信息安全project文档

By12302010060彭智

Preoject设计思路：

使用java api中的socket来实现客户端与服务端的通信，使用ObjectInputStream与ObjectOutputStream来实现信息流的封装传递，当客户端启动时，会通过while(true)进入无限循环不停的等待从客户端接收信息，而后服务端会对传递过来的一个封装的数据类进行分析，通过getType()分类操作，再传给某些客户端；客户端也是这样通过while(true)进入循环不停的等待服务器发过来的信息，同样判断传过来的封装类的getType()来分类操作。

针对评分点的特别说明：

Reliable key distribution：在用户注册时，将改用户的密钥设置为”snow”+用户名，可以保证唯一性，同时也可以通过维护用户名来维护密钥。

Reliable account querying：在添加好友界面中可以根据用户名查找指定用户

Secure key exchange：在聊天发起时，源用户告知目的用户其密钥，并与目的用户协商使用统一密钥：”snow”+源用户名+目的用户名

Secure message sending：目的用户收到信息之后会发给源目标已经接收成功

Integrity of message：在加密时对二进制数据加入校验码，在解密时根据校验码判断是否被修改。（校验码为对byte[]中的每一项进行反向加，即溢出时+0x1，最后时对加的总和进行反码，即为校验码）

Efficiency：加密使用的是AES算法，加密速度很快

**程序的正确性：**

Registration and key distribution：注册时密钥设置为snow+用户名，由于用户名唯一，所以可以保证唯一性。

Friending and authentication：添加好友需要告知目的用户，等待其同意，即为身份验证

Key generation and exchange：注册时生成，密钥为snow+用户名，密钥设置为snow+用户名

Message encryption and decryption：信息使用AES加密和解密

Message integrity：在加密时对二进制加入校验码，在解密时根据校验码判断是否被修改。

**Quality of source co de 10%**

File sending 已实现

亮点：使用wireshark可以体现已经加密

接收的Type的类型：

0：上线。

服务端：添加入上线用户列表，判断其在线好友，给在线好友发通知告知其上线

客户端：更新在线好友列表

-1：下线。

服务端：将其从上线用户列表删除

客户端：无

1：聊天。

服务端：判断是否为好友，若不是则设定接收方为发送方返回，若是则发送给接收方

客户端：判断发送方与接收方的Name是否相同，相同则弹出（他还不是你的好友，不能发起聊天），若不同则为好友发过来的信息，对其中的info进行解密打印出来。

2：请求接收文件。

服务端：判断是否为好友，若不是则设定接收方为发送方返回，若是则发送给用户

客户端：判断发送方与接收方的Name是否相同，相同则弹出（他还不是你的好友，不能发送文件）

3：确定接收文件。

服务端：发送文件给客户端

客户端：目标用户愿意接收文件，源用户开始读取本地文件并发送到网络上

4：中途取消接收文件。

服务端：发送给目标用户

客户端：打印取消信息

5：登陆。

服务端：核对用户名密码，返回告知用户是否正确

客户端：正确则跳转到聊天界面，否则提示密码不正确

6：注册。

服务端：判断用户是否存在，不存在则将其信息加入，返回给用户注册成功

客户端：若注册成功则返回登陆界面

7：添加好友。

服务端：若info为addFriend，则判断是否已经是好友，若不是则通知目标用户：源用户想添加其为好友；若info为addFriendBack，则加入服务端维护的所有用户的好友列表中，并返回源用户。

客户端：若info为addFriend，则弹出提示：源用户想添加其为好友，若同意，则设置info为addFriendBack，返回客户端；若info为addFriendBack，则直接添加入在线好友列表中。

用户聊天窗口事件：

发送：将输入框中的String使用AES算法加密发送给服务端

添加好友：弹出新窗口，列表显示所有在线用户，支持搜索指定在线用户

文件传送信息栏：动态显示文件传送百分比

好友列表：选择聊天、传送文件的在线好友

关闭（右上角的X）：向服务端发送关闭请求

用户登录界面：

昵称

密码

登录：向服务端发送登录请求

注册：跳转到注册界面

用户注册界面：

昵称

用户密码

确认密码

返回：返回到登陆界面

注册：向服务端发送注册请求