

**课 程 设 计 报 告**

**题目： 密码学课程设计**

**课程名称： 密码学课程设计**

**专业班级：**

**学 号：**

**姓 名：**

**指导教师：**

**报告日期： 2023年 月 日**

**网络空间安全学院**

目 录

[一、设计过程 1](#_Toc1321052424)

[1.1 分组密码快速实现 1](#_Toc120559999)

[1.2线性密码分析 2](#_Toc1776453173)

[1.3差分密码分析 3](#_Toc169168843)

[1.4算法增强 4](#_Toc1392542335)

[1.5 RSA参宿计算 4](#_Toc874607508)

[1.6模重复平方计算 5](#_Toc1225575735)

[1.7中国剩余定理 5](#_Toc1171293658)

[1.8 HASH函数与彩虹表 6](#_Toc1834920679)

[1.9数字信封 7](#_Toc1640656454)

[二、实验心得 7](#_Toc1721163234)

[三、对课程设计内容和过程的建议 8](#_Toc925443090)

## 一、设计过程

### 1.1 分组密码快速实现

（1）设计内容

根据SPN加解密过程，快速实现教材例题3.1所给的S盒与P盒的SPN加解密函数，对于输入的32比特密钥和16比特明文，能够快速将明文进行加密输出密文，并将密文最后比特位反转后快速解密得出明文。

（2）设计过程

在教材3.1例题中，S盒与P盒均已固定，加密过程共有5轮，前3轮每一次需要进行与轮密钥异或、S盒代换、P盒置换3次操作，第4轮只进行与轮密钥异或、S盒代换两次操作，第5轮只做与轮密钥异或1次操作。

与加密过程相似，但是解密过程使用的S’盒与P’盒为加密时S盒与P盒的逆，第1轮解密需要与轮密钥异或1次操作,第2轮解密需要S’盒代换、与轮密钥异或2次操作，最后3轮解密需要P’盒置换、S’盒代换、与轮密钥异或3次操作。

题目要求进行快速实现，在1000ms内完成4x106加密和解密操作，除了使用快速输入输出外，还需要降低加密解密代码的时间消耗，加解密过程中的3中操作分别为异或、S盒代换、P盒置换，其中异或运算次数无法减少，优化的可能性较低，应此主要优化在S盒代换和P盒置换上。

在未优化的情况下，S盒代换使用查表实现，4位输入4位输出，16位输入需要进行4次查表，可将其合并为1次查表，16位输入16位输出。P盒置换由复杂的位运算实现，但考虑到其16位输入16位输出，因此可将其使用查表实现。进一步分析，P盒置换前需要进行S盒代换，因此可以将S盒代换查表与P盒置换查表进行合并，进行一次查表操作即可。在程序开始运行时提前计算出SP代换置换所需的16位输入16位输出，在加密过程中直接查表。

与加密过程相似，解密时将查表过程合并，并提前计算出所使用的表，在解密时直接查表。

1. 小结

根据上述思路实现算法，程序在650ms通过了4x106加密和解密操作，符合题目要求。但算法仅对16位长度输入实现了加解密操作，对长度更大的数据的加解密暂未实现。

### 1.2线性密码分析

1. 设计内容

使用1.1中实现的SPN加解密算法，对输入的8000对使用相同32位密钥加密的16位明密文进行线性分析，破解加密所用的32位密钥。程序需要在1000ms内破解100组数据输入，需要快速实现。

1. 设计过程

使用线性分析只能对最后一轮加密的轮密钥进行分析，应尽可能多得得到最后一轮加密的轮密钥，对于剩余位数则需要进行枚举。根据文档可进行两次线性分析，第一次分析枚举密钥的k6和k8，对8000对明密文对计算偏差x5⊕x7⊕x8⊕u6⊕u8⊕u14⊕u16并统计偏差最大值，最大值附近的k6和k8分别为4位候选子密钥，共8位子密钥。第二次分析枚举密钥的k5和k7，对8000对明密文对计算偏差x5⊕x6⊕u2⊕u4⊕u6⊕u8⊕u10⊕u12并统计偏差最大值，最大值附近的k5和k7分别为4位候选子密钥，共8位子密钥。

上述的两轮线性分析得到了最后一轮加密使用的16位候选子密钥，对于剩余的16位子密钥进行枚举，组合成完整的32位密钥，对密钥进行加密测试，找到满足所有8000对明密文的密钥。

