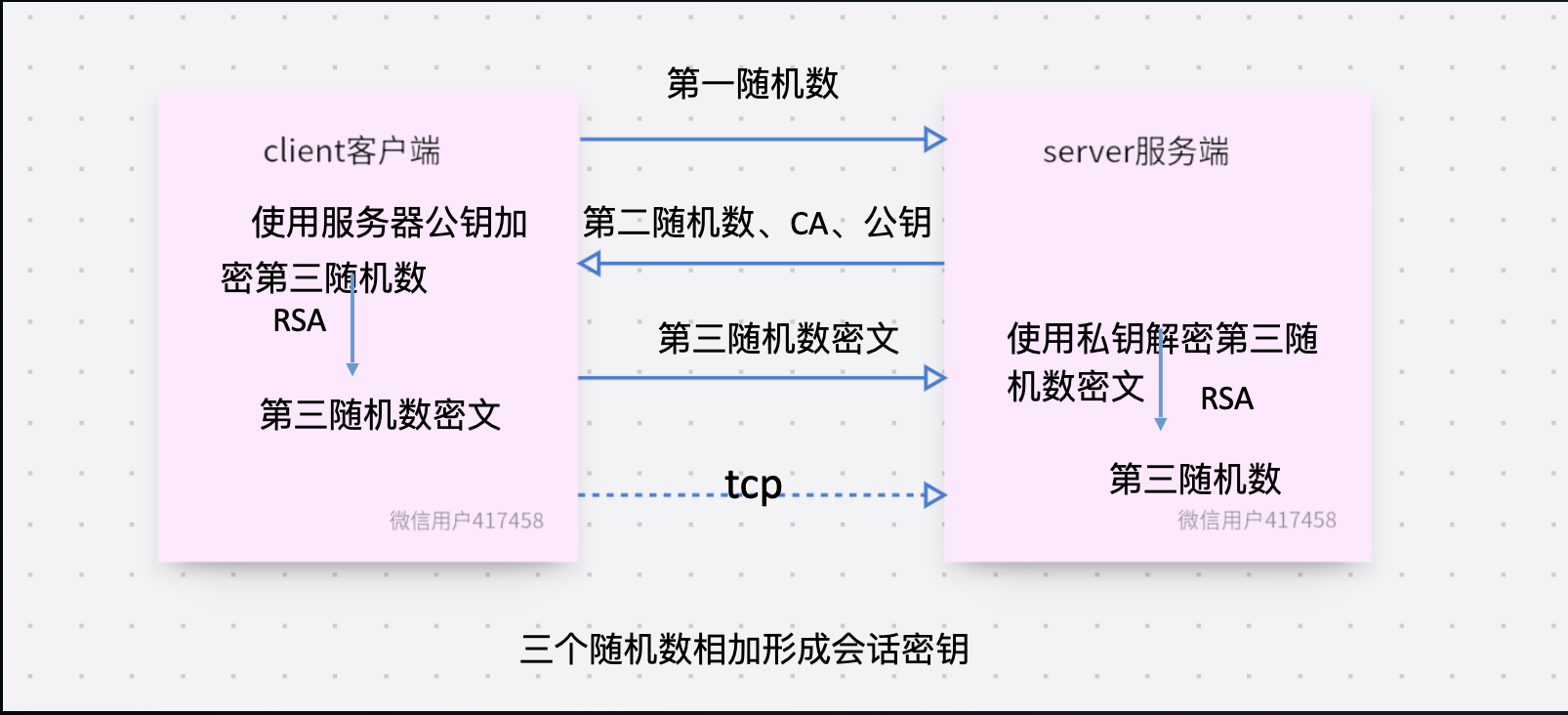
提供安全认证、加密传输和加密存储功能的网络加密磁盘空间

一、任务分工说明

| 姓名 + 学号 | 任务分工 |
| --- | --- |
| 邓致远 2020211820 | 负责协调、组织人员，分配成员任务，监督项目进展；讲解Git-hub协作使用、统一配置安装数据库；解决开发过程中遇到的小bug和小问题。 |
| 喻志威 2021212003 | 负责服务器部分代码的搭建和前后端的协调集成；设计数据库表；解决联调过程中出现的bug和缺陷，对已有代码进行完善优化，确保整个项目成功运行。 |
| 孔祥阳 2021211813, 刘怡文 2020212535, 何一可 2021211945 | 负责后端信道加密、文件加密、以及群组加密部分的代码，设计讨论项目使用的加密方案编码实现，并与前端和服务器人员对接。 |
| 闫宇 2021211928, 熊兵2021211948 | 负责前端UI界面的设计与开发，包括登录注册界面，个人主页界面和群功能界面；同时与后端和服务器进行对接并解决联调过程中出现的问题。 |
| 黄棋烨 2021211850, 王喻骢 2021211934 | 负责文件上传下载功能和群密钥功能的实现与集合；同时协助前后端讨论具体实现思路，完成部分功能的代码开发。 |
| 彭琬荟 2021211955 | 负责项目文档和汇报PPT的编写与整理，将项目代码转换为文字和图表的直观形式并呈现出来。 |

