

SMBA 分组密码算法 设计描述

兴唐通信科技有限公司

2019 年 2 月

目 次

1	概述.....	1
2	符号、运算符和约定.....	1
2.1	符号和运算符列表.....	1
2.2	比特、字节和子块顺序约定.....	2
3	加解密过程.....	2
3.1	加密过程.....	2
3.2	解密过程.....	2
4	轮函数及其组件.....	3
4.1	轮函数 $F[K]$	3
4.2	白化变换 $W[K]$	3
4.3	换位变换 P	3
4.3.1	换位变换 P_{64}	3
4.3.2	换位变换 P_{128}	4
4.4	代替变换 S	4
4.4.1	代替变换 S_{32}	4
4.4.2	代替变换 S_{64}	4
4.4.3	代替变换 S_{128}	4
4.5	线性变换 L	5
4.5.1	线性变换 L_{32}	5
4.5.2	线性变换 L_{64}	5
4.5.3	线性变换 L_{128}	5
5	密钥扩展.....	6
5.1	变换 α 、 β 、 γ	6
5.1.1	变换 α_{64} 、 β_{64} 、 γ_{64}	6
5.1.2	变换 α_{128} 、 β_{128} 、 γ_{128}	6
5.2	SMBA-128-128 的子密钥生成.....	7
5.3	SMBA-128-256 的子密钥生成.....	8
5.4	SMBA-256-256 的子密钥生成.....	8

6	S 盒 S_0 和 S_1.....	9
7	样本数据.....	10
7.1	SMBA-128-128 样本数据	10
7.2	SMBA-128-256 样本数据	11
7.3	SMBA-256-256 样本数据	12
8	密码算法设计框图.....	15
8.1	加密过程图	15
8.2	解密过程图	16
8.3	SMBA-128 轮函数图	17
8.4	SMBA-256 轮函数图	18
8.5	SMBA-128-128 和 SMBA-256-256 密钥扩展轮变换图	18
8.6	SMBA-128-256 密钥扩展轮变换图	19
9	附录 S 盒 S_0、S_1 构造过程.....	19
9.1	S_0 构造过程.....	19
9.1.1	有限域多项式基上的计算	19
9.1.2	S_0 构造图示.....	20
9.1.3	S_0 构造过程描述.....	20
9.2	S_1 构造过程.....	21
9.2.1	有限域正规基上的计算	21
9.2.2	S_1 构造图示.....	22
9.2.3	S_1 构造过程描述.....	22

1 概述

SMBA 是一个分组密码算法，分组长度支持 128 和 256 比特。分组长度为 128 比特时密钥长度支持 128 和 256 比特，分组长度为 256 比特时密钥长度为 256 比特。分组长度（比特数，记为 BL）和密钥长度（比特数，记为 KL）对应的 SMBA 的迭代轮数（记为 N）如下表所示。

密钥长度 分组长度	128 比特	256 比特
128 比特	18	24
256 比特	—	24

分组长度为 128 比特的 SMBA 记为 SMBA-128, 其中密钥长度为 128 或 256 比特时也分别记为 SMBA-128-128 或 SMBA-128-256。分组长度为 256 比特的 SMBA 记为 SMBA-256, 也可记为 SMBA-256-256。

需要说明的是，下文所述的图 1~图 6 都在“8 算法设计图”中给出。

2 符号、运算符和约定

2.1 符号和运算符列表

0x	用前缀0x表示16进制数。
=	赋值运算符，也可表示相等。
\oplus	逐比特异或运算符。
&	逐比特与运算符，其运算优先级高于 \oplus 。
\vee	逐比特或运算符。
	字符串连接运算符。
$\lll n$	操作数循环左移n比特运算符。
$MSB_t(x)$	由x的最左（即最高）t比特组成的t比特数。
$LSB_t(x)$	由x的最右（即最低）t比特组成的t比特数。
M^T	向量或矩阵M的转置。
0^n	n比特0组成的比特串。

- 变换复合运算符，即对于变换 f_1 、 f_2 和变量 x ： $f_2 \circ f_1(x) = f_2(f_1(x))$ 。
- BL 分组长度，比特数。
- KL 密钥长度，比特数。
- N 迭代轮数。

2.2 比特、字节和子块顺序约定

所有数据变量用最高比特（或子块）在左边，最低比特（或子块）在右边表示，而且从左到右从0开始编号。例如，一个 n 比特数从左到右分别称为它的第0比特、第1比特、第2比特、...、第 $n-1$ 比特。当一个变量被分成几个子块时，最左边（最高）子块由原始数据的最高部分组成，等等一直到最低。

当密码算法的输入（或输出）是字节流时，我们按地址小的是高位（即左边），地址大的是低位（即右边）的顺序把每4（或2）个字节组成一个32（或16）比特字。32（或16）比特字之间的顺序是先左后右，即存储时，左边地址小，右边地址大。

约定比特串 $x_0||x_1||x_2||\dots||x_{n-1}$ 、整数 $2^{n-1}x_0+2^{n-2}x_1+2^{n-3}x_2+\dots+2x_{n-2}+x_{n-1}$ 、行向量 $(x_0,x_1,x_2,\dots,x_{n-1})$ 、列向量 $(x_0,x_1,x_2,\dots,x_{n-1})^T$ 是等价的，可互相转化。

3 加解密过程

3.1 加密过程

SMBA 的加密过程（参见图 1） $\text{ciphertext} = E_{\text{Key}}(\text{plaintext})$ 如下：

(1) 将 BL 比特明文 plaintext 分为各 $BL/2$ 比特的左、右两部分 x_0 和 x_1 ，即 $\text{plaintext} = x_0||x_1$ 。

