ZUC密码分析实验

1901210403 胡兆杰

本次实验是计算祖冲之密码算法两个S盒的差分分布表和线性逼近表。我用了老师上传的zuc_core.c 中的S盒数据，并使用python实现。

差分分析表的实现

祖冲之密码算法的S盒的输入有8个比特，其中前四个表示行数，后四个表示列数，置换时将输入换成S盒中相应位置的值。

要构造差分分析表，我们需要两个明文x和x*，其中x $\oplus$ x* 为一个定值x'，再将x和x*分别输入S盒运算，得到输出为y和y*。计算y$\oplus$y*的值y'，统计y'中值的分布情况。在实验中，我们让x'取遍00000000到11111111的所有值，对于每一个x'，让x也从00000000取到11111111，计算相应的x*，再计算y和y*，最终得到y'，统计从00000000到11111111中的每个值在y'中的出现次数，最终将所有结果汇总，形成差分分析表。

在实现中，由于观察到书上的例子中有许多项的出现次数是0，所以为了节省存储空间，在每一轮统计中，我用一个字典只存储出现次数大于一次的值和他们的出现次数，最后在打印输出时再将不在字典中的数的出现次数都置0。

线性逼近表的实现

线性逼近表是用于线性分析的表，和差分分析表稍有不同。

线性逼近表分析的是明文X和X经过S盒的输出Y的某些位组成的随机变量。表中有a和b两种元素，a表示X的某些位组成的数，b表示Y的某些位组成的数。比如a为3则表示X_7和X_8为1，b为4则表示Y_6为1，则此时的随机变量即为$X_7 \oplus X_8 \oplus Y_6$。我们首先统计从00000000到11111111的S盒输出的值，再逐条统计满足$X_7 \oplus X_8 \oplus Y_6 = 0$的个数并记录。和差分分析表一样，我们仍然要遍历所有的值，将所有结果汇总起来，最后形成一张256*256的表。

在实现中，我们先将整数转换成8位的二进制数，找出为1的位置，在S盒置换表中将这些位置上的值进行异或，如果结果为0，则计数器加1，直到统计完所有的值。

实验结果

为更好的展示实验结果，我将生成的表输出到excel文件中，下面是文件的部分截图。

ddt_s0

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb
0x0	256	0	0	0	0	0	0	0	0	0	0	0
0x1	0	0	0	2	4	2	0	0	4	0	0	2
0x2	0	2	0	0	0	2	0	2	4	2	4	2
0x3	0	0	0	0	0	0	0	0	0	0	4	0
0x4	0	2	4	2	4	0	0	0	4	2	0	2
0x5	0	2	0	0	0	0	0	2	0	0	0	2
0x6	0	0	0	0	0	0	0	2	0	2	0	0
0x7	0	0	4	0	0	0	0	0	0	0	0	2
0x8	0	4	4	4	4	4	0	4	0	4	0	4
0x9	0	0	0	0	0	0	0	0	0	0	0	0
0xa	0	2	0	0	0	2	4	0	0	2	4	0
0xb	0	4	0	0	0	4	0	0	0	0	0	0
0xc	0	0	0	0	0	0	4	0	0	4	0	4
0xd	0	0	0	0	0	0	0	0	0	0	0	0
0xe	0	4	0	0	4	0	0	0	0	0	0	0
0xf	0	0	0	0	0	0	4	0	0	0	0	0
0x10	0	4	0	4	0	4	0	4	0	4	0	4
0x11	0	2	0	2	0	0	4	0	0	2	0	2
0x12	0	2	0	2	0	2	0	0	4	2	4	0
0x13	0	0	0	0	0	0	0	0	0	0	4	0
0x14	0	2	4	2	4	0	0	0	4	2	0	2
0x15	0	2	4	2	4	0	0	0	0	2	0	2
0x16	0	0	0	0	4	0	0	2	0	2	0	0
0x17	0	2	0	0	0	0	0	2	4	0	0	0
0x18	0	4	4	4	4	4	0	4	0	4	0	4
0x19	0	0	0	0	0	0	0	0	0	0	0	0
0x1a	0	4	4	0	0	0	0	0	4	0	0	0

