****

本科毕业设计（论文）

|  |  |
| --- | --- |
| 题目： | **LBlock密码的实现与分析** |
|  |  |

学生姓名：

所在系别： 管理系

专 业： 信息管理与信息系统

学 号： 20124304

指导教师： 教授

**2016 年 6 月**

北京电子科技学院

本科毕业设计（论文）原创性及知识产权声明

本人郑重声明：所呈交的毕业设计（论文）是本人在导师的指导下取得的成果。对本设计（论文）的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。因本毕业论文引起的法律结果完全由本人承担。

本毕业设计（论文）成果归北京电子科技学院所有。

特此声明

毕业设计（论文）作者签名：

年 月 日

**【摘 要】**随着信息技术的发展，物联网技术的普及，数据安全问题引起人们密切关注，物联网的应用组件大致是由计算能力较弱的微型处理设备组成，因其运算量小、存储空间有限，导致许多物联网中的传统密码已经不能满足日益发展的数据安全需求，如高级加密标准（AES）。为了保护数据安全，提高数据运算量和存储空间，出现了许多轻量级密码算法。其中LBlock算法是由我国吴文玲等人在2011年ANCS会议上提出来，与其他轻量级分组密码相比，LBlock算法执行效率更高，适应性更强。LBlock算法是一种32轮、分组长度为64比特、主密钥为 80比特的类 Feistel结构的轻量级分组密码。在VC6.0软件环境下对LBlock算法进行实现，然后在此基础上结合TCP/IP协议以及Hash函数对其进行实用化程序设计。

**【关键词】** 轻量级分组密码 LBlock算法 算法实现 安全性分析

**【Abstract】**The information technology is developing very quickly, data security issues become increasingly prominent in the field of IOT caused serious concern, networking System components more computing power is relatively weak micro-processing equipment, its limited computing, storage capacity, leading to a traditional password, such as the advanced encryption standard (AES), we have been unable to better solve the problem of data security of its existence. To protect your data secure computing resource constrained environment, there have been a number of lightweight cryptographic algorithms. The LBlock proposed by a lightweight block cipher, compared with other lightweight block cipher LBlock software and hardware implementation of more efficient and adaptable. LBlock algorithm 32, the packet length is 64 bits, the master key for the 80-bit block cipher lightweight class Feistel structure. In VC6.0 software environment for Mickey algorithm to achieve, and then combined on the basis of TCP / IP protocols as well as its practical function Hash programming.

**【Keywords】**Lightweight block cipher LBlock algorithm achieve security

目 录

前 言

步入21世纪信息时代，互联网，移动通信快速发展，人们对信息数据的安全关注度越来越高。最近几年，科技发展带来的信息进步是显而易见的，RFID技术和数据传输渗透到人们工作和生活的点点滴滴，由于轻量级分组密码算法结构较简单，密钥长度相对较短，硬件实现相对容易，能耗少，对设备的硬件要求也低，并对信息数据提供很大强度的安全保障，很适应微型设备，最近这几年轻量级分组密码有了迅速的发展，并在人们的工作和生活中起着十分重要的作用。

近代密码学，尤其是进入二十一世纪以后，由于信息安全的不断深入，信息安全的重要性愈发突出。密码技术作为信息安全最重要的中心部分，涉及密码算法的部件设计、各种算法的安全性分析还有数字签名等各种不同的领域，受到全世界各国的普遍重视。

分组密码在对称密码学中起着十分重要作用，进入20世纪90年代，分组密码的理论得到普及，应用得到迅速发展。分组密码的原理大致如下：在加密过程中，明文被划分成相等长度的，并使用同一个加密算法和密钥对一组组明文数据进行加密。设明文数据序列为p1,p2,…，每m个组成一组明文P= (p1,p2,…,pm)'加密密钥为K= (k1,k2,...,kt)，明文P在加密密钥的控制下转换成一组密文C = (c1,c2,...,cn)，简记为C=Ek（P）。

轻量级分组密码LBlock算法是由吴文玲等人在2011年ANCS会议上提出来的，其分组长度是64比特，密钥的长度是80比特，是32轮迭代分组算法， 采用Feistel算法结构，加密算法和解密算法互逆，使用的轮密钥顺序相反，其次对密钥扩展算法进行了独创设计，采用的是Feistel结构和SPN结构。因此分组密码在现实生活中有着非常广泛的应用，研究LBlock算法的实现与分析具有很高理论价值和实用价值。

1 绪论

轻量级分组密码算法是分组密码[1]其中一种，分组密法算法的设计人大都受到DES算法和AES算法设计原理的影响。但轻量级分组密码不仅可以保障数据的安全，还提高的执行效率，以便适应特殊复杂环境，本质上，轻量级分组密码算法足一种有限轮的带密钥的置换，其大致原理是将一定长度的明文数据经过分组，转换为相同长度的密文。分组密码算法主要有两种设计结构：其一是Feistel网络结构，不是所有分组都进入轮函数，只有一半分组才可以率先实现加解密，并且加解密互逆，使用的硬件资源较少，所以在计算能力低的环境中使用是十分有效的；其二是代替-置换网络(SPN)结构，每一轮包括混淆层和扩散层。该结构的最大优点是能够从理论上给出最低差分特征概率和最佳线性逼近优势的边界。本课题通过对分组密码算法LBlock[2]进行实现与分析，理解分组密码算法的设计原理，进一步掌握现代分组密码学的知识，学习各种分组密码攻击方法，优化该算法的某些性能。

## 1.1分组密码

1.1.1分组密码基本原理

分组密码实现的具体过程如下：将明文数据表示的数字序列，划分成相等长度，每组长度为n，可理解是n的矢量，然后对每组明文数列进行相同的加密计算，根据相应的密钥，变换成等长的输出序列。解密过程是加密的逆运算。扩散（diffusion）和扰乱（confusion）是影响密码安全的主要因素。扩散可以使明文中的单个字节或数据影响解密文件中的多个数据，从而使明文的统计特征丢失，影响运算准确性，等同于明文的统计特征被扩散。扰乱可以让密钥与密文的统计特征丢失，使信息数据之间的关系变得尤为复杂，从而提高了使用统计方法进行攻击的难度。解决扰乱的办法可以通过代换算法来实现。

