Дифференциальный криптоанализ

Построение дифференциального пути для XSPL-шифра

Построение дифференциального пути для заданных исходных значений

$$\alpha = \begin{pmatrix} 3 & 3 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \text{Матрица L=} \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

$\prod$= (15,9,1,7,13,12,2,8,6,5,14,3,0,11,4,10)

XSPL-шифр

Этот шифр состоит из следующих преобразований:

- сложение по модулю 2;
- преобразование замены или подстановки. Обозначается S-преобразование;
- преобразование перестановки. Обозначается P-преобразование;
- линейное преобразование. Обозначается L-преобразование.

Рассмотрим два раунда шифрования:

1 Раунд:

Шаг 1: α xor key

$$\Delta_{11} = \begin{pmatrix} 3 & 3 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Шаг 2: Sbox

Воспользуемся таблицей DDT (Difference Distribution Table).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	16,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
1	0,	2,	0,	2,	2,	0,	4,	0,	0,	0,	2,	2,	0,	2,	0,	0,
2	0,	2,	0,	0,	2,	0,	2,	0,	2,	0,	0,	0,	0,	0,	6,	2,
3	0,	0,	0,	0,	0,	6,	0,	0,	4,	0,	2,	2,	0,	0,	2,	0,
4	2,	0,	2,	2,	0,	2,	2,	0,	0,	2,	0,	0,	0,	0,	2,	2,
5	0,	0,	0,	2,	4,	4,	0,	0,	0,	2,	0,	0,	0,	4,	0,	0,
6	0,	2,	0,	0,	0,	0,	0,	0,	4,	0,	0,	2,	2,	2,	2,	2,
7	0,	0,	0,	2,	0,	2,	0,	2,	0,	0,	2,	4,	2,	2,	0,	0,
8	0,	0,	2,	0,	2,	0,	0,	2,	0,	2,	0,	0,	4,	2,	0,	2,
9	0,	0,	2,	0,	0,	0,	4,	0,	2,	2,	2,	0,	2,	0,	0,	2,
10	0,	2,	4,	4,	0,	0,	2,	2,	0,	0,	2,	0,	0,	0,	0,	0,
11	0,	2,	2,	0,	2,	0,	0,	4,	2,	2,	0,	0,	2,	0,	0,	0,
12	0,	0,	2,	0,	0,	0,	0,	0,	0,	2,	0,	4,	2,	2,	0,	4,
13	0,	2,	0,	0,	2,	0,	2,	0,	2,	4,	2,	2,	0,	0,	0,	0,
14	0,	4,	0,	4,	2,	0,	0,	0,	0,	0,	0,	0,	2,	2,	0,	2,
15	0,	0,	2,	0,	0,	2,	0,	6,	0,	0,	2,	0,	0,	0,	4,	0,

Рис.1-- DDT (Difference Distribution Table).

Программный код для реализации DDT таблицы для заданной подстановки приложен ниже.

Оба активных полубайта равны 3, значит наиболее подходящим значением в нашем случае являются β=5.

Выберем β=5.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	16,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
1	0,	2,	0,	2,	2,	0,	4,	0,	0,	0,	2,	2,	0,	2,	0,	0,
2	0,	2,	0,	0,	2,	0,	2,	0,	2,	0,	0,	0,	0,	0,	6,	2,
3	0,	0,	0,	0,	0,	6,	0,	0,	4,	0,	2,	2,	0,	0,	2,	0,
4	2,	0,	2,	2,	0,	2,	2,	0,	0,	2,	0,	0,	0,	0,	2,	2,
5	0,	0,	0,	2,	4,	4,	0,	0,	0,	2,	0,	0,	0,	4,	0,	0,
6	0,	2,	0,	0,	0,	0,	0,	0,	4,	0,	0,	2,	2,	2,	2,	2,
7	0,	0,	0,	2,	0,	2,	0,	2,	0,	0,	2,	4,	2,	2,	0,	0,
8	0,	0,	2,	0,	2,	0,	0,	2,	0,	2,	0,	0,	4,	2,	0,	2,
9	0,	0,	2,	0,	0,	0,	4,	0,	2,	2,	2,	0,	2,	0,	0,	2,
10	0,	2,	4,	4,	0,	0,	2,	2,	0,	0,	2,	0,	0,	0,	0,	0,
11	0,	2,	2,	0,	2,	0,	0,	4,	2,	2,	0,	0,	2,	0,	0,	0,
12	0,	0,	2,	0,	0,	0,	0,	0,	0,	2,	0,	4,	2,	2,	0,	4,
13	0,	2,	0,	0,	2,	0,	2,	0,	2,	4,	2,	2,	0,	0,	0,	0,
14	0,	4,	0,	4,	2,	0,	0,	0,	0,	0,	0,	0,	2,	2,	0,	2,
15	0,	0,	2,	0,	0,	2,	0,	6,	0,	0,	2,	0,	0,	0,	4,	0,

Рис.2 –Выбор значения β

$$P_1 = \left(\frac{6}{16}\right) \times \left(\frac{6}{16}\right) = \frac{9}{64}$$

$$\Delta_{12}=\prod(\alpha)=\begin{pmatrix}5 & 5 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}$$

Шаг 3: Перестановка (транспонирование матрицы)

$$\Delta_{13}=\begin{pmatrix}5 & 0 & 0\\ 5 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}$$