根据上面的思路，先是第一轮分析 256x8000次循环，保留16个偏差最大的子密钥进行第二轮分析 16x256x8000次循环，经过测试，第一轮分析用时约5ms,第二轮分析用时约100ms，显然计算次数过多，时间消耗太大，因此需要优化循环的次数。由于两轮分析的最后一轮轮密钥都是每4位一组，相互独立，子密钥的取值只会有0~15，第一轮的随机变量可以分为两组x5⊕x7⊕x8⊕u6⊕u8，u14⊕u16，第二轮的随机变量分为三组u2⊕u4，x5⊕x6⊕u6⊕u8，u10⊕u12，每组之间都是独立的，所以可以只循环16x8000次，将计算结果保存起来，后面计算完整随机变量再查表。

经过上面的优化，计算上述表格的时间大约在0.7ms，剩下的第一轮分析时间大约在1.3ms，第二轮分析时间在22ms，相较于优化前明显变快，但是仍然不满足题目需求。分析发现时间消耗主要花费计算随机变量时循环的跳转上，因此需要减少循环的次数，由于上述分组计算随机变量的结果均为0或1，实际只占用1bit内存即可存储，但每次循环都要在内存中完整取一次数，因此考虑将64次计算结果进行一次储存，将64次查表优化为1次查表，最后计算随机变量只需要异或64位整数并计算异或结果中1的个数，该计算可分为4次查表实现。经过优化，第二轮分析的循环次数减少到16x256x125，时间消耗减少到了0.4ms。

1. 小结

根据上述思路实现代码，并对程序代码进行优化，程序能够正确分析出每组数据所使用的加密密钥，并且通过第十个测试点的100组数据用时为420ms，满足快速实现的要求。由于对随机变量的计算进行了拆分，并且优化了位运算的计算，代码实际可读性变差，不能直观地展现线性分析的过程。

### 1.3差分密码分析

1. 设计内容

使用1.1实现的SPN加解密算法，使用差分分析的方式，对0x0000至0xffff对应的明文加密得到的密文进行分析，破解加密时使用到的32位密钥，需要在1000ms内实现对500组密文的破解。

1. 设计过程

差分密码分析是通过分析明文对的差值对密文对差值的影响来恢复某些密钥比特的分析方法，构造若干对明文串对，使每对明文的异或结果相同，观察相应的密文异或结果。

与线性分析类似，差分分析也只能对最后一轮子密钥进行分析，剩余的密钥位需要进行枚举。根据文档资料，对最后一轮子密钥分析的过程可以分为两轮分析，第一轮分析使明文对异或结果x’=0x0b00，密文对异或，再与子密钥进行异或，统计u’=0x0606出现次数，最后统计所有子密钥中u’=0x0606出现次数最多的子密钥，将其作为k6和k8的候选子密钥。第二轮分析使明文对异或结果x’=0x0500，密文对异或，再与子密钥进行异或，统计u’=0x6060出现次数，最后统计所有子密钥中u’=0x6060出现次数最多的子密钥，将其作为k5和k7的候选子密钥。两轮分析中，得到的候选子密钥k5、k6、k7、k8都是4位长度，可组合为最后一轮加密的完整16位子密钥，剩余的16位子密钥通过枚举得到。

根据上述思路实现代码后，需要对程序进行优化，降低算法的时间消耗。经过分析可以发现，在第一轮计算中，k6和k8相互独立，第二轮计算中，k5和k7相互独立，并且两轮计算也相互独立，因此只需要从0～255枚举k，将k作为子密钥参与异或运算，子密文异或后与子密钥异或结果为0x6时，将其结果存储为1，原本需要循环65536x65536次的循环只需要循环65536x16次即可。上述计算后，还需要进行两轮计算统计u’=0x0606和u’=0x6060出现的次数得到候选子密钥k5、k6、k7、k8，只需要从0~16枚举k5、k6、k7、k8，查表上述存储的表并将结果相与，统计相与结果中1的个数即可得到u’=0x0606和u’=0x6060出现的次数，出现次数最大的子密钥即候选子密钥。不难看出，上述操作中存储结果1时，只需要使用1bit,因此可以使用线性分析中压缩存储的思路，将每64次操作的结果进行一次存储，将64次查表操作优化为1次查表操作。

