Fishingrod 算法描述

Fishingrod 算法输入为 128 比特，轮秘钥K_i为 64 比特，输出为 128 比特,轮数为 18 轮，轮函数结构图如图 1 所示。输入左右分支L_i, R_i均为 64 比特，左分支L_i与轮秘钥K_i做“与”运算结果亦或上右分支R_i的结果作为轮函数F的输入。F 的输出亦或上L_i得到一轮加密的右分支输出R_{i+1}，F 的输出与轮秘钥K_i做“与”运算结果亦或上右分支R_i得到一轮加密的右分支输出L_{i+1}。

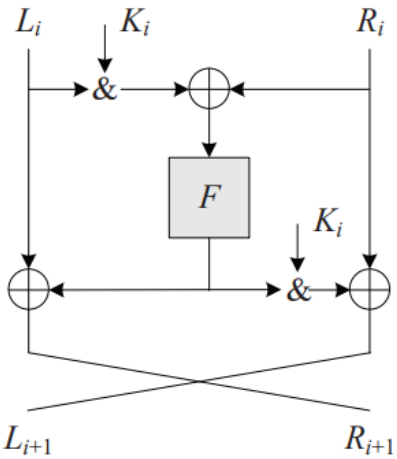

图 1 Fishingrod 轮函数结构图

F 函数

F 函数输入为 64 比特，输出为 64 比特，F 函数包含三个操作：通过 S 层(S)，循环移位(SR)，列混合(MC),即$\mathrm{F} = \mathrm{MC} \cdot \mathrm{SR} \cdot \mathrm{S}$。

S：S 层由 8 个 8 比特 S 盒组成，64 比特通过 S 盒，S 盒使用 AES 的 S 盒。经过 S 盒子运算后的状态表示为：

$$S_7, S_6, S_5, S_4, S_3 S_2, S_1, S_0$$

SR：将 S 层的输出循环左移两个字节。经过移位后的状态表示为：$S_5, S_4, S_3 S_2, S_1, S_0, S_7, S_6$

MC：采用 AES 的 MC 矩阵进行列混淆操作，具体运算如下：

$$\begin{bmatrix} A_7 & A_3 \\ A_6 & A_2 \\ A_5 & A_1 \\ A_4 & A_0 \end{bmatrix} \leftarrow \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} S_5 & S_1 \\ S_4 & S_0 \\ S_3 & S_7 \\ S_2 & S_6 \end{bmatrix}$$

密钥扩展

In this part, we are going to give an example which is based on the linear feedback shift register (LFSR):

The key schedule expands the 128-bit cipher key into 16 64-bit round-keys. Let $(k_0, k_1, k_2, k_3) \in \mathbb{F}_2^{32\times 4}$ be the 128-bit cipher key. Then,

$$k_{i+4} = \mathcal{D}(k_{i+3} \oplus k_i), \tag{1}$$

where $i = 0, 1, \ldots 13$ and $\mathcal{D}(x) = x \oplus (x_{\lll 2}) \oplus (x_{\lll 11})$.

According to Section 5.1, we define the round keys as $K_{2i} = k_{2i+1}||k_{2i}$ and $K_{2i+1} = \overline{K_{2i}}$. For this linear code, we have the following theorem:

给定初始的(k_0, k_1, k_2, k_3)，通过 LFSR 生成$(k_0, \ldots, k_{17})$。

可以计算出所有偶数轮的密钥

$$K_0 = k_1|k_0, K_2 = k_3|k_2, \ldots, K_{16} = k_{17}|k_{16}$$

奇数轮密钥等于偶数轮密钥的补，可以计算出轮密钥$K_0, \ldots, K_{17}$