(2) 对于 $i=0,1,\dots,N-1$

$$x_{i+2} = x_i \oplus F[EK_i](x_{i+1})$$

(3) x_{N+1} 和 x_N 连接成 BL 比特密文 ciphertext ，即 $\text{ciphertext} = x_{N+1}||x_N$ 。

其中， F 为轮函数， EK_i （ $0 \leq i \leq N-1$ ）为 N 个 $BL/2$ 比特的加密子密钥，由密钥 Key 用密钥扩展派生。

3.2 解密过程

SMBA 的解密过程（参见图 2） $\text{plaintext} = D_{\text{Key}}(\text{ciphertext})$ 如下：

(1) 将 BL 比特密文 **ciphertext** 分为各 BL/2 比特的左、右两部分 x_0 和 x_1 ，即

$$\text{ciphertext} = x_0 || x_1。$$

(2) 对于 $i=0,1,\dots,N-1$

$$x_{i+2} = x_i \oplus F[DK_i](x_{i+1})$$

(3) x_{N+1} 和 x_N 连接成 BL 比特明文 **plaintext**，即 $\text{plaintext} = x_{N+1} || x_N。$

其中，F为轮函数， DK_i ($0 \leq i \leq N-1$) 为N个BL/2比特的解密子密钥，由密钥Key用密钥扩展派生。

解密子密钥与加密子密钥满足如下关系： $DK_i = EK_{N-1-i}$ ， $0 \leq i \leq N-1$ 。

4 轮函数及其组件

4.1 轮函数 F[K]

轮函数 $y=F[K](x)$ 的输入是一个BL/2比特数 x 和一个BL/2比特子密钥 K ，输出是一个BL/2比特数 y ， $y=F[K](x)=L(S(P(W[K](x))))$ ，即 $F[K]=L \circ S \circ P \circ W[K]$ （参见图3、图4）。各变换 $W[K]$ 、 P 、 S 、 L 下面定义。

4.2 白化变换 W[K]

对于BL/2比特数 x 和BL/2比特子密钥 K ， $W[K](x)=x \oplus K。$

4.3 换位变换 P

换位变换 P 对于 SMBA-128 使用换位变换 P_{64} ，对于 SMBA-256 使用换位变换 P_{128} 。

4.3.1 换位变换 P_{64}

P_{64} 是一个 64 比特数到 64 比特数的变换，具体为：把 64 比特数 x 分为 8 个 8 比特数 x_i ， $0 \leq i \leq 7$ ，即

$$x = x_0 || x_1 || x_2 || x_3 || x_4 || x_5 || x_6 || x_7$$

令 $y_i = x_{\tau(i)}$ ，则

$$P_{64}(x) = y_0 || y_1 || y_2 || y_3 || y_4 || y_5 || y_6 || y_7$$

其中 $\tau(i)$ ， $0 \leq i \leq 7$ ，依次为：3,4,5,6, 0,1,2,7。

4.3.2 换位变换 P_{128}

P_{128} 是一个 128 比特数到 128 比特数的变换，具体为：把 128 比特数 x 分为 16 个 8 比特数 x_i ， $0 \leq i \leq 15$ ，即

$$x = x_0 || x_1 || x_2 || x_3 || x_4 || x_5 || x_6 || x_7 || x_8 || x_9 || x_{10} || x_{11} || x_{12} || x_{13} || x_{14} || x_{15}$$

令 $y_i = x_{pi(i)}$ ，则

$$P_{128}(x) = y_0 || y_1 || y_2 || y_3 || y_4 || y_5 || y_6 || y_7 || y_8 || y_9 || y_{10} || y_{11} || y_{12} || y_{13} || y_{14} || y_{15}$$

其中 $pi(i)$ ， $0 \leq i \leq 15$ ，依次为：3,4,5,6, 10,11,12,13, 8,9,14,15, 0,1,2,7。

4.4 代替变换 S

代替变换 S 对于 SMBA-128 使用代替变换 S_{64} ，对于 SMBA-256 使用代替变换 S_{128} 。

4.4.1 代替变换 S_{32}

S_{32} 是一个 32 比特数到 32 比特数的变换，具体为：将 32 比特数 x 分为 4 个 8 比特数，即 $x = x_0 || x_1 || x_2 || x_3$ ，令

$$y_0 = S_0(x_0), \quad y_1 = S_1(x_1), \quad y_2 = S_0(x_2), \quad y_3 = S_1(x_3)$$

则 $S_{32}(x) = y_0 || y_1 || y_2 || y_3$ 。

其中， S_0 、 S_1 都是 8 比特到 8 比特的代替盒（S 盒）。

4.4.2 代替变换 S_{64}

S_{64} 是一个 64 比特数到 64 比特数的变换，具体为：将 64 比特数 x 分为 2 个 32 比特数，即 $x = x_0 || x_1$ ，令

$$y_0 = S_{32}(x_0), \quad y_1 = S_{32}(x_1)$$

则 $S_{64}(x) = y_0 || y_1$ 。

4.4.3 代替变换 S_{128}

S_{128} 是一个 128 比特数到 128 比特数的变换，具体为：将 128 比特数 x 分为 2 个 64 比特数，即 $x = x_0 || x_1$ ，令

