client 端注册登陆 Client 端数据包生成函数，AS 端解包函数
C->AS EKc,v[data] 文件传输的 Client 端生成包函数和 V 端解包函数
AS->C EKc,v[data] 文件传输的 V 端生成包函数和 Client 端解包函数

模块设计任务分工

1. 客户端和服务端 socket 连接，服务端的 IO 复用。
2. 文件传输基本功能实现和 UI 设计。
3. 数据库设计和更新接口函数
4. DES&RSA&hash 加解密函数接口编写

2.3.2 任务分工

小组成员	任务一	任务二	任务三
李全可	Kerberos 认证 C 和 TGS 的生成包和解包函数	文件下载的生成包和解包函数	Client 端的编写
黄成杰	Kerberos 认证 C 和 AS（用户注册）的生成包和解包函数	文件上传的生成包和解包函数	AS，TGS 以及 V 端的编写，数据库访问设计与数据库设计

陈文兵	Kerberos 认证 C 和 AS (用户登录) 的生成包 和解包函数	文件刷新的生成包和 解包函数	UI 的设计
袁道雄	Kerberos 认证 C 和 V 的生成包和解包函数	文件删除的生成包和 解包函数	UI 的设计

我的任务分工：

1. Kerberos 认证过程，C—>TGS 和 TGS—>C 认证包生成包解包函数。
2. Client 端线程和功能函数的编写。
2. 文件服务过程，文件下载相关包的生成包和解包函数。
3. 文件分块和合并函数。

第三章 kerberos 认证设计

§3.1 C->TGS IDv|Tickettgs|Authenticator

3.1.1 流程

1. Client 收到报文后利用校验位检查报文完整性，获取前 4 个字节得知报文类型。
2. 使用解包函数解密获取报文中的 Kc,tgs、IDtgs、TS2、lifetime2 和 Tickettgs。
3. 获取 IDv，文件服务器的 IP 地址。
4. 获取 ADc，Client 自己的 IP 地址。
5. 获取 TS3，系统当前时间。
6. 拼接 IDc、ADc 和 TS3 用 Kc,tgs 加密后生成 Authenticator。
7. 拼接 IDv|Tickettgs|Authenticator。
8. 添加包头包尾发给 TGS 服务器。

3.1.2 报文格式

报文类型 ST	数据长度	数据包	校验位
4 字节 (1005)	1 字节	0~127 字节	1 字节

数据包		
IDv	Tickettgs	Authenticator
32 位二进制	上一步的票据	33 字节

Authenticator (Kc,tgs 加密)		
IDc	ADc	TS3
8 字节	32 位二进制 (4 字节)	13 字节

IDv: 文件服务器 IP 地址。

TS3: 时间戳，当前系统时间。

3.1.3 函数接口

Client 解包函数 (TGS->C)

```
public String[] GetTGStoC(String cpackage,String EKctgs)
```


Client 生成包函数 (C→TGS)

```
public String CtoTGS(String IDv,String Tickettgs,String IDc,String ADc,String EKctgs)
```

§3.2 TGS->C EKc,tgs[Kc,v|IDv|TS4|Ticketv]

3.2.1 流程

总流程

1. TGS 收到报文后利用校验位检查报文完整性，获取前 4 个字节得知报文类型。
2. 解析 Client 发来的包，获取 IDv、Tickettgs、Authenticator。
3. 用 EKtgs（TGS 的私钥）解析 Tickettgs 获取 Kc,tgs,IDc,ADc 等。
4. 用 Kc,tgs 解析 Authenticator 获取 IDc,ADc 等。
5. 检验数据信息，Tickettgs 是否失效等。
6. 随机生成一个 Kc,v，用于接下来 Client 和 V 会话的 session key。
7. 获取 IDv，文件应用服务器 IP 地址。
8. 获取 TS4，时间戳，当前系统时间。
9. 生成 Ticketv，Client 用于访问 V 的票据。
10. 拼接各字段，采用 EKc,tgs 加密。
11. 添加包头包尾，发给 Client。