此外，题目所给数据每组包含明文从0x0000~0xffff的65536组数据，实际进行计算时，并不需要如此大的数据量，只需要0x3000组数据即可完成差分分析，经过上述优化以及对输入数据的精简，程序破解一组输入的时间约为1ms，能够在1000ms内通过全部500组测试数据。

1. 小结

根据以上思路实现算法，程序能够对输入的密文对进行破解，得到完整的32位加密密钥，并且程序性能能够满足要求，在600ms通过了全部的500组测试数据。算法的缺点方面与线性分析算法类似，由于代码的优化，使得原本差分分析的特征不够明显，不能直观地表示差分分析的过程。

### 1.4算法增强

1. 设计内容

选择合适的密钥长度、分组长度、S盒代换、P盒置换和加密轮数，对1.1所实现的SPN加密算法进行改进，提高加密的安全性，使输出密文能够达到随机数的检测标准，在保证算法安全性的前提下，还需要保证算法有较高的效率，在1000ms完成对227比特数据的加密。

1. 设计过程

题目输入密钥长度为128bit，数据长度为227bit，可以将每组加密的数据长度更改为64bit，因此轮密钥长度也需要修改为为64bit，仍然保持5轮加密，前3轮每一次需要进行与轮密钥异或、S盒代换、P盒置换3次操作，第4轮只进行与轮密钥异或、S盒代换两次操作，第5轮只做与轮密钥异或1次操作。

除了加长每组加密的数据长度，还可以更换安全性更高的S代换盒，原始算法中使用的S盒为4bit输入4bit输出，将其修改为8bit输入8bit输出的S盒。P置换盒也需要进行修改，原始算法P盒是对16位数据进行置换，因分组长度变为了64，P盒修改为对64位数据进行置换。

另外，添加CBC的工作模式，将0x2ab1c3272de1f9a7作为初始密文向量，每组数据加密前，将明文与上一组数据加密得到的密文进行异或，将该组明文加密得到的密文保存为密文向量，用于与下一组加密的明文进行异或。

1. 小结

通过对SPN算法的破解和改进，了解了影响SPN加密算法安全性的一些因素，通过修改的密钥长度、分组长度、S盒代换、P盒置换和加密轮数等方式，提高了原始加密算法的安全性，使加密后的密文能够达到随机数检测标准。

### 1.5 RSA参数计算

1. 设计内容

利用TASSL中的大整数加法、剑法、乘法、模运算等基本运算，实现对大整数的求逆运算，求最大公因数运算等，并利用实现的运算函数检查RSA加密参数p、q、e的合法性。

1. 设计过程

对于求模逆运算的实现，可以使用辗转相除算法递归实现，该算法接受两个主要参数，整数e和模数m，计算商dv = e / m，余数rem = e mod m，当余数rem为0时说明e的模逆为1，停止递归并返回r1 = 1、r0 = 0。如果余数rem不为0，则进行递归求余数rem关于模数e的逆，递归结束后返回r1 = dv \* r1 + r0，r0  = r1。算法的返回值r1即为e关于模数m的逆。

对于大整数求最大公因数的算法，与常规GCD算法相同，同样使用递归算法实现gcd算法接收两个参数a与b，当b为0时结束递归，返回值r = a，当b不为0时，余数rem = a mod b，使用gcd计算b与rem的最大公因数。

关于RSA参数的合法性检查，需要进行多次判断。首先需要检查参数p、q是否未素数，当p、q不为素数时，RSA参数不合法，检测素数可以使用BN\_is\_prime\_ex函数实现；RSA的模数m = (p - 1) \* (q - 1)，需要检查模数m的有效位数，m的有效位数至少需要为1000才能满足合法性；除了模m的位数不能太小，e的有效位数同样不能太小，至少需要为16；需要保证m与e互素，当m与e不互素时，RSA加密参数不合法；p - q的位数需要满足一定的条件，不能太小。

根据上述求逆运算算法、求最大公因数算法和RSA参数检验的思路，利用TASSL中的大整数运算函数实现上述算法。

1. 小结

根据上述思路实现算法后，能够实现对大整数的求逆运算和求最大公因数运算，利用实现的求逆运算算法和求最大公因数算法，完成了对RSA参数合法性的检查，能够检查出RSA参数选择上常见的一些错误，提高RSA加密的安全性。

### 1.6模重复平方计算

1. 设计内容

模平方重复算法是一种高效计算幂运算的算法，利用TASSL中的大整数加法、剑法、乘法、模运算等基本运算，实现模平方算法expmod，对输入的四个大整数e、m、p、q，计算me(mod*pq*)。

