

**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： 信息安全1901**

**学 号： U201911657**

**姓 名： 李文重**

**指导教师： 汤学明**

**报告日期： 2021年10月12**

**教师评语：**

**分数：**

**网络空间安全学院**

目录

[一、设计过程 1](#_Toc82811542)

[1.1 SPN实现（每一个题目不要超过一面） 1](#_Toc82811543)

[1.2线性分析 1](#_Toc82811544)

[二、实验心得 1](#_Toc82811545)

[三、对课程设计内容和过程的建议 1](#_Toc82811546)

## 一、设计过程

### 1.1 SPN实现（每一个题目不要超过一面）

（1）设计内容

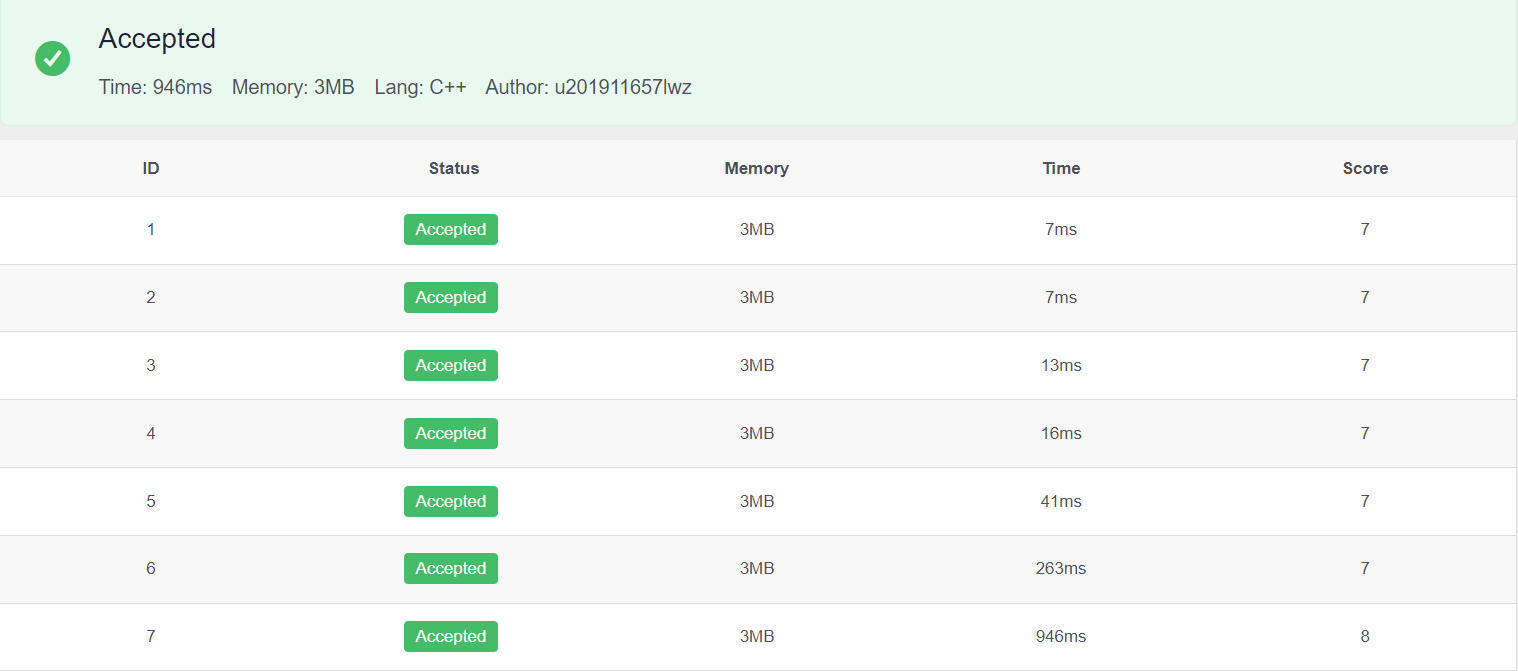
这个题目是使用课本上给出的S盒和P盒，完成一个SPN，并且优化他的效率。OJ的评测方法则是给出一个明文和一个密钥，然后进行加密，把最后一bit取反之后再进行解密。输出两个操作之后的结果。最多可能会需要进行百万级别的加、解密操作。