生成 Ticketv 流程

1. 获取 Kc,v。
2. 获取 IDc。
3. 获取 ADc。
4. 获取 IDv。
5. 获取 TS4，上一步的时间戳。
6. 获取 LifeTime4，该票据额生命周期。
7. 采用 EKv（V 的公钥）加密。
8. 生成 Ticketv EKv[Kc,v|IDc|ADc|IDv|TS4|LifeTime4]。

3.2.2 报文格式

报文类型 ST	数据长度	数据包	校验位
4 字节 (1006)	1 字节	0~127 字节	1 字节

数据包 (EKc,tgs 对称钥加密)			
Kc,v	IDv	TS4	Ticketv
64 位二进制	4 字节	13 字节	58 字节

Tickettgs (EKv 加密)					
Kc,v	IDc	ADc	IDv	TS4	Lifetime4
64 位二进制	8 字节	4 字节	4 字节	13 字节	13 字节

报文类型 (ST:1006) : String 类型 4 个字节。

校验位: 使用异或校验位算法, 一个 byte 类型, 验证报文的完整性。

Kc,v: 随机生成的 64 位二进制密钥, 用于接下来 Client 和 V 会话的 session keys。

3.2.3 函数接口

TGS 解包函数 (C→TGS)

```
public String[] GetCtoTGS(String cpackage,String EKtgs)
```

TGS 生成包函数 (TGS→C)

```
public String TGStoC(String Kcv,String EKctgs,String EKv,String IDv,String IDc,String ADc)
```

第四章 模块设计

§4.1 kerberos Client 端设计

4.1.1 功能描述

1. 分别连接 AS,TGS 和 V 完成 kerberos 认证。
2. 向 V 发送上传的文件的数据包，接收、检查和分析 V 发来接受结果控制包。
3. 向 V 发送共享文件请求的数据包，接收、检查和分析 V 发来的文件数据包，向 V 发送文件接受结果的控制包。
4. 向 V 发送删除私人文件请求的数据包，接收、检查和分析 V 发来的结果响应控制包。
5. 向 V 发送共享文件目录请求的数据包，接收、检查和分析 V 发来共享文件目录数据包。

4.1.2 全局变量

IDc	Kc	ADc	IDas	IDtgs	IDv	Ec,tgs	Ec,v
用户账号	用户密码	用户的 IP 地址	AS 服务器 IP 地址	TGS 服务器 IP 地址	V 服务器 IP 地址	C 与 TGS 会话密钥	C 与 V 会话密钥

4.1.3 输入输出

输出：

1001 号请求注册数据包

1003 号 kerberos 认证包

1005 号 kerberos 认证包

1007 号 kerberos 认证包

1009 号上传文件数据包

1011 号下载共享文件请求数据包

1015 号刷新共享文件夹请求数据包

1117 号共享文件块传输结果控制包

输入:

1002 号注册结果控制包

1004 号 kerberos 认证包

1006 号 kerberos 认证包

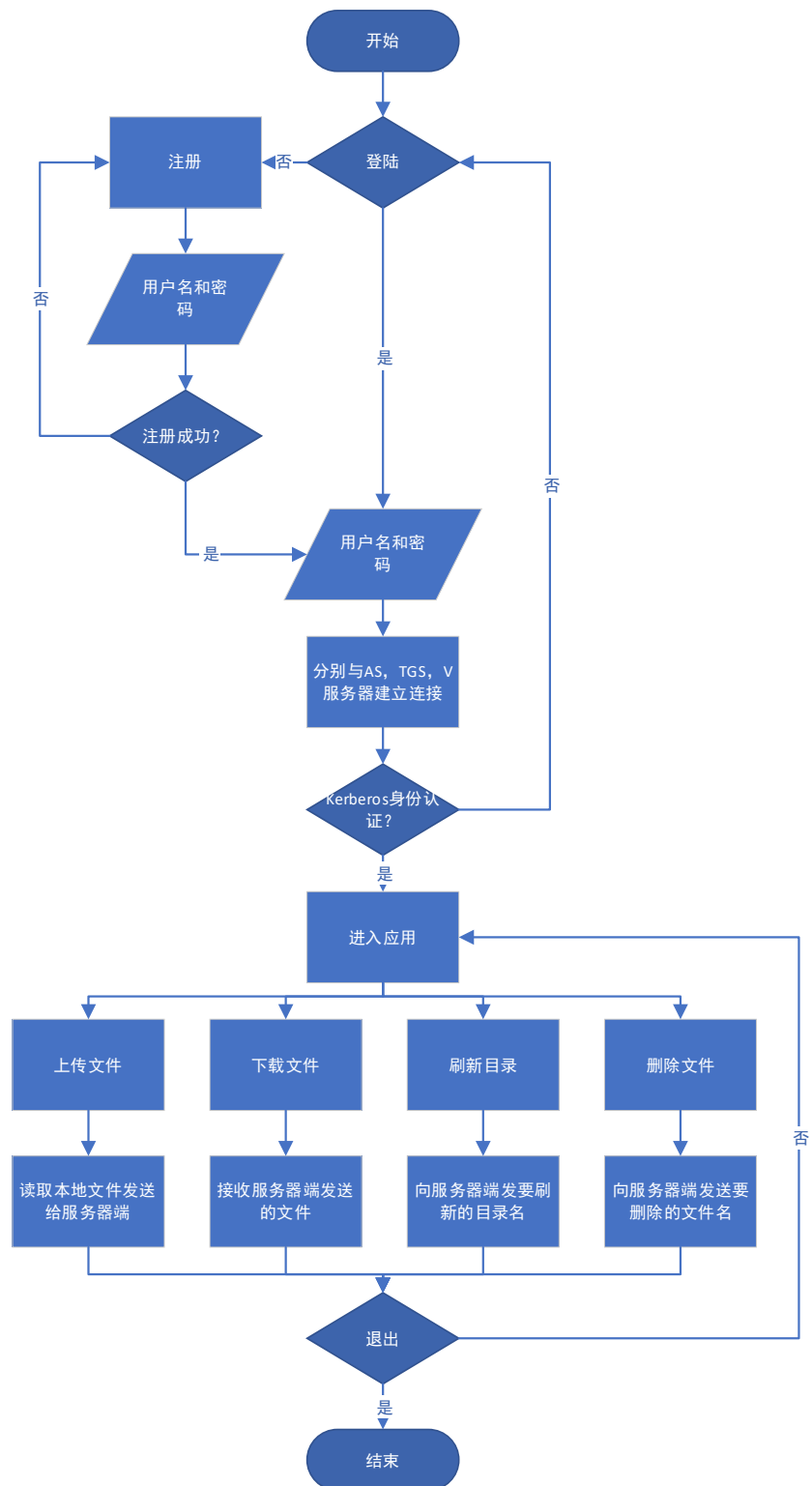
1008 号 kerberos 认证包

1109 号文件块传输结果控制包

1017 号共享文件数据包

1019 号共享文件目录数据包

4.1.4 程序流程



4.1.5 函数接口

//初始化用户进程

public void init()

//kerberos 认证

public boolean verify(String ID_c, String K_c)

//注册

public boolean register(String ID_c, String K_c)

//上传文件

public boolean uploadfile1 (String filename)

//下载文件

public boolean downloadfile1(String filename)

//刷新目录

public String[] getdirectory()

//删除文件

public boolean deletefile(String filename)

