轻量级分组密码 RECTANGLE

RECTANGLE 基于 SPN 网络设计，其分组长度为 64 比特，密钥长度分为 80 比特和 128 比特两个版本。本文对 RECTANGLE 算法基本结构、轮函数、密钥扩展算法做基本介绍。

1. 符号和术语

文中用到的符号及术语如下：

1) RC[t]：第 t 轮的轮常量。
2) ||：比特位(串)的连接，若 $a \| b$，则连接后 a 为高比特位(串)，b 为低比特位(串)；
3) <<<：循环左移操作，$X <<< n$ 表示将 X 循环左移 n 比特。
4) ⊕：按位异或。
5) $LSB_n(X)$：取 X 最低 (最右边) n 比特。
6) $MSB_n(X)$：取 X 最高 (最左边) n 比特。
7) 种子密钥：密码算法的初始密钥，用于密钥扩展算法的输入。
8) 轮子密钥：迭代型密码算法每一轮的子密钥，通过密钥扩展算法得到。

2. 算法状态

设 64 比特明文、密文、轮子密钥以及中间结果统称为一个算法状态。RECTANGLE 64 比特算法状态 $W = w_{63} \| w_{62} \| \cdots \| w_2 \| w_1 \| w_0$ 统一按照图 1 a)排列为 4×16 的矩阵，其中每个元素对应一个比特；图 1 b)为对应的二维坐标表示方法，其中 $a_{0,0}$ 表示最低比特位，$a_{0,1}$ 表示次低比特位，……，$a_{3,15}$ 表示最高比特位。定义：

$$\begin{bmatrix} w_{15} & \cdots & w_2 & w_1 & w_0 \\ w_{31} & \cdots & w_{18} & w_{17} & w_{16} \\ w_{47} & \cdots & w_{34} & w_{33} & w_{32} \\ w_{63} & \cdots & w_{50} & w_{49} & w_{48} \end{bmatrix} \qquad \begin{bmatrix} a_{0,15} & \cdots & a_{0,2} & a_{0,1} & a_{0,0} \\ a_{1,15} & \cdots & a_{1,2} & a_{1,1} & a_{1,0} \\ a_{2,15} & \cdots & a_{2,2} & a_{2,1} & a_{2,0} \\ a_{3,15} & \cdots & a_{3,2} & a_{3,1} & a_{3,0} \end{bmatrix}$$

a) 算法状态　　b) 二维坐标表示

图1. RECTANGLE 算法状态及二维坐标表示

1) $R_i = a_{i,15} \| \cdots \| a_{i,2} \| a_{i,1} \| a_{i,0}$ 表示算法状态的第 i 行，其中 $0 \le i \le 3$；
2) $C_j = a_{3,j} \| a_{2,j} \| a_{1,j} \| a_{0,j}$ 表示算法状态的第 j 列，其中 $0 \le j \le 15$。

3. 轮函数

RECTANGLE 一共有 25 轮，轮函数由轮密钥加(AddRoundKey)、列变换(SubColumn)、行移位(ShiftRow)三个模块组成，最后一轮之后增加一次轮密钥加操作。

1) 轮密钥加：64 比特的算法状态与轮子密钥按位异或。
2) 列变换：对每一列进行 S 盒置换，即$C_j^{'} = S(C_j)$，其中$0 \le j \le 15$，S 盒置换表如表 1 所示。

表1 RECTANGLE S 盒置换表

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	6	5	C	A	1	E	7	9	B	0	3	D	8	F	4	2

3) 行移位：对每一行进行循环左移，其中第 0 行保持不变，第 1 行循环左移 1 比特，第 2 行循环左移 12 比特，第 3 行循环左移 13 比特，如下所示。

$$R'_0 = R_0 <<< 0$$
$$R'_1 = R_1 <<< 1$$
$$R'_2 = R_2 <<< 12$$
$$R'_3 = R_3 <<< 13$$

4. 密钥扩展算法

- **80 比特密钥**

80 比特种子密钥$V = v_{79} \| \cdots \| v_2 \| v_1 \| v_0$排列成$5 \times 16$的矩阵，图 2 给出了密钥状态及其二维坐标表示。设$R_i = k_{i,15} \| ... \| k_{i,2} \| k_{i,1} \| k_{i,0}$表示第$i$行，其中$0 \le i \le 4$；$K_r$表示第$r$轮的轮子密钥，其中$0 \le r \le 25$。则第零轮的轮子密钥$K_0 = R_3 \| R_2 \| R_1 \| R_0$。接着循环执行如下操作 25 次，每循环一次之后将前 4 行作为新一轮的轮子密钥。