$$y_0 = S_{64}(x_0), \quad y_1 = S_{64}(x_1)$$

则 $S_{128}(x)=y_0||y_1$ 。

4.5 线性变换 L

线性变换 L 对于 SMBA-128 使用线性变换 L_{64} ，对于 SMBA-256 使用线性变换 L_{128} 。

4.5.1 线性变换 L_{32}

L_{32} 是32比特数到32比特数的变换，具体为：将32比特数 x 分为4个8比特数，即 $x=x_0||x_1||x_2||x_3$ ，令

$$y_0= x_0\oplus x_2\oplus x_3$$

$$y_1= x_1\oplus x_2\oplus x_3$$

$$y_2= x_0\oplus x_1\oplus x_2$$

$$y_3= x_0\oplus x_1\oplus x_3$$

则 $L_{32}(x)=y_0||y_1||y_2||y_3$ 。

4.5.2 线性变换 L_{64}

L_{64} 是一个64比特数到64比特数的变换，具体为：将64比特数 x 分为2个32比特数，即 $x= x_0||x_1$ ，令

$$z_0= L_{32}(x_0), \quad z_1= L_{32}(x_1)$$

$$u= z_0\oplus z_1$$

$$v= u\lll 9$$

$$y_0= z_0\oplus v, \quad y_1= z_1\oplus v$$

则 $L_{64}(x)=y_0||y_1$ 。

4.5.3 线性变换 L_{128}

L_{128} 是一个128比特数到128比特数的变换，具体为：将128比特数 x 分为2个64比特数，即 $x= x_0||x_1$ ，令

$$y_0= L_{64}(x_0), \quad y_1= L_{64}(x_1)$$

则 $L_{128}(x)=y_0||y_1$ 。

5 密钥扩展

密钥扩展算法根据密钥 Key，生成 N 个 BL/2 比特加密子密钥 EK_i 和 N 个 BL/2 比特解密子密钥 DK_i，0≤i≤N-1。

常数 C 为自然对数的底 e 的小数部分的前 128 比特，用 16 进制表示为

$$C=0xb7e15162, 0x8aed2a6a, 0xbf715880, 0x9cf4f3c7$$

5.1 变换 α 、 β 、 γ

变换 α 、 β 、 γ 对于 SMBA-128-128 和 SMBA-128-256 使用 α_{64} 、 β_{64} 、 γ_{64} ，对于 SMBA-256-256 使用 α_{128} 、 β_{128} 、 γ_{128} 。

5.1.1 变换 α_{64} 、 β_{64} 、 γ_{64}

α_{64} 、 β_{64} 、 γ_{64} 都是 64 比特到 64 比特的变换，分别如下。

(1) 把 64 比特数 x 分为 8 个 8 比特数 x_i ，0≤i≤7，即

$$x=x_0||x_1||x_2||x_3||x_4||x_5||x_6||x_7$$

令 $y_i=x_{\alpha_{64}(i)}$ ，则

$$\alpha_{64}(x)=y_0||y_1||y_2||y_3||y_4||y_5||y_6||y_7$$

其中 $\alpha_{64}(i)$ ，0≤i≤7，依次为：2,7,1,4,6,3,5,0。

(2) $\beta_{64}(x)=x<<<16$

(3) 把 64 比特数 x 分为 8 个 8 比特数 x_i ，0≤i≤7，即

$$x=x_0||x_1||x_2||x_3||x_4||x_5||x_6||x_7$$

令 $y_i=S_{i \bmod 2}(x_i)$ ，则

$$\gamma_{64}(x)=y_0||y_1||y_2||y_3||y_4||y_5||y_6||y_7$$

5.1.2 变换 α_{128} 、 β_{128} 、 γ_{128}

α_{128} 、 β_{128} 、 γ_{128} 都是 128 比特到 128 比特的变换，分别如下。

(1) 把 128 比特数 x 分为 16 个 8 比特数 x_i ，0≤i≤15，即

$$x=x_0||x_1||x_2||x_3||x_4||x_5||x_6||x_7||x_8||x_9||x_{10}||x_{11}||x_{12}||x_{13}||x_{14}||x_{15}$$

令 $y_i=x_{\alpha_{128}(i)}$ ，则

$$\alpha_{128}(x)=y_0||y_1||y_2||y_3||y_4||y_5||y_6||y_7||y_8||y_9||y_{10}||y_{11}||y_{12}||y_{13}||y_{14}||y_{15}$$

其中 $\alpha_{128}(i)$ ， $0 \leq i \leq 15$ ，依次为：13,4,12,0,8,10,2,6,14,3,11,15,7,9,1,5。

$$(2) \beta_{128}(x)=x \ll 32$$

(3) 把 128 比特数 x 分为 16 个 8 比特数 x_i ， $0 \leq i \leq 15$ ，即

$$x=x_0||x_1||x_2||x_3||x_4||x_5||x_6||x_7||x_8||x_9||x_{10}||x_{11}||x_{12}||x_{13}||x_{14}||x_{15}$$

令 $y_i=S_{i \bmod 2}(x_i)$ ，则

$$\gamma_{128}(x)=y_0||y_1||y_2||y_3||y_4||y_5||y_6||y_7||y_8||y_9||y_{10}||y_{11}||y_{12}||y_{13}||y_{14}||y_{15}。$$

5.2 SMBA-128-128 的子密钥生成

加密子密钥和解密子密钥按如下方式生成：

(1) $U_0=MSB_{64}(Key)$ ， $V_0=LSB_{64}(Key)$ ；

(2) $D=N||BlockLen||KeyLen||0^{40}$ ，这里 $BlockLen=BL/8$ 为分组长度（字节数），
 $KeyLen=KL/8$ 为密钥长度（字节数）， N 、 $BlockLen$ 、 $KeyLen$ 都用 8 比特表示。

(3) $C_0=MSB_{64}(C) \oplus D$ ；

(4) 进行 N 个密钥扩展轮变换（参见图 5），即

对于 $i=0$ 到 $N-1$

$$\left\{ \begin{array}{l} X_i=U_i \oplus C_i \\ Y_i=\alpha_{64}(X_i) \\ Z_i=\gamma_{64}(Y_i) \\ EK_i = DK_{N-1-i} = Z_i \\ U_{i+1}=Z_i \oplus V_i \\ V_{i+1}=\beta_{64}(U_i) \\ C_{i+1}=C_i \ll 23 \end{array} \right.$$