AES是分组密码中最典型的算法之一。AES的征集开启了分组密码研究的新大门，大约有15个AES候选算法，也体现了如今分组密码设计实现的水平。分组密码所采用的整体结构有如下三种：包括Feistel结构，典型的代表有CAST-256算法、DEAL算法、DFC/E2算法等，SP网络，典型的代表有Safer+算法、Serpent算法等，还包括其他密码结构，典型的代表有Frog算法和HPC算法。DES是典型的Feistel结构代表，Feistel因其的公布而广为流传人知，并且被许多分组密码所采用。Feistel结构的有其独特的好处，可以使得加解密相似，这一点在实现中尤其重要。其次，SP网络的特点是扩散特性尤其突出。在现有的分组密码中，所有的基本运算包括异或、加、减、乘及数据循环等。查表运算为DES的安全打下了基础，通过仔细地选择S-盒，能有效地抗击线性和差分密码分析攻击，提供好的数据及密钥比特的雪崩特性。而S-盒的设计与分析实现是分组密码设计实现中的重要组成部分，它直接影响密码算法的安全性，对S-盒的设计并没有一个确定的要求，但一般提高S-盒的非线性度、函数的代数次数以及项数。

1.1.2分组密码S盒

S盒(substi tution Box)首次出现在Lucifer算法中，随后因DES的使用而广为流行。S盒是许多分组密码算法中唯一的非线性部件，因此，它的密码强度决定了整个密码算法的安全强度，它的工作速度决定了整个算法的制乱速度。特别地，使用高强度的S盒对于增强Feistel型密码，SPN型密码和IDEA型密码的安全性起着至关重要的作用。对S盒的深入研究不仅有助于迭代分组密码的设计，而且对于以非线性变换为核心的密码算法的分析有相当价值，同时也有助于快速寻求满足某些特定密码需求的新的密码函数。

目前对分组密码的设计主要集中在非线性S盒的设计、置换方法的选择和寻找好的密钥扩展方案[3]。S盒的设计和构造主要集中在S盒的设计准则和构造方法上，S盒的设计准则主要有：非线性度、差分均匀度、代数次数及项数分布、扩散准则，平衡性以及相关免疫性。基于上述设计准则，人们提出了许多S盒构造方法，如随机提取并测试、使用数学函数构造等方法。

## 1.2 分组密码发展现状

1.2.1国外发展现状

分组密码是现代密码学研究中的一个重要分支，其诞生和发展有着重要的理论价值和广泛的实用背景。最初出现的密码本质上是序列密码，它已广泛地应用于军事、外交等方面。分组密码是以二十世纪七十年代DES(Data Encryptioil Standard)的出现为标志，在这以后得到了广泛的讨论和发展。

由于DES的密钥量只有56bits，在计算机技术高度发展的今天，DES受到严重的安全威胁，事实上，1997年人们已成功破译由DES加密后的信息。1997年9月，美国国家标准技术研究所(NIST：National Institute of Standards and Technology)为了履行其法定职责，发起了一场推选用于保护敏感的联邦信息的对称密钥加密算法活动。1998年8月，NIST宣布接受十五个候选算法并邀请全世界密码研究界协助分析这些候选算法，包括对每个算法的安全性和效率特性进行初步检验。NIST考察了这些初步的研究结果，并且选定MARS、RC6、RijndaelI、Serpentl和Twofish五个算法作为参加决赛的算法，经公众对决赛算法进行更进一步的分析评论，2000年10月，NIST决定推荐Rijndael作为高级加密标准(AES：Advanced Encryption Standard)，2001年11月，Rijndael被采纳作为高级加密标准。继美国推出AES计划以后，欧洲于2000年1月启动了新欧洲签名、完整性和加密计划一NESSIE(New European Schemes for Signatures，Integrity and Encryption)计划，以适应21世纪信息安全发展的全面需求。该计划为期三年，投资33亿欧元，主要目的就是通过公开征集和进行公开的、透明的测试、评估提出一套高效的密码标准，以保持欧洲工业界在密码学研究领域的领先地位。2001年底，NESSIE工作组在十七种候选的分组密码算法中，选定了IDEA、Khazad、MISTYl、Camellia、SHACAL、RC6、SAFER++等七种分组密码算法为分组密码的第二轮决赛算法，2003年2月27日，NESSIE工作组公布了包括分组密码、认证码、杂凑函数和数字签名等在内的十七个标准算法，其中M1STYl、Camellia、SHACAL三个分组密码算法连同AES算法Rijndael一起作为欧洲新世纪的分组密码标准算法。由此可见，分组密码的设计与分析是目前国际信息安全领域的前沿热点问题。

1.2.2国内发展现状

轻量级分組密码算法作为一种特殊的分组密码算法，它们在硬件实现、加密速度、运行功耗等方面与AES等高强度密码算法相比有明显的优势，更适合物联网微型计算设备使用。常见的轻量级分组密码算法DESL、PRESENT，HIGHT， mCrypton，LED，MIBS，LBl0CK，KLEIN，KATAN等等。

现在的轻量级分组密码算法大都受到DES和AES设汁原理的影响。例如A. Bogdanov等人在CHES2007上提出的轻量级分组算法PRESENT,轮函数就是代替-置换网络(SPN)结构，它借鉴了AES最终候选算法Serpent的设计思路。该算法共31轮，分组长度为64比特，密钥规模分为80比特和128比特两种，分别称为PRESENT-80和PIIESENT-128。与其它轻量级的分组算法相比，它的硬件执行效率更高，是轻量级分组算法中的佼佼者。轻量级分组密码算法MIBS是M. Izadi等人在CANS 2009上首次提出，使用了 Feistel的算法结构。该算法针对RFID标签设计，它的硬件资源使用很少，远合在计算资源受到限制的环境下使用。MIBS算法共有32轮迭代运算，其分组长度为64比特，密钥长度为64比特和80比特。2009年，在CHES2009会议上Cannkre等人提山了一种适用于硬件实现的轻量级分组密码KATAN/KTANTAN,该结构类似于流密码算法的加密结构，使用了位的逻辑操作和移位操作。由于该分组密码在0.13nm工艺下具有硬件成本低、执行效率高等特点,非常适合RFID标签等对硬件成本和执行效率要求苛刻的环境。KATAN/KTANTAN的密钥长度为80比特，需进行254轮的加密，分组长度有32比特，48比特,64比特三种。

