编号：

****

毕业设计说明书

题 目： **文件加密系统**

学 院： 计算机与信息安全学院

专 业： 信息安全

学生姓名： 王光汉

学 号： 1300340125

指导教师： 姚罡

职 称：

**题目类型：**□理论研究 □实验研究 □工程设计 □工程技术研究 ☑软件开发

2017年9月30日

摘 要

文件加密的需求在如今越来越敏感的社会中，在信息安全意识越来越重视的当下变得越来越重要。它不再只是军事需要，已然成为我们日常的必须基础。没有信息安全的现代以及未来人就像是生活在远古时代的智人害怕野兽一般害怕信息的泄露、网络攻击。可以说，信息安全工具已然成为现代以及未来人类的生活的必须品，没有它新人类将寸步难行，如履薄冰。我们现在的很多信息网络高速的发达的国家和地区已经越来越离不开信息安全技术了。没有了信息安全技术就没有我们今天便利的生活。它正在服务着我们生活的方方面面，从线上到线下，从支付宝到微信再到其他的电子商务等等都离不开信息安全技术的支撑。在未来他们支撑和服务的国家和地区以及领域将越来越大。届时，我们就会与信息安全技术工具紧紧联系在一起了。

总而言之，我们会越来越依赖于信息安全技术与工具。

文件加密技术作为信息安全的一种手段一个工具，它对于我们生活的意义不言而喻。网络上很多宣传一秒加密超大的文件的文件加密工具，其实就是简单对源文件进行名字更改再隐藏而已。这种行为简直就是不把用户的信息安全当回事，只对不懂技术和君子有效。

因此，研究文件加密技术，研究其中涉及的算法以及对即将需要用到的工具环境进行研究也是必然的。研究的加密算法包括两大类：非对称加密算法、对称加密算法。对应实际的算法是RSA算法和AES算法。用到的工具是Qt creator5.9以及本身自带的库文件，库文件包括基本GUI库以及线程库。开发操作系统环境是linux mint17.3。还研究第三方开发库OpenSSL的使用。最终设计和实现了简洁易用的CDF（code deoce files）文件加密系统软件。

关键词：文件加密；Qt；OpenSSL

**Abstract**

There is a kind of automatic access system that use automatic indemnification technology to identify user’s ID and rights, and according to user’s rights to control the door.

**Key words** Word1;Word2

目 录

引言 1

开发背景 1

文件加密发展历史 1

1 需求分析 2

1.1 核心需求分析 2

1.2 多线程需求分析 3

1.3 图形化界面需求分析 3

2 设计方案与主要技术 3

2.1 设计方案 3

2.1.1 CDFC标志 4

2.1.2 KEY\_IV\_LEN\_SUM 4

2.1.3 ENCODE\_KEY 5

2.1.4 ENCODE\_IV 5

2.1.5 ENCODE\_DATA 5

2.2 主要使用到的技术 5

2.2.1 AES算法 6

2.2.2 RSA算法 6

3 具体实现 6

3.1 文件加密子系统 6

3.1.1 选择要加密的文件 6

3.1.2 选择公钥文件 9

3.2 文件解密子系统 10

3.3 RSA密钥对文件生成子系统 10

4 测试 10

5 总结 10

谢 辞 12

参考文献 13

附 录 14

## 引言

## 开发背景

从未有哪个时代信息传播速度能与今天匹敌。由此，也产生了过去从未有过的如此的普遍的信息传播的安全性需求。它如今与我们的日常生活息息相关，特别是现代化发达城市，人们每天的生活都与信息的安全传输息息相关。从早上的起来之后到路边摊买油条豆浆或其他面食，到乘地铁或公交车到公司上班，再到回家之后上网买东西等等这一些列的购物过程不管是线上还是线上，不管是支付宝还是微信，都离不开信息的安全传输这个需求。可以毫不夸张的说没有信息安全的传输技术支持，没有加密算法的发展进步，就没有今天世界各地的繁荣的电子商务的发展。

同样电子文件的安全性在今天的环境下越来越显得重要。没有经过加密的电子文件在网络上传输，极易被不良用心这截获，轻者个人隐私没泄露，重者个人财产甚至生命受到威胁。从新闻上不断爆出的各种明星艳照门事件以及各式各样网络安全事件可以不难看出这种需求的必要性。

