安全通信模拟程序系统设计方案

姓名：顾若兰 学号：194790 学院：网络空间安全 专业：计算机专硕

1. 系统需求

网络的出现使信息的获取、传播、处理和利用更加的高效和便捷，也极大的便利了我们的生活，在这次突然爆发的疫情期间，信息化抗疫更是功不可没。但网络通信也存在很多安全隐患，网络通信中信息在公共的信道传递，容易受到主动或被动攻击，从而导致信息泄露甚至更加严重的损失。

一个基本的安全通信模型，需要实现信息加密，完整性验证和鉴别。本安全通信模拟系统采用了基于分组模式的 DES、AES-128 对称加密算法进行数据加密，具有计算量较小，运行速度较快的优点；采用 MD5、SHA-1 哈希算法得出信息的散列值以实现完整性验证；采用 RSA 非对称加密算法对密钥和散列值进行加密，RSA 加密算法虽然具有很强的安全性，但是在程序运行过程中，其耗时较长，计算量非常大，对硬件要求较高。

2. 系统框架

本系统框架如图 3.1 所示，共分为三个功能模块，分别是信息加密、信息完整性检验和非对称加密模块。

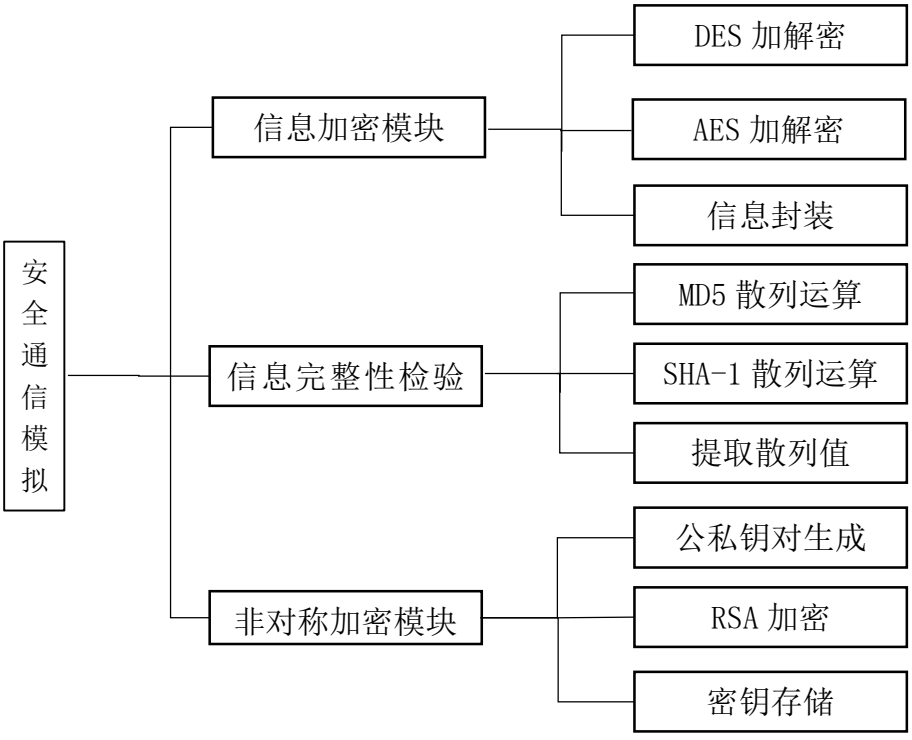

图 3.1 系统框架图

3. 系统说明

开发语言：C++

图形化界面语言：MFC

开发平台：Visual Studio 2012

开发/运行环境：Windows

4. 系统工作流程

系统大致工作流程如图 4.1 所示。

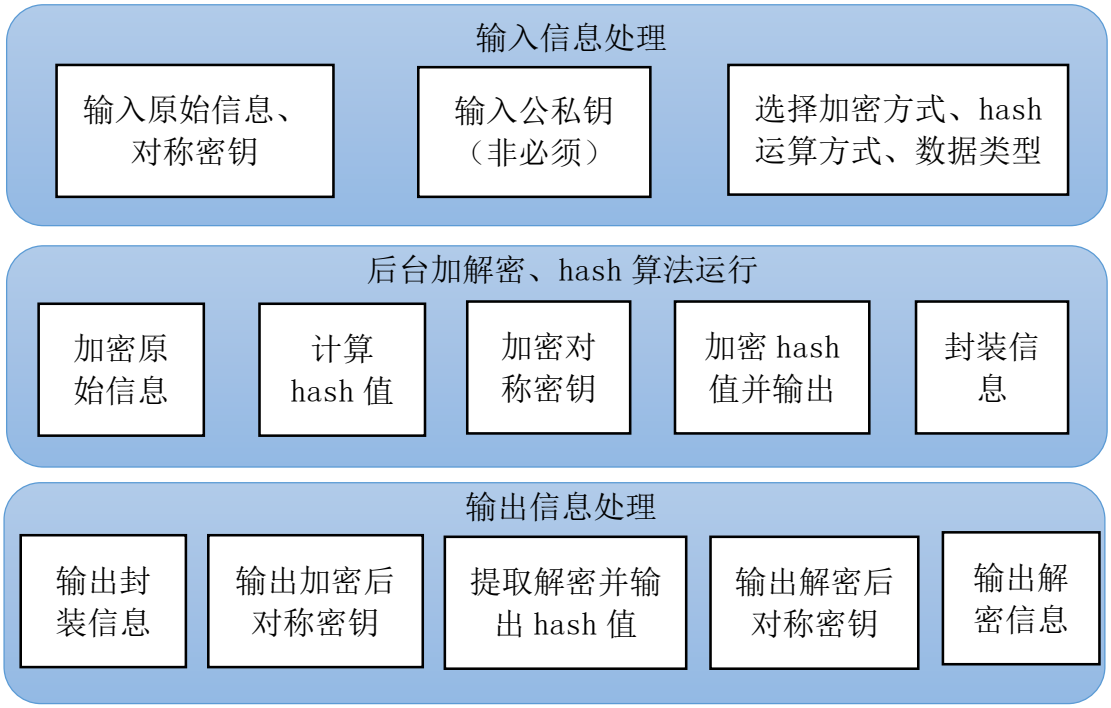

图 4.1 系统大致工作流程图

系统工作的详细流程如图 4.2 所示，其中 InMsg 表示原始待发送信息信息，L(Inmsg)表示原始待发信息长度，K 表示对称密钥,L(K)表示。直角方框表示表示输入输出，菱形表示判断，圆角方框代表程序执行，Y 代表是，N 代表否。