1. 设计过程

模平方重复算法利用指数的二进制表示和模运算的性质，将指数的计算过程分解和重复利用，从而减少计算量。计算me(mod*pq*)时，首先初始化结果变量r = 1，将指数e转换为二进制表示形式，得到一个由0和1组成的位串，e = en\*2n + en-1\*2n-1 + ... + e1\*21 + e0\*20。从二进制串的最高位开始逐位处理，如果当前位为1，则将结果变量r乘以底数m，并对模数pq取余，即r = (r \* m)(mod*pq*)，如果当前位为0,则不进行操作，对底数m进行平方操作，并对模数pq取余，即m = (m \* m)(mod*pq*)，继续处理下一位，直到处理完所有的位数，返回结果变量r，r即为模平方运算me(mod*pq*)的结果。

在上述算法中，所有的运算都是大整数运算，使用TASSL中的大整数算法能够很方便得实现，主要运算位两次模乘运算，使用大整数的模乘运算速度较慢，不能满足题目对于算法速度的要求，因此需要使用更快的模乘算法，TASSL中实现了蒙哥马利模乘算法BN\_mod\_mul\_montgomery，可以加速模乘的运算，使用蒙哥马利模乘运算时，需要将大整数转换为蒙哥马利数。除了使用蒙哥马利模乘运算，还可以在运算中使用上下文加速算法。

1. 小结

通过使用上述的模平方算法和蒙哥马利模乘算法，实现expmod算法，对输入的四个大整数进行计算并输出me(mod*pq*)。由于算法使用的蒙哥马利算法由TASSL内部实现，未自己实现模乘加速算法，所以对与模乘加速算法的细节还存在部分不了解。

### 1.7中国剩余定理

1. 设计内容

利用1.5中实现的求逆算法，根据加密密钥e计算RSA解密密钥d，利用1.6中实现的模幂运算和中国剩余定理，实现RSA的解密，更具RSA的参数和计算得到的解密密钥d对输入的密文进行解密。

1. 设计过程

在RSA加解密过程中，解密密钥d是加密密钥e关于模(p-1)(1-1)的逆，因此计算解密d可以使用1.5中实现的求逆算法进行计算。RSA的解密，即计算cd(mod*pq*)，该计算可以使用1.6中实现的模重复平方算法实现，但直接使用模重复平方算法计算cd(mod*pq*)时，模数*pq*较大，计算时间较慢，由于模数*pq*为两个素数p和q的乘积，因此可以使用中国剩余定理减小模重复平方算法的模数，从而加速RSA解密过程。

使用中国剩余定理计算时，首先计算x1 = cd(mod*q*)，p关于模q的逆p’，计算m1 = x1 \* p \* p’(mod*pq*)，再计算x2 = cd(mod*p*)，q关于模p的逆q’，计算m1 = x2 \* q \* q’(mod*pq*)，最后计算m = m1 + m2(mod*pq*)，m即为解密得到的明文。

1. 小结

根据上述分析思路实现RSA解密算法，对于输入的大整数p、q、e，能够计算出RSA解密时所使用的解密密钥d，并能够使用该解密密钥d对输入的密文进行解密得到明文。

### 1.8 HASH函数与彩虹表

1. 设计内容

给定R函数和调用10000次SHA1函数和R函数生成的m条彩虹表链头和链尾，查找SHA1值对应的口令是否在彩虹表中。输出该SHA1值对应的口令，若找不到匹配的链表，则输出"None"。

1. 设计过程

题目给出m条调用了10000次SHA1函数和R函数生成的彩虹表的链头和链尾，查找指定SHA1值h对应的口令是否在这m条彩虹表链中。彩虹表的查找过程与生成彩虹表的过程类似，对h重复使用R函数和SHA1函数，若使用R函数得到的字符串与m条彩虹表链尾中的某一条相等，则选中该链，从该链的链头出发，重复使用SHA1函数和R函数，当中间字符串s使用SHA1函数后的值与h相等时，s即为所求的口令。