（2）设计过程

最开始我的想法是没有考虑太多算法效率的问题，直接实现了网络就是。结果就是运行效率惨不忍睹，根本没法通过。所以我就开始了优化，容易想到的优化首先是可以预处理小的S或者P盒为大的S盒，但是这样我原本设计的通过四个int型来存储四个半字节的方法就显得有些麻烦，而且每次都要进行合成再分解就会导致一笔不小的时间开销。因此我修改成直接使用一个int来存储中间步骤中的4个半字节，然后直接查表进行加解密。

这种方法比起之前速度已经快了不少，但是仍然不足以通过，时间差一点点。考虑到S、P盒在运用上实际上是很有规律的，比如在加密过程中，S盒总是和P盒连在一起，因此把他们的关系矩阵进行一个复合就能省下一半的时间。因此我就在原来的基础上做出了打的SP盒表，终于把时间卡到了1000ms以内。

（3）小结



我觉得这道题目我最大的不足之处就在于，为了尽可能地优化时间效能，而失去了一个很大的可扩展性，现在我的函数如果要处理更大的初始明文，或者更大的S，P盒就可能会有问题。因为更大的明文可能会挑战int型的存储上限，而更大的S、P盒则可能要求更大的空间去存储表格，因此需要进行代码的重构。而这个问题很快就会在第四题中出现。

### 1.2线性分析

（1）设计内容

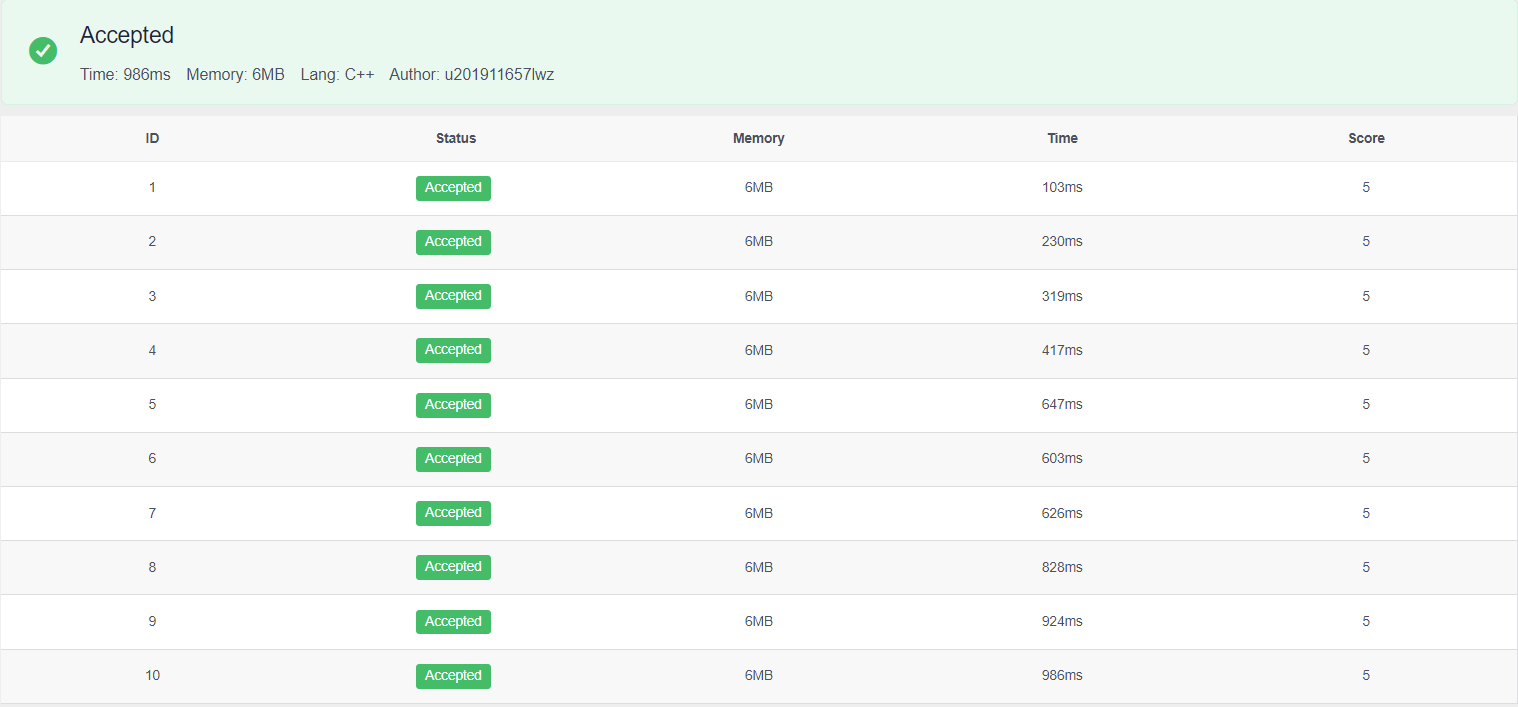
对于上文所示的SPN，要进行一次线性攻击，来进行密钥的破解。每一组需要破译的密钥都给出了8000组用此密钥进行加密的明密文对。而需要破译的组数可能会很大，达到1000组。

