Линейный криптоанализ

Построение линейного пути для XSPL-шифра

Построение линейного пути для заданных исходных значений

$\alpha = \begin{pmatrix} 3 & 3 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ 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)

1. Описание структуры шифра

XSPL-шифр

Этот шифр состоит из следующих преобразований:

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

1.1 Подстановка

В нашем шифре мы разбиваем 36-битный блок данных на девять 4-битных подблока. Каждый подблок формирует вход в S-бокс 9×9 (подстановка с 4 входными и 4 выходными битами), который может быть легко реализован с помощью табличного поиска шестнадцати 4-битных значений, индексированных целым числом, представленным 4 входными битами. Наиболее фундаментальным свойством S-box является то, что он является нелинейным отображением, т. е. выходные биты не могут быть представлены в виде линейной операции над входными битами.

Для нашего шифра мы будем использовать одно и то же нелинейное отображение для всех S-блоков.

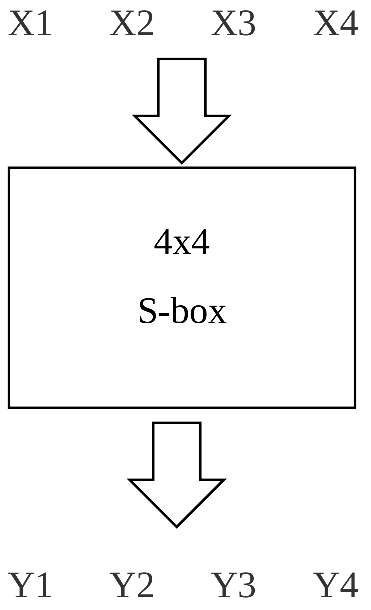
Атаки линейного и дифференциального криптоанализа одинаково применимы к тому, существует ли одно отображение или все S-боксы являются различными отображениями. Отображение, выбранное для нашего шифра, приведено в таблице 1.

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

Таблица 1. S-box (in hexademical)

Рассмотрим S-образное представление с входом X (подблок i) = [X1 X2 X3 X4] и соответствующим выходом Y (подблок i) = [Y1 Y2 Y3 Y4]. Все линейные аппроксимации могут быть исследованы для определения их полезности путем вычисления вероятностного смещения для каждого. Следовательно, мы рассматриваем все выражения вида уравнения (1), где X-вход и Y - выход S-бокса.

$X_{n1}+X_{n1}+\ldots + X_{nm}+Y_{p1}+Y_{p2} +\ldots+Y_{pt}=0$, где «+»- сложение по модулю два



LAT (Linear Approximation Table)

Полное перечисление всех линейных аппроксимаций S-бокса в нашем шифре приведено в таблице линейных аппроксимаций таблицы 4. Каждый элемент таблицы представляет собой число совпадений между линейным уравнением, представленным в шестнадцатеричном виде как "Входная сумма" и сумма выходных битов, представленных в шестнадцатеричном виде как "Выходная сумма" минус 8. Следовательно, деление значения элемента на 16 дает вероятностное смещение для конкретной линейной комбинации входных и выходных битов. Шестнадцатеричное значение, представляющее собой сумму, при просмотре в двоичном виде значение указывает на переменные, участвующие в сумме. Для линейной комбинация входных переменных, представленных как: $a_1 \cdot X_1 \oplus a_2 \cdot X_2 \oplus a_3 \cdot X_3 \oplus a_4 \cdot X_4$, где $a_n \in \{0,1\}$ и "·" логическое умножение (конъюнкция)

Аналогичным образом, для линейной комбинации выходной последовательности бит - $b_1 \cdot У_1 \oplus b_2 \cdot У_2 \oplus b_3 \cdot У_3 \oplus b_4 \cdot У_4$? где BI $\in \{0,1\}$, в шестнадцатеричной системе счисления представляет двоичный вектор b1b2b3b4.

Следовательно, смещение линейного уравнения $X_1 \oplus X_3 \oplus X_4 = Y_2 \oplus Y_4$ (шестнадцатеричный вход 11 и шестнадцатеричный выход 4) равен $\frac{-4}{16} = \frac{-2}{8}$, а вероятность того, что линейное уравнение верно $1/2 - 2/8 = 2/8$.

