

密码学第三次作业

张津婵-1901210582

1. DDT 和 LAT 表的计算

实现思路：

查阅公开信息，得到 ZUC 算法的 S0, S1 两个 S 盒信息。每个 S 盒中有 256 个 16 进制数。为方便计算，定义函数 `int2bin(x)`，用于将 int 类型的数字转换为对应的 8 位二进制串，并初始化 256*256 大小的二维数组 DDT 和 LAT 用于存储差分分析表和线性逼近表。定义 `get_DDT` 和 `get_LAT` 得到 DDT 和 LAT。定义 `saveTable(table, filename)` 和 `printTable(table)`，存储和打印二维数组 DDT 和 LAT。

差分分析表的计算-----`get_DDT (s)`

令 $x = x' \oplus x''$, $\text{diff} = y' \oplus y''$

其中 x' 和 x'' 表示输入明文， $y' \oplus y''$ 表示对应的输出

对于某固定的 x ，可通过遍历找到满足 $x = x' \oplus x''$ 的 x' 、 x'' ，通过 S 盒置换得到对应的 y' 、 y'' ，再异或计算即可得到 $\text{diff} = y' \oplus y''$ 。通过 for 循环，遍历 x 从 0 到 255，遍历 diff 从 0 到 255，在 `DDT[x][diff]` 进行计数，遍历结束即可得到最终结果。

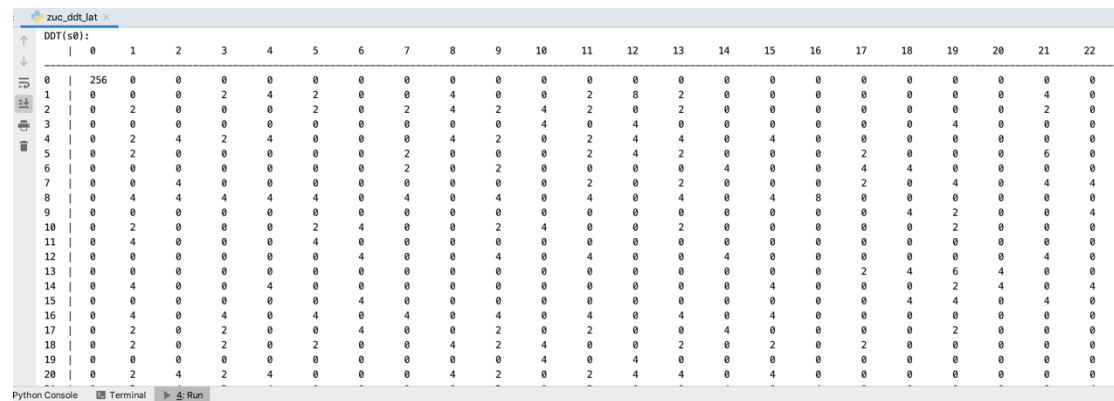
线性逼近表的计算-----`get_LAT(s)`

每个 x 、 y 都可表示为 8 位的二进制数。定义函数 `fun(i, x, j, y)` 进行异或计算（如， $i=5$ (00000101)， $x = (x_1x_2x_3x_4x_5x_6x_7x_8)$ ，将 i 和 x 进行按位异或）

要得到 LAT 表，需要统计所有可能的 $x_i \oplus y_j = 0$ 。遍历时， y 由 x 进行 S 盒置换得到，并使用 `count` 对 `fun` 的异或结果进行统计，最后令 `LAT[i][j] = 128 - count`。遍历结束即可得到最终结果。