Шаг 4: Линейное преобразование (умножение на матрицу L)

$$\Delta_{14}=\begin{pmatrix}5 & 0 & 0\\ 5 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}\times\begin{pmatrix}0 & 1 & 1\\ 1 & 0 & 1\\ 1 & 1 & 0\end{pmatrix}=\begin{pmatrix}0 & 5 & 5\\ 0 & 5 & 5\\ 0 & 0 & 0\end{pmatrix}$$

2 Раунд:

Шаг 1: $\alpha(\Delta_{14})$ xor key

$$\Delta_{21}=\begin{pmatrix}0 & 5 & 5\\ 0 & 5 & 5\\ 0 & 0 & 0\end{pmatrix}$$

Шаг 2: Sbox

Воспользуемся DDT (Difference Distribution Table)

Все четыре активных полубайта равны 5, значит наиболее подходящими значениями в нашем случае являются β=4,5,13. Если бы мы рассматривали третий раунд шифрования, то на данном этапе мы могли бы выбрать более выгодные значения β, чтобы потом у него при замене в S-box’е было одно значение, превосходящее по вероятности над остальными, поэтому заметим, что для α=4 вероятности у значений β очень хорошо распределены между собой, для α=5 –три значения равновероятных, и наконец для α=13- есть значение β=9, которое превосходит остальных.

Выберем β=13.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	16,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
1	0,	2,	0,	2,	2,	0,	4,	0,	0,	0,	2,	2,	0,	2,	0,	0,
2	0,	2,	0,	0,	2,	0,	2,	0,	2,	0,	0,	0,	0,	0,	6,	2,
3	0,	0,	0,	0,	0,	6,	0,	0,	4,	0,	2,	2,	0,	0,	2,	0,
4	2,	0,	2,	2,	0,	2,	2,	0,	0,	2,	0,	0,	0,	0,	2,	2,
5	0,	0,	0,	2,	4,	4,	0,	0,	0,	2,	0,	0,	0,	4,	0,	0,
6	0,	2,	0,	0,	0,	0,	0,	0,	4,	0,	0,	2,	2,	2,	2,	2,
7	0,	0,	0,	2,	0,	2,	0,	2,	0,	0,	2,	4,	2,	2,	0,	0,
8	0,	0,	2,	0,	2,	0,	0,	2,	0,	2,	0,	0,	4,	2,	0,	2,
9	0,	0,	2,	0,	0,	0,	4,	0,	2,	2,	2,	0,	2,	0,	0,	2,
10	0,	2,	4,	4,	0,	0,	2,	2,	0,	0,	2,	0,	0,	0,	0,	0,
11	0,	2,	2,	0,	2,	0,	0,	4,	2,	2,	0,	0,	2,	0,	0,	0,
12	0,	0,	2,	0,	0,	0,	0,	0,	0,	2,	0,	4,	2,	2,	0,	4,
13	0,	2,	0,	0,	2,	0,	2,	0,	2,	4,	2,	2,	0,	0,	0,	0,
14	0,	4,	0,	4,	2,	0,	0,	0,	0,	0,	0,	0,	2,	2,	0,	2,
15	0,	0,	2,	0,	0,	2,	0,	6,	0,	0,	2,	0,	0,	0,	4,	0,

Рис.3 –Выбор значения β

$$P_2 = \left(\frac{|4|}{16}\right) \times \left(\frac{|4|}{16}\right) \times \left(\frac{|4|}{16}\right) \times \left(\frac{|4|}{16}\right) = \left(\frac{1}{4}\right)^4$$

$$\Delta_{22} = \prod(\alpha) = \begin{pmatrix} 0 & 13 & 13 \\ 0 & 13 & 13 \\ 0 & 0 & 0 \end{pmatrix}$$

Шаг 3: Перестановка (транспонирование матрицы)

$$\Delta_{23} = \begin{pmatrix} 0 & 0 & 0 \\ 13 & 13 & 0 \\ 13 & 13 & 0 \end{pmatrix}$$

Шаг 4: Линейное преобразование (умножение на матрицу L)

$$\Delta_{24} = \begin{pmatrix} 0 & 0 & 0 \\ 13 & 13 & 0 \\ 13 & 13 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 13 & 13 & 26 \\ 13 & 13 & 26 \end{pmatrix}$$

Итоговая вероятность:

$$P = P_1 \times P_2 = \frac{9}{16384}$$

То есть нам необходимо 16384 пар (открытый текст – зашифрованный текст), чтобы в 9 случаях дифференциальный путь был равен нашему.

Приложение

Программный код, вычисляющий DDT по заданной подстановке
∏= (15,9,1,7,13,12,2,8,6,5,14,3,0,11,4,10)

```
from array import *
S=array('i',[15,9,1,7,13,12,2,8,6,5,14,3,0,11,14,10])
Array_X = [0] * 16
for i in range(0,16):
    Array_X[i] = [0] * 16

print("
","%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%4d%5d%4d%4d%4d%4d%4d%4d"%(0,1,2,3,4,5,6,7,8,
9,10,11,12,13,14,15))

for a in range(0,16):
    print('%3d' % a, end="")
    value=0
    for x in range(0,16):

        P=S[x]^S[x^a]
        Array_X[a][P]+=1


    for j in range(len(Array_X[i])):
        print ("%3d" % Array_X[a][j], end=',')
    print()
```