文件的安全依赖于文件的加密技术或者说加密算法与加密过程设计。

## 文件加密发展历史

世界上很多高科技的发展总是随着军事需要发展，文件加密也不例外。最早的文件加密被运用与军事情报传输文件的加密。历史上关于加密文件的记载载于周朝兵书《六韬.龙韬》中的《阴符》和《阴书》。其中记载如何使用八等长度的符来表达不同的消息、情报、指令。以现代的技术来衡量，他们那时的加密算法属于现代密码学中的替代法。所谓的替代法就是把原本可以轻易看懂的信息用其他的信息替代变成难以看懂也是毫无意义的符号。但是，阴书中所载的方法只能加密很少量的信息，表达很简短的含义。

无独有偶，在古代的西方国家也有将文件加密大规模运用在战争军事情报之中。希罗多德的《历史》有载，公元前五世纪，希腊城邦和波斯帝国频频交战，希腊城邦使用移位法对军事情报文件进行加密处理。这使得就算是波斯帝国截获了他们的情报也无法破解。由此导致无法作军事部署。

在此之后又产生了其他的加密算法用于加密军事情报。但是，主要以移位法和替代法为基础。同时，这些算法也依赖与算法的隐蔽性。而不管是简单的移位法还是替代法，随着时代的发展变迁，科学技术的进步这些算法越来越变得脆弱。由此便产生了现代的更科学的加密算法。

自香农在20世纪发表的《秘密体制的通信理论》那篇文章，加密算法才由古典的算法进化到现代加密算法。这种进化的表现在于把加密算法的重心转移到应用数学上，而传统的古典的加密算法的重心在于模式的改变。于是，渐渐诞生现今我们知道的三类加密算法：非对称加密算法，对称加密算法，以及散列函数。这三类加密算法是当今各式各样电子产品信息安全的支撑。

非对称加密算法是指加密秘钥与解密秘钥是相互独立的，也就是说秘钥分为两份一份用于加密一份用于解密，但是这两份秘钥是不同的。该算法主要运用于秘钥分发与数字签名。主要的实现有RSA, EI Gamal, ECC, Knapsack等等。

对称加密算法是指加密秘钥是一样的，唯一的。也就说只能用同一个秘钥加密和解密一个信息数据。主要的实现有 DES 算法、AES 算法、3DES 算法、TDEA 算法、Blowfish 算法、RC5 算法、IDEA 算法等。

散列算法，又称哈希函数。是指一种从任何一种数据中创建小的数字“指纹”的方法。散列函数把消息或数据压缩成摘要，使得数据量变小，将数据的格式固定下来。

但是，又随着科学技术的进步现在的加密算法的安全性也渐渐受到挑战。例如，现在的计算机的计算力越来越强特别是正在研发当中的量子计算机。据量子计算机研究科学家称未来只要你拥有一台54个量子位的计算机你就可以轻松破解现在的各种高强度的加密算法完成现在看起来不可能的事。因此新的加密算法也在研究当中，以应对未来的量子计算机的挑战。

在此情况下研究文件加密技术是非常有益于温习旧知识并了解未来[1]的趋势的。

文件加密系统软件主要结合RSA与AES的优点设计出一个可用的、更方便进行密钥交换的、加密速度相对RSA算法要更快、加密数据长度更长的易于使用的软件系统。

# 需求分析

进行软件系统开发最开始，最重要的工作是需求分析。如果这部分做不好，那之后很可能要走很多弯路，例如需求分析错了，那之前的工作都白费了——一切都得重来。所以，接下来的工作是进行文件加密系统的需求分析。

## 核心需求分析

文件加密软件系统主要以文件加密和文件解密为主要的两个模块。它们是文件加密软件系统的核心部分。但是，为了使得软件的使用更加简洁。应该加入RSA密钥对生成模块，以防使用命令行或其他的第三方软件生成RSA密钥对操作太过复杂，不利于用户使用。因此，其实，最终就逻辑部分软件系统是由三大子模块组成：文件加密、文件解密、RSA密钥对生成。系统组成模块示意图图 1‑1如所示。



图 1-1

## 多线程需求分析

当然，加密软件只有以上那三个基本功能是不太实用的，例如当需要加密多个文件时，是不可能选择加密完一个文件之后再选择其他的文件加密的，需要可以同时加密多个文件，即使这种同时只是一种假象。所以要加入多线程，只有加入多线程才能实现一次加载多个文件进行加解密。

## 图形化界面需求分析

但是，仅仅是这些也是不可能实用的，特别是针对一些非技术人员。因为他们习惯于使用GUI，也就是图像化界面。他们习惯于点击一下就行，看按钮就知道是干嘛的图形化界面。所以，加密软件系统必须要有GUI。这一点，为了少走弯路，选择Qt进行开发。毕竟这个工具的文档还是很全的，虽然很多时候最新版的中文文档还没有翻译出来。但是，有文档总是比没有文档好的。