**1.3分组密码研究现实意义**

密码体制是信息安全的重要保障，而分组密码是密码体制的重要组成部分。通过研究LBlock分组密码重点研究，具有很高价值的研究意义：

1.分组密码因为其体积小，运行速度快，适用于高速、大规模的网络数据传输、移动通信，与现代网络、移动通信发展趋势相吻合。

2.分组密码符合现在密码算法发展趋势，具有很高的综合性、标准型。综合性是指分组密码不仅具有其本身的优点，在设计时一般也会流密码等其它的一些密码算法的优点。

3.分组密码还有很大的发展空间。分组密码中很多关键性问题还有待解决。

4.分组密码的可证明安全性还有很大空缺。在可证明安全性上，公钥密码走到了前列。而分组密码的可证明安全性可以说是才刚刚起步。

5.目前，设计分组密码的热点是设计带有认证功能的分组密码。机密性与完整性是密码算法设计的两大目标。而在分组密码发展过程中，机密性的发展已经略显成熟，但完整性却有待提高。

**1.4本文章节安排**

本文的主要研究对象是我国轻量级密码算法LBlock分组密码算法。此次毕业设计的创新点有两处：1.在实现LBlock分组密码算法的基础上，结合其它一些技术手段，编写出基于LBlock分组密码算法的实用化程序。2.对LBlock分组密码算法研究分析的基础上，尝试对LBlock分组密码进行某些分析。本文的结构安排如下：

第一章简要介绍分组密码，给出分组密码的基本原理、国内外关于分组密码研究的现状以及对本课题研究的现实意义。

第二章介绍LBlock分组密码算法以及实现该算法的重要工具MFC。本章详细介绍了LBlock分组密码算法的原理及其运算过程，并介绍了MFC的基本特性。

第三章主要介绍实用化程序中使用的两种主要手段，TCP/IP协议以及基于MD5算法的HASH值文件摘要计算。

第四章对LBlock分组密码进行了一些分析，并给出了一个对LBlock分组密码算法的分析。

最后综合全文，得出结论。并对在毕业设计中提供帮助的各位老师表示感谢。

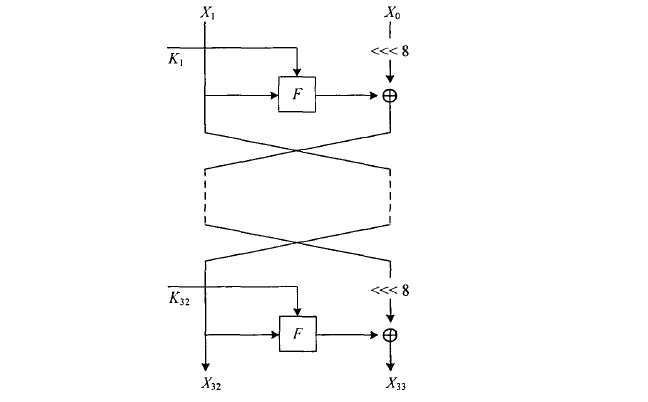
2 LBlock分组密码

**2.1LBlock分组密码的算法理论**

LBlock密码是吴文玲等人在2011年ANCS会议上提出来的一种轻量级分组密码算法。该算法分组长度为64比特，密钥长度为80比特。它使用了变形的Feistel结构，设计轮数为32轮。LBlock分组密码算法由三部分构成：加密算法、解密算法和密钥扩展算法。其硬件实现性能为：1320GE，在lOOKHz的频率下吞吐量为200kbps。

**2.1.1LBlock算法的加密过程**

LBlock加密算法是一个32轮迭代的变形Feistel结构的算法，令要加密的64比特明文M = X1||X0，加密后得到的密文C=X33||X32，整个加密过程如图2.1所示。



**图2.1 LBlock密码算法加密过程**

**2.1.2LBlock算法的输入输出**

对一个64比特明文用M = X1||X0来表示，则数据的加密过程如下:

1.For i=2,3,...,33,do

Xi =F(Xi-1，Ki-1)（Xi-2<<<8）左移

2.输出C= X32 - ||X33作为对应的64比特密文。

**2.1.3LBlock算法的F函数**

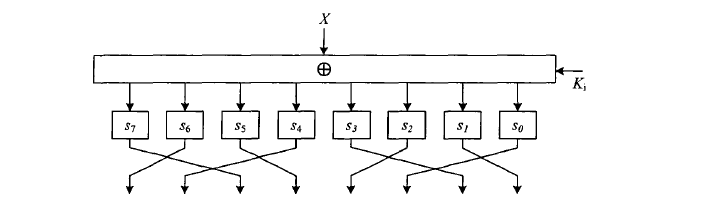
每轮所使用的F函数根据如下定义：

轮函数F：轮函数F是由混淆函数S和扩散函数P组成，组成方式如下：

F{0，1}32\*{0，1}32 {0,1}32

（X，Ki） U=P（S（XKi））

F函数如图2.2所示



**图2.2 LBlock密码算法F函数**

**2.1.4LBlock算法的S函数**

混淆函数S：混淆函数S是非线性层，是轮函数F的重要部分，它由8个S盒组成，每个S盒为4x4比特：

S{0，1}32 {0，1}32

Y=Y7||Y6||Y5||Y4||Y3||Y2||Y1||Y0 Z=Z7||Z6||Z5||Z4||Z3||Z2||Z1||Z0

Z7=s7(Y7)，Z6=s6(Y6)，Z5=s5(Y5)，Z4=s4(Y4)，Z3=s3(Y3)，Z2=s2(Y2)，Z1=s1(Y1)，Z0=s0(Y0)，

**2.1.5LBlock算法的P函数**

扩散函数P：是8个4比特字组成的置换，表示如下：

S{0，1}32 {0，1}32

Z=Z7||Z6||Z5||Z4||Z3||Z2||Z1||Z0 U=U7||U6||U5||U4||U3||U2||U1||U0

U7=Z 6U6=Z4U5=Z7U4=Z5 U3=Z2U2=Z0U1=Z1

**2.1.6LBlock算法的S盒**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S0 | 14 | 9 | 15 | 0 | 13 | 4 | 10 | 11 | 1 | 2 | 8 | 3 | 7 | 6 | 12 | 5 |
| S1 | 4 | 11 | 14 | 9 | 15 | 13 | 0 | 10 | 7 | 12 | 5 | 6 | 2 | 8 | 1 | 3 |
| S2 | 1 | 14 | 7 | 12 | 15 | 13 | 0 | 6 | 11 | 5 | 9 | 3 | 2 | 4 | 8 | 10 |
| S3 | 7 | 6 | 8 | 11 | 0 | 15 | 3 | 14 | 9 | 10 | 12 | 13 | 5 | 2 | 4 | 1 |
| S4 | 14 | 5 | 15 | 0 | 7 | 2 | 12 | 13 | 1 | 8 | 4 | 9 | 11 | 10 | 6 | 3 |
| S5 | 2 | 13 | 11 | 12 | 15 | 14 | 0 | 9 | 7 | 10 | 6 | 3 | 1 | 8 | 4 | 5 |
| S6 | 11 | 9 | 4 | 14 | 0 | 15 | 10 | 13 | 6 | 12 | 5 | 7 | 3 | 8 | 1 | 2 |
| S7 | 13 | 10 | 15 | 0 | 14 | 4 | 9 | 11 | 2 | 1 | 8 | 3 | 7 | 5 | 12 | 6 |
| S8 | 8 | 7 | 14 | 5 | 15 | 13 | 0 | 6 | 11 | 12 | 9 | 10 | 2 | 4 | 1 | 3 |
| S9 | 11 | 5 | 15 | 0 | 7 | 2 | 9 | 13 | 4 | 8 | 1 | 12 | 14 | 10 | 3 | 6 |