（2）设计过程

首先考虑到根据课本上的结论，8000组数据应该足以破解偏差达到的线性攻击，而如果偏差更小就很有可能破译不了。所以在选择线性攻击的时候偏不能太小，这就限制了很有可能并不能选到一组最理想的线性分析，即与第一组所分析的半字节正好互补，且偏差比较大。

因为自己最开始找线性分析并不是很顺利，往往都是最后有三个半字节，而且偏差小于的（虽然后来在实际测试中发现，如果三个半字节的密钥中有一个是固定且正确的的话，跑出来的结果甚至比预计的两倍还要高）。我想到利用计算机来完成这一部分的工作，基础思路也很简单，就以第一组的8000组明密文对，枚举线性攻击的掩码（或许可以这么说），然后判断实际中的偏差，最后选中其作为第二组攻击尝试着攻击第二组密钥。程序运行了几个小时之后，给出了几个还算不错的答案，于是我就选出了其中的一个作为第二组线性攻击。这个攻击依然包含了U4中的三个字节，但是已经足够了。在此基础之上，这道题还有一部分面向数据的调参，比如要枚举前多少大的偏差对应的密钥才能认为当前失败。每个密钥的验证需要进行几次加密都成功才能被判断为成功。最终我面向数据（？）的找到了一个能保证正确性前提下的最少的次数。

1. 小结



这个题目虽然险之又险的通过了，但是因为我是面向数据改的参数，所以这个程序很有可能并不具有通过随机测试的能力（也有可能有）。而且同样的这道题目我的代码也存在难以适应其他规模SPN的问题。不过这道题目应该是极大的锻炼了我的卡常数意识，在最初的代码里我有一些无用循环，还有多次使用不直接预处理等等的问题，好像只过了最多两个点。后来一步一步的思考判断优化，才成了现在代码的样子。

### 1.3差分分析

1. 设计内容

目的类似于上一题的线性分析，只不过这次采用差分分析的方法进行，差分分析所需要的数据量要大于线性分析，但是差分分析的效果显著的好于线性分析。每一组中会给出0x0000到0xffff的对应密文。最多会有500组密钥需要破解。

1. 设计过程

手动找了一组差分分析链，然后结合课本上的例子，再第二题基础上稍微改动一下。一交，直接RE了。然后也不知道RE是什么原因，就开始二分查错结果查了一遍，各个部分交上去都能RE，哪一部分都会挂，这就完全超出我认知范围了，连续好几天都没突破RE。后来在和别的同学讨论的时候，他告诉我，可能是头文件有问题，之后我又二分头文件，发现iostream.h这个头文件会出错，于是就重构了以下代码，去掉了必须要iostream的地方，又经历了一番艰难的改错，最后终于能过前几个点了。之后借用系统中的时间函数，进行了运行时间分析，发现实际上原本我预计时间会很长的地方反而时间很短，倒是读入时间很长，于是就优化了读入做了一个循环展开，但是通过不了。最后一个同学提醒我修改一下差分分析函数里面的循环枚举顺序，就可以稍作很多步重复操作，因为一个循环中，如果涉及到到的都是同一个明密文对，显然可以优化成只计算一次。

（3）小结

### 1.4 SPN增强

（1）设计内容

（2）设计过程

（3）小结

### 1.5 RSA参数计算

（1）设计内容

（2）设计过程

（3）小结

### 1.6模重复平方

（1）设计内容

（2）设计过程

（3）小结

### 1.7中国剩余定理

（1）设计内容

（2）设计过程

（3）小结

### 1.8 PKCS7

（1）设计内容

（2）设计过程

（3）小结

### 1.9 彩虹表

（1）设计内容

（2）设计过程

（3）小结

…

## 二、实验心得

## 三、对课程设计内容和过程的建议