# 设计方案与主要技术

设计采用Qt GUI库、Qt多线程库以及OpenSSL接口相结合，打造出一款简洁易用的文件加密软件系统。关键技术是采用对称加密算法AES与非对称加密算法RSA相结合进行文件加解密。当然了这两个算法都有现成的开源实现代码那就是OpenSSL开发库接口。其中主要使用OpenSSL中EVP函数进行相关加解密操作。

## 设计方案

最基本也是最关键的设计在于加解密的协议设计，具体而言就是加密文件的格式设计以及所采用的加密算法的组合。加密文件的格式是关键点的原因在于如果格式设计不好就很有可能加密之后再也无法成功解密。所以加密文件格式的关键在于要从全局考虑加密文件格式，而不是仅仅考虑能顺利加密文件。本系统所设计的加密文件的格式如所表 2‑1示。

表 2‑1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 第0到第3字节 | 第4到第5字节 | 第6到第M个字节 | 第M到N个字节 | 第N到最后一个字节 |
| CDFC标志。 | KEY\_IV\_LEN\_SUM | ENCODE\_KEY | ENCODE\_IV | ENCODE\_DATA |

### CDFC标志

CDFC标志是由4个大写的引文字母：C、D、F、C构成的标志。该标志是几个单词的缩写它们分别是：code、decode、file、code。意思是加解密文件系统的加密标志。该标志主要用于标识当前文件是被本系统软件加密之后的文件，防止错误对其他未被本系统软件进行加密的文件进行解密从而导致不可预料的错误。这种设计主要模仿参照linux的可执行文件格式以及Windows下可执行文件格式的做法。例如在Linux操作系统中标准可执行文件格式中会使用E、L、F（Executable and Linking Format）这几个大写的英文字母作为最基本的文件格式标识指定文件为Linux下可执行文件。Windows系统也是类似的做法，它在可执行文件中写入MZ用于标志当前文件为Windows下的可执行文件。当然，对于一个成熟的系统而言只是简单的在文件中的特定位置写入特殊的标志作为特定文件的做法是不完善的不足够的。一旦其他文件在特殊情况下出现相同的标志，但是它又只是普通的字符就很有可能导致不可预料的错误。所以它们必然采用更加精确的成熟的方案来真正确认指定的文件是否是可执行文件的。但是，为了降低本设计的复杂度，本设计只是使用该标志来进行简单的文件识别，没有进一步采用其他更稳定的更可靠地方案来进行文件识别。这是本设计存在的缺陷。

### KEY\_IV\_LEN\_SUM

KEY\_IV\_LEN\_SUM是指被RSA算法使用公钥文件对AES加密算法所需的秘钥进行加密之后的长度和被RSA算法使用公钥文件对AES加密算法所需的IV值进行加密之后的长度的和。它是一个十六进制值，占两个字节。之所以只占两个字节，是因为本系统软件只采用1024位的RSA加解密标准对数据进行加密。并且在RSA加密算法中只要给定秘钥的长度，密文的长度是确定的。因为加密公式：

所以C最大值是n-1,不可能超过n的位数。但是从实际使用的角度来说实际上加密之后的密文长度为1024比特。因为现代标准基本计算机都是以8比特为一个字节进行标准化存储数据的，所以不可能出现1023比特的存储方式。因为1023不能被8整除。而使用1024的RSA加密算法对数据进行加密最后密文长度最长为1024比特也就是128字节。所以最终被RSA使用公钥进行加密之后的密文长度都是128字节是确定的。也因此没有使用分别的字节分别表示记录两个数据的长度。并且还有一点，那就是一个字节可以表示的正整数最大值为255，所以只能采用两个字节表示这两者数据长度之和。

### ENCODE\_KEY

ENCODE\_KEY是指被加密之后的AES秘钥的数据。它是被RSA使用公钥采用1024加密算法进行加密之后的数据。总共128个字节，这128个字节大多数是不可显示的字符。所以，大多数情况下使用简单文本编辑器打开被本系统软件加密之后的文件可以看到这一段数据是一个个的不可识别的乱码。一般采用十六进制文本编辑器查看比较合适。

### ENCODE\_IV

ENCODE\_IV是指被加密之后的AES秘钥的IV值的数据。它是被RSA使用公钥采用1024加密算法进行加密之后的数据。总共128个字节，这128个字节大多数是不可显示的字符。所以，大多数情况下使用简单文本编辑器打开被本系统软件加密之后的文件可以看到这一段数据是一个个的不可识别的乱码。一般采用十六进制文本编辑器查看比较合适。