Можно отметить некоторые основные свойства таблицы линейной аппроксимации. Например, вероятность того, что любая сумма непустого подмножества выходных битов равна сумме без входных битов, равна ровно 1/2, поскольку любая линейная комбинация выходных битов должна

иметь равное число нулей и единиц для биективного S-бокса. Кроме того, линейная комбинация без выходных битов всегда будет равна линейной комбинации без входных битов, приводящие к смещению +1/2 и табличному значению +8 в верхнем левом углу. Следовательно, верхняя строка таблицы-это все нули, за исключением самого левого значения. Аналогично, первая колонка — это все нули, за исключением самого верхнего значения. Можно также отметить, что сумма любой строки или любого столбца должна быть либо +8, либо -8.

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

Таблица 2. LAT

1.2 Перестановка

Раунд перестановки- это просто транспозиция подблоков между собой или перестановка позиций подблоков. Перестановка приведена в таблице 2 (где номера обозначают позиции подблоков, т.е. 1 самый левый подблок (4 бита) и 16- крайний правый). В данном случае, если входной текст представить, как матрицу, состоящую из 9 подблоков 4-битных значений, то данный этап можно объяснить иначе-это транспонирование исходной матрицы.

input	1	2	3	4	5	6	7	8	9
output	1	4	7	2	5	8	3	6	9

Таблица 3. Перестановка

1.3 Линейное преобразование L

В рассматриваемом нами шифре под данным этапом подразумевается перемножение входной матрицы на матрицу L.

Матрица $L=\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$

2.Рассмотрим три 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 LAT (Linear Approximation Table).

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

Рис.1--LAT (Linear Approximation Table)

Программный код для реализации LAT таблицы для заданной подстановки приложен далее.

Оба активных полубайта равны 3, значит наиболее подходящими значениями в нашем случае являются β=5;13, у которых преобладание |ε|=4.

Выберем β=13.

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

Рис.2 –Выбор значения β

$$P_1 = \left(\frac{|-4|}{16}\right) \times \left(\frac{|-4|}{16}\right) = \frac{1}{16}$$

$$\Delta_{12} = \prod(\alpha) = \begin{pmatrix} 13 & 13 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Шаг 3: Перестановка (транспонирование матрицы)

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

Шаг 4: Линейное преобразование (умножение на матрицу L)

$$\Delta_{14} = \begin{pmatrix} 13 & 0 & 0 \\ 13 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 13 & 13 \\ 0 & 13 & 13 \\ 0 & 0 & 0 \end{pmatrix}$$

2 Раунд:

Шаг 1: $\alpha(\Delta_{14})$ xor key

$$\Delta_{21}=\begin{pmatrix}0 & 13 & 13\\ 0 & 13 & 13\\ 0 & 0 & 0\end{pmatrix}$$

Шаг 2: Sbox

Воспользуемся LAT (Linear Approximation Table)

Все четыре активных полубайта равны 13, значит наиболее подходящими значениями в нашем случае являются β=2;3, у которых преобладание |ε|=4.

Выберем β=2.

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

Рис.3 –Выбор значения β

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

$$\Delta_{22}=\prod(\alpha)=\begin{pmatrix}0 & 2 & 2\\ 0 & 2 & 2\\ 0 & 0 & 0\end{pmatrix}$$

Шаг 3: Перестановка (транспонирование матрицы)

$$\Delta_{23}=\begin{pmatrix} 0 & 0 & 0 \\ 2 & 2 & 0 \\ 2 & 2 & 0 \end{pmatrix}$$

Шаг 4: Линейное преобразование (умножение на матрицу L)

$$\Delta_{24}=\begin{pmatrix} 0 & 0 & 0 \\ 2 & 2 & 0 \\ 2 & 2 & 0 \end{pmatrix}\times\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}=\begin{pmatrix} 0 & 0 & 0 \\ 2 & 2 & 4 \\ 2 & 2 & 4 \end{pmatrix}$$

3 Раунд:

Шаг 1: $\alpha(\Delta_{24})$ xor key

$$\Delta_{31}= \begin{pmatrix} 0 & 0 & 0 \\ 2 & 2 & 4 \\ 2 & 2 & 4 \end{pmatrix}$$

Шаг 2: Sbox

Воспользуемся LAT (Linear Approximation Table).