5.3 SMBA-128-256 的子密钥生成

加密子密钥和解密子密钥按如下方式生成：

- (1) $U_{0,0}||U_{0,1}||U_{0,2}||U_{0,3}=\text{Key}$ ，其中 $U_{0,i}$ 的长度都是 64 比特， $0 \leq i \leq 3$ ；
- (2) $D=N||\text{BlockLen}||\text{KeyLen}||0^{40}$ ，这里 $\text{BlockLen}=\text{BL}/8$ 为分组长度（字节数）， $\text{KeyLen}=\text{KL}/8$ 为密钥长度（字节数）， N 、 BlockLen 、 KeyLen 都用 8 比特表示。
- (3) $C_0 = \text{MSB}_{64}(C) \oplus D$ ；
- (4) 进行 N 个密钥扩展轮变换（参见图 6），即

对于 $i=0$ 到 $N-1$

$$\left\{ \begin{array}{l} X_i = U_{i,0} \oplus C_i \\ Y_i = \alpha_{64}(X_i) \\ Z_i = \gamma_{64}(Y_i) \\ EK_i = DK_{N-1-i} = Z_i \\ U_{i+1,0} = Z_i \oplus U_{i,1} \\ U_{i+1,1} = U_{i,2} \\ U_{i+1,2} = U_{i,3} \\ U_{i+1,3} = \beta_{64}(U_{i,0}) \\ C_{i+1} = C_i \lll 23 \end{array} \right.$$

5.4 SMBA-256-256 的子密钥生成

加密子密钥和解密子密钥按如下方式生成：

- (1) $U_0 = \text{MSB}_{128}(\text{Key})$ ， $V_0 = \text{LSB}_{128}(\text{Key})$ ；
- (2) $D=N||\text{BlockLen}||\text{KeyLen}||0^{104}$ ，这里 $\text{BlockLen}=\text{BL}/8$ 为分组长度（字节数）， $\text{KeyLen}=\text{KL}/8$ 为密钥长度（字节数）， N 、 BlockLen 、 KeyLen 都用 8 比特表示。
- (3) $C_0 = C \oplus D$ ；
- (4) 进行 N 个密钥扩展轮变换（参见图 5），即

对于 $i=0$ 到 $N-1$

{

$$X_i = U_i \oplus C_i$$

$$Y_i = \alpha_{128}(X_i)$$

$$Z_i = \gamma_{128}(Y_i)$$

$$EK_i = DK_{N-1-i} = Z_i$$

$$U_{i+1} = Z_i \oplus V_i$$

$$V_{i+1} = \beta_{128}(U_i)$$

$$C_{i+1} = C_i \lll 57$$

}

6 S 盒 S_0 和 S_1

S 盒 S_0 和 S_1 的构造过程见附录。用 16 进制表示的代替盒 S_0 和 S_1 分别如下：

S_0 :

0x18, 0x32, 0x6b, 0x69, 0x9e, 0xcb, 0x0b, 0x2f, 0x03, 0x94, 0x9f, 0x21, 0xc8, 0x86, 0x26, 0xf4,
 0x46, 0x67, 0x0d, 0x6d, 0x39, 0xc0, 0x09, 0x29, 0xa1, 0xda, 0x82, 0xf3, 0x78, 0x6a, 0xee, 0x3b,
 0x84, 0x87, 0xd1, 0xf8, 0x70, 0xb9, 0xa3, 0x3f, 0xfe, 0x3d, 0x37, 0xf7, 0xfb, 0x48, 0xc6, 0xbc,
 0x65, 0x4e, 0x6e, 0x64, 0xaf, 0x91, 0xdb, 0x1c, 0x71, 0x8e, 0x59, 0x8d, 0x01, 0x74, 0xe3, 0xa8,
 0xb4, 0xbf, 0x4b, 0x2d, 0xea, 0x95, 0x6f, 0x2b, 0x5f, 0xb3, 0x98, 0x57, 0xfd, 0x24, 0x85, 0xe9,
 0x5d, 0x6c, 0x4d, 0x4f, 0x13, 0xe1, 0x49, 0x0f, 0xdc, 0x36, 0x55, 0x3c, 0xef, 0x11, 0x50, 0x4a,
 0x73, 0x5b, 0x8a, 0xa9, 0x92, 0x58, 0x35, 0x80, 0x38, 0x04, 0x06, 0x12, 0xb7, 0xc7, 0x68, 0xd2,
 0x45, 0x66, 0xcd, 0x47, 0xac, 0xba, 0xf2, 0x16, 0x4c, 0xb2, 0x5c, 0x99, 0xc2, 0x8b, 0x23, 0x7c,
 0xe5, 0x9b, 0xab, 0x42, 0xad, 0x05, 0x28, 0xce, 0xca, 0x5e, 0x79, 0x2e, 0xf5, 0x72, 0x43, 0x77,
 0x96, 0x33, 0x7a, 0xa4, 0x40, 0xde, 0x75, 0xf9, 0x89, 0xd9, 0xb5, 0x1a, 0x9a, 0xb6, 0xd6, 0x8f,
 0xb0, 0xe4, 0x62, 0x88, 0x7e, 0x83, 0x15, 0x3a, 0x2c, 0x51, 0x56, 0xeb, 0x90, 0xf1, 0x1b, 0x00,
 0x9d, 0x19, 0x52, 0xa7, 0x34, 0x53, 0x97, 0x25, 0xbe, 0x30, 0xaa, 0xf0, 0xa2, 0x02, 0x0c, 0xe7,
 0xd3, 0x08, 0xc1, 0xf6, 0xa0, 0x76, 0x10, 0x1f, 0x61, 0xa5, 0x14, 0xec, 0x22, 0x31, 0xd8, 0xbb,
 0xc5, 0xe8, 0x54, 0x93, 0x3e, 0x7b, 0x9c, 0x27, 0xdd, 0x1d, 0xcc, 0xe2, 0x0e, 0xe6, 0x81, 0x20,
 0x2a, 0xfa, 0xff, 0xe0, 0x07, 0xae, 0xcf, 0x0a, 0xed, 0x1e, 0xa6, 0x41, 0x7f, 0x63, 0x5a, 0xfc,
 0xd5, 0xc9, 0xd4, 0xb8, 0x60, 0xd7, 0x7d, 0xd0, 0xc4, 0xc3, 0x44, 0x17, 0xdf, 0x8c, 0xb1, 0xbd.