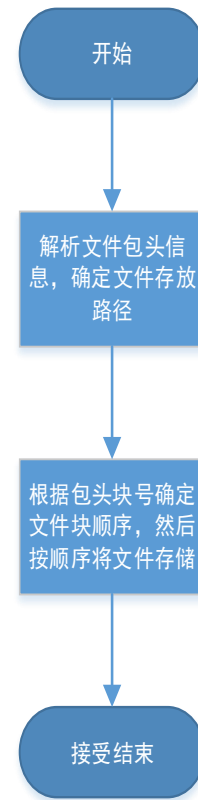
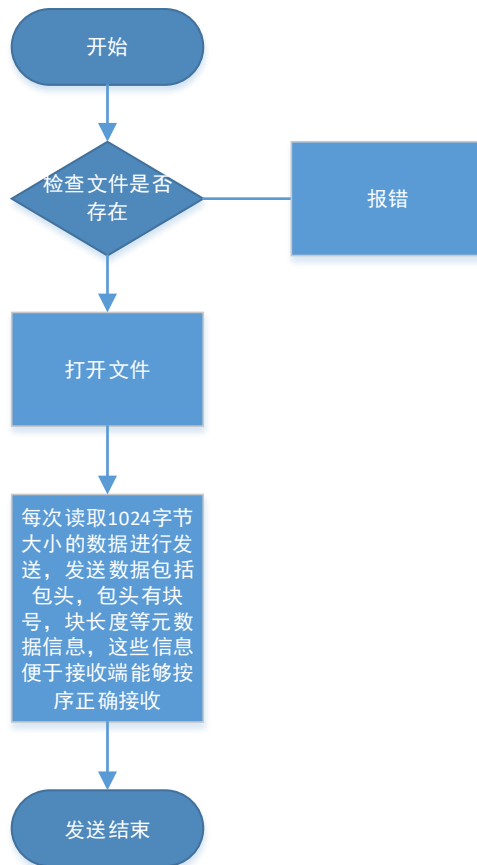
§4.2 文件下载

4.2.1 功能描述

文件于客户端和服务端的互相传输，用户选择文件下载，服务端将指定文件传输给客户端，客户端将文件数据保存到本地，完成文件下载。

4.2.2 功能描述

1. 客户端想要下载时，会先向服务端发送一个请求数据包，这个请求数据包内容包括下载文件的文件名，当前登录客户端的用户 ID 等信息。
2. 服务端会先检查用户是否具有下载该文件的权限（共享文件是公开的，任意用户都可以下载），然后发送一个回应的数据包给客户端（包括文件的路径地址，文件大小等信息）。
3. 客户端收到数据包后就会调用相应的文件传输接口函数，将文件进行传输。
4. 当用户发送或者接收的文件过大，则进行分块处理，然后再进行发送或者接收。（这个逻辑包括在 `upload_File` 和 `download_File` 函数当中）
5. 每次发送文件时，都会分块读取文件，然后再拼接上包头信息（包括块号，块大小等信息），然后接收文件时，会先解析文件块的包头，然后提取出块号信息，再进行文件写入，避免文件次序发生错误。
6. 在文件传输过程中，所有数据传输会在末尾加上校验位，用于检验传输的完整性。如若数据完整性出现问题，立即请求重传，如若数据完整，则给出正常的反馈，请求下一个包数据。（该逻辑在 `upload_File` 和 `download_File` 文件中）。



4.2.3 函数接口

public String Client_Filedownload(String filename,String Kcv)

功能：用户向服务端发送下载文件请求包，请求包生成函数

参数：filename: 请求的文件名

Kcv: C 和 V 认证产生的 session key

返回值：数据包

//服务器解析客户端发来的文件下载请求包

public String Server_GetFiledownload(String cpackage ,String Kcv)

功能：服务端解析客户端发来的文件下载请求包

参数：cpackage: 文件下载请求包

Kcv: C 和 V 认证产生的 session key

返回值：请求包中的各字段

public byte[][] Server_filebyte(String filename,String Kcv)

功能：服务端向用户发送文件数据包

参数：filename: 请求的文件名

Kcv: C 和 V 认证产生的 session key

返回值：文件数据包组

//客户端解析服务端发来文件数据包

public byte[][] Client_getfilebyte(byte[] filedata,String Kcv)

功能：客户端解析服务端发来的文件数据包

参数：filedata: 一组文件数据包

Kcv: C 和 V 认证产生的 session key

返回值：文件数据原字节

public byte[][] filesplit_byte(String filename)

功能：将文件进行分块

参数：filename: 文件绝对路径

返回值：文件字节数据组

public boolean filemerge_byte(byte[][]filedata,String filename)

