P2P

开发说明文档

牛云飞

学号：17031211734

计算机学院 工程三班 研一

P2P 架构及实现原理

本项目使用了一种结合了非对称加密的集中式 P2P 架构，保证了通信网络健壮性的同时又保证了隐秘性。

这个 P2P 网络中有三个基本的角色，即 Server、Alice 和 Bob，其中 Alice 和 Bob 是 P2P 通信的双方，Server 是集中式网络的服务器。

在 Alice 和 Bob 通信之前，必须通过安全信道完成密钥的交换，以保证在 P2P 网络中的通信安全。

Alice/Bob 端通信流程：

1. 开启通信客户端；
2. 向 Server 发送用提前和 Bob 交换过的密钥对加密的本机 IP/域名；
3. 等待通信中另一方的连接，或者向 Server 请求客户端记录表，并通过之前交换过的密钥对进行解密，如果发现合法的 IP/域名信息则作为对方的地址发起连接；
4. 连接成功，使用预交换的密钥对通信，或连接不成功，等待新的连接或寻找新的合法地址。

Server 端通信流程：

1. 开启通信服务端；
2. 初始化客户端记录表；
3. 等待任意 Client 发送加密的 IP/域名信息，将从 Client 收到的 IP/域名信息加入客户端记录表中；
4. 等待 Client 请求客户端记录表，向 Client 发送客户端记录表。

Server 端的所有发送和请求都不需要 Client 进行身份认证，同时因为客户端记录表中的任一个值都经过非对称加密，所以 Client 只能从中找到自己的通信对象的信息，而无法得到其他用户的信息，保证了每一个用户的网络地址都只有信任的通信对象可以得知，Server 和网络上的其他用户都无法获知，保证了通信的隐秘、安全。

重点技术说明

1. WebService 技术

Alice 和 Bob 端的通信使用了 WebService 架构，每个通信客户端都运行一个 WebService 的 Server 端，通信又向通信对象发起服务请求。具体实现上采用了 JAX-WS 框架。

2. MTOM（Message Transmission Optimization Mechanism）消息优化传输机制

针对 Base64 编码情况带来的开销提出的解决方案。当数据量小的时候，SOAP 依然使用 XML 进行消息的传递。

消息传输优化机制 (MTOM) 标准允许将消息中包含的大型数据元素外部化，并将其作为无任何特殊编码的二进制数据随消息一起传送。MTOM 消息会打包为多部分/相关 MIME 序列，放在 SOAP 消息中一起传送。

但是在大量数据情况下，如果数据依然进行 Base64 编码,会带来 33%的额外开销，这样的情况对于大量数据交换的情况是无法容忍的。MTOM 就是针对 SOAP 消息传输的基础上提出的改进办法。对于大量数据的传递，不会进行进行 Base64 编码，而是直接以附件的二进制原始数据的形式封装在 SOAP 消息的 MIME 部分，进行传输。SOAP 消息通过指向随其发送的 MIME 部分来引用二进制内容，另外包括 SOAP 基本的 XML 数据，这些还是 Base64 编码。因为此模型与简单邮件协议 SMTP 模型基本一致。

MTOM 通过简化大量数据的编码过程，从而提高数据的处理效率。因为 SOAP 消息等必要的信息，MTOM 也有一些必要的开销。MTOM 仅在二进制数据元素的大小超过大约 1 KB 时，才能体现出其优势。

