**AES SBox的线性逼近表的分析**

1. **线性逼近表：**

设ai∈{0,1} , bi∈{0,1}, i=1,2,3,4,5,6,7,8 为了记号的紧凑,把每一个二元向量(a1,a2,a3,a4,a5,a6,a7,a8)和(b1,b2,b3,b4,b5,b6,b7,b8)看做一个十六进制数字(这些数字分别叫做输入和与输出和）。这样，这256×256个随机变量里的每一个就用一对十六进制数字来表示。设NL(a,b)表示满足如下条件的二进制16元组(x1,x2,x3,x4,x5,x6,x7,x8,y1,y2,y3,y4,y5,y6,y7,y8)的个数：

(y1,y2,y3,y4,y5,y6,y7,y8)=(x1,x2,x3,x4,x5,x6,x7,x8)及(⊕aixi)⊕(⊕ biyi)=0 。

包含所有NL值的一张表叫做线性逼近表

1. **实验关键代码：**

(1)首先让每一个(x1,x2,x3,x4,x5,x6,x7,x8)与(a1,a2,a3,a4,a5,a6,a7,a8)进行与操作,同样让每一个(y1,y2,y3,y4,y5,y6,y7,y8)与(b1,b2,b3,b4,b5,b6,b7,b8)进行与操作,这样只有对应的ai=1和bi=1的xi和yi的值被保留下来,其他的xi和yi全变成0

for (int a = 0; a <= 255; a++) {

for (int b = 0; b <= 255; b++) {

for (int x = 0; x <= 255; x++) {

//与a进行与操作，从而只让与a为1的位置相对应的位置留了下来，其他位置均为0

//又因为与0异或，相当于没操作，所以可以直接将八位全进行异或

int n = x & a;

int m = SubBytes(x) & b;

if (XOR(n, m) == 0) {

table[a][b]++;

}

}

}

}

1. 又因为与0进行异或相当于没进行操作，所以当完成(1)操作之后，进行如下操作 (x1⊕x2⊕x3⊕x4⊕x5⊕x6⊕x7⊕x8)⊕(y1⊕y2⊕y3⊕y4⊕y5⊕y6⊕y7⊕y8),实际上就相当于只对保留值的xi和yi进行了异或。
2. 为了实现将一个16进制数的每个二进制位进行异或，我采用了以下思想。首先将16进制数与0x01进行与操作，从而只保留了最低位，再将该数每次右移一位，循环以上操作，即可实现让每个位均出过在最低位并进行异或。

//两个字节的八位相互异或

int XOR(byte n ,byte m) {

int tmp1= 0;

for (int i = 1; i <= 8; i++) {

tmp1 ^= n & 1;//与1进行与操作，从而得到最低位

n=n >> 1;//右移，由此来依次得到每一位

}

//同理

int tmp2 = 0;

for (int i = 1; i <= 8; i++) {

tmp2 ^= m & 1;

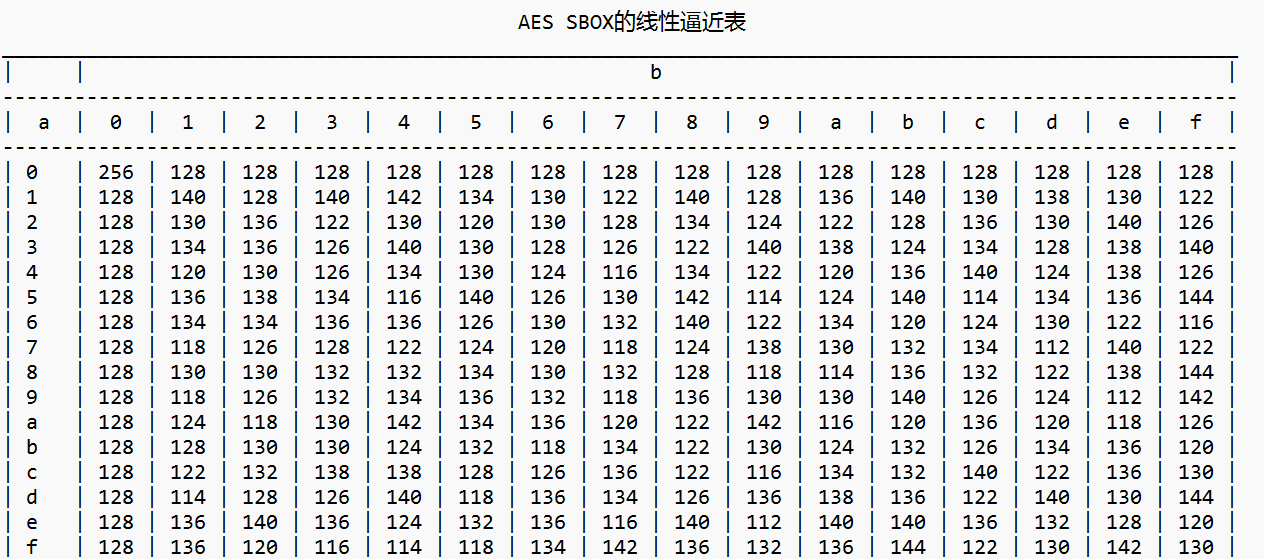
m=m >> 1;