由于使用的R函数共有100个，且h在链中出现的位置不确定，对h第一次使用R函数时，需要枚举所有的R函数，重复使用R函数和SHA1函数的次数不超过10000次。每次使用R函数后需要将得到的字符串s’与m条链表的链尾进行比较，找到与s’相同的链尾，如果使用遍历所有的m个链尾字符串并使用strcmp进行比较，查找的时间消耗过大，无法满足性能要求，为了降低遍历比较的时间消耗，可以使用树的形式存储链尾字符串，每一层存储字符串的一个字符，最后一层存储链的序号，当查找字符串是否在m条链尾时，只需要逐个字符在树中查找，如果该字符串在树中，则可以直接找到所属链的序号，只需要进行8次查找即可完成遍历比较的操作。

当字符串与链尾匹配后，得到所属链表的序号，对链头字符串重复使用SHA1函数和R函数，比较SHA1函数后得到的SHA1值与输入SHA1值是否相等，当相等时，进行SHA1函数的字符串s就是SHA1值对应的明文口令。

1. 小结

根据上述算法分析实现彩虹表查找程序，程序能够实现在m条链中查找指定SHA1值对应的明文口令，且经过字符串查找优化，程序性能提高，能够较快地查找彩虹表。

### 1.9数字信封

1. 设计内容

已知可信根CA证书和用户B的私钥，对于输入的PKCS#7数字信封，对其发送者身份进行检验，检查发送者身份是否正确，并且对数字信封进行解密，获取数字信封包含的明文信息。

1. 设计过程

题目已知可信根CA证书和用户B的私钥，需要对输入的PKCS#7数字信封进行发送者身份验证和使用私钥进行解密。对于PKCS#7消息的处理，可以实用TASSL库实现，首先检查数字信封是否签名，如果未进行签名，无法确定发送者身份，验证不通过，其次使用用户B的私钥进行解密，检查数字信封是否能够解密。完成解密后，提取数字信封中的签名信息，用可信根CA证书对每个签名信息进行校验，若签名信息无法通过校验，表明数字信封发送者身份不正确。

1. 小结

使用TASSL库实现上述的校验和解密操作，对输入的PKCS#7进行校验和解密，程序能够完成对PKCS#7发送者身份的校验，发送者身份正确的数字信封，能够解密提取出数字信封中的明文字符串。

## 实验心得

本次实验涵盖了多个密码学主题，包括SPN分组密码，RSA公钥密码，HASH哈希算法和数字签名等。根据SPN的加解密原理实现了加解密程序，并通过线性密码分析和差分密码分析两种方式破解SPN加密秘钥，了解到了破解分组加密的常用方式，知道了与分组密码安全性相关的因素，如秘钥长度、分组长度、加密轮数等，通过修改与SPN加密安全性相关的因素，实现了对原SPN算法的增强，实现了更安全的分组加密。在RSA公钥密码中，通过使用TASSL提供的大整数计算函数，完成了大整数求逆、大整数求最大公因数、大整数求模幂运算等，并学习了如何使用中国剩余定理、辗转相除法、蒙哥马利模除对RSA加解密过程进行加速。此外还学习了使用彩虹表查找HASH值对应的明文口令，对数字签名进行校验。

本次实验除了学习了多个密码学主题内容，还通过代码实现了加密、解密、秘钥破解等过程，学会了如何在保证算法正确性的前提下，使用算法知识对程序代码进行优化，加快程序的运行速度，以此在有限的时间内完成尽可能多的加解密操作。

在RSA公钥加密和PKCS#7数字信封内容上，学习了密码学常用库openssl/tassl的使用，学会了使用基础的大整数运算实现其他的大整数运算，并通过密码学知识对运算进行加速，学习了对PKCS#7数字信封签名的校验和对数字信封内容的解密，了解了数字签名的作用和使用方式。

## 三、对课程设计内容和过程的建议

在课程内容上，使用多道习题对密码学不同的内容进行学习和测试，让学生以编程的方式学习密码学的内容，可以在学习知识的同时提高学生的动手操作能力、编程能力。但是习题的资料并不能引导学生完成题目内容，资料内涉及的算法为对实现原理进行详解，例如SPN线性分析中，对于为何要计算偏差，为何从偏差最大可以得到子密钥未作解释，导致虽然实现了程序能够满足题目要求，但是仍然对线性密码分析的过程不了解。此外，在题目所给测试集方面，往往测试集内容较少，不能涵盖所有的测试情况，在题目答案出现错误时不能反馈出错原因等，使得程序可能在特定的测试集无法通过测试等。

总的来说，本次课程设计的内容安排合适，题目存在部分难度，但是可以提高学生编程解决问题的能力，对于课程中出现的部分问题，希望在后续课程中能够改进。