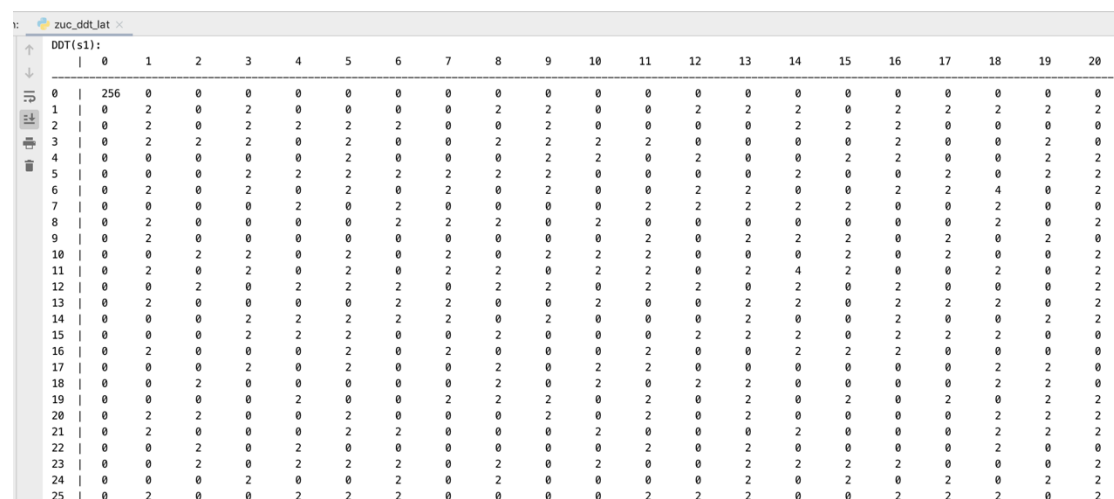
运行结果：

DDT (s0) python 输出截图：（完整结果见 DDT1.txt）



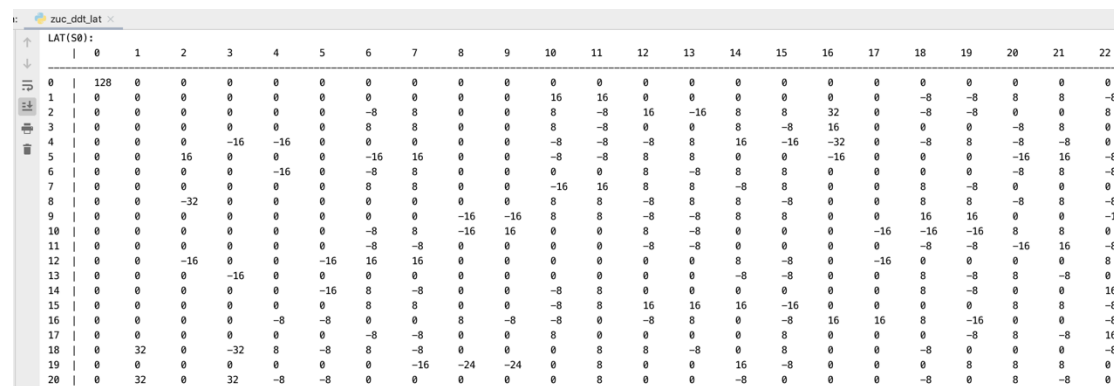
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	4	2	0	0	4	0	0	2	8	2	0	0	0	0	0	0	0	4	0
2	0	2	0	0	0	2	0	2	4	2	4	2	0	2	0	0	0	0	0	0	0	2	0
3	0	0	0	0	0	0	0	0	0	4	0	4	0	0	0	0	0	0	0	4	0	0	0
4	0	2	4	2	4	0	0	4	2	0	2	4	4	0	4	0	0	0	0	0	0	0	0
5	0	2	0	0	0	0	0	2	0	0	0	2	4	2	0	0	2	0	0	0	0	6	0
6	0	0	0	0	0	0	0	2	0	2	0	0	0	4	0	0	4	4	0	0	0	0	0
7	0	0	4	0	0	0	0	0	0	0	2	0	2	0	0	0	2	0	4	0	4	4	4
8	0	4	4	4	4	4	0	4	0	4	0	4	0	4	0	4	8	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	2	0	0	0	4
10	0	2	0	0	0	2	4	0	0	2	4	0	0	2	0	0	0	0	2	0	0	0	0
11	0	4	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	4	0	0	4	0	4	0	0	4	0	0	0	0	0	4	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	6	4	0	0
14	0	4	0	0	4	0	0	0	0	0	0	0	0	0	4	0	0	0	2	4	0	4	4
15	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	4	4	0	4	0	0
16	0	4	0	4	0	4	0	4	0	4	0	4	0	4	0	4	0	0	0	0	0	0	0
17	0	2	0	2	0	2	0	4	0	2	0	2	0	4	0	0	0	0	0	2	0	0	0
18	0	2	0	2	0	2	0	0	4	2	4	0	0	2	0	2	0	2	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	4	0	4	0	0	0	0	0	0	0	0	0	0	0
20	0	2	4	2	4	0	0	0	4	2	0	2	4	4	0	4	0	0	0	0	0	0	0