S_1 :

0x6a, 0x05, 0xe8, 0x16, 0x7f, 0x37, 0x68, 0x54, 0xbe, 0x01, 0xc1, 0x87, 0xbf, 0xb3, 0xd1, 0x44,
 0xf8, 0xf9, 0xd4, 0xf5, 0x1f, 0x77, 0xa5, 0x3b, 0x04, 0x92, 0x62, 0xd7, 0xc6, 0xa7, 0x0c, 0xb1,
 0xbd, 0x5c, 0xff, 0xf4, 0x17, 0x3d, 0xa3, 0x32, 0x9e, 0x09, 0x8c, 0x63, 0x66, 0x10, 0xe9, 0x47,

0xa8, 0xfe, 0x3c, 0xc0, 0x7d, 0x1d, 0xfd, 0x4d, 0x4c, 0xd8, 0xd9, 0x9c, 0x34, 0xd0, 0x25, 0xeb,
0x91, 0x61, 0xe0, 0x1b, 0x95, 0x98, 0xe5, 0xba, 0xe3, 0x64, 0xad, 0x0f, 0x73, 0x19, 0x39, 0xbb,
0x00, 0xe7, 0xf2, 0x3a, 0x43, 0x79, 0x8d, 0xf0, 0x78, 0x23, 0xc4, 0xa0, 0x0b, 0x03, 0xb4, 0x52,
0x69, 0xf1, 0xf6, 0xc9, 0x0d, 0xb6, 0x8a, 0xa9, 0x27, 0xf3, 0x86, 0x88, 0xea, 0xc2, 0x29, 0x14,
0x8f, 0x22, 0x9b, 0xc7, 0x26, 0xed, 0xa2, 0xab, 0xd5, 0xaa, 0xc5, 0x6d, 0x60, 0x42, 0x65, 0xa6,
0x99, 0xb2, 0xac, 0x81, 0x3f, 0x5d, 0xa1, 0x72, 0x74, 0x30, 0xc3, 0xef, 0x9a, 0x1e, 0x7b, 0x2a,
0x67, 0x48, 0x24, 0x31, 0x35, 0x15, 0xf7, 0x2f, 0x89, 0x20, 0x40, 0xdd, 0x50, 0x2e, 0x08, 0x11,
0xde, 0x80, 0x5b, 0x6c, 0x5f, 0x75, 0x4a, 0x7e, 0x5e, 0x70, 0xce, 0x6b, 0xcf, 0x84, 0xda, 0xec,
0x76, 0xb7, 0x83, 0xee, 0x55, 0x57, 0xcd, 0x51, 0x8b, 0x02, 0xb9, 0xbc, 0xd3, 0xb5, 0xae, 0x90,
0x46, 0xe2, 0x71, 0xc8, 0x6e, 0x85, 0x8e, 0x36, 0x38, 0x2b, 0x18, 0x0a, 0xb0, 0xfa, 0x3e, 0x21,
0x59, 0x7c, 0x7a, 0xb8, 0x6f, 0x96, 0xcc, 0x1c, 0xe1, 0x2c, 0x06, 0xfc, 0x97, 0x9d, 0xe4, 0x07,
0xca, 0x0e, 0xcb, 0xd6, 0x4b, 0x13, 0x4f, 0x49, 0xdf, 0x82, 0x4e, 0xdc, 0x5a, 0x1a, 0xdb, 0x53,
0x56, 0x33, 0xaf, 0xfb, 0x9f, 0xd2, 0x2d, 0x41, 0x45, 0x93, 0x58, 0x28, 0xe6, 0xa4, 0x12, 0x94.

7 样本数据

7.1 SMBA-128-128 样本数据

第1组样本数据:

密钥: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

明文: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

密文: 0x21, 0x9a, 0x3b, 0x8f, 0x61, 0x47, 0x98, 0x64, 0x8f, 0xc4, 0xd6, 0xd1, 0x9d, 0x72, 0x86, 0x6d,

第2组样本数据:

密钥: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

明文: 0x21, 0x9a, 0x3b, 0x8f, 0x61, 0x47, 0x98, 0x64, 0x8f, 0xc4, 0xd6, 0xd1, 0x9d, 0x72, 0x86, 0x6d,

密文: 0x1e, 0xd5, 0x77, 0x29, 0x53, 0x7a, 0xaf, 0x45, 0x10, 0x67, 0x92, 0x64, 0x1a, 0x4f, 0x21, 0x7e,

第3组样本数据:

密钥: 0x21, 0x9a, 0x3b, 0x8f, 0x61, 0x47, 0x98, 0x64, 0x8f, 0xc4, 0xd6, 0xd1, 0x9d, 0x72, 0x86, 0x6d,

明文: 0x1e, 0xd5, 0x77, 0x29, 0x53, 0x7a, 0xaf, 0x45, 0x10, 0x67, 0x92, 0x64, 0x1a, 0x4f, 0x21, 0x7e,

密文: 0x66, 0xc5, 0xa2, 0x73, 0x42, 0x0c, 0x3c, 0x23, 0x84, 0x8a, 0xf4, 0x2e, 0x8f, 0xb3, 0x9e, 0xed,

第4组样本数据:

密钥: 0x1e, 0xd5, 0x77, 0x29, 0x53, 0x7a, 0xaf, 0x45, 0x10, 0x67, 0x92, 0x64, 0x1a, 0x4f, 0x21, 0x7e,

明文: 0x66, 0xc5, 0xa2, 0x73, 0x42, 0x0c, 0x3c, 0x23, 0x84, 0x8a, 0xf4, 0x2e, 0x8f, 0xb3, 0x9e, 0xed,