$$\begin{bmatrix} v_{15} & \cdots & v_2 & v_1 & v_0 \\ v_{31} & \cdots & v_{18} & v_{17} & v_{16} \\ v_{47} & \cdots & v_{34} & v_{33} & v_{32} \\ v_{63} & \cdots & v_{50} & v_{49} & v_{48} \\ v_{79} & \cdots & v_{66} & v_{65} & v_{64} \end{bmatrix} \qquad \begin{bmatrix} k_{0,15} & \cdots & k_{0,2} & k_{0,1} & k_{0,0} \\ k_{1,15} & \cdots & k_{1,2} & k_{1,1} & k_{1,0} \\ k_{2,15} & \cdots & k_{2,2} & k_{2,1} & k_{2,0} \\ k_{3,15} & \cdots & k_{3,2} & k_{3,1} & k_{3,0} \\ k_{4,15} & \cdots & k_{4,2} & k_{4,1} & k_{4,0} \end{bmatrix}$$

图2. 80 比特密钥状态及其二维坐标表示

a) S 盒操作：对前 4 行的最低 4 列分别进行 S 盒操作。

$k'_{3,j} \| k'_{2,j} \| k'_{1,j} \| k'_{0,j} = S(k_{3,j} \| k_{2,j} \| k_{1,j} \| k_{0,j})$，其中$0 \le j \le 3$。

b) 一轮 5 分支广义 Feistel 变换。

$$R'_0 = (R_0 <<< 8) \oplus R_1$$
$$R'_1 = R_2$$
$$R'_2 = R_3$$
$$R'_3 = (R_3 <<< 12) \oplus R_4$$
$$R'_4 = R_0$$

c) 轮常量异或。

$k'_{0,4} \| k'_{0,3} \| k'_{0,2} \| k'_{0,1} \| k'_{0,0} = (k_{0,4} \| k_{0,3} \| k_{0,2} \| k_{0,1} \| k_{0,0}) \oplus RC[t]$，$RC[t]$对应第$t$轮的轮常量，其中$1 \le t \le 25$，长度为 5 比特，通过线性反馈移位寄存器得到。

- **128 比特密钥**

图 3 对应 128 比特密钥状态的二维坐标表示。设$R_i = k_{i,31} \| ... \| k_{i,2} \| k_{i,1} \| k_{i,0}$用来表示第$i$行，$0 \le i \le 3$；$K_r$表示第$r$轮的轮子密钥，$0 \le r \le 25$。则第零轮的轮子密钥$K_0 = LSB_{16}(R_3) \| LSB_{16}(R_2) \| LSB_{16}(R_1) \| LSB_{16}(R_0)$。接着循环执行如下步骤 25 此，每次产生一个轮子密钥。

$$\begin{bmatrix} k_{0,31} & \cdots & k_{0,2} & k_{0,1} & k_{0,0} \\ k_{1,31} & \cdots & k_{1,2} & k_{1,1} & k_{1,0} \\ k_{2,31} & \cdots & k_{2,2} & k_{2,1} & k_{2,0} \\ k_{3,31} & \cdots & k_{3,2} & k_{3,1} & k_{3,0} \end{bmatrix}$$

图3. 128 比特密钥状态二维坐标表示

a) S 盒操作：对最低 8 列分别进行 S 盒操作。

$k'_{3,j} \| k'_{2,j} \| k'_{1,j} \| k'_{0,j} = S(k_{3,j} \| k_{2,j} \| k_{1,j} \| k_{0,j})$，其中$0 \le j \le 7$。

b) 一轮 4 分支广义 Feistel 变换。

$$R'_0 = (R_0 <<< 8) \oplus R_1$$
$$R'_1 = R_2$$
$$R'_2 = (R_2 <<< 16) \oplus R_3$$
$$R'_3 = R_0$$

c) 轮常量异或：与 80 比特密钥的轮常量异或一致。