二、作品技术原理介绍

1.三次握手建立安全信道

图示：

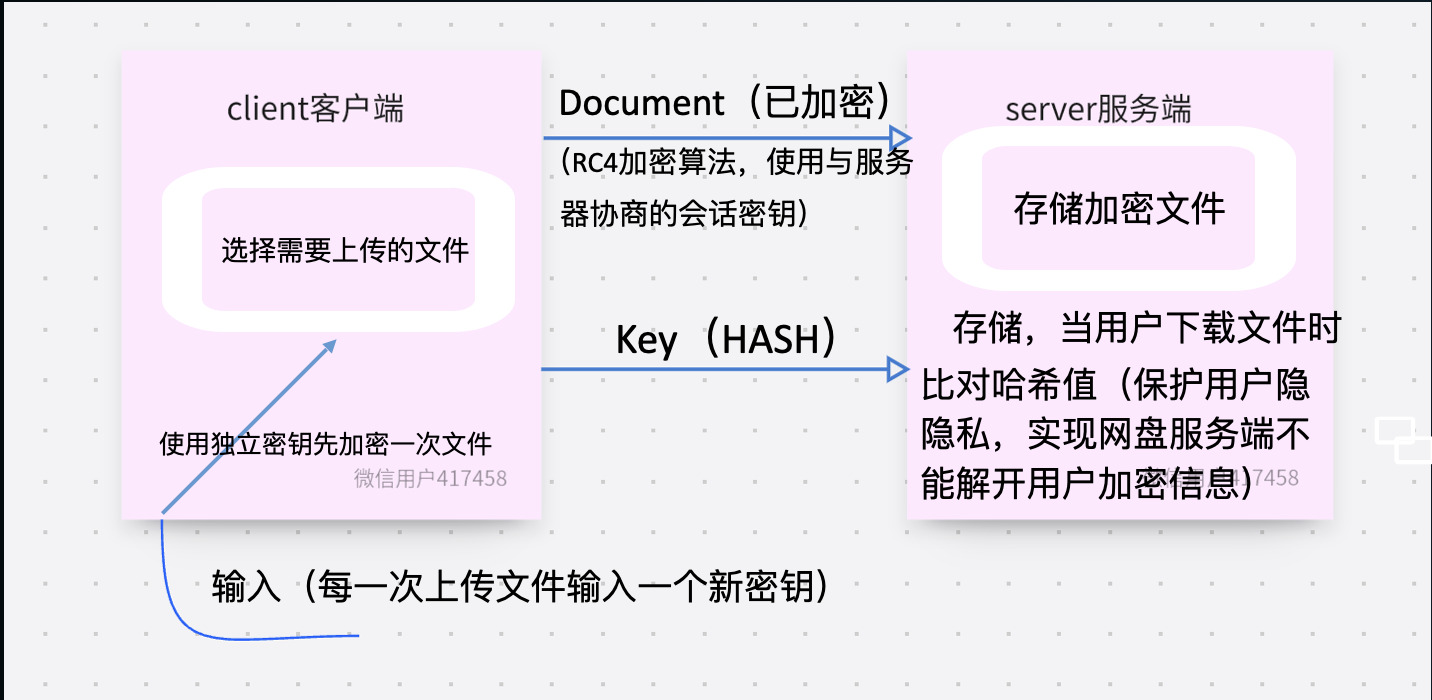
原理解释：

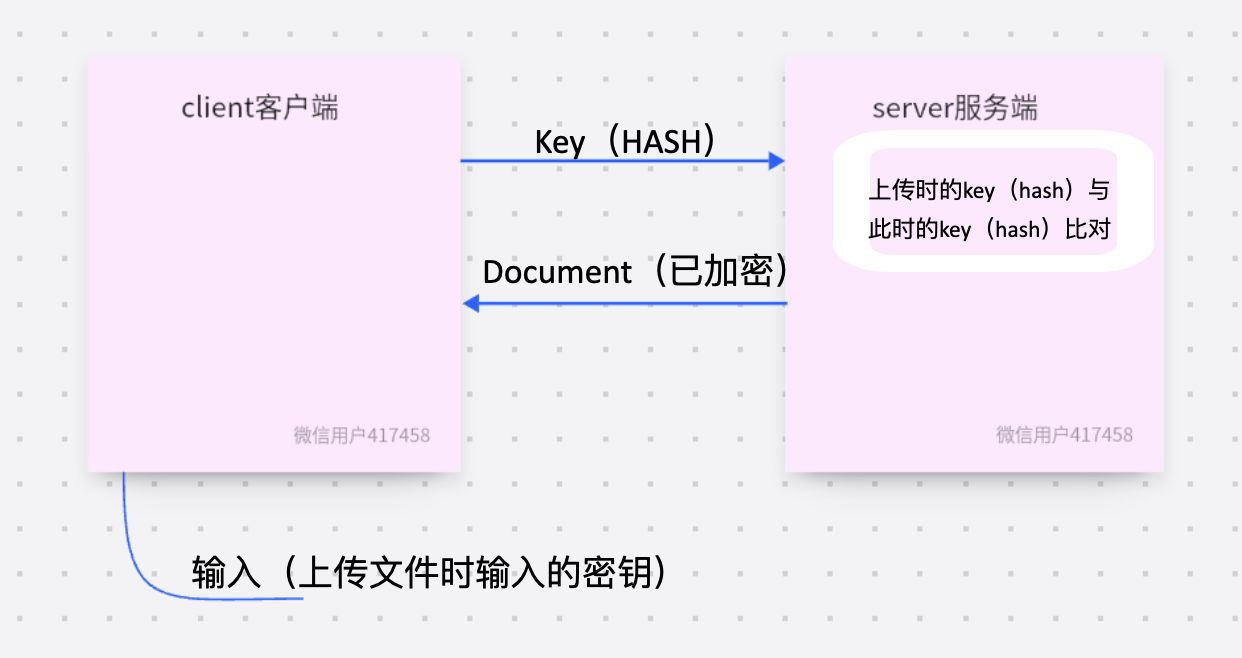
为了建立一个安全、可信的通信信道，我们小组在该项目中采用了“三次握手”的方式。首先client向server以明文形式发送一个第一随机数，server收到第一随机数后会向client以明文形式发回一个第二随机数，还有CA以及服务器公钥，client使用服务器公钥以RSA加密算法加密第三随机数得到第三随机数密文，将其发给server，server使用私钥解密得到第三随机数。此时client、server双方均有第一、第二、第三随机数的明文，将这三个随机数相加形成会话密钥，有效防范了中间人攻击，确保了整个通信过程的安全性、可靠性。