功能：将文件进行合并

参数：filedata: 文件字节数据组

filename: 合并的文件名

返回值：合并结果

4.2.4 数据包格式

下载共享文件（C→V）

数据包（Kc,v 加密）(1011)	
文件名	
16 字节	

共享文件发给用户（V→C）

数据包（Kc,v 加密）（1017）			
文件名	文件总块数	文件块编号	文件内容
16 字节	1 字节	1 字节	0~109 字节

共享文件块接受情况(C→V)

数据包 (Kc,v 加密) (1117)
状态信息
2 字节 (传输成功 11/传输失败 10/传输结束 00)

§4.3 加解密设计

4.3.1 DES 加密算法

根据 Kerberos 认证 要求，除了包头包尾外，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与服务器之间数据包的传递。

4.3.2 RSA 加密算法

RSA 加密算法用在用户注册时和 AS 数据交流的加密和票据的加密（对方的公钥加密）。

4.3.3 hash 函数

用户口令经过 hash 后存储至数据库，服务器对于用户口令的检查是通过口令的 hash 值与数据库对比得出。本次课设采用 MD5 单向散列算法。

4.3.4 函数接口

DES 加解密

public DES(String key)

功能：构造函数，传入对称密钥

参数：key: 密钥字符串

返回值：无

public String encrypt_string(String message)

功能：加密字符串

参数：message: 要加密的字符串

返回值：加密后的字符串

public String decrypt_string(String message)

功能：解密字符串

参数：message: 要解密的字符串

返回值：解密后的字符串

public byte[] encrypt_byte(byte[] message1)

功能：加密字节数组

参数：message: 要加密的字节数组

返回值：加密后的字节数组

public byte[] decrypt_byte(byte[] message)

功能：解密字节数组

参数：message: 要解密的字节数组

返回值：解密后的字节数组

Hash 函数

public String generateHash(String input)

功能：hash 字符串

参数：input: 要 hash 的字符串

返回值：hash 后的字符串

第五章 运行展示

§5.1 AS 端

用户注册

```
AS server start at:Mon Sep 14 17:07:31 CST 2020
connected with address:192.168.43.172
AS 接收到 Client的报文1001061724740654005967470372924104091931249299977225829289884439221235553333121954657
packet_type:1001

*****
AS收到client请求注册包，包的内容为：
hencjjjj
12345678
数据库驱动加载成功
数据库连接成功
FindIDidc:hencjjjj
FindIDsql:select Kc from myusers where IDc='hencjjjj'
该idc不存在
FindIDidc:hencjjjj
FindIDsql:select Kc from myusers where IDc='hencjjjj'
该idc不存在
Insertidc:hencjjjj
Insertsql:insert into myusers values('hencjjjj','25d55ad283aa400af464c76d713c07ad')
true
注册成功
*****
```

用户登陆

```
connected with address:192.168.43.172
AS 接收到 Client的报文10030037hencjjjj 192.168.43.244 1600074568830
packet_type:1003

*****
AS收到client请求认证包，包的内容为：
hencjjjj
192.168.43.244
1600074568830
AS端验证client成功
数据库驱动加载成功
数据库连接成功
FindIDidc:hencjjjj
FindIDsql:select Kc from myusers where IDc='hencjjjj'
kc:25d55ad283aa400af464c76d713c07ad
K_TGS:default1
k_c_tgs:?.54YbZ
idc:hencjjjj
ip_client:192.168.43.172
ID_TGS:192.168.43.244
*****
```

§5.2 TGS 端

认证报文

TGS 接收到 Client 的报文 10050167192.168.43.49 w5lp0RBXSaNNrQpk+G23rT3BNvUbj55LjxpU4v2ZUY7npN99PF1N+eEg0GVInRVLky5woE21HAISyIA71C9+M2A1H+puj
packet_type:1005

收到 client 请求验证 TGS 包, 包的内容为:

192.168.43.49

\NUR4wP

chen1234

192.168.43.172

192.168.43.244

1600073352464

10000000

chen1234

192.168.43.172

1600073350846

TGS 端验证 client 成功

k_c_v: Htj?IKX

Kc_tgs_Tickettgs: \NUR4wP

EKv: default2

IDv: 192.168.43.49

IDc_Tickettgs: chen1234

ADc_Tickettgs: 192.168.43.172

§5.3 C 端

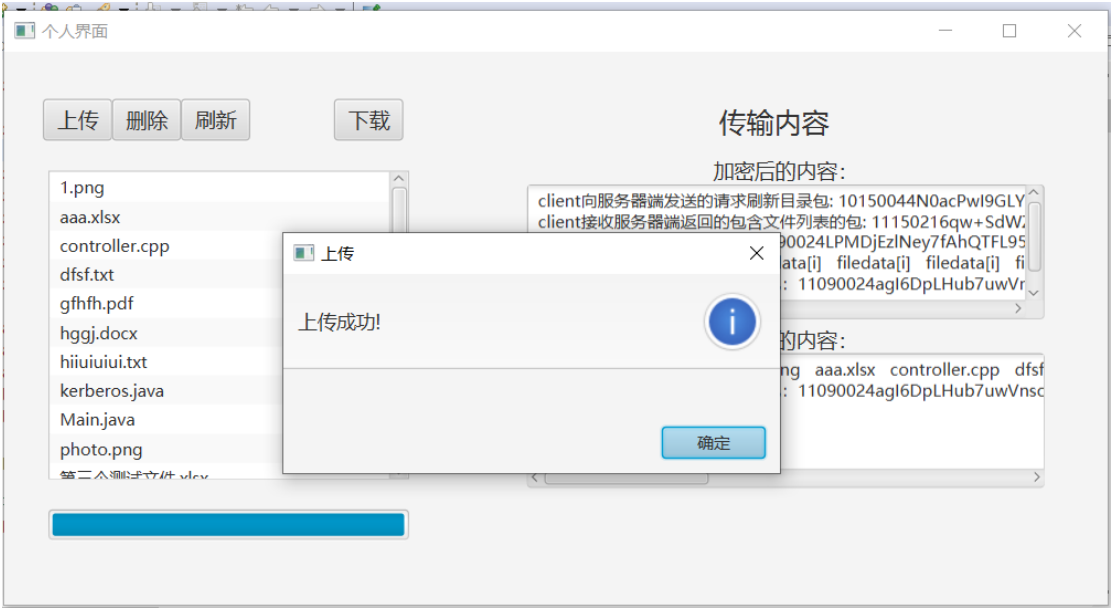
注册



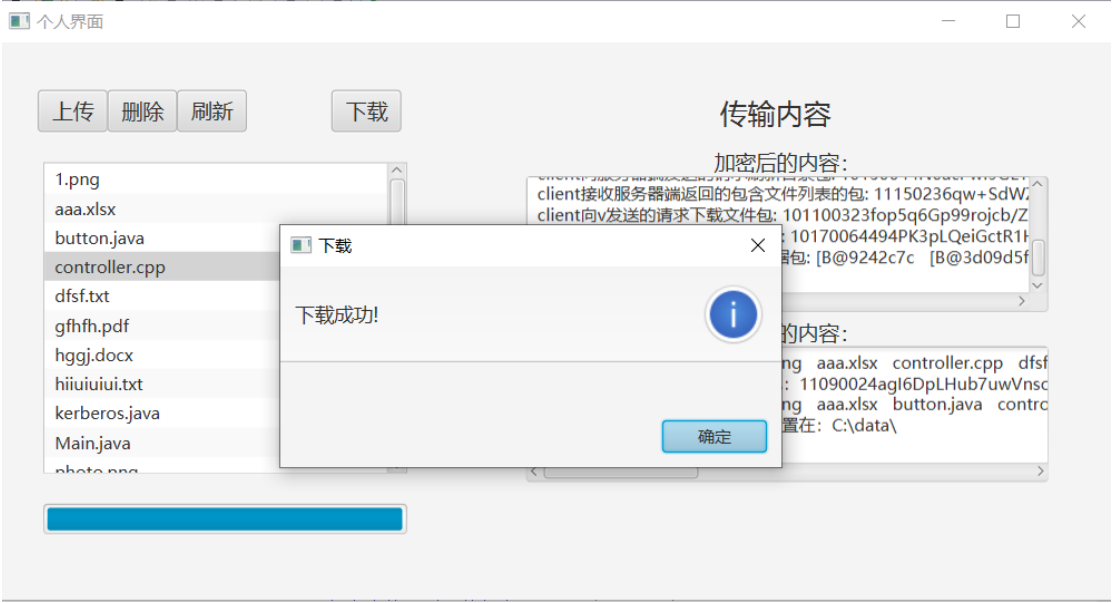
登陆



文件上传



文件下载



文件刷新



文件删除



§5.4 V 端

Kerberos 认证














```
connected with address:192.168.43.172
V 接收到 Client的报文10070153E8u/L8RGbxhNrQpk+G23rT3BNvUbj55LjxpU4v2ZUY7npN99PF1N+aTxxKD+s9Y9ba6QeFQ4X2EipnnW3WhPuWav10d/wo1I 4pQ098n7GoGWIHV6
packet_type:1007
packet_type:1007
IF_VERIFY:0000

*****
*****
收到用户请求验证V的包，包的内容为：
Htj?IKX
chen1234
192.168.43.172
192.168.43.49
1600073353328
10000000
chen1234
192.168.43.172
1600073350873
V端信息验证成功
k_c_v_Tickettgs:Htj?IKX
*****
```

文件上传

```
V 接收到 Client的报文1009LPMDjEz1NexDipNPc16c34AdwfnwufRs
packet_type:1009
packet_type:1009
IF_VERIFY:1111

*****
收到client请求上传文件的1号数据包，该包无实际内容，仅仅包含包类型
收到client请求上传文件的2号数据包，包的内容为：
文件名：button.java
文件总块数：4
1
2
3
用户：192.168.43.172已经上传文件button.java到服务器
*****
```

名称	修改日期	类型	大小
 gfhfh.pdf	2019/9/22 21:18	PDF 文件	5 KB
 1.png	2020/9/14 12:20	PNG 文件	89 KB
 aaa.xlsx	2020/9/14 13:52	Microsoft Excel ...	54 KB
 hggj.docx	2020/9/14 13:53	Microsoft Word ...	13 KB