DDT (s1) python 输出截图：（完整结果见 DDT2.txt）



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	0	2	0	0	0	0	2	2	0	0	2	2	2	2	2	2	2	2	2
2	0	2	0	2	2	2	2	2	0	2	0	0	0	0	2	2	2	0	0	0	0
3	0	2	2	2	0	2	0	0	2	2	2	2	2	0	0	0	2	0	0	2	0
4	0	0	0	0	0	2	0	0	0	2	2	0	2	0	2	0	2	0	0	2	2
5	0	0	0	2	2	2	2	2	2	2	0	0	0	0	2	0	0	2	0	2	2
6	0	2	0	2	0	2	0	2	0	2	0	0	2	2	0	2	2	4	0	0	2
7	0	0	0	0	2	0	2	0	0	0	2	2	2	2	2	2	0	2	0	2	0
8	0	2	0	0	0	0	2	2	2	2	0	2	0	0	0	0	0	2	0	2	2
9	0	2	0	0	0	0	0	0	0	0	2	0	2	2	2	2	0	2	0	2	0
10	0	0	2	2	0	2	0	2	0	2	2	2	0	0	0	2	0	2	0	0	2
11	0	2	0	2	0	2	0	2	2	2	2	2	2	2	2	2	0	2	0	2	2
12	0	0	2	0	2	2	2	2	2	2	0	2	2	2	2	0	2	0	0	0	2
13	0	2	0	0	0	0	2	2	0	0	2	0	0	2	2	0	2	2	2	2	2
14	0	0	0	2	2	2	2	2	0	2	0	0	0	2	0	2	0	2	0	2	2
15	0	0	0	2	2	2	0	2	0	0	0	2	2	2	2	2	2	2	2	0	0
16	0	2	0	0	0	2	0	2	0	0	0	2	0	0	2	2	2	0	0	0	0
17	0	0	0	2	0	2	0	0	2	0	2	2	0	0	0	0	0	0	2	2	0
18	0	0	2	0	0	0	0	0	2	0	2	0	2	2	0	0	0	0	2	2	0
19	0	0	0	0	2	0	0	2	2	2	0	2	0	2	0	2	0	2	0	2	2
20	0	2	2	0	0	2	0	0	0	2	0	2	0	2	0	0	0	0	2	2	2
21	0	2	0	0	0	2	2	0	0	2	0	0	0	2	0	0	0	0	2	2	2
22	0	0	2	0	2	0	0	0	0	0	2	0	2	0	0	0	0	0	2	0	0
23	0	0	2	0	2	2	2	0	2	0	2	0	0	2	2	2	2	0	0	0	2
24	0	0	0	2	0	0	2	0	2	0	0	0	0	2	0	2	0	2	0	2	2
25	0	2	0	0	2	2	2	0	0	0	0	2	2	2	0	0	2	2	2	0	2

LAT (s0) python 输出截图：（完整结果见 LAT1.txt）



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	128	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	16	16	0	0	0	0	0	0	0	0	0	8	8	-8
2	0	0	0	0	0	-8	8	0	0	8	-8	16	-16	8	8	32	0	-8	-8	0	0	8	8
3	0	0	0	0	0	0	8	0	0	8	-8	0	0	8	-8	16	0	0	0	-8	8	0	0
4	0	0	0	-16	-16	0	0	0	0	-8	-8	-8	8	16	-16	-32	0	-8	8	-8	-8	0	0
5	0	0	16	0	0	0	-16	16	0	0	-8	-8	8	8	0	0	-16	0	0	-16	16	-8	-8
6	0	0	0	0	-16	0	-8	8	0	0	0	8	8	-8	8	0	0	0	0	-8	8	-8	0
7	0	0	0	0	0	0	8	8	0	0	-16	16	8	8	-8	8	0	0	8	-8	0	0	0
8	0	0	-32	0	0	0	0	0	0	8	8	-8	8	8	-8	0	0	8	8	-8	8	-8	-8
9	0	0	0	0	0	0	0	-16	-16	8	8	-8	-8	8	8	0	0	16	16	0	0	-16	0
10	0	0	0	0	0	-8	8	-16	16	0	0	8	-8	-8	0	0	-16	-16	8	8	0	0	0
11	0	0	0	0	0	0	-8	0	0	0	0	-8	-8	0	0	0	0	-8	-8	-16	16	-8	8
12	0	0	-16	0	0	-16	16	0	0	0	0	0	0	8	-8	0	-16	0	0	0	0	8	8
13	0	0	0	-16	0	0	0	0	0	0	0	0	0	0	-8	0	0	0	0	0	-8	0	0
14	0	0	0	0	0	-16	8	-8	0	0	-8	8	0	0	0	0	0	0	8	-8	0	0	16
15	0	0	0	0	0	0	8	8	0	0	-8	8	16	16	-16	0	0	0	0	8	8	-8	0
16	0	0	0	0	-8	-8	0	8	-8	-8	0	-8	8	0	-8	16	16	8	-16	0	0	-8	0
17	0	0	0	0	0	0	-8	0	0	8	0	0	0	0	8	0	0	0	-8	8	-8	16	0
18	0	32	0	-32	8	-8	8	-8	0	0	8	8	-8	0	8	0	0	-8	0	0	0	0	-8
19	0	0	0	0	0	0	0	-16	-24	-24	0	8	0	0	16	-8	0	0	0	8	8	8	0
20	0	32	0	32	-8	-8	0	0	0	0	8	0	0	-8	0	0	0	-8	0	8	-8	0	0