Активные полубайта равны 2 и 4. Наиболее подходящими значениями для α=2 являются β=9;13, у которых преобладание $|\varepsilon|$=4. Для α=4 – β=1;6

Выберем β=9 и β=1 соответственно.

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

Рис.4 –Выбор значения β

$$\Delta_{32}=\prod(\alpha)=\begin{pmatrix}0 & 0 & 0\\ 9 & 9 & 1\\ 9 & 9 & 1\end{pmatrix}$$

$$P_3=\left(\frac{|4|}{16}\right)\times\left(\frac{|4|}{16}\right)\times\left(\frac{|4|}{16}\right)\times\left(\frac{|4|}{16}\right)\times\left(\frac{|-4|}{16}\right)\times\left(\frac{|-4|}{16}\right)=\left(\frac{1}{4}\right)^6$$

Шаг 3: Перестановка (транспонирование матрицы)

$$\Delta_{33}=\begin{pmatrix}0 & 9 & 9\\ 0 & 9 & 9\\ 0 & 1 & 1\end{pmatrix}$$

Шаг 4: Линейное преобразование (умножение на матрицу L)

$$\Delta_{34}=\begin{pmatrix}0 & 0 & 0\\ 2 & 2 & 0\\ 2 & 2 & 0\end{pmatrix}\times\begin{pmatrix}0 & 1 & 1\\ 1 & 0 & 1\\ 1 & 1 & 0\end{pmatrix}=\begin{pmatrix}18 & 9 & 9\\ 18 & 9 & 9\\ 2 & 1 & 1\end{pmatrix}$$

Итоговая вероятность:

$$P = P_1 \times P_2 \times P_3 = \frac{1}{16777216} = \frac{1}{2^{24}}$$

То есть нам необходимо 16 777 216 пар (открытый текст – зашифрованный текст), чтобы хотя бы в одном случае линейный путь был равен нашему.

Приложение

Программный код, вычисляющий LAT 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',[15,9,1,7,13,12,2,8,6,5,14,3,0,11,4,10])
M=array('i',[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15])

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

for x in range(0,16):
    print ('%3d'%x,end="")
    x1= x%2
    if (x1==1):
        X1=x1
    else:
        X1 = 0

    x2= (x // 2) % 2
    if (x2==1):
        X2 = x2
    else:
        X2 = 0


    x3=(x//4) % 2
    if (x3==1):
        X3 = x3
    else:
        X3= 0
```

```
x4= x // 8
if (x4==1):
    X4 = x4
else:
    X4=0

for j in range(0, 16):

    value=X4*(M[j]//8   ) ^ X3*( (M[j]//4)%2 ) ^ X2*( (M[j]// 2) % 2) ^ X1*( M[j]%2)
    Array_X[x][j]=value

for y in range(0, 16):
        Sovpadenie = 0
        y1 = y % 2
        if (y1 == 1):
            Y1 = y1
        else:
            Y1 = 0

        y2 = (y // 2) % 2
        if (y2 == 1):
            Y2 = y2
        else:
            Y2 = 0

        y3 = (y // 4) % 2
        if (y3 == 1):
            Y3 = y3
        else:
            Y3 = 0

        y4 = y // 8
        if (y4 == 1):
            Y4 = y4
        else:
            Y4 = 0

        for i in range(0, 16):

            P = Y4 * (S[i] // 8) ^ Y3 * ((S[i] // 4) % 2) ^ Y2 * ((S[i] // 2) % 2) ^ Y1 * (S[i] % 2)
            if (P == Array_X[x][i]):
                Sovpadenie += 1
        print("%3d" % (Sovpadenie-8),end=",")
print('')
```