 第二个测试文件.xlsx	2020/9/14 13:53	Microsoft Excel ...	8 KB
 第三个测试文件.xlsx	2020/9/14 14:35	Microsoft Excel ...	6 KB
 photo.png	2020/9/14 14:43	PNG 文件	330 KB
 dfsf.txt	2020/9/14 14:44	文本文档	0 KB
 kerberos.java	2020/9/14 14:47	JAVA 文件	15 KB
 controller.cpp	2020/9/14 14:49	C++ 源	15 KB
 hiiuiui.txt	2020/9/14 16:18	文本文档	1 KB
 Main.java	2020/9/14 16:18	JAVA 文件	1 KB
 button.java	2020/9/14 16:53	JAVA 文件	4 KB

文件下载

```
V 接收到 Client的报文101100323fop5q6Gp99rojb/ZbToWtjzi1aIdr/  
packet_type:1011  
packet_type:1011  
IF_VERIFY:1111
```

收到client用户请求下载文件的包，包的内容为：

file_name:controller.cpp

文件D:\test\file_disk\controller.cpp已经发送给用户：192.168.43.172

文件刷新

```
V 接收到 Client的报文10150044N0acPwI9GLY3Rpw/Aj0YtmB+VGYcCKxH+Lnw+C4thI8=  
packet_type:1015  
packet_type:1015  
IF_VERIFY:1111
```

收到client用户请求刷新文件的包







目录：D:\test\file_disk\已经成功刷新

名称	修改日期	类型	大小
 1.png	2020/9/14 12:20	PNG 文件	89 KB
 aaa.xlsx	2020/9/14 13:52	Microsoft Excel ...	54 KB
 controller.cpp	2020/9/14 14:49	C++ 源	15 KB
 dfsf.txt	2020/9/14 14:44	文本文档	0 KB
 gfhfh.pdf	2019/9/22 21:18	PDF 文件	5 KB
 hggj.docx	2020/9/14 13:53	Microsoft Word ...	13 KB
 hiiuiui.txt	2020/9/14 16:18	文本文档	1 KB
 kerberos.java	2020/9/14 14:47	JAVA 文件	15 KB
 Main.java	2020/9/14 16:18	JAVA 文件	1 KB
 photo.png	2020/9/14 14:43	PNG 文件	330 KB
 第二个测试文件.xlsx	2020/9/14 13:53	Microsoft Excel ...	8 KB
 第三个测试文件.xlsx	2020/9/14 14:35	Microsoft Excel ...	6 KB