密文: 0x32, 0x10, 0x80, 0xcd, 0x29, 0x39, 0x4e, 0x91, 0xb0, 0x0e, 0xfc, 0xd5, 0x97, 0xa4, 0xa2, 0xa3,

第5组样本数据:

密钥: 0x66, 0xc5, 0xa2, 0x73, 0x42, 0x0c, 0x3c, 0x23, 0x84, 0x8a, 0xf4, 0x2e, 0x8f, 0xb3, 0x9e, 0xed,

明文: 0x32, 0x10, 0x80, 0xcd, 0x29, 0x39, 0x4e, 0x91, 0xb0, 0x0e, 0xfc, 0xd5, 0x97, 0xa4, 0xa2, 0xa3,

密文: 0x98, 0xc9, 0xe0, 0x38, 0x42, 0x8f, 0x45, 0xe5, 0xf3, 0xf5, 0xc2, 0x35, 0x9d, 0x72, 0x7b, 0x71,

7.2 SMBA-128-256 样本数据

第1组样本数据:

密钥: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

明文: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

密文: 0x40, 0x76, 0x91, 0x2d, 0xfc, 0x2b, 0x9b, 0xe2, 0x20, 0xb9, 0x76, 0x27, 0xe1, 0x72, 0xf3, 0x33,

第2组样本数据:

密钥: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

明文: 0x40, 0x76, 0x91, 0x2d, 0xfc, 0x2b, 0x9b, 0xe2, 0x20, 0xb9, 0x76, 0x27, 0xe1, 0x72, 0xf3, 0x33,

密文: 0x93, 0x7b, 0x88, 0xd4, 0x88, 0x71, 0x18, 0x28, 0x90, 0x4d, 0x14, 0x6b, 0x8b, 0x60, 0x27, 0x37,

第3组样本数据:

密钥: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x40, 0x76, 0x91, 0x2d, 0xfc, 0x2b, 0x9b, 0xe2, 0x20, 0xb9, 0x76, 0x27, 0xe1, 0x72, 0xf3, 0x33,
明文: 0x93, 0x7b, 0x88, 0xd4, 0x88, 0x71, 0x18, 0x28, 0x90, 0x4d, 0x14, 0x6b, 0x8b, 0x60, 0x27, 0x37,
密文: 0x47, 0xc7, 0x8d, 0x6e, 0xad, 0xd6, 0xb6, 0x19, 0x2b, 0x7c, 0x01, 0xf0, 0xe6, 0x72, 0x05, 0x40,

第4组样本数据:

密钥: 0x40, 0x76, 0x91, 0x2d, 0xfc, 0x2b, 0x9b, 0xe2, 0x20, 0xb9, 0x76, 0x27, 0xe1, 0x72, 0xf3, 0x33,
0x93, 0x7b, 0x88, 0xd4, 0x88, 0x71, 0x18, 0x28, 0x90, 0x4d, 0x14, 0x6b, 0x8b, 0x60, 0x27, 0x37,
明文: 0x47, 0xc7, 0x8d, 0x6e, 0xad, 0xd6, 0xb6, 0x19, 0x2b, 0x7c, 0x01, 0xf0, 0xe6, 0x72, 0x05, 0x40,
密文: 0xef, 0xf3, 0xb8, 0x2f, 0x5a, 0xf5, 0x4d, 0x60, 0xb0, 0xb5, 0xca, 0x67, 0xae, 0x8a, 0xb6, 0x14,

第5组样本数据:

密钥: 0x93, 0x7b, 0x88, 0xd4, 0x88, 0x71, 0x18, 0x28, 0x90, 0x4d, 0x14, 0x6b, 0x8b, 0x60, 0x27, 0x37,
0x47, 0xc7, 0x8d, 0x6e, 0xad, 0xd6, 0xb6, 0x19, 0x2b, 0x7c, 0x01, 0xf0, 0xe6, 0x72, 0x05, 0x40,
明文: 0xef, 0xf3, 0xb8, 0x2f, 0x5a, 0xf5, 0x4d, 0x60, 0xb0, 0xb5, 0xca, 0x67, 0xae, 0x8a, 0xb6, 0x14,
密文: 0x39, 0xfd, 0x4f, 0x48, 0x34, 0x3b, 0xda, 0xd7, 0x3e, 0xc4, 0x01, 0xf1, 0xde, 0x55, 0x2a, 0x01,

7.3 SMBA-256-256 样本数据

第1组样本数据:

密钥: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
明文: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
密文: 0x1b, 0x8b, 0x33, 0x77, 0x1d, 0x34, 0xa3, 0x64, 0xd2, 0x89, 0x20, 0x99, 0xb3, 0xd9, 0x0b, 0x1f,

0x43, 0x62, 0x50, 0xea, 0x73, 0x6c, 0x45, 0x30, 0xa3, 0xd3, 0xf8, 0xa4, 0xab, 0x1c, 0xb7, 0x59,

第2组样本数据:

密钥: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

明文: 0x1b, 0x8b, 0x33, 0x77, 0x1d, 0x34, 0xa3, 0x64, 0xd2, 0x89, 0x20, 0x99, 0xb3, 0xd9, 0x0b, 0x1f,

0x43, 0x62, 0x50, 0xea, 0x73, 0x6c, 0x45, 0x30, 0xa3, 0xd3, 0xf8, 0xa4, 0xab, 0x1c, 0xb7, 0x59,

密文: 0xbb, 0x46, 0x09, 0x4e, 0x8d, 0xd7, 0x12, 0xb3, 0x65, 0xf0, 0x29, 0x36, 0xd0, 0x27, 0x77, 0xea,

0xa0, 0xf8, 0xa0, 0xab, 0xe9, 0x26, 0x98, 0x82, 0x5b, 0xa2, 0x41, 0xa7, 0xe5, 0x98, 0x09, 0x04,