Xj=（F（Xj+1，Kj+1）Xj+2）>>>8

**2.1.7LBlock算法的密钥扩展算法**

LBlock密码算法的密钥扩展算法：将80比特主密钥K存入密钥寄存器中，定义密钥寄存器K=k79k78k77k76。。。。。k1k0，输出密钥寄存器K中的最左边的32比特作为轮密钥K1，然后进行如下操作：

1. For i=l,2,...,31，根据如下方式更新密钥寄存器K

(a) K<<<29

(b){ k79k78k77k76}=s9{ k79k78k77k76。}

{ k75k74k73k72}=s8{ k75k74k73k72}

(c) { k50k49k48k47k46}{i}2

(d)输出密钥寄存器K中的最左边32比特作为轮密钥Ki+1，其中S8、S9是两个4 X 4比特S盒。

**2.2MFC窗口界面设计**

在初步实现算法以后，将算法编入程序，用实践检测理论。在里这主要实现了一个基于MFC窗口化程序[4]以及LBlock分组密码算法的加解密程序，使其带有输入密钥，输入密钥。使用生成的密钥流，与选择的.txt文件或手动输入明文进行加密，并生成密文.txt文件；解密时，输入相同的密钥，与选择打开的密文.txt文件解密为明文，并显示的程序。在实现过程中，使用MFC窗口化编程使程序可视化。

**2.2.1Windows编程**

Windows程序一般分为三类，具体如下表（表2.2）所示：

**表2.2 Windows程序分类**

|  |  |  |
| --- | --- | --- |
| 程序分类 | 面向对象/面向过程 | 程序界面 |
| 控制台程序 | 面向过程 | 命令行界面 |
| Win32程序 | 面向过程 | 窗口程序 |
| MFC程序 | 面向对象 | 窗口程序 |

MFC程序是一种事件驱动的编程模式[5]，它基于Windows消息机制[7]：

Windows是一种可实现多个任务通过窗口运行的OS。在Windows中随便某一时刻，资源都是由其统一进行管理、分配。所以我们可以看到在这一时刻，会有多个应用程序正在运行。也就是说，在Windows中并不是一个程序享用所有的资源。

Windows在运行过程中就像守卫一样随时观察用户的动作，在观察过程中不断分析这个动作是与某一个应用程序存在关系，随后将这些动作信息发送给这个与用户动作存在关系的应用程序。

与此同时，在Windows运行过程中，运行的应用程序也在随时等待着Windows发送的消息。如果这个应用程序发现它有还没有处理的信息，就从消息队列中拿取这条信息并对它进行分析，然后根据消息的内容作出不同的响应回馈用户之前所做的操作。

**2.2.2 MFC**

MFC是Microsoft Foundation Class的缩写，是在Microsoft Windows操作系统里的一个重要的应用程序框架，它的本质其实是作为C++语言与Windows的接口存在的[8]。它的实际工作是通过选择MFC类库中的合适的类，并调用这个合适的类下的合适的函数完成用以相应功能。

从宏观意义上看，MFC其实给出了我们需要的应用程序的轮廓，还给出了标准的实现用户接口的方式[6]，而编程人员做的则是把需要的应用程序的模式、功能填入这个轮廓。一般编程工具VS提供AppWizard用以生成最原始的应用程序轮廓；资源管理器用以更加直接的设计用户接口；ClassWizard用来协助将代码等填入既定框架。

MFC具有以下特性[9]：

1.封装了Windows API（Windows提供给应用程序与操作系统的编程接口，即应用程序可以使用Windows系统的函数。程序可以在任何时候调用Windows API函数，最大程度的利用了Windows。

2.提供了一些通用类的支持，如列表、数组和映射类的集合、字符串类、时间日期类、文件访问类等。

3.具有丰富的用户界面特性。支持单文档视图（SDI）结构、多文档视图（MDI）结构、对话框、菜单、常用控件、打印预览和打印。

4.支持动态链接库.DLL，使DLL的编写越来越容易。

5.支持数据库的访问。提供了多种数据库开发技术，主要包括ODBC、API、MFC ODBC、DAO、OLE DB和ADO。

6.支持网络和通信。使用ISAPI（网络服务接口）[7]、Windows Socket以及封装的Win32 Internet接口，使网络通信变成更为轻松。

7.支持OLE（对象链接和嵌入）。支持对象嵌入、对象链接和自动化的OLE服务器和容器。

8.支持ActiveX控件。可以方便的生成和使用ActiveX控件。

MFC是实现C++程序窗口化可视的最关键部件，它能够与动态链接库相互配合实现界面的设计。尽管随着信息技术的发展，MFC很有可能被取代，但它作为使C++程序真正可用化且作为编程人员发展的基础等的重要作用是不可替代的。

**2.2.3对话框**

对话框作为程序窗口化、可视化的最重要部分，它的作用是作为一个接口，将Windows下的应用程序中从用户所得到的数据、信息等，进行相应的处理，并将处理结果输出到屏幕等输出设备。对话框从宏观看由两个部分组成：对话框资源[8]，通过对话框编辑器来实现，可以编辑对话框相关的参数，例如其大小、位置、样式等；对话框类，它的功能是帮助用户建立一个能够与对话框资源产生联系的类，在MFC中，一般由CDialog类派生。

而对话框则有两种类型，分别为模态对话框与非模态对话框。简单理解，模态对话框是指，当该对话框运行时，应用程序会暂停，并等待该对话框运行完毕并关闭；非模态对话框是指，在该对话框运行过程中，可以允许应用程序进行其它任务。