在本项目中，MTOM 被用来实现文件传输。

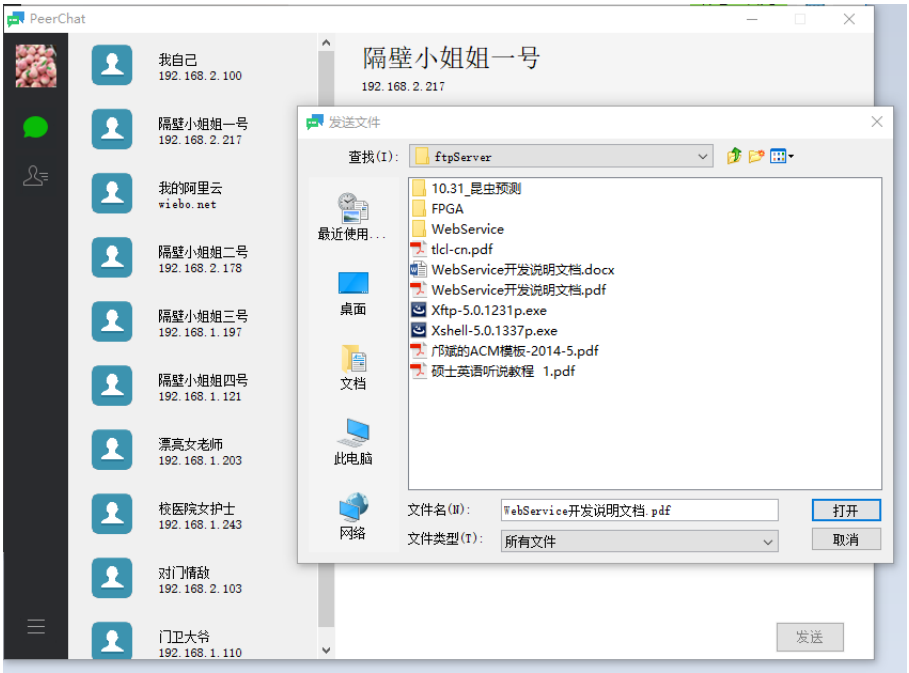
3. JAVA Swing

本项目使用 Java Swing 构建用户界面。

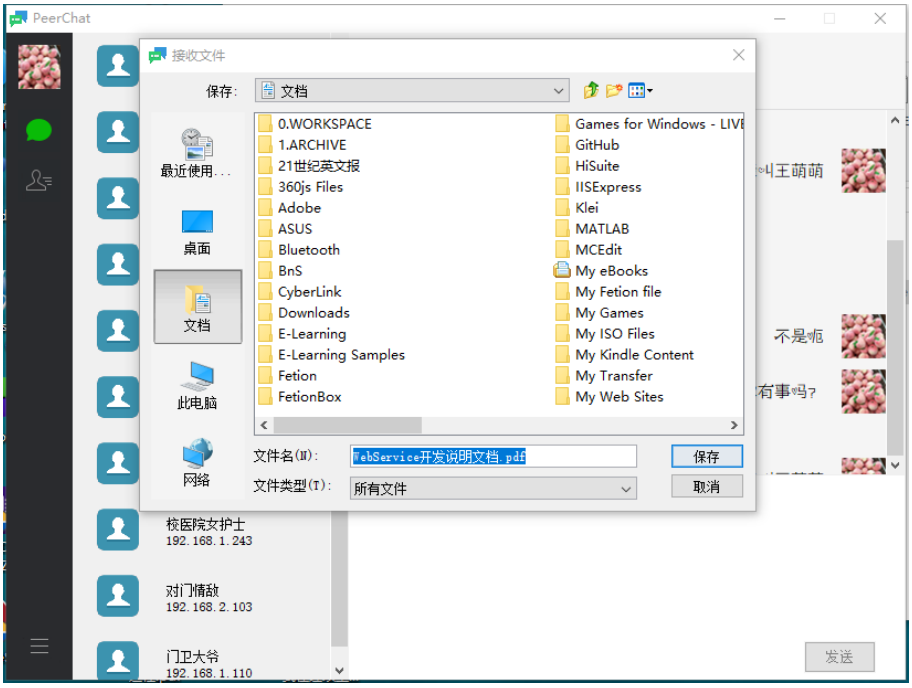
运行结果截图及文字说明

如上图是本 P2P 聊天软件的界面，界面设计整体上仿照微信 PC 版。

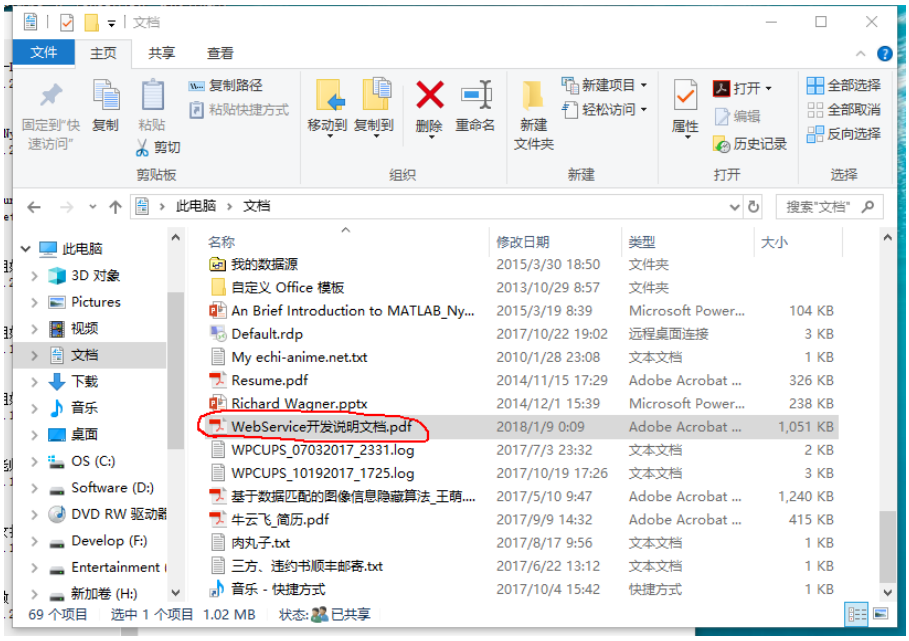
左侧黑色竖栏为工具栏，有联系人目录、添加联系人、选项等功能；工具栏左侧为联系人栏，显示了目前记录的联系人，包括头像、昵称、IP/域名，点击相应的联系人可以和相应联系人开始聊天；右侧是聊天栏，最上侧显示了当前聊天的联系人的昵称和 IP/域名，中间显示了当前的消息记录，下侧是编辑栏，可以编辑和发送消息，同时还有发送文件按钮。



如图是点击编辑栏上侧的发送文件按钮后弹出的文件选择窗口。



如上图是文件接收方在收到文件后弹出的保存窗口，点击“保存”后文件成功保存。



如上图是成功收到并保存的文件。

开发中遇到的问题

1. 不熟悉在 WebService 中的文件传输

   为了能实现快速开发，我使用了 WebService 实现通信，但是传统的 WebService 调用无法传输大文件，在查阅了一些资料之后才找到了 MTOM 这种将文件作为 MIME 附件传送的方案。

2. 对于用 Java Swing 构建复杂 UI 缺乏经验

   这个 P2P 聊天软件的 UI 比较复杂，调用和数据交互多样，这为 UI 的开发带来很多困难，尤其是聊天窗口中的消息气泡显示很难实现。

3. 对公钥加密的实现不熟悉

   过去对公钥加密只至于了解，没有动手实现过相关项目，这也增加了临时的工作量。