### ENCODE\_DATA

ENCODE\_DATA是指被加密之后的文件的数据。这部分的数据的长度是不确定的，所以在文件最后部分。这利于文件读写数据。这部分的数据是被AES使用AES秘钥（未加密之前的）和AES的IV值进行加密的数据。这部分数据大部分也是不可显示的字符，所以大部分情况下使用标准文本编辑器打开是会出现各式各样的乱码的，需要使用专门的十六进制文本编辑器打开查看才能开到实际数据。

## 主要使用到的技术

本系统软件主要使用了OpenSSL中的AES加密算法与RSA1024加密算法的EVP封装接口。以下主要对这两个算法本身进行简单叙述。

### AES算法

### RSA算法

# 具体实现

文件加密系统的实现包括三个子系统：文件加密子系统、文件解密子系统以及RSA密钥对文件的生成子系统。每个子系统相对独立，相互之间互不干扰。这是现代软件系统的基本设计方案，有利于扩展和维护。以下将对本系统三个子系统的方方面面的实现进行论述。

## 文件加密子系统

文件加密子系统的最主要的工作是对文件进行加密，因此能不能顺利加密文件是这其中最关键的核心。

而能不能顺利加密文件的关键不仅在于加密协议的设计更要在细节做好设计。

加密文件的关键步骤是：选择要加密的文件、选择公钥文件、输入AES秘钥和AES的IV值、选择目标文件夹、使用RSA公钥加密AES秘钥并写入目标文件、使用RSA公钥加密AES的IV值并写入目标文件、使用AES加密文件并写入目标文件、保存目标文件到指定目标文件夹。其过程如图 3‑1所示。以下将按照基本步骤论述各个步骤的实现。

### 选择要加密的文件

本来按照步骤是不需要一次性选择多个文件，但是为了实际使用的方便。需要实现用户可以一次选择多个文件之后进行多个文件同时进行加密操作。这就需要使用一个链表保存用户选择的文件的文件名的字符串（包括文件绝对路径的文件名）。这需要链表结构以及对应的操作支持，本设计采用Qt经过包装后提供的QStringList类的对象实例保存。这个类提供了很多必备链表操作。它可以当做链表来使用也可以当队列来使用。

为了设计的简洁性，本操作选择在主界面的文件加密按钮监听到点击事件后触发弹出的窗口进行。但是为了选择文件的可定制性，要选择文件还需要再点弹窗上的选择文件按钮才能触发，从而弹出一个选择文件的对话框让用户选择要加密的文件。用户可以在多个文件夹之间切换选择文件。但是为了降低设计的复杂度，用户最多只能在同一个文件夹下选择多个文件，而不能在多个文件夹选择多个文件。



图 3-1

为了保证用户操作的正确性，如果用户在点击选择文件按钮之后弹出的文件选择对话框中没有选择任何文件就会弹出一个提示窗口提示用户未选择任何的要加密的文件，并将用户的光标锁定在选择的文件的按钮上。

在用户选择完要加密的文件之后把用户选择的文件的名字（包括文件的绝对路劲的文件名）加入QstringList对象实例files中，并在当前界面的下方以列表的形式展示要加密的文件的文件名（包括文件的绝对路劲的文件名）。让用户清楚自己要加密的文件到底是那些，从而减少操作失误。如果用户看了下方的文件列表之后，认为自己选择错了可以在当前的界面上再点击选择文件按钮再次触发文件选择对话框选择文件。这种实现方案虽然也能实现大多数的场景需求，但是毕竟不可以在多个文件之间选择多个文件，这样在需要在多个文件夹中选择多个不同的文件时，就会变得很棘手，有点麻烦。当然了这种需求应该很少出现的。

### 选择公钥文件

选择公钥文件是指选择RSA加密算法所需的公钥文件。按照可选性原则，本来需要设计成用户可以多种不同位数的公钥文件。但是为了降低设计的复杂性和操作的复杂度。默认只支持RSA算法的1024位的加密所需的公钥文件。而且目前没有办法在加密前检测公钥所对应的RSA位数，所以不能判断用户所选的公钥文件是否合法。只能等到加密时，OpenSSL的EVP函数报错才能知道是不是合法的公钥文件。这是当前本系统软件存在的缺陷。

按照常理应该设计成用户可以分别为各个文件选择相对应的公钥文件用于之后对AES秘钥和AES的IV值进行加密。但是，为了降低设计的复杂度与用户操作的复杂度。本设计选择只支持使用同一个公钥文件对本次选择要加密的文件的AES秘钥和AES的IV进行加密。这既降低了设计的复杂度又降低了用户操作的复杂度。况且以上提到的这种需求很少出现。所以这是一个平衡了设计难度与操作复杂度以及软件系统的完整性设计。