2.利用哈希函数值进行比对，有效保护客户隐私

图示：

上传过程：



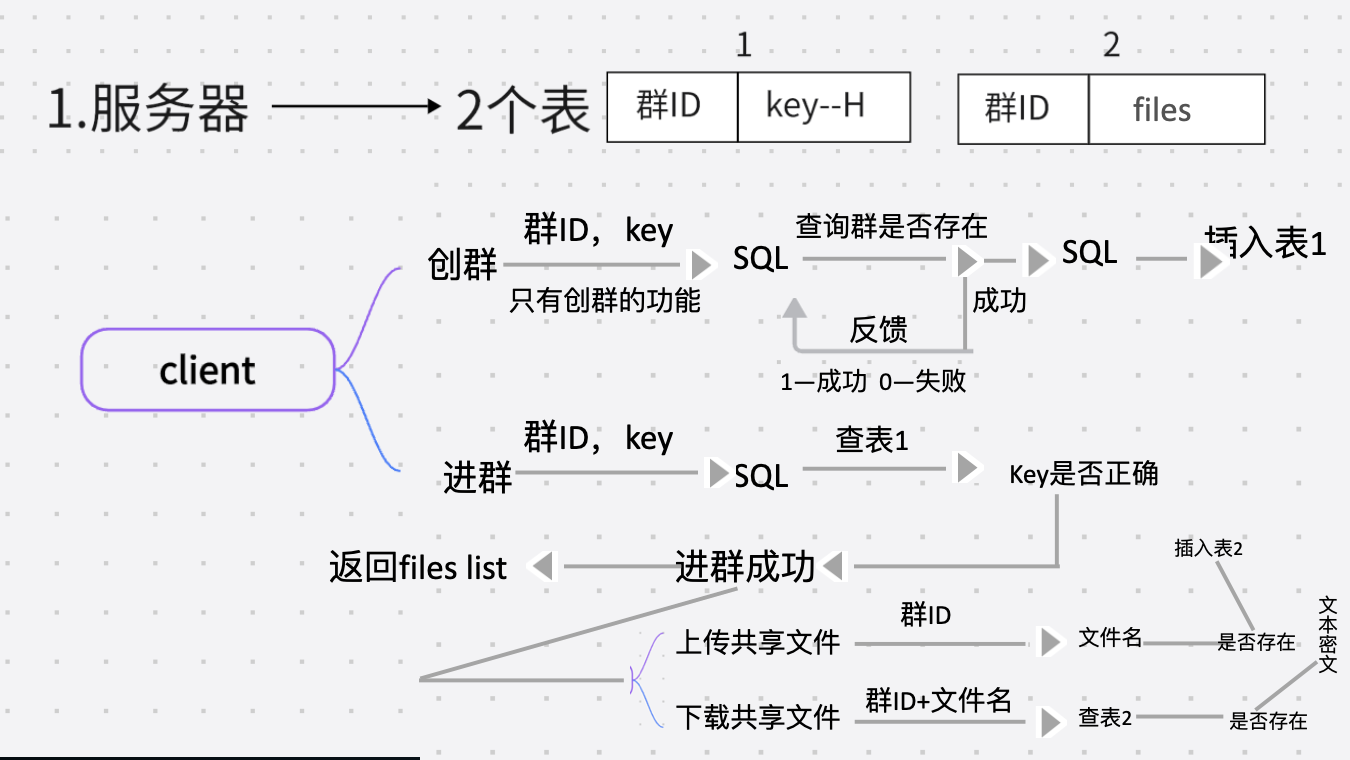
下载过程：

原理解释：

当一个客户选择上传文件时，会出现输入新密钥的弹窗，输入密钥后会使用这个独立密钥先加密一次文件，实现了“一次一密”的功能，再通过安全信道，使用建立安全信道时client与server协商好的会话密钥，运用RC4加密算法发送已被加密的文件，server收到文件后进行保存。

与文件一同发送的还有户选择上传文件时输入新密钥的哈希值，当用户下载文件时，需要再此输入上传时输入的密钥，并将密钥的哈希值发送给server，经server比对后再返回加密文件。

在上传、下载中设计输入新密钥，并发送密钥哈希值这一环节时利用了哈希函数的以下性质：①计算简单,但从输出反推输入极其困难。也就是说知道输出很难推断出原始输入；②输入改变一点,输出可能完全不同。即抗碰撞性,这也是衡量哈希算法质量的重要标准；③具有确定性,同一个输入总能得到同样的输出；④哈希值不会泄露输入的数据信息。只能用来检验数据是否被篡改过；⑤计算速度快,对任意长度的输入都能很快得到固定长度的输出；⑥即使是知道算法细节,也很难找到两个不同输入导致同样输出的情况。 这样可以保证server也不能知道文件内容，合理地保护了客户隐私。

3 .建表与群组、群文件的上传、下载

原理解释：

在server建立两个表，属性如上图所示。client端有创群和进群两种功能，当用户选择创群时，需要输入群ID和一个新密钥，通过SQL查询群是否存在，若存在则创群失败，若群ID不存在，则在表1中插入一个新的元祖；当用户选择进群，则需要输入对应群的群ID和密钥，根据表1查询群是否存在，密钥是否正确，若群ID，key均正确则进群成功。客户进群成功后，可以选择上传或者是下载共享文件，当用户选择上传共享文件，则需输入群ID，查询该群内是否已经保存有相同文件名的文件，若已存在则返回失败，若不存在，则能成功上传；当用户选择下载共享文件，需发送群ID和文件名，经过查询表2，若该文件存在则可得到文本密文，文件不存在则返回失败。

在这里可以把群看作是一个抽象意义的“人”，群密钥相当于是一个人上传文件时输入的新密钥，通过这个群密钥划分、规定出了一片共享空间。

值得注意的是，server保存的是群密钥的hash值，原理与保存客户上传文件时输入密钥的哈希值一致，可以保证server无法“看见”群内文件

三、概要设计报告

1. 系统概述

#### 1.1 目标

开发一个安全的网络存储空间系统，支持用户登录认证、数据加密传输和加密存储，以满足互联网、无线网络和移动通信等用户的需求。

#### 1.2 主要功能

1. 用户管理：支持多用户登录和管理，每个用户拥有自己的独立存储空间。
2. 登录认证：实现安全的登录认证机制，支持多种认证方式，包括口令、数字证书、短信、Email、生物特征识别和微信/支付宝身份识别。
3. 数据传输：采用加密传输机制，确保数据传输的机密性和完整性，防止篡改和中间人攻击。
4. 存储管理：实现本地加密存储，使用一次一密加密传输机制，支持安全的加密数据找回。
5. 用户组管理：用户可以组成多个用户组，每个用户组有独立的存储空间，不允许跨用户组访问。
6. 多终端登录：支持用户在多个终端上登录自己的存储空间。
7. 数据解密：只有创建存储空间的用户能够解密其存储空间的加密数据，服务器无法解密用户数据。

2. 系统架构

#### 2.1 客户端

客户端主要功能包括用户登录认证、数据加密传输和解密、本地数据加密存储。

* 登录认证：根据用户选择的认证方式进行认证，使用安全的口令找回机制。
* 数据传输：采用自行实现的一次一密加密传输机制，确保数据传输的安全性。
* 本地存储：数据在本地存储时进行加密，只有用户可以解密数据。

#### 2.2 服务器

服务器主要功能包括用户管理、用户组管理、认证、数据传输管理、存储管理。

* 用户管理：管理用户信息，包括用户组划分和权限管理。
* 用户组管理：用户可以创建和管理用户组，每个用户组有独立的存储空间。
* 认证：负责验证用户的登录认证请求，支持多种认证方式。
* 数据传输管理：管理数据的加密传输，确保数据的机密性和完整性。
* 存储管理：管理用户存储空间，包括数据的加密和解密。

3. 技术实现

#### 3.1 网络编程

使用 Python 的 socket 编程技术建立客户端和服务器之间的通信。

#### 3.2 加密算法

采用对称加密算法（如 AES）和非对称加密算法（如 RSA）实现数据的加密传输和存储。对称加密用于加密传输中的数据，非对称加密用于加密存储中的密钥和认证。

#### 3.3 安全认证

实现多种登录认证方式，包括口令、数字证书、短信、Email、生物特征识别和微信/支付宝身份识别。为口令认证提供找回机制，确保找回口令后仍能解密数据。

#### 3.4 数据传输

自行实现基于会话的一次一密加密传输机制，防止数据泄露和中间人攻击。

#### 3.5 存储管理

数据在本地存储时进行加密，只有用户能够解密数据。服务器无法解密用户数据，确保数据的安全性。

4. 测试环境

在一台计算机上安装该系统作为服务器，共享一个安全存储空间。其他计算机可以通过互联网、无线网络或移动通信等方式连接并安全登录该网络存储空间，实现数据信息的安全传输和加密存储。

四、详细设计报告

client文件夹下的app1.py文件

#### 概要设计：

#### 基于Tkinter的登录注册系统的实现。系统提供了用户名密码登录和邮箱验证码登录两种方式。在登录过程中，会将用户输入的信息发送给服务器进行验证，并根据验证结果弹出相应的提示框。注册功能会将用户输入的信息发送给服务器进行注册，并根据注册结果弹出相应的提示框。忘记密码功能会发送验证码给用户手机，并根据用户输入的验证码和新密码进行密码找回操作。