ddt_s1

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb
0x0	256	0	0	0	0	0	0	0	0	0	0	0
0x1	0	2	0	2	0	0	0	0	2	2	0	0
0x2	0	2	0	2	2	2	2	0	0	2	0	0
0x3	0	2	2	2	0	2	0	0	2	2	2	2
0x4	0	0	0	0	0	2	0	0	0	2	2	0
0x5	0	0	0	2	2	2	2	2	2	2	0	0
0x6	0	2	0	2	0	2	0	2	0	2	0	0
0x7	0	0	0	0	2	0	2	0	0	0	0	2
0x8	0	2	0	0	0	0	2	2	2	0	2	0
0x9	0	2	0	0	0	0	0	0	0	0	0	2
0xa	0	0	2	2	0	2	0	2	0	2	2	2
0xb	0	2	0	2	0	2	0	2	2	0	2	2
0xc	0	0	2	0	2	2	2	0	2	2	0	2
0xd	0	2	0	0	0	0	2	2	0	0	2	0
0xe	0	0	0	2	2	2	2	2	0	2	0	0
0xf	0	0	0	2	2	2	0	0	2	0	0	0
0x10	0	2	0	0	0	2	0	2	0	0	0	2
0x11	0	0	0	2	0	2	0	0	2	0	2	2
0x12	0	0	2	0	0	0	0	0	2	0	2	0
0x13	0	0	0	0	2	0	0	2	2	2	0	2
0x14	0	2	2	0	0	2	0	0	0	2	0	2
0x15	0	2	0	0	0	2	2	0	0	0	2	0
0x16	0	0	2	0	2	0	0	0	0	0	0	2
0x17	0	0	2	0	2	2	2	0	2	0	2	0
0x18	0	0	0	2	0	0	2	0	2	0	0	0
0x19	0	2	0	0	2	2	2	0	0	0	0	2
0x1a	0	2	2	0	0	0	0	2	0	2	0	2
0x1b	0	2	0	2	0	2	0	2	2	2	0	0

lat_s0

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb
0x0	256	256	256	128	256	128	128	128	256	128	128	128
0x1	128	128	128	128	128	128	128	128	128	128	144	144
0x2	128	128	128	128	128	128	120	136	128	128	136	120
0x3	128	128	128	128	128	128	136	136	128	128	136	120
0x4	128	128	128	112	112	128	128	128	128	128	120	120
0x5	128	128	144	128	128	128	112	144	128	128	120	120
0x6	128	128	128	128	112	128	120	136	128	128	128	128
0x7	128	128	128	128	128	128	136	136	128	128	112	144
0x8	128	128	96	128	128	128	128	128	128	128	136	136
0x9	128	128	128	128	128	128	128	128	112	112	136	136
0xa	128	128	128	128	128	128	120	136	112	144	128	128
0xb	128	128	128	128	128	128	120	120	128	128	128	128
0xc	128	128	112	128	128	112	144	144	128	128	128	128
0xd	128	128	128	112	128	128	128	128	128	128	128	128
0xe	128	128	128	128	128	112	136	120	128	128	120	136
0xf	128	128	128	128	128	128	136	136	128	128	120	136
0x10	128	128	128	128	120	120	128	128	136	120	120	128
0x11	128	128	128	128	128	128	120	120	128	128	136	128
0x12	128	160	128	96	136	120	136	120	128	128	128	136
0x13	128	128	128	128	128	128	128	112	104	104	128	136
0x14	128	160	128	160	120	120	128	128	128	128	128	136
0x15	128	128	128	128	128	128	120	136	136	136	144	120
0x16	128	128	128	128	120	104	136	136	120	136	120	128
0x17	128	128	128	128	128	128	144	128	128	128	120	144

lat_s1

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd
0x0	256	256	256	128	256	128	128	128	256	128	128	128	128	128
0x1	128	134	126	140	136	130	126	136	138	140	140	134	134	116
0x2	128	118	130	116	134	132	124	134	132	138	126	136	126	116
0x3	128	136	120	140	130	118	134	126	122	142	142	118	140	140
0x4	128	140	122	130	114	134	124	140	132	120	134	118	138	126
0x5	128	122	140	130	130	136	126	136	114	136	122	140	124	126
0x6	128	130	132	134	124	126	124	126	136	122	132	126	124	126
0x7	128	124	126	130	136	136	114	130	114	130	124	140	142	122
0x8	128	130	136	126	114	116	130	128	132	114	140	118	134	116
0x9	128	124	142	126	126	126	124	128	126	114	120	136	120	120
0xa	128	116	126	122	136	124	130	118	132	140	130	130	136	120
0xb	128	130	132	118	112	118	120	142	114	116	130	116	130	120
0xc	128	130	138	116	136	130	130	124	128	118	130	136	116	122
0xd	128	132	132	136	124	140	112	128	118	130	126	130	122	130
0xe	128	124	120	136	122	142	126	126	128	128	136	140	130	138
0xf	128	130	130	120	130	128	128	114	122	132	120	134	120	134
0x10	128	116	142	138	132	120	142	138	134	126	116	116	114	114
0x11	128	114	128	122	128	134	116	138	124	118	128	130	128	126
0x12	128	126	120	130	134	116	118	128	130	140	122	120	116	142
0x13	128	112	130	134	126	114	128	128	132	132	122	134	114	126
0x14	128	128	120	116	142	126	130	142	122	134	122	130	140	120
0x15	128	134	126	120	138	132	116	138	116	114	126	128	126	120
0x16	128	122	130	132	116	126	134	120	126	132	128	118	138	120
0x17	128	116	144	140	116	140	132	140	132	128	136	116	116	132
0x18	128	138	126	124	142	120	128	134	118	136	116	114	140	118
0x19	128	116	120	136	134	142	138	126	132	116	136	124	118	138
0x1a	128	120	132	124	128	128	116	140	142	142	138	138	130	122