如之前提到的那样，在用户点击了主界面的加密文件按钮之后，就会触发一个响应事件。这个事件会触发主界面的加密文件函数。在这之后会实例化一个继承于Qt基本对话框的加密文件基本配置对话框。在这个对话框上不止有上一步提到的选择文件按钮也包含有选择公钥文件按钮。

点击该按钮会触发弹出一个选择RSA公钥文件的对话框。用户可以在多个文件夹之间来回切换选择RSA公钥文件。但是，有一点值得注意的是，所选的公钥之后要解密文件必须选择与当前相对应的RSA私钥文件，否则无法成功解密被本系统软件加密之后的文件。

## 文件解密子系统

## RSA密钥对文件生成子系统

# 测试

# 总结

通过几个月的不断学习、温习、试验，毕设终于完成。在准备完成这次毕业设计之前试了其他的工具组合，但是最终还是选择现在这个方案。主要是因为其他的组合方案要么文档不齐，要不相互之间融合不是很好。对于文档不齐就好像是夜里抹黑走路，一点都没有安全感。对于相互之间如何不太好的组合，就好像是强扭的瓜，不甜。

毕设不仅是对已有的知识的检验，更是对个人执行能力的考验。很多时候，在真正的实践之前，会觉得整个流程尽在眼前。但是，实际上真正的实践的时候会不断在各种细节徘徊。有时一个你以为很简单的点，却可能会揪住你几天。所以，实践才是检验真理的唯一标准。

本次我的毕业设计是设计一个文件加密系统。这是对我的linux C编程能力、linux c++编程能力、软件工程能力、密码编码学与网络安全知识、数据结构与算法等等能力的最好的考验。

在本次毕业设计的需求分析阶段，我就知道本次工程难点大概会在哪些方向。本来以为只要研究多一点，应该也不难。但在运用多线程时遇到了很奇怪的问题。调试也调不出来。况且，在软件开发中，多线程调试是最难的。最后，发现是OpenSSL开发库的EVP动态加载问题。这种问题，在平时你不去实践是不会知道的点。所以，只有经过类似本次系统设计才能了解到这种工程问题及解决办法。

实现阶段，也遇到其他问题。例如要展现加解密文件的状态，就要不只是显示要加解密的文件的文件名，还要显示进度。但是，刷新进度条在多线程下意外出问题。也是不知道是什么情况。最终，发现只有保存进度条的对象指针才能不出问题。还有就是OpenSSL的EVP使用问题。最开始的时候，加密或解密大文件都是每次加密或解密一定大小的数据块，然后加密或解密完成就关闭EVP动态库，下次再打开。本来按照逻辑说这是没有问题的。但是，按照EVP的加解密过程来说，这是有问题的。要加密或解密一个文件就必须从头到尾直到完成才能退出。这个问题针对多线程。最后只有把加解密一个文件在一个过程完成，才能通过。

总而言之，这次的毕业设计总算完成了。在这个过程中经历了很多苦与痛，但也学会了很多之前不知道的知识。

# 谢 辞

从确定论文选题至今，我的本科毕业论文已经顺利完成。在此，我要特别感谢我的指导老师——计算机与信息安全学院姚罡老师。从当初选定论文题目到搜集资料，从确定论文大体框架到进行开题报告，从修改初稿二稿到最终的定稿，老师给了我极悉心的指导。老师是一位十分认真严谨的老师，对我的论文要求十分严格，不论是内容格式，还是标点符号都进行了严格的把关。可以说，老师对我的论文指导尽职尽责、呕心沥血。这使得我的论文能够如期、保质保量地完成。在写毕业论文的这段日子里，老师严谨的治学态度、渊博的知识结构、精益求精的工作作风以及诲人不倦的高尚师德给我留下了深刻的印象，并将使我受益终身。在此，我向敬爱的老师致以最崇高的敬意与最衷心的谢意。

此外，还要感谢我身边的朋友和同学，感谢他们在论文写作过程中对我的指导、帮助和支持，感谢他们的的宝贵建议，感谢所有关心、支持、帮助过我的良师益友。

最后，向在百忙中抽出时间对本文进行评审并提出宝贵意见的各位老师表示衷心地感谢！

# 参考文献

[1] 余彩霞, 姚晔. 基于多级安全加密的电子文件流转中的访问控制研究[J]. 档案学通讯, 2017(02): 58–63.

# 附 录