#### app1.py文件流程：

#### 图片 2

| 函数名 | 描述 | 参数 | 返回值 | 调用方式 |
| --- | --- | --- | --- | --- |
| send\_message | 用于向服务器发送消息。 | safe\_socket（安全的 socket 连接对象）和 message（要发送的消息对象）。 | 无 | 在登录、注册、密码找回等函数中，将消息对象作为参数调用该函数，发送消息给服务器。 |
| receive\_message | 用于接收服务器返回的消息。根据消息类型进行条件判断。 | safe\_socket（安全的 socket 连接对象）。 | 服务器返回的消息。 | 在登录、注册、密码找回等函数中，调用该函数接收服务器返回的消息。 |
| login | 创建登录窗口并绑定登录函数。 | 无 | 无 | 在主循环中点击登录按钮时调用该函数，执行用户登录逻辑。 |
| login\_by\_username | 用户名密码登录函数，用于发送用户名和密码信息给服务器进行验证。 | 无 | 无 | 在用户名密码登录窗口中，点击登录按钮时调用该函数，接收服务器返回的消息并判断是否成功登录。如果成功，初始化 PersonalInfoWindow 类进入主页，否则返回错误信息。 |
| login\_by\_email | 邮箱验证码登录函数，用于发送邮箱和验证码信息给服务器进行验证。 | 无 | 无 | 在邮箱验证码登录窗口中，点击登录按钮时调用该函数，接收服务器返回的消息并判断是否成功登录。如果成功，初始化 PersonalInfoWindow 类进入主页，否则返回错误信息。 |
| get\_email\_verification\_code | 获取邮箱验证码函数，用于请求服务器发送邮箱验证码。 | 无 | 无 | 在邮箱验证码登录窗口中，点击获取验证码按钮时调用该函数，接收服务器返回的消息并判断是否成功发送验证码。 |
| register | 创建注册窗口并绑定注册函数。 | 无 | 无 | 在主循环中点击注册按钮时调用该函数，执行用户注册逻辑。 |
| register\_user | 用户注册函数，用于发送用户名、密码、邮箱和电话号码信息给服务器进行注册。 | 无 | 无 | 在注册窗口中，点击注册按钮时调用该函数，接收服务器返回的消息并判断是否成功注册。如果成功，初始化 PersonalInfoWindow 类进入主页，否则返回错误信息。 |
| forgot\_password | 创建忘记密码窗口，要求用户输入手机号码、新密码和手机验证码，并绑定获取验证码和确认修改函数。 | 无 | 无 | 在主循环中点击忘记密码按钮时调用该函数，执行忘记密码逻辑。 |
| get\_phone\_verification\_code | 获取手机验证码函数，用于请求服务器发送手机验证码。 | 无 | 无 | 在忘记密码窗口中，点击获取验证码按钮时调用该函数，接收服务器返回的消息并判断是否成功发送验证码。 |
| forget\_the\_password | 密码找回函数，用于发送手机号码、验证码和新密码信息给服务器进行密码找回。 | 无 | 无 | 在忘记密码窗口中，点击确认修改按钮时调用该函数，接收服务器返回的消息并判断是否成功修改密码。 |

* 加密部分函数分析

#### communication.py

#### 这个文件创建了一个EncryptedSocket类，包含了三类方法：用于实现类似TLS安全信道建立的server\_shakehands和client\_shakehands两种握手方法；建立好安全信道，获取了信道上会话密钥后的发送和接收方法send\_encrypt和recv\_decrypt;在TLS握手中检验证书有效性的certificate\_permit方法。

#### EncryptedSocket类：

#### \_\_init\_\_(self, socket, address=None, port=None):该方法对address进行判断，若address为默认值None，则判定调用者为客户端并调用client\_shakehands函数，若不为None，则判定调用者为服务器并调用server\_shakehands函数

##### 参数：socket是已经与通信对象建立好连接的套接字，

##### 返回值：无

##### 调用方式：在初始化EncryptedSocket类时调用

#### server\_shakehands(self):首先调用create\_rsa\_key方法生成服务器的公私钥，然后建立套接字等待客户端连接，如果接收到的消息不为空，则对消息按规定的格式进行解析，若消息头为Client\_Hello，则接收客户端传来的第一随机数，并生成第二随机数，然后将第二随机数，服务器的数字证书与服务器公钥发送给对应客户端；若消息头为client\_shakehands\_done，则解密客户端发来的预主密钥，并与第一随机数第二随机数拼接在一起生成会话密钥，然后发送server\_shakehands\_done消息告诉客户端握手结束

##### 参数：无

##### 返回值：无

##### 调用方式：EncryptedSocket类初始化时调用

#### client\_shakehands(self): 首先发送Client\_Hello消息与客户端本地生成的第一随机数，然后等待服务器的响应，若响应消息头为Server\_Hello，则接收服务器传来的第二随机数与证书，然后对证书进行验证，不通过抛出异常，通过则接收服务器公钥，并用该公钥对本地生成的预主密钥进行加密，以client\_shakehands\_done为消息头将预主密钥发送给服务器；若消息头为server\_shakehands\_done则直接退出循环结束该函数

##### 参数：无

##### 返回值：无

##### 调用方式：EncryptedSocket类初始化时调用

#### send\_encrypt(self, data:bytes):在通信时，规定发送的内容要么为铭文普通信息（字符串类型），要么为加密文件的密文字节流。若是前者，则需编码为utf-8后发送，若是后者，直接单独发送。无论如何传入的均为字节流，均会在该方法中再次被使用握手确定的信道上密钥进行加密，加密后的密文会通过socket原生的send方法发送

##### 参数：待发送bytes类型的数据data

##### 返回值：无

##### 调用方式：在通信过程中，会使用EncryptedSocket建立一个类，发送内容时调用这个类的本方法，传入需发送内容即可实现发送

#### recv\_decrypt(self):建立好安全信道后，所有接收到的内容全部用会话密钥进行了一次一密。本方法在接收到对方发来的消息后，直接使用会话密钥进行一次解密，并且返回，而解密出的内容可能是普通信息，也可能是加密后文件字节流。这个方法不关心之后的操作，终端在接收到信息后自行确认对接收到的内容应进行何种操作。

##### 参数：无

##### 返回值：原信息在调用send\_encrypt方法前的形式

##### 调用方式：在监听时调用，将接收到的东西返回，用户或服务器自行进行之后的操作。

#### certificate\_permit(self, cert\_data): 该方法接受一个包含X.509证书数据的字符串并判断该证书数据是否有效。该方法的流程如下：首先，将传入的证书数据编码为字节数据。然后，将字节数据解析为X.509证书对象。接着，检查证书是否在有效期内，如果不在有效期内，则输出相应的消息。最后，如果证书在有效期内，尝试验证证书的签名，如果验证成功，则输出有关证书的信息，否则输出验证失败的消息。其中各个方法的分析如下：

#### （1）接收证书数据：

#### certificate\_data = cert\_data.encode(): 这一行将传入的cert\_data字符串编码为字节数据，以便后续的处理。

#### certificate = x509.load\_pem\_x509\_certificate(certificate\_data, default\_backend()): 这一行使用cryptography库的x509模块加载传入的PEM格式的X.509证书数据，将其解析为X.509证书对象，并将其存储在certificate变量中。