在执行MFC窗口化程序时，对话框程序的运行过程如下：

1.实例化应用程序类的对象（CWinApp）

2.调用应用程序类中的Initlnstance函数

3.定义主对话框类的对象

4.调用对话框类的构造函数

5.调用对话框类的DoModal函数

6.调用对话框对象中的OnlnitDialog函数初始化对话框

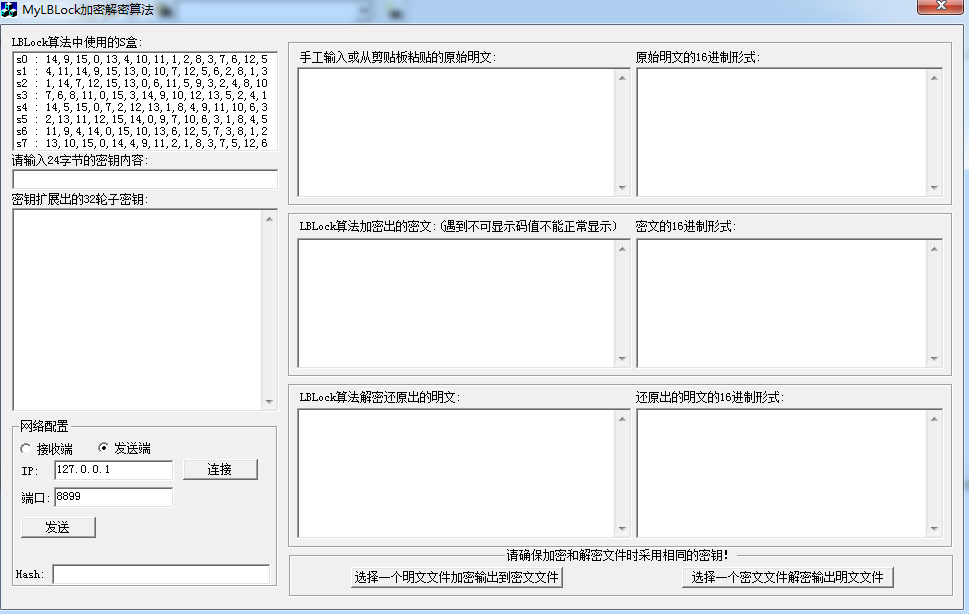
7.调用Onpaint函数，绘制窗口

8.对话框退出后，返回DoModal函数状态

9.返回Intlnstance函数，结束应用程序。

3 LBlock分组密码的实用性软件实现

在初步理解LBlock分组密码算法原理以及使用软件对其实现后，集合其他技术实现一款可以实现.txt文件传输，并计算其HASH并进行比对的实用程序。在输入密钥后，选择加解密的文档。整个程序分为发送端与接收端两个部分。程序主显示界面如下图（图3.1，图3.2）所示：



**图3.1 发送端**



**图3.2 接收端**

**3.1HASH函数**

随着信息安全的不断发展，对于消息的完整性的需求也在不断提高。 其中，HASH函数就是一种计算消息完整性的函数。HASH函数[9]能够计算文件的消息摘要，即将一段任意长的文件映射计算成一固定长的输出串。这个软件使用的是基于MD5算法的HASH函数。

伴随着1949年Shannon理论的出现，密码学的发展步入现代密码学阶段，它主要分为三个分支：流密码、分组密码以及HASH函数[10]。HASH函数因为它的特性，所以被广泛应用于消息认证、数字签名等方面。

简单地讲，HASH函数它的作用是对消息进行了“压缩”，将任意长的消息映射为一定长度的HASH值。HASH函数的定义为，可以看出HASH函数是“一对多”的映射，从数学角度可以看出它的定义域要比它的值域大得多。

HASH函数根据其设计方法的不同大致可以分为以下三类：

1.使用分组密码模式的HASH函数。这种HASH函数只是说它模仿了分组密码的模式，而不是使用带有分组密码的组件。这种HASH函数的优点在于由于它与分组密码的模式很类似，而分组密码的发展又比较充分，所以可以直接利用某些分组密码的既定成果。而它的缺点在于其效率比较低，而且也不能完全贴合于分组密码的安全性。

2.使用离散对数构造的HASH函数。这种HASH大量使用类似逻辑或、与等的二进制运算，而这种运算本质上构成非线性是困难的，必须与其它运算结合。其效率很低，并且安全性比较脆弱。

3.标准HASH函数。这类HASH函数，一般指MD系列以及SHA系列。它们最大的特点就是直接构造HASH函数，所以它的效率要比前两种高很多。它的缺点是缺乏扩散性，并且我国山东大学王小云等人的研究对其安全性提出了质疑。

**3.1.1HASH函数特征**

HASH函数最重要的作用是用来计算消息的完整性。完整性是指没有经过授权不能对信息进行编辑、修改或者插入等的特征。它的作用使用来保证接受的数据是实体发出并且是完整的。它的原理是可以将任意长度的消息映射成特定长度的消息值，类似与人类的指纹。通过验证这个特殊标记，原消息的完整性便会得知。

在现实中，HASH函数应当满足如下一些要求[11]：

1.对于给定的数据，容易计算出其摘要值。

2.对于已知的消息摘要，用消息摘；要恢复原文是困难的。

3.不可能从两个不同输入消息映射一个消息摘要。即两个不同消息不能具有相同的消息摘要。

**3.1.2MD5算法**

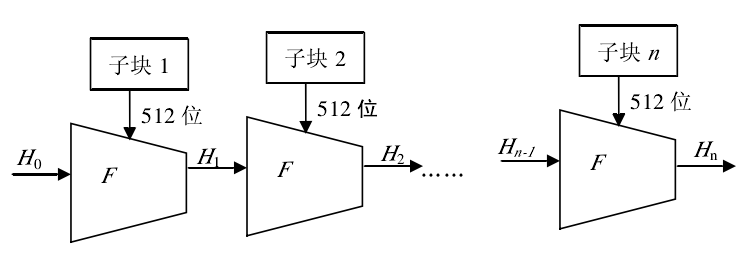
HASH函数根据其是否有密钥输入可以分为不带密钥的HASH函数以及带密钥的HASH函数（消息认证码）。其中的MD5算法属于不带密钥的HASH函数，它以任意长的消息作为输入，输出为128位的摘要。1990年出现了使用布尔预算及其它基本的算术运算方法，采用迭代机构，并用其思想，产生了三轮压缩的迭代函数MD4算法[12]。在1992年，在对MD4算法改进的基础上，产生了MD5算法。MD5算法是一种非常常见的单向算法，MD即信息摘要算法，它可以将任意长度的信息映射成为固定长度的信息摘要。MD5算法相比于MD4算法，增加了“安全带”的概念，这使得MD5算法比起MD4算法更加安全，但同时增加了运算速度。