LAT (s1) python 输出截图：（完整结果见 LAT2.txt）

zuc_ddt_lat																							
LAT(S1):																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	128	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	6	-2	12	8	2	-2	8	10	12	12	6	6	-12	0	-2	6	-8	4	-2	2	-16	0
2	0	-10	2	-12	6	4	-4	6	4	10	-2	8	-2	-12	12	6	12	-2	6	12	-6	4	-8
3	0	8	-8	12	2	-10	6	-2	-6	14	14	-10	12	12	12	-8	10	-14	10	-10	8	-12	12
4	0	12	-6	2	-14	6	-4	12	4	-8	6	-10	10	-2	12	12	-12	16	-10	-2	-14	-2	-12
5	0	-6	12	2	2	8	-2	8	-14	8	-6	12	-4	-2	-12	-6	10	-8	-2	8	-4	-10	-8
6	0	2	4	6	-4	-2	-4	-2	8	-6	4	-2	-4	-2	-12	-2	12	10	8	-2	4	-6	-4
7	0	-4	-2	2	8	8	-14	2	-14	2	-4	12	14	-6	-12	-8	-14	6	-8	-4	-6	2	-12
8	0	2	8	-2	-14	-12	2	0	4	-14	12	-10	6	-12	6	-8	0	2	0	-2	-6	4	-6
9	0	-4	14	-2	-2	-2	-4	0	-2	-14	-8	8	-8	2	-2	6	-10	12	-8	0	4	6	6
10	0	-12	-2	-6	8	-4	2	-10	4	12	2	2	8	-8	2	-14	4	4	10	2	12	-12	6
11	0	2	4	-10	-16	-10	-8	14	-14	-12	2	-12	2	-8	6	12	2	12	-2	8	-10	-12	6
12	0	2	10	-12	8	2	2	-4	0	-10	2	8	-12	-6	-10	-12	4	-2	-2	8	8	-6	-6
13	0	4	4	8	-4	12	-16	0	-10	2	-2	2	-6	2	2	2	2	-6	-10	6	-2	-6	2
14	0	-4	-8	8	-6	14	-2	-2	0	0	8	12	2	10	6	10	-12	4	4	8	-6	2	-2
15	0	2	2	-8	2	0	0	-14	-6	4	-8	6	-8	6	2	4	2	-4	12	10	4	-14	10
16	0	-12	14	10	4	-8	14	10	6	-2	-12	-12	-14	-14	12	-12	-4	8	-10	-6	4	-8	10
17	0	-14	0	-6	0	6	-12	10	-4	-10	0	2	0	-2	-8	-10	-6	0	6	4	10	-12	2
18	0	-2	-8	2	6	-12	-10	0	2	12	-6	-8	-12	14	-12	2	0	-6	-4	-6	10	12	-2
19	0	-16	2	6	-2	-14	0	0	4	4	-6	6	-14	-2	0	-8	-14	-10	12	-8	-12	-8	-14

2. 为什么最后增加 Key Mixing 操作？

由于 S 盒是可逆的，如果不在最后一轮进行异或轮密钥，那么攻击者拿到密文即可通过 S 盒进行解密，得到 S 盒的输入。在最后增加 Key Mixing 操作提高了密码算法的安全性。