#### （2）验证证书有效期:

#### current\_time = datetime.datetime.utcnow(): 获取当前的UTC时间，用于后续验证证书是否在有效期内。

#### if current\_time < certificate.not\_valid\_before or current\_time > certificate.not\_valid\_after:: 检查当前时间是否在证书的有效期范围内。如果当前时间早not\_valid\_before或晚not\_valid\_after，则证书被认为无效并输出一条消息表明证书已过期或尚未生效。

#### （3）证书验证:

#### certificate.public\_key().verify(...): 使用证书的公钥来验证证书的签名是否有效。如果验证成功，表示证书有效，将输出一条消息，显示证书颁发者、颁发者、以及有效期开始和结束时间。如果验证失败，将捕获异常，并输出一条包含错误信息的消息，表示证书验证失败。

#### - 参数：一个包含X.509证书数据的字符串

#### - 返回值：无

#### - 调用方式：在tls握手中由客户端调用该方法来对服务器证书进行验证

#### 

#### encryption\_utils.py

#### 这个文件的内容是加密时调用的函数，主要包含了三大类：实现类似tls的安全信道中调用的RSA相关函数；在加密信道中以及对文件进行加密的RC4相关函数；一些其他函数，如获取hash，获取随机数等。

#### RSA部分

#### create\_rsa\_key():该方法首先调用Crypto.Random.new().read()方法得到一个随机数生成器，然后将生成器传入Crypto.RSA.generate()方法生成一对2048位的公私钥，最后将公钥和私钥分别导出并返回

##### 参数：无

##### 返回值：rsa算法的公钥和私钥

##### 调用方式：在进行TLS握手时由服务器端调用

#### rsa\_encrypt(data, pub\_key):调用Crypto.RSA.importKey()函数导入公钥pub\_key，用该公钥创建一个PKCS1\_v1\_5加密对象，再用这个加密对象对明文data进行加密，最后返回base64编码后的密文

##### 参数：待加密的明文字节流data，用来加密的公钥字节流pub\_key

##### 返回值：密文字节流data

##### 调用方式：在TLS握手过程中，由客户端调用，用服务器公钥对预主密钥进行RSA加密

#### rsa\_dcrypt(data, priv\_key): 调用Crypto.RSA.importKey()函数导入私钥priv\_key，用该私钥创建一个PKCS1\_v1\_5解密对象，再用这个解密对象对base64解码后的密文data进行解密，最后返回明文

##### 参数：待解密的密文字节流data，用来解密的私钥字节流priv\_key

##### 返回值：明文字节流data

##### 调用方式：在TLS握手过程中，由服务器调用，用服务器私钥对客户端传来的预主密钥进行RSA解密

#### RC4部分

#### rc4\_key\_schedule(seed\_key\_str):传入一个字符串类型的种子密钥，返回列表类型的RC4 S盒

##### 参数：str类型的加密密钥

##### 返回值：list类型的s\_box

##### 调用方式：与 generate\_rc4\_keystream，rc4\_en\_de\_crypt函数一起调用，常常需要将他们三个一起调用来实现加密

#### generate\_rc4\_keystream(S: list, data\_length: int)传入一个list类型的S盒和需加密的内容的长度（这个需加密的内容格式是字节流bytes）

##### 参数： list类型的s\_box，int类型的需加密内容长度data\_length

##### 返回值：bytes类型的字节流

##### 调用方式：调用完rc4\_key\_schedule生成s\_box后，用该函数生成密钥字节流

#### rc4\_en\_de\_crypt(data: bytes, key: bytes)：传入字节流以及密钥流，进行按位异或的加解密。

##### 参数：字节流类型的待加/解密数据data，字节流类型的RC4密钥流

##### 返回值：bytes类型的加/解密后的数据

##### 调用方式：key\_schedule，generate\_rc4\_keystream函数调用完后生成了密钥流，传入密钥流和待加/解密数据，返回明文或密文字节流

#### rc4\_file(file\_path, seed\_key:str)：加密文件。传入待加密文件的绝对路径file\_path和进行加密的密钥seed\_key,依次调用以上三个函数将选定的文件进行加密，返回字节流类型的加密数据。

##### 参数：str类型的file\_path,str类型的加密密钥

##### 返回值：bytes类型的加密文件数据

##### 调用方式：考虑到加密文件的需求相对固定，为文件的加密单独写了一个函数，将较为独立的三个RC4函数在本函数中调用，实现接口的方便调用，传入的两个变量易于获得，返回的加密数据直接发送即可。

#### 其他部分

#### get\_hash(data):对传入的参数data进行sha256哈希运算，返回哈希值

##### 参数：待哈希的字节流data

##### 返回值：哈希运算后的字节流data

##### 调用方式：在生成rc4的密钥流前调用，获得种子密钥的哈希值；在创建群组和登录群组时调用获得群密码的哈希值；在下载文件时调用获取加密文件密钥的哈希值

#### get\_random\_number():生成长度为16，字符范围为0-9的随机字符串

##### 参数：无

##### 返回值：字符串number

##### 调用方式：在TLS握手过程中，通信双方调用，分别用来生成第一随机数，第二随机数和预主密钥

#### create\_session\_key(a, b, c):传入第一随机数，第二随机数和预主密钥，并将它们按顺序拼接后返回

##### 参数：依次为第一随机数，第二随机数，预主密钥

##### 返回值：会话密钥

##### 调用方式：TLS握手过程中，由服务器端调用，生成协商好的会话密钥communication.py

##### communication.py文件分析

| 函数名 | 描述 | 参数 | 返回值 | 调用方式 |
| --- | --- | --- | --- | --- |
| \_\_init\_\_(self, socket, address=None, port=None) | 初始化 EncryptedSocket 类，根据 address 判断调用者是客户端还是服务器。 | socket（已建立连接的套接字）, address（可选，通信地址）, port（可选，通信端口） | 无 | 在初始化 EncryptedSocket 类时调用。 |
| server\_shakehands(self) | 服务器进行 TLS 握手，生成公私钥，等待客户端连接并建立安全信道。 | 无 | 无 | 在 EncryptedSocket 类初始化时由服务器调用。 |
| client\_shakehands(self) | 客户端进行 TLS 握手，发送 Client\_Hello 消息，验证服务器证书，建立安全信道。 | 无 | 无 | 在 EncryptedSocket 类初始化时由客户端调用。 |
| send\_encrypt(self, data:bytes) | 通过安全信道发送加密后的数据。根据是否为文件字节流，进行加密和发送。 | data（待发送的字节流数据） | 无 | 在通信时使用 EncryptedSocket 类的实例调用。 |
| recv\_decrypt(self) | 通过安全信道接收加密的数据，解密后返回原始数据。 | 无 | 原始数据 | 在监听时使用 EncryptedSocket 类的实例调用。 |
| certificate\_permit(self, cert\_data) | 验证 X.509 证书的有效性。检查证书有效期和验证签名。 | cert\_data（X.509 证书数据） | 无 | 在 TLS 握手中由客户端调用该方法对服务器证书进行验证。 |