MD5运算时，首先进行初始化，将信息分割为512位的分组，再将每一分组继续分割为32位的十六个分组。再由算法进行一系列变换，得到四个32位的分组。最后，由这四个32位分组组合而成一个128位的值。具体过程如下：

1.为给定信息填充。方法为在其最后一位添加一个1，并添加若干0，是整个序列模512的值为448。

2.用给定的四个32位整数参数（链接变量）为MD5算法做初始化[13]。

3.开始进行“四轮循环”操作，循环次数即为512位分组的组数。算法包括模232加法、循环左移位一个包括四个布尔函数组成的。



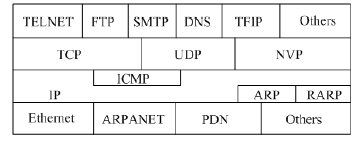
**图3.3 MD5算法流程图**

MD5算法是一种相对安全且易用的加密算法。相较于MD4算法，它牺牲速度来提供更高的安全性。被广泛应用于信息安全领域。

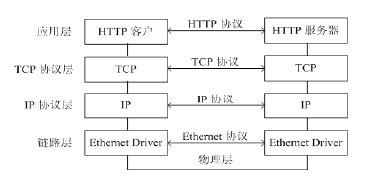
本软件中使用MD5算法分别对接收端和发送端的密文进行计算，并得到其摘要值，并进行一致性判断，从而得出在文件传输过程中有没有被未经授权的进行过修改、编辑等非法操作。

**3.2基于TCP协议的文件传输**

TCP/IP(Transmission Control Protocol/Internet Protocol)即为传输控制/因特网协议，它的思想是通过网络连接层，并在其各种低层协议上建立一种网络连接，能解决不同机器、不同网络互联问题，使用户的通信更加方便的一种网络协议。它是当今世界使用最为广泛，且不受某些硬件限制的网络通信协议。TCP/IP的结构为层级结构，并且具体分为四层，分别为应用层、传输层（TCP层）、网络层（IP层）以及链路层[14]。它的底层为高层提供服务且每一层都具有特定的功能，它的结构如图3.4所示。应用层的主要作用是确定应用程序具体要使用那些协议，它也是TCP/IP协议中最大的子集；传输层它的主要作用是为两个不同的具有网络通信的主机上的不同应用程序提供端口间的通信；网络层作用是使整个分组在网络中的活动被协调统一；链路层的作用则是给它的上层协议接收发送非配数据包。图3.5给出了TCP/IP协议的一种协议层模型，我们可以看出其每一层都一一对应。当今信息技术与计算机技术的发展，诞生了无论规模最大还是使用人数最多的Internet，而TCP/IP协议则是Internet的基础。也正是因为这个原因，使得采用TCP/IP的网络信息技术具有更广阔的发展趋势。



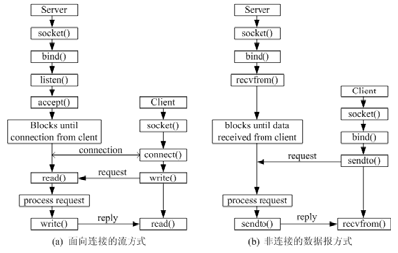
**图3.4 TCP/IP结构图**

****

**图3.5 TCP/IP协议层模型**

**3.2.1Socket套接字**

Socket本质其实是一种通信端点，它使接收和发送数据变为可能。Socket可以分为Stream Sockets（流套接字）和Datagram Sockets（数据报套接字）[15]，它们的通信方式如图3.6所示。其中Stream Sockets提供了一种可传输大量字节流数据，且传输过程是误差错的通信方式；Datagram Sockets提供了一种可双向传输，但不能保证可靠性的通信方式。

****

**图3.6 两种Socket通信方式**

从另一方面讲，Socket其实是一种数据结构：作用是搭建一条通路，用于连通本没有互联的进程，用以实现它们的接收、发送数据。而实现互联的Socket，其连接的两端都可以实现数据的传输。它在TCP/IP协议层级结构中，位于第二层传输层中。Socket的最大功能是它屏蔽了具体应用程序和不同操作系统的差异，使得在不同操作系统、不同应用程序之间传输数据成为可能。

我们知道，两个地点实现通讯，要在两个地点设立电话。Socket的作用就类似于“电话”的作用,它是进行网络通信的两个节点中可以被应用程序访问的逻辑实体。

在本软件中使用了基于Stream Sockets通信方式的C/S模式的文件传输结构，用于保证数据的完整。

**3.2.2Winsock通信机制**

应用程序之间的网络通信，究其本质，其实是应用程序之间能够通过利用相同的网络通信协议来完成网络通信以及数据的传输。也就是说，具有网络通信的应用程序是与Windows Sockets API进行交互的，而不是直接的与TCP/IP进行交互[16]。在Windows下，不同应用程序以及Winsock都是在其用户模式下完成工作的，所以Windows不能够只依靠Winsock完成不同网络之间的通信，而且还需要底层的支持。在VC6.0中，我们可以直接调用WinSockAPI，而不用去考虑TCP/IP细节等。

Socket核心驱动程序以及传输驱动程序接口（TDI，Transport Driver Interface）是Winsock与传输协议之间的通信纽带，它们的主要作用是负责Windows核心态的网络通信。具有网络通信的应用程序的Socket核心驱动程序的接口便是Winsock；而TDI则是Socket核心驱动程序与传输协议间的核心、缓冲区及其管理等的核心模式，使其能够提供Socket核心驱动程序给应用程序。

Winsock通过应用程序实现发送与接收数据时，其实并非由Winsock收发数据，本质上由AFDSYS（核心模式驱动程序）通过对缓冲区进行管理，进行发送与接收数据。AFDSYS在这里起一个“平台的”作用，也就是说，发送端发送数据到AFDSYS后，立刻返回，而AFDSYS保存至缓冲区，并在后台将数据发送出去。同理，在接收端，首先由AFDSYS接收数据，并将接收的数据储存于自身缓冲区，后台命令应用程序来拿取数据。之后缓冲区的数据存入到所使用应用程序缓冲区。其中，AFDSYS中的接收缓冲区与发送缓冲区的大小一致，都为8192字节。但在实际应用中，它们的大小都可以根据应用程序的实际需求来确定。