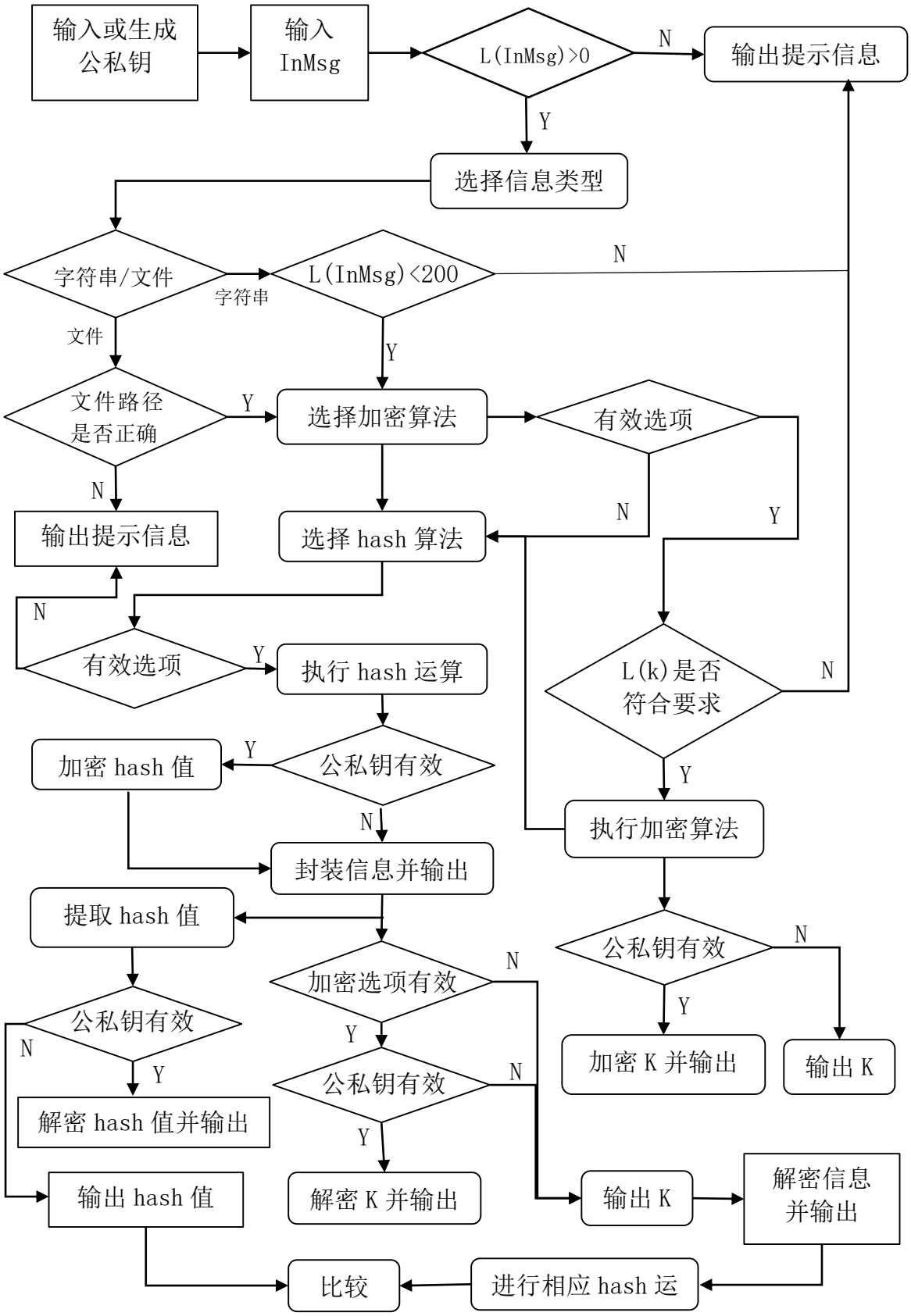

图 4.2 系统详细工作流程图

5. 界面设计

主界面如图 5.1 所示，界面主要分为输入信息区域、按钮选项区域和输出信息区域，信息框从左至右按照安全通信中的信息生成顺序排列，输入框和输出框采用了不同的外观样式以便区分。文件类型、加密算法选择、散列运算选择使用下拉框。

图 5.1 主界面

提示信息界面如图 5.2 所示，提示文本信息根据不同情况生成：

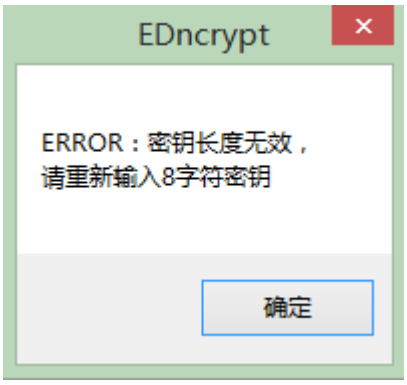

图 5.2 提示信息界面

6. 功能模块

本系统主要分为三个功能模块：

1.1 信息加密功能模块

输入要求：（1）原始信息:需在“请输入字符串/文件路径”文本框中输入，基于 ASICii 码的字符串，可以是 txt 文件路径或字符串，文件路径需要为不带类型后缀的绝对文件路径，例如：D:\Visual Studio 2012\Projects\EDncrypt\TestFile\file1;（2）对称密钥:十进制数字形式密钥，DES 加密密钥需为 8 字节，AES 加密密钥为 16 字节；（3）加密算法选择:下拉框

中可选择 DES、AES，未选择则不使用加密算法；（4）输入信息类型选择：下拉框中可选择字符串、文件；，选择则不对输入信息做任何处理。

接口函数：用户输入相应信息后点击“安全通信模拟”按钮即可实现数据的加解密，其函数接口为 OnBnClickedEncrypt()，位于 EDncryptDlg.cpp 文件中。DES 加密函数定义为 desEncrypt()，解密函数定义为 desDecrypt()中，位于 DES.h 头文件中；AES 加密函数定义为 aesEncrypt()，解密函数定义为 aesDecrypt()，位于 AES.h 头文件中。