encryption\_utils.py

| 函数名 | 描述 | 参数 | 返回值 | 调用方式 |
| --- | --- | --- | --- | --- |
| create\_rsa\_key() | 创建 RSA 公私钥对。 | 无 | RSA 公钥和私钥 | 在 TLS 握手时由服务器端调用以生成公私钥对。 |
| rsa\_encrypt(data, pub\_key) | 使用 RSA 公钥对数据进行加密。 | data（待加密的数据），pub\_key（RSA 公钥） | 加密后的数据（密文字节流） | 在 TLS 握手时由客户端调用，使用服务器公钥对预主密钥进行 RSA 加密。 |
| rsa\_decrypt(data, priv\_key) | 使用 RSA 私钥对数据进行解密。 | data（待解密的数据），priv\_key（RSA 私钥） | 解密后的数据（明文字节流） | 在 TLS 握手时由服务器端调用，使用服务器私钥对客户端传来的预主密钥进行 RSA 解密。 |
| rc4\_key\_schedule(seed\_key\_str) | 生成 RC4 S 盒，根据种子密钥字符串。 | seed\_key\_str（RC4 种子密钥字符串） | RC4 S 盒（列表） | 用于生成 RC4 密钥流的一部分，通常与 generate\_rc4\_keystream 和 rc4\_en\_de\_crypt 一起使用。 |
| generate\_rc4\_keystream(S: list, data\_length: int) | 生成 RC4 密钥流，用于加解密数据。 | S（RC4 S 盒），data\_length（需加密内容长度） | RC4 密钥流（字节流） | 用于 RC4 加解密的密钥生成，通常与 rc4\_key\_schedule 和 rc4\_en\_de\_crypt 一起使用。 |

#### Personal\_Info\_Window.py文件分析

#### **文件概述：**

#### 主要实现了登录注册后的用户使用界面，以及上传、下载以及群组相关功能在客户端的处理逻辑，并提供了发送和接收数据的接口。

#### 在个人功能方面：生成该使用界面时，会向服务器请求该用户已上传的文件列表并显示；点击上传会读取用户要上传的文件和对应文件密钥，向服务器发送请求上传文件的消息，收到服务器正确回复后，将加密后的文件和文件密钥的哈希值发送给服务器；选中已上传文件列表中的文件并点击下载会读取用户输入的文件密钥，向服务器发送请求下载选中文件的消息，服务器比对密钥哈希值正确后，等待接收文件，解密并保存。

#### 在群组功能方面：点击创建共享群组会读取用户输入的群名和群密钥，向服务器申请创建群组，根据创建结果弹出相应的提示框；点击登入共享群组会读取用户输入的群名和群密钥，服务器确认群名、群密钥正确后，进入共享群组界面；进入共享群组界面，会向服务器请求该群组已上传的文件列表并显示；点击上传共享文件会读取用户要上传的文件，向服务器发送请求上传文件的消息，收到服务器正确回复后，将用群密钥加密后的文件发送给服务器；选中文件列表中文件并点击下载共享文件会向服务器发送下载请求，接收到服务器正确回复后，等待接收文件，解密并保存。

#### 定义了清空组件，用于个人/群组功能中UI绘制；定义了发送、接收方法，用于个人/群组功能中与服务器的交互。

#### **个人功能模块：**

#### def \_\_init\_\_(self, username, client):

#### 参数：Personal\_Info\_Window对象，用户名，安全sockt连接对象

#### 返回值：无

#### 调用方法：登录后创建Personal\_Info\_Window对象，绘制个人使用界面UI，调用show\_personal\_info方法显示个人信息，并提供以下按钮：显示文件存储（调用show\_file\_storage方法）；创建共享文件（调用create\_group方法）；登入共享群组（调用login\_group方法）。

#### def show\_personal\_info(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：进入个人使用界面时自动调用该方法，在显示区域展示用户名等信息。

#### def show\_file\_storage(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：点击显示文件存储后调用该方法，先调用clear\_content\_frame()方法清空原有组件。向服务器请求该用户已经上传的文件列表，接收后展示在文件列表。并提供以下按钮：上传文件（调用upload\_file方法）；下载文件（调用download\_file方法）；刷新（调用refresh\_file方法）。

#### def upload\_file(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：点击上传文件后，打开文件选择对话框，选中要上传的文件，获取该文件的文件名和路径，然后弹出输入框，读取用户输入的文件加密密钥。向服务器发送上传文件请求，请求内容包含文件加密密钥的hash值和文件名，服务器检查没有同名文件后会返回确认信息，收到确认信息后调用send\_encrypt方法将所选文件用加密密钥加密，并发送加密后的文件数据。

#### def download\_file(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：选中文件列表中的文件并点击下载文件后，打开文件选择对话框，选择要保存的文件路径和文件名，弹出输入框读取用户输入的文件密钥。向服务器发送下载文件请求，请求内容包含文件名和文件密钥的hash值，服务器比较对应文件的密钥hash值相同后会返回正确信息，客户端收到正确信息后再向服务器发送确认信息并等待接收文件。收到文件后，调用rc4\_en\_de\_crypt()方法使用文件密钥解密接收到的加密数据，并按照用户输入的文件路径和文件名进行保存。

#### def refresh\_file(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：重新调用show\_file\_storage()方法，起到更新文件列表的作用。

#### **群组功能模块：**

#### def create\_group(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：点击创建共享群组后调用，弹出输入框读取用户输入的群名和群密钥，调用create\_group\_button与服务器交互。

#### def create\_group\_button(self, group\_name, group\_key):

#### 参数：Personal\_Info\_Window对象，群名，群密钥。

#### 返回值：无

#### 调用方法： 向服务器发送创建共享群组请求，根据创建结果显示相应提示框。

#### def login\_group(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：点击登入共享群组后调用，弹出输入框读取用户输入的群名和群密钥，调用create\_group\_button方法与客户端进行交互。

#### def login\_group\_button(self, group\_name, group\_key):

#### 参数：Personal\_Info\_Window对象，群名，群密钥

#### 返回值：无

#### 调用方法：向服务器发送登入共享群组请求，若群组存在且密钥正确服务器返回正确信息，收到正确信息后调用show\_shared\_files方法。

#### def show\_shared\_files(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：登入共享群组成功后调用该方法，先调用clear\_content\_frame()方法清空原有组件。向服务器请求该用户已经上传的文件列表，接收后展示在共享文件列表。并提供以下按钮：上传共享文件（调用upload\_shared\_file方法）；下载共享文件（调用download\_shared\_file方法）；刷新（调用refresh\_shared\_file方法）。

#### def upload\_shared\_file(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：点击上传共享文件后，打开文件选择对话框，选中要上传的文件，获取该文件的文件名和路径。向服务器发送上传文件请求，请求内容包含文件名，服务器检查没有同名文件后会返回确认信息，收到确认信息后调用send\_encrypt方法将所选文件用群密钥加密，并发送加密后的文件数据。