最后，这个程序是基于TCP/IP协议，以及Winsock编程原理进行实现的。它采用客户端/服务器（C/S，Client/Server）模式。但为了实际使用，在不影响其本质的情况下，将客户端与服务器（接收端与发送端）合并成了一个程序。

4 LBlock分组密码分析

在完成基于LBlock分组密码的基本算法的实现，以及在实现算法的基础上，对其可视化并加入更多的功能使其更加实用。之后，尝试对其进行分析，以便对LBlock分组密码它的特性以及安全性有更加直观了解。

**4．1LBlock分组密码**

由于RFID技术在工作和生活中的广泛应用，自从LBlock分组密码算法在2011年ANCS会议上由吴文玲等人提出以来。很多研究分组密码算法的科研工作者都对它的安全性进行了分析。

Marine Minier[17]等人于2012年提出了LBlock密码算法的相关密钥不可能差分攻击，Yu Sasaki等人于2012年提出了 LBlock密码算法的积分攻击，Shusheng Liu等人于2012年提出了 LBlock密码算法相关密钥差分分析，Ya Liu等人于2012年提出了LBlock密码算法的不可能差分分析，Jiageng Chen等人于2013年提出了LBlock密码算法的差分攻击，Hadi Soleimany等人于2013年提出LBlock密码算法的零相关线性分析，具体攻击结果由表4.1给出。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 攻击方法 | 轮数 | 数据 | 时间 | 内存 |
| 零相关线性分析 | 22 | 264 | 270.54 | 264 |
| 积分攻击 | 22 | 261 | 270 | 263 |
| 差分攻击 | 17 | 259.75 | 267.52 |  |
| 相关密钥差分攻击 | 22 | 264.1 | 267 |  |
| 不可能差分攻击 | 21 | 262.5 | 278.7 |  |
| 相关密钥不可能差分攻击 | 22 | 247 | 270 |  |

**4.2LBlock分组密码的分析方法**

**4.2.15 轮 LBlock 的多项式表示**

利用符号计算数学软件 Mathematica7.0 给出 LBlock 第 1 轮~第 5 轮每个输出比特的多项式表达式(以明文比特和主密钥比特为自变量)。首先给出 LBlock 的非线性组件 S 盒的代数表达式[18]。然后给出轮子密钥的代数表达式。有了这些表达式后，可以直接用 Mathematica 编出LBlock 的纯符号加密程序，其程序流程和用 C 语言编写LBlock 的加密程序一样。

**4.2.2S 盒的代数表达式**

设 X =x3x2x1x0表示 S 盒的输入，利用文献[8]的算法即可求得输出。这里只给出 s0的输出s0 ( X )=y3 y2 y 1y0的代数表示式，具体如下：

y3=1+x0x1+x0x2+x3+x1x3+x0x2x3

y2=1+x0+x0x2+x1x2+x3+x0x3+x2x3+x0x2x3+x1x2x3

y1=1+x0+x2 +x0x2+x1x2+x3

y0=x0+x1+x2+x3+ x2x3

**4.2.3轮 子密钥的代数表达式**

根据 S 盒的代数表达式，将对应的S 盒的代数表达式作为子函数， 就可以得到轮密钥用主密钥表示的代数表达式。

程序输入：

(1)MasterKey：80 比特，用纯符号表示。

(2)round：轮数(由于轮数较大时，轮子密钥的表达式也比较复杂，因此设定轮数作为参数来控制只生成加密函数需要的轮子密钥。本文只能给出 5 轮 LBlock 的代数表达式，因此，轮子密钥的代数表达式也只需要生成5 轮)。

程序输出：

round 个轮子密钥。每个轮子密钥都是 64 比特。每个比特都是用主密钥表示的代数表达式。其中，s9和 s8用得到的多项式表示进行定义

有了 S 盒和轮子密钥的代数表达式，就可以像编写轮子密钥的生成程序那样编写 LBlock 加密函数的 Mathematica 代码。 将加密程序用明文符号和主密钥符号代入，就可以得到LBlock 第 1 轮~第 5 轮输出的代数表达式。第 1 轮~第 3 轮输出的最低比特的代数表达式如下：

第 1 轮：

p32

第 2轮：

k52+ k54+k55+p24+p36+p38+(k52+p36)(k54+p38)+(k53+p37)(k54+p38)+p39

第三轮，第 4 轮和第 5 轮输出的代数表达式太复杂，因此，这里不再给出。

**4.3基于代数表达式的 1~5 轮 LBlock 完全性分析**

根据上述LBlock输出的代数表达式，可以直接分析安全性[19]，利用 Mathematica 内置命令Variables 给出第 1 轮~第 5 轮 LBlock 输出的每个比特所含的自变量。

主密钥变量个数如下：

第一轮：

{4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0}

第 2 轮：

{8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4, 4,4,4,4,4,4,4,4}

第3轮

{16,16,16,16,16,16,16,16,16,16,16,16,13,13,13,13,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8}

第 4 轮：

{27,27,27,27,25,25,25,25,25,25,25,25,24,24,24,24,29,29,29,29,30,30,30,30,25,25,25,25,29,29,29,29,16,16,16,16,16,16,16,16,16,16,16,16,13,13,13,13,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16}

第 5 轮：

{41,41,41,41,41,41,41,41,45,45,45,45,41,41,41,41,49,49,49,49,49,49,49,49,44,44,44,44,44,44,44,44,27,27,27,27,25,25, 25,25,25,25,25,25,24,24,24,24,29,29,29,29,30,30,30,30,25,25, 25,25,29,29,29,29}

明文变量个数：

第 1 轮：

{5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}

第 2 轮：

{9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5}

第 3 轮：

{17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9}

第 4 轮：

{29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,17,17,17,17,17,17, 17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17, 17,17,17,17,17,17}

第 5 轮：

{45,45,43,45,45,45,45,43,45,45,45,45,45,45,45,45,45,45,43,45,45,45,45,43,45,45,45,45,45,45,45,45,29,29,29,29,29,29, 29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29,29, 29,29,29,29,29,29}

由上述的变量个数可以看出，5 轮 LBlock 的输出的任何比特都远未达到完全性，这是因为 LBlock 的明文和主密钥分别是 64 比特和 80 比特。第 1 轮~第 5 轮输出的任何比特都只和部分明文与部分主密钥有关。