第3组样本数据:

密钥: 0x1b, 0x8b, 0x33, 0x77, 0x1d, 0x34, 0xa3, 0x64, 0xd2, 0x89, 0x20, 0x99, 0xb3, 0xd9, 0x0b, 0x1f,

0x43, 0x62, 0x50, 0xea, 0x73, 0x6c, 0x45, 0x30, 0xa3, 0xd3, 0xf8, 0xa4, 0xab, 0x1c, 0xb7, 0x59,

明文: 0xbb, 0x46, 0x09, 0x4e, 0x8d, 0xd7, 0x12, 0xb3, 0x65, 0xf0, 0x29, 0x36, 0xd0, 0x27, 0x77, 0xea,

0xa0, 0xf8, 0xa0, 0xab, 0xe9, 0x26, 0x98, 0x82, 0x5b, 0xa2, 0x41, 0xa7, 0xe5, 0x98, 0x09, 0x04,

密文: 0xac, 0xca, 0x96, 0x73, 0xff, 0xa5, 0xea, 0x6a, 0xbf, 0x4c, 0x32, 0xc7, 0x37, 0x04, 0x49, 0xa0,

0x39, 0x02, 0x48, 0xac, 0xc1, 0x21, 0x61, 0x00, 0x9a, 0x57, 0xca, 0x24, 0xb9, 0x49, 0xa2, 0xe3,

第4组样本数据:

密钥: 0xbb, 0x46, 0x09, 0x4e, 0x8d, 0xd7, 0x12, 0xb3, 0x65, 0xf0, 0x29, 0x36, 0xd0, 0x27, 0x77, 0xea,

0xa0, 0xf8, 0xa0, 0xab, 0xe9, 0x26, 0x98, 0x82, 0x5b, 0xa2, 0x41, 0xa7, 0xe5, 0x98, 0x09, 0x04,

明文: 0xac, 0xca, 0x96, 0x73, 0xff, 0xa5, 0xea, 0x6a, 0xbf, 0x4c, 0x32, 0xc7, 0x37, 0x04, 0x49, 0xa0,

0x39, 0x02, 0x48, 0xac, 0xc1, 0x21, 0x61, 0x00, 0x9a, 0x57, 0xca, 0x24, 0xb9, 0x49, 0xa2, 0xe3,

密文: 0x9c, 0x36, 0x26, 0x31, 0x05, 0x93, 0x38, 0x27, 0xa3, 0x07, 0x9a, 0x69, 0x29, 0xee, 0x13, 0x2f,

0x49, 0xbb, 0xf3, 0x5e, 0x47, 0x52, 0x35, 0xb3, 0x3b, 0x94, 0x31, 0x8d, 0x7f, 0xcd, 0x3d, 0xc6,

第5组样本数据:

密钥: 0xac, 0xca, 0x96, 0x73, 0xff, 0xa5, 0xea, 0x6a, 0xbf, 0x4c, 0x32, 0xc7, 0x37, 0x04, 0x49, 0xa0,

0x39, 0x02, 0x48, 0xac, 0xc1, 0x21, 0x61, 0x00, 0x9a, 0x57, 0xca, 0x24, 0xb9, 0x49, 0xa2, 0xe3,

明文: 0x9c, 0x36, 0x26, 0x31, 0x05, 0x93, 0x38, 0x27, 0xa3, 0x07, 0x9a, 0x69, 0x29, 0xee, 0x13, 0x2f,

0x49, 0xbb, 0xf3, 0x5e, 0x47, 0x52, 0x35, 0xb3, 0x3b, 0x94, 0x31, 0x8d, 0x7f, 0xcd, 0x3d, 0xc6,

密文: 0x4c, 0x1c, 0x1f, 0x69, 0x38, 0x2e, 0x71, 0x37, 0x5e, 0xe0, 0xb1, 0x03, 0x37, 0xae, 0x4d, 0x09,

0xf9, 0xbb, 0xc4, 0x15, 0x0e, 0x05, 0x78, 0x50, 0x4e, 0x83, 0x79, 0xee, 0xcd, 0x5a, 0x11, 0x6b,