#### def download\_shared\_file(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：选中共享文件列表中的文件并点击下载文件后，打开文件选择对话框，选择要保存的文件路径和文件名。向服务器发送下载文件请求，请求内容包含文件名，服务器返回正确信息，客户端收到正确信息后再向服务器发送确认信息并等待接收文件。收到文件后，调用rc4\_en\_de\_crypt方法使用群密钥解密接收到的加密数据，并按照用户输入的文件路径和文件名进行保存。

#### def refresh\_shared\_file(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：重新调用show\_file\_storage()方法，起到更新共享文件列表的作用。

#### 清空组件和发送接收：

#### def clear\_content\_frame(self):

#### 参数：Personal\_Info\_Window对象

#### 返回值：无

#### 调用方法：生成个人和群组文件界面时调用该方法，清除已有组件以便UI绘制以及信息更新。

#### def send\_message(safe\_socket,message):

#### 参数：安全socket对象，要发送的信息

#### 返回值：无

#### 调用方法：调用safe\_socket对象的send\_encrypt方法，使用安全信道将信息发送给服务器。

#### def receive\_message(safe\_socket):

#### 参数：安全socket对象，要接收的信息

#### 返回值：无

#### 调用方法：调用safe\_socket对象的recv\_decrypt方法，使用安全信道接收服务器消息。

#### message\_type.py文件中的函数逻辑实现：

#### register(username, password, email, phone)：实现注册功能

#### 参数：username, password, email, phone分别是client传来的用户名、密码的哈希值、邮箱地址和电话号码

#### 返回值：若用户已存在则返回0，若邮箱地址已存在返回1，否则返回2，表示注册成功。

#### 调用方式：当解析到client传来的message\_type为register时，调用该函数。

#### login\_password(username, password)：实现通过密码登录的功能

#### 参数：username, password分别是client传来的用户名、密码的哈希值

#### 返回值：若用户不存在则返回0，若密码错误返回1，否则返回2，表示登录成功。

#### 调用方式：当解析到client传来的message\_type为login\_password时，调用该函数。

#### get\_email\_code(email)：实现发送邮箱验证码的功能

#### 参数：email分别是client传来的邮箱地址

#### 返回值：若邮箱不存在则返回0，否则返回1，表示发送成功。

#### 调用方式：当解析到client传来的message\_type为get\_email\_code时，调用该函数。

#### login\_email(email, auth\_code)：实现邮箱验证登录的功能

#### 参数：email, auth\_code分别是client传来的邮箱地址以及邮箱验证码

#### 返回值：将client输入的邮箱验证码与服务器本地生成的验证码比较，若相等则返回login\_check\_email(email)，若不相等则返回1，表示验证码不正确。若在输入邮箱获得验证码之后把邮箱篡改掉，则返回0。

#### 调用方式：当解析到client传来的message\_type为login\_email时，调用该函数。

#### get\_phone\_code(phone)：实现发送手机验证码的功能

#### 参数：phone分别是client传来的电话号码

#### 返回值：若手机号码不存在，返回0，否则返回1 。

调用方式：当解析到client传来的message\_type为get\_phone\_code时，调用该函数。

#### change\_password(phone, auth\_code, password\_1, password\_2)：实现修改密码的功能

#### 参数：phone, auth\_code, password\_1, password\_2分别是client传来的电话号码、验证码、要修改的密码以及确认密码。

#### 返回值：若修改成功，则返回2；若两次输入的密码不一致，则返回1；若验证码输入错误，则返回0 。

调用方式：当解析到client传来的message\_type为change\_password时，调用该函数。

#### getrate\_random()：实现修改密码的功能

#### 参数：无

#### 返回值：4位数字字符串

调用方式：当调用get\_email\_code和get\_phone\_code时，调用该函数。

#### md5s(strs)：实现使用md5算法将字符串压缩的功能

#### 参数：strs是传入的一个字符串

#### 返回值：字符串的哈希值

调用方式：当调用send\_phone时，调用该函数。

#### send\_email(email, email\_message)：实现发送邮件的功能

#### 参数：email, email\_message分别是邮件的收件方以及验证码

#### 返回值：若发送成功则返回True，否则返回Flase。

调用方式：当调用get\_email\_code时，调用该函数。

#### send\_phone(phone, auth\_code)：实现短信的功能

#### 参数：phone, auth\_code分别是短信的收件方以及验证码

#### 返回值：无

调用方式：当调用get\_phone\_code时，调用该函数。

#### make\_folder(path, folder\_name)：实现创建文件夹的功能

#### 参数：path, folder\_name分别是文件路径和文件夹名称

#### 返回值：无

调用方式：当调用save\_file时，调用该函数。

#### save\_file\_check(username,filename,key\_hash)：实现检查是否文件名已经存在的功能

#### 参数：username,filename,key\_hash分别是用户名、文件名和文件密钥的哈希值

#### 返回值：当文件名已存在时返回0，当文件名不存在时返回’ok’

调用方式：当解析到client传来的message\_type为upload时，调用该函数。

#### save\_file(username, filename, data)：实现保存client传来的文件密文的功能

#### 参数：username,filename,data分别是用户名、文件名和文件密文

#### 返回值：当文件保存成功时返回True，当文件保存失败时返回False

调用方式：当解析到client传来的message\_type为upload并且收到save\_file\_check函数返回值为’ok’时，调用该函数。

#### save\_file(username, filename, data)：实现保存client传来的文件密文的功能

#### 参数：username,filename,data分别是用户名、文件名和文件密文

#### 返回值：当文件保存成功时返回True，当文件保存失败时返回False

调用方式：当解析到client传来的message\_type为upload并且收到save\_file\_check函数返回值为’ok’时，调用该函数。

#### find\_file(username)：实现返回用户所存储的文件名列表的功能

#### 参数：username是用户名

#### 返回值：当用户有存储文件时返回文件列表，当无存储文件时返回0

调用方式：当解析到client传来的message\_type为get\_list时，调用该函数。

#### send\_file(username, filename, key\_hash)：实现返回用户所存储的文件密文的功能

#### 参数：username, filename, key\_hash是用户名、文件名以及文件密钥的哈希值

#### 返回值：当文件不存在时返回0，当文件密钥不正确时返回1，当获得权限下载时返回2

调用方式：当解析到client传来的message\_type为download时，调用该函数。

#### create\_group(group\_name,group\_key)：实现创建群空间的功能

#### 参数：group\_name,group\_key是群名和群邀请码的哈希值

#### 返回值：当群名已存在时返回0，当创建群成功时返回2

调用方式：当解析到client传来的message\_type为create\_group时，调用该函数。

#### login\_group(group\_name,group\_key)：实现登录群空间的功能

#### 参数：group\_name,group\_key是群名和群邀请码的哈希值

#### 返回值：当群名不存在时返回0，当群邀请码不正确时返回1，当群邀请码正确时返回2

调用方式：当解析到client传来的message\_type为login\_group时，调用该函数。

19、get\_group\_list(group\_name)：实现得到群文件列表的功能

参数：group\_name是群名

返回值：当群文件不存在时返回0，当群文件存在时，返回群文件列表

调用方式：当解析到client传来的message\_type为get\_group\_list时，调用该函数。

20、upload\_group(group\_name, filename)：实现群文件上传的功能

参数：group\_name，filename是群名和文件名

返回值：当群文件已存在时返回0，当群文件上传成功时，返回’ok’调用方式：当解析到client传来的message\_type为upload\_group时，调用该函数。