图 6.1 文件加密

输出信息：（1）输入数据为字符串时：“传输信息”文本框输出加密并且封装后的信息，前四位为加密后信息的长度，后面分别为加密后信息和 hash 值；“解密信息”文本框输出解密后的字符串。（2）输入数据为文件时，封装后的信息存入新建文本文件中，文件名为原文件名加上“_enc”，文件存放在与原始文件相同的文件夹中，文件路径输出在“传输信息”文本框中，封装信息的前 16 位为加密后信息长度，后面分别是加密后信息和 hash 值；解密后的字符串同样存入文本文件中，文件名为为原文件名加上 “_dnc”，文件存放在与原始文件相同的文件夹中，文件路径输出在“解密信息”文本框中。

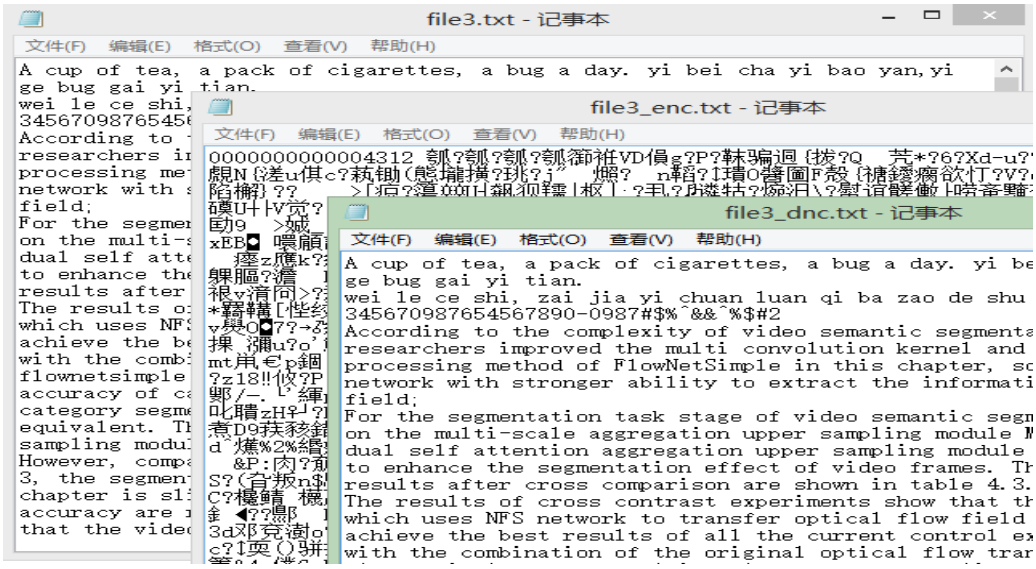

图 6.2 文件加密结果

1.2 信息完整性验证模块

输入要求：在“请输入字符串/文件路径”文本框中要输入原始信息，另外需要进行散列算法选择，在下拉框中可选择 MD5 或 SHA_1 散列算法，未选择则不使用散列算法。

接口函数：同样的，用户输入相应信息后点击“安全通信模拟按”钮即可实现数据的散列值计算和封装信息散列值的提取;MD5 散列函数定义为 hashMD5(),位于 MD5.h 头文件中;SHA-1 散列函数定义为 hashSHA_1(),位于 SHA_1.h 头文件中。

输出信息：在“散列值（10 进制）”文本框中输出十进制字符串形式散列值。在“解密提取散列值”文本框中输出从封装信息中提取出的散列值，可以将解密后的信息再次进行散列值计算同提取的散列值进行比较以验证数据的完整性和可靠性。

1.3 非对称加密模块

（1）公私钥生成：用户点击“生成公私钥”按钮即可生成公私钥，其函数接口为 OnBnClickedGenePrPu()，公私钥输出在“传输信息”文本框中。考虑到公私钥生成的时间,在程序中对生成随机公钥的次数进行限制,所以可能会生成失败,重试几次即可生成公私钥对。

图 6.3 公私钥对生成

（2）非对称加密：在用户输入了有效的公私钥后点击“安全通信模拟”按钮后可实现对对称密钥和 hash 值的加解密，加密后的密钥输出在对应的“RSA 加密后对称密钥”文本框中，解密密钥参与解密信息计算中，加密后的 hash 值存入封装信息中，解密提取的 hash 值输出在“解密提取散列值”中。