文件删除

V 接收到 Client的报文101200327kuFia3LQGtz1gxV2IAj8sJB5KGRDq2a
packet_type:1012
packet_type:1012
IF_VERIFY:1111

收到client用户请求删除文件的包
文件:D:\test\file_disk\aaa.xlsx已经成功删除

 gfhfh.pdf	2019/9/22 21:18	PDF 文件	5 KB
 1.png	2020/9/14 12:20	PNG 文件	89 KB
 hggj.docx	2020/9/14 13:53	Microsoft Word ...	13 KB
 第二个测试文件.xlsx	2020/9/14 13:53	Microsoft Excel ...	8 KB
 第三个测试文件.xlsx	2020/9/14 14:35	Microsoft Excel ...	6 KB
 photo.png	2020/9/14 14:43	PNG 文件	330 KB
 dfsf.txt	2020/9/14 14:44	文本文档	0 KB
 kerberos.java	2020/9/14 14:47	JAVA 文件	15 KB
 controller.cpp	2020/9/14 14:49	C++ 源	15 KB
 hiiuiui.txt	2020/9/14 16:18	文本文档	1 KB
 Main.java	2020/9/14 16:18	JAVA 文件	1 KB
 button.java	2020/9/14 16:53	JAVA 文件	4 KB

第六章 感想

在本次课设中，作为小组的组长，从初期编写小组的设计文档，网上查询相关的设计文档格式，从整体构思到局部功能，再从局部功能到整体构思，一点点的编写和完善设计文档，避免出现文档的逻辑错误，使每一个功能和模块都更加精炼，避免不必要的冗余，遇到关键问题和组员积极进行交流，交换意见。设计出初稿后提交给导师，老师给出意见，大家再进行讨论，解决存在的问题。整个过程使自己对设计文档的编写有了较为深的认识。

在应用认证实现过程中，在相关加解密约定后，我们组首先分工的是 `kerberos` 认证部分，我负责的是 `C` 到 `TGS` 认证过程的生成包解包函数，在小组充分讨论后函数参数和返回值等后，进行自己解包生成包函数的编写，编写后在自己本机上实验成功，接着小组解包生成包函数汇总后，我们拼接了认证函数和注册函数后进行了联机测试，最终认证和登陆测试成功，那一刻还是挺激动的。

最后是文件应用服务的分工，我负责的是文件下载服务相关包的生成包解包函数，此前我参照网上文件分块和合并的处理，并结合我们文件服务的要求，花费的大量的时间封装出了适合我们的文件分块和合并函数，并改写出了 `DES` 字节加解密的接口函数，用于文件数据包加解密，经过本机调试后，文件恢复正常，真的有种苦尽甘来的感觉。完成分块和合并处理后我编写出了文件下载相关包生成包解包函数，小组合并后，整体文件服务所需要的各种包就写完了。最后 `C` 端要和 `UI` 结合的原因，我设计封装好的功能函数和适当的全局变量供 `UI` 触发事件调用，在联调过程出现了很多问题，大家一起讨论和调试找出问题，提高了团队协作能力和检错能力，加深了对团队编程的理解。

回顾起此课设，在这段日子里，从每一天的早上到半夜，日复一日的完成编程任务、检错和解决错误，虽然很累，但是也学到很多很多的东西，不仅巩固了以前所学过的知识，而且学到了很多在书本上所没有学到过的知识。对网络的认证和加密和安全措施有个更深的理解，更提高了团队协作编程的能力，也熟悉了从理论到实践的过程，专业能力得到进一步的加强！