21、download\_group(group\_name, filename)：实现群文件下载的功能

参数：group\_name，filename是群名和文件名

返回值：当群文件不存在时返回0，当群文件中没有找到用户想下载的文件时，返回1，当群文件中有用户想下载的文件时，返回2。

调用方式：当解析到client传来的message\_type为download\_group时，调用该函数。

22、send\_back(safe\_socket, message\_return)：实现通过socket消息回传的功能

参数：safe\_socket, message\_return是套接字对象和想要回传的字符串

返回值：无

调用方式：当解析到client传来的message 需要做出回应时，调用该函数。

#### sql\_init.py文件中的函数逻辑实现：

#### 1、create\_user(username,password,email,phone)：实现将新用户插入users表的功能

参数：username,password,email,phone分别是用户名、密码的哈希值、邮箱地址以及电话号码

返回值：当用户名已存在时返回0，当邮箱地址已存在时返回1，当插入成功时，返回2。

调用方式：当调用register函数时，调用该函数。

#### 2、login\_check\_password(username,password)：实现用户用密码登录验证的功能

参数：username,password分别是用户名、密码的哈希值

返回值：当用户名不存在时返回0，当密码不正确时返回1，当登录成功时，返回2。

调用方式：当调用login\_password函数时，调用该函数。

#### 3、login\_check\_email(email)：实现用户用邮箱登录验证的功能

参数：email是邮箱地址

返回值：当邮箱不存在时返回0，当邮箱存在时返回需要被发送邮箱验证码的用户名。

调用方式：当调用login\_email函数时，调用该函数。

#### 4、login\_check\_phone(phone)：实现检查手机号是否存在对应用户的功能

参数：phone是电话号码

返回值：当用户不存在时返回0，当用户存在时返回需1 。

调用方式：当调用get\_phone\_code函数时，调用该函数。

#### 5、change\_my\_password(phone,new\_password)：实现更改用户密码的功能

参数：phone，new\_password是电话号码和新密码的哈希值

返回值：修改成功返回2

调用方式：当调用change\_password函数时，调用该函数。

#### 6、insert\_file(username,filename,hash\_value)：实现将用户想上传的文件的信息插入files表的功能

参数：username,filename,hash\_value是用户名、文件名和文件密钥的哈希值

返回值：当文件已存在时，返回0，当文件信息已插入成功时返回2

调用方式：当调用save\_file\_check函数时，调用该函数。

#### 7、find(username)：实现将用户所存储的文件列表返回的功能

参数：username是用户名

返回值：当文件不存在时，返回None，当文件存在时返回文件列表

调用方式：当调用find\_file函数时，调用该函数。

#### 8、get\_file(username, filename, hash\_value)：实现检验用户是否有权限下载文件的功能

参数：username,filename,hash\_value是用户名、文件名和文件密钥的哈希值

返回值：当该用户没有存储文件时返回0，当文件密钥不正确时返回1，否则返回2，表示获得权限

调用方式：当调用send\_file函数时，调用该函数。

#### 9、create\_group\_in\_sql(spacename,hash\_value)：实现检验群是否能够创建的功能

参数：spacename,hash\_value是群空间名和邀请码的哈希值

返回值：当群名已存在时返回0，当创建成功时返回2

调用方式：当调用create\_group函数时，调用该函数。

#### 10、login\_group\_in\_sql(group\_name,group\_key)：实现检验用户是否能够登录群的功能

参数：group\_name,group\_key是群空间名和邀请码的哈希值

返回值：当群名不存在时返回0，当邀请码不正确时返回1，否则返回2，表示登录成功

调用方式：当调用login\_group函数时，调用该函数。

#### 11、get\_group\_list\_in\_sql(group\_name)：实现返回群文件列表的功能

参数：group\_name是群空间名

返回值：当群文件不存在时返回None，否则返回文件列表

调用方式：当调用get\_group\_list函数时，调用该函数。

#### 12、upload\_group\_in\_sql(group\_name, filename)：实现上传群文件的功能

参数：group\_name, filename是群空间名和想要上传的文件名

返回值：当群文件已存在时返回0，否则返回2，表示上传成功

调用方式：当调用upload\_group函数时，调用该函数。

#### 13、download\_group\_in\_sql(group\_name, filename)：实现下载群文件的功能

参数：group\_name, filename是群空间名和想要下载的文件名

返回值：当群文件为空时返回0，当群文件存在但找不到想要下载的文件返回1，否则返回2表示下载成功

调用方式：当调用download\_group函数时，调用该函数。

五、测试分析报告

* 注册功能

fig:

3546e5bb43f911e5e995473041ef02c

* 用户名+密码登录

fig:

b17549299472abb26563a75df3b49c0

* 邮箱+验证码登录

fig:

18fee328cddfcca88963eb974ed0a3b

fig:

5ff6cdbb493e30b290d2d4b17aa115e

* 手机号+验证码重置密码

fig:

06e1ec9a12978a0d76e55c161441a95

* 个人用户上传文件

fig:

aa56e0f2e40cb1388c25c8d3c129ce9

fig:

966d584b9470b12c0b6a75d8ac0062e

fig:

1eca007548997329a259651ccd985f4

fig:

4637b06d5c739ab0f032c808e5da007

* 个人用户下载文件

fig:

af1966a654c7543b5b2b2b7c60dc52e

fig:

6e0babae927e4cd2c0762d96c2d2dcb

fig:

a7b12905abf944c2f27f967ac18a309

* 创建共享文件群

fig:

e5795395e08633095be9e305953abe1

fig:

87e39186e994f90bbca4d013ae6ee74

fig:

fd6cea8faedf66b1e75144de8794a26

* 上传文件到共享群中

fig:

89a497f2e2ff5b168faed0877ddae82

fig:

7320d691cdfdc022d71bdeb948d7ac5

* 新建用户登录共享群并查看共享文件

fig:

62d8b3ce60040b416c606c773bc650d

fig:

3a1a02d2abbec15e0eab35a06f5484e

fig:

36b37dca198d6fa806bb4f22c524da3

fig:

64e1a90bf63fd9cebcbf4ee062aea10

六、程序编译和安装使用文档

1. 打开项目源码后，首先修改数据库配置，在server文件夹下的config.py中修改数据库名称，用户名，密码，地址。之后修改服务器IP地址，若使用本机做服务器则不变。
2. 打开server文件夹下的protocol文件夹下的communication.py文件，在第57行和第170行分别修改路径为本地的证书和私钥路径，证书和私钥文件与communication.py位于同一目录中。之后同样修改client文件夹下protocol文件夹下的communication.py文件。
3. 运行server文件夹下的main.py文件，建立数据库表
4. 进入server文件夹下的server.py文件，启动服务器，提示服务器运行中，开始监听。
5. 新建一个命令行终端，cd client进入client目录，输入python app1.py即可启动客户端。