结 论

从1949年Shannon理论的提出，现代密码学得到了飞速的发展，得到了长足的进步。分组密码，作为现代密码学的重要分支，随着二十一世纪初欧洲两大密码计划的顺利进行，也产生了很多具有重要意义的分组密码，也对信息安全等信息技术的发展产生重要的推进作用。LBlock分组密码作为我国分组密码设计最终成功产物之一，研究它具有很重要的现实意义。LBlock分组密码的寄存器位数都是80位。在LBlock分组密码中非线性反馈移位寄存器S则使算法具有了良好的非线性以及随机性[20]。寄存器通过特定位形成相互控制，获得了更好的非线性特征。在输出密钥流时，密钥流是通过这两个寄存器的第一位进行异或得来的。

在实现LBlock分组密码的过程中，制作了一款基于LBlock分组密码算法的文件加解密传输软件。这款软件在实现算法的基础上，使用MFC类对其进行可视化窗口设计，使用基于TCP/IP协议C/S模式进行网络传输，并使用基于MD5-HASH算法计算其文件摘要值，验证其文件完整性的实用软件。

之后，对其进行了更深入的分析。LBlock分组密码算法是远未达到完全性，因为 LBlock 的明文和主密钥分别是 64 比特和 80 比特。第 1 轮~第 5 轮输出的任何比特都只和部分明文与部分主密钥有关。

LBlock分组密码算法，作为我国研究的分组密码算法，其占用资源小，易于实现，且具有良好的随机性以及安全性，所以，它在现代密码学的发展过程中必将具有重要的地位，在现实中也将具有更广泛地使用，研究它也使我们对现代密码学的发展动态有了很好的掌控。

致 谢

大学生涯随着时光飞逝，也要临近尾声，学有所成，即将要离开母校感慨颇多。四年的经历，有过失败后的泪水，有过不易时的感动，也有过成功时的笑颜。

大学四年里，感谢我的学校、老师、同学。我的系主任、各位老师，在大学这样一个过渡的时期中如同灯塔一般，为我的学习生活指明了方向，让我能够学习专业的学术知识，能够去实践后掌握专业技能，让我在最后的毕业论文中发挥所学；脱离了父母的温室，室友如同手足兄弟一般是它们让我有了一个强有力的充满正能量的团队，在论文创作过程中相互学习，相互补充；大学如同小社会一般，是我成长中步入真正社会的第一步。而北京电子科技学院更是我步入社会的基石。在学校，与同学一起，将所学理论转化为实践，组织并参与各类活动，为我能够更加独立进入社会工作奠定夯实基础。正是因为这些经历我才成长了，明白了“独立”的真正含义。临近毕业，对于我生活了四年的学校，很是不舍，对于未来我也定会奋斗。

其次，我要由衷感谢对我此次毕业论文创作中提供很大帮助的各位老师。本论文是在我的导师 老师以及学院专业老师们的悉心指导下完成的。从毕业设计开题， 老师对我们认真负责的态度，对学术严谨的精神让我非常敬佩。同时在毕业设计创作中，能够发现问题，马上指导，立刻解决，为我最终完成论文给予很多建议和帮助。从题目的选定准备开题答辩到论文的最终完成， 老师都始终给我细致指导和鼓励，对于在毕业设计中出现的问题，首先要求我主动寻求解决方法，在不能解决的基础上给我提供了大量的意见，更在毕业设计中给我提供了大量的学习资料。在此谨向 老师致以诚挚的感谢！

最后，从开题直至论文材料最后归总结束，我要向在百忙之中评阅本论文的所有老师表示感谢，正是因为你们以极高的专业眼光和悉心的指导来帮助我分析解决问题，我才能将本论文圆满完成。所以在此要对你们致以诚挚地感谢！

参考文献

[1]吴文玲，冯赍国，张文涛。分组密码的设计与分析(第2版)[M].北京：治华大学出版社，2009.

[2]Wu Wenling, Zhang Lei. LBlock：A Lightweight Block Cipher[C]// Proceedings of ACNS’11. Heidelberg, Germany：Springer, 2011: 327-344.

[3]SoleimanyH,Nyberg K. Zero-Correlation Linear Cryptanalysis of Reduced-Round LBlock[J]. IACR Cryptology ePrint Archive, 2012,2012:570.

[4]冯登国，裴定一，密码学导引。科学出版社,1999.

[5]M. Matsui. New Block Encryption Algorithm MISTY. FSE 1997, Lecture Notes in Computer Science, Vol. 1267, pp. 54-68, 1997.

[6]M. Matsui. Linear Cryptanalysis Method for DES Cipher. EUROCRYPT 1993, Lecture Notes in Computer Science, Vol. 765, pp. 386-397, 1994.

[7]陈文宇.Windows编程中的面向对象技术[D].电子科技大学计算机科学与工程学院，2002.

[8]徐璇，姜明新，黄静等.基于MFC的工程软件界面设计[J].电子设计工程.2011，19（21）：11-13.

[9]范志先，孙殿柱，李延瑞等.采用MFC开发用户交互界面的新方法[J].工程图学学报.2008，4：160-163.

[10]刘飞.HASH函数研究与设计[D].南京航空航天大学，2012.

[11] 张绍兰.几类密码HASH函数的设计与安全性分析[D].北京邮电大学，2011.

[12]Courtois N T, Pieprzyk J. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. [A]. ASIACRYPT 2002. Springer, 2002, LNCS 2501: 267-287.

[13] 晓玲.MD5加密算法的研究及应用[J].信息技术.2010，5（7）：145-151.

[14] ]束长宝，于照，张继勇.基于TCP/IP的网络通信及其应用[J].网络与通信.2006，5（12）：157-159.

[15]Chen J, Miyaji A. Differential Cryptanalysis and Boomerang Crypt-analysis of LBlock[M]//Security Engineering and Intelligence Informatics.Springer Berlin Heidelberg, 2013: 1-15.

[16]梁敏，陈万军.用VC实现基于TCP/IP的局域网内点对点的大文件传输.西北民族大学学报(自然科学版)。2005，26（57）：72-76.

[17]Biham E, Biryukov A, and Shamir A, Miss in the Middle Attacks on IDEA and Khufu. Fast Software Encryption — FSE 1999.Springer,1999,LNCS 1636: 124—138.

[18]Dinur I,Shamir A. Applying Cube Attacks to Stream Ciphers in Realistic Scenarios [J]. Cryptography and Communications, 2012,4(3/4): 217-232.

[19]Preneel B, Bosselaers A, Rijmen V. Comments by the NESSIE Project on the AES Finalists[EB/OL]. [2010-11-24].

[20]Feistel H.Cryptography and Computer Privacy[J]. ScientificAmerican, 1973, 228(5): 15-23.