7. 数据存储方式

输入字符串时以基于 ASIC 码的字符串存储，内存动态分配。输入文件时输出

txt 文件，本地储存，路径与输入文件路径一致。

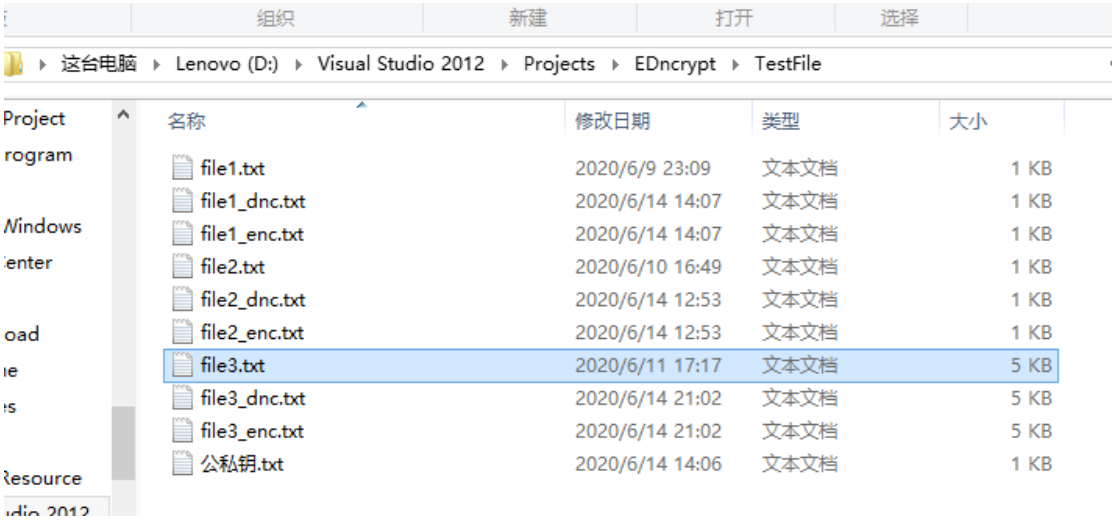

图 7.1 文件存储

在公私钥生成过程中定义了 BigInt 类，实质上是由字符‘0’-‘9’组成的字符串，对于公私钥的存储，需要使用数据库，下面给出公私钥存储的数据库表设计：

表 7.1 公私钥存储表

表项	数据类型	是否主键	外键
用户 ID	INT64	是	是
公钥	String	否	否
私钥	String	否	否
信任用户	INT64	否	是

8. 待改进和优化部分

（1）对 Unicode、中文等其他字符集的支持：在各编写加解密、hash 算法时是基于二进制和 ASICii 码的计算，但是在后续整合过程中发现字符集的不同会使加解密信息不一致，而且基于 ASICii 码的加解密已不能满足实际的需要。

（2）RSA 算法的改进：要实现密钥位数较长的 RSA 算法需要定义大整数类，本系统中定义了 BigInt 类,用一个字节存储一位十进制数据,普通的 Unsigned Int 类型四字节可表示到 4294967295，但 BingInt 类型中只能表示到 9999，存储空间的利用率大大降低。在 RSA 程序中对求幂等运算都进行了优化，但当 RSA 参与运算时，运行速度明显变慢，当增加初始运算的两个大质数 p、q 的长度时程序

会明显卡顿甚至导致电脑卡死，这一点可能有硬件设备的性能有限的原因，但RSA 程序的优化也非常有必要。

9. 总结

在完成本次大作业的过程中，我遇到了很多困难同样也获得了很多知识和技能。在实现 DES、AES_1 对称加密算法，MD5、SHA-1 散列算法、RSA 非对称加密算法的过程中我对其算法原理和流程有了更直观的理解，也感受到了各种算法的性能区别。同时此次在此次大作业中我也习得很多的编程技巧，比如在这些算法中经常会巧妙地使用位运算，这是在以往的编程过程中很少用到的，以及学习了MFC 等。因为时间、精力、自身目前能力和硬件性能等条件有限，本系统还存还很多不足，日后我也会不断进行完善。最后对任课老师黄老师表达诚挚的感谢，在老师的课堂上我学到了很全面的密码学知识，非常感谢老师在疫情期间一直以来的线上授课和答疑解惑。