}

return tmp1 ^ tmp2;

}

1. **实验部分结果:**



查看完整结果，请查看压缩包中txt文件

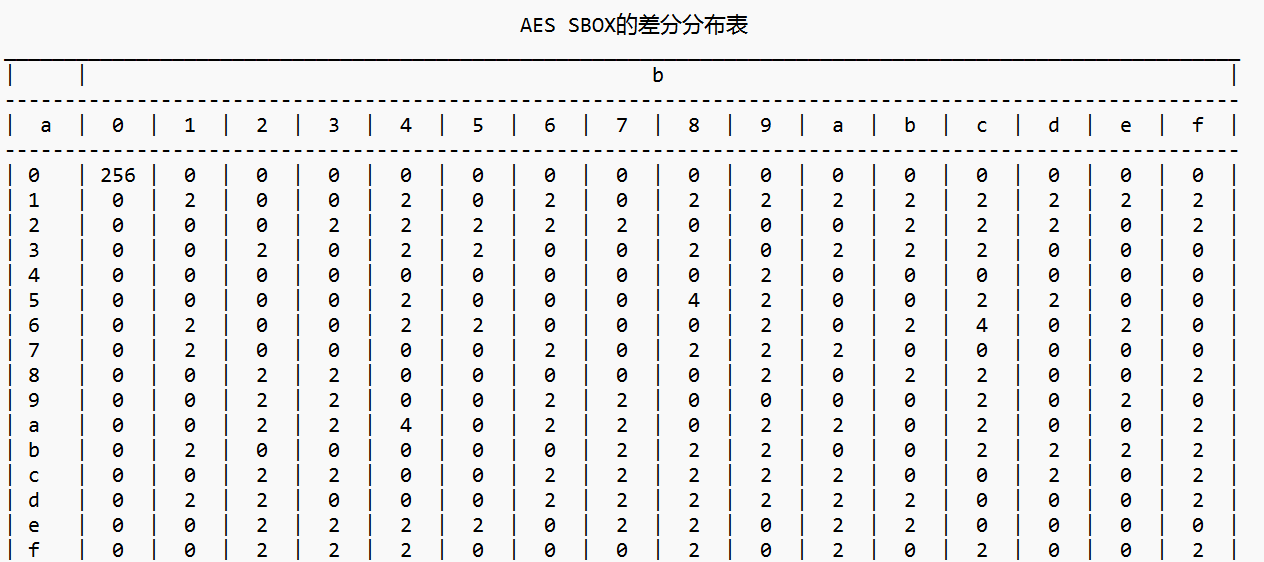
1

1. **实验结果分析**

AES的S盒的线性逼近表与SPN的有一点相似，就是在a=0的那一行和b=0的那一列，除了只有一个256，剩余均为128。此外，线性逼近表比较均匀，最大值为144（除了256），最小值为112，与128的差值的绝对值不超过16，而基数为256，所以最大的偏差也才为16/256=0.0625 。而明密文对数目T要接近 ,线性攻击才能成功，由此可见明密文对数量要非常庞大才行，所以AES对线性攻击还是比较安全的。

1. **差分分布表**

**部分结果：**

****

所有结果详见压缩包txt文件

从生成的差分分布表来看，以0、2为主，少部分出现4。和SPN相比，AES显然更不容易被差分攻击攻击成功。