8 密码算法设计框图

8.1 加密过程图

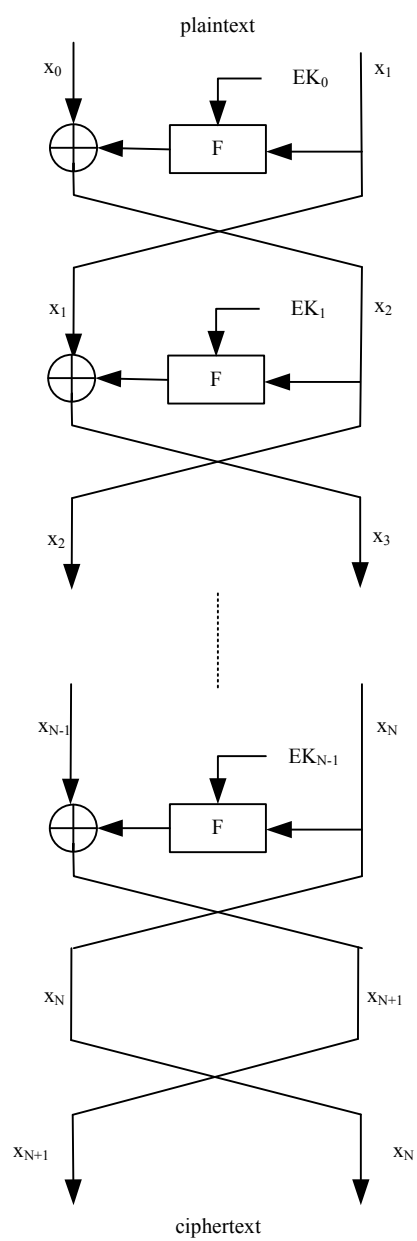


图 1 加密过程图

8.2 解密过程图

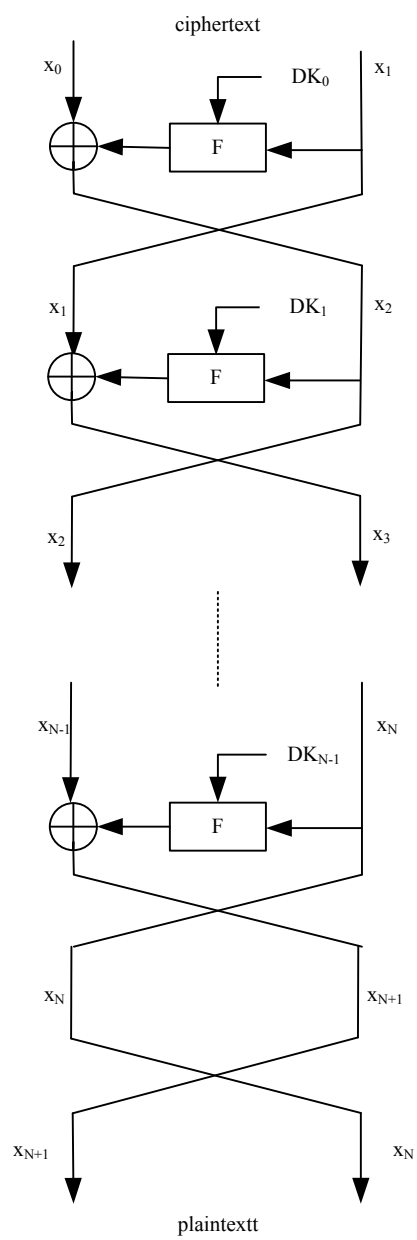


图 2 解密过程图

8.3 SMBA-128 轮函数图

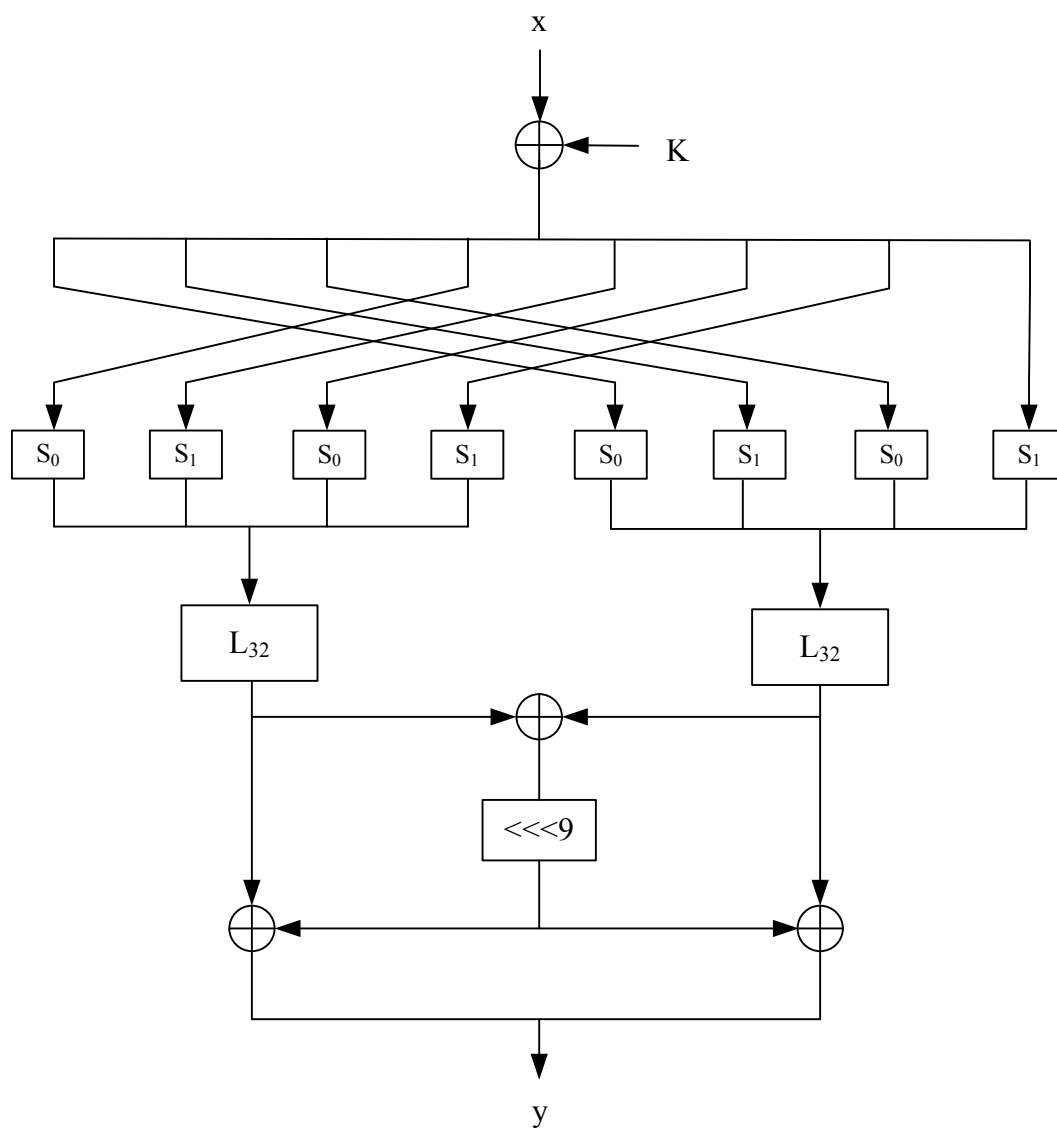


图 3 SMBA-128 轮函数图

8.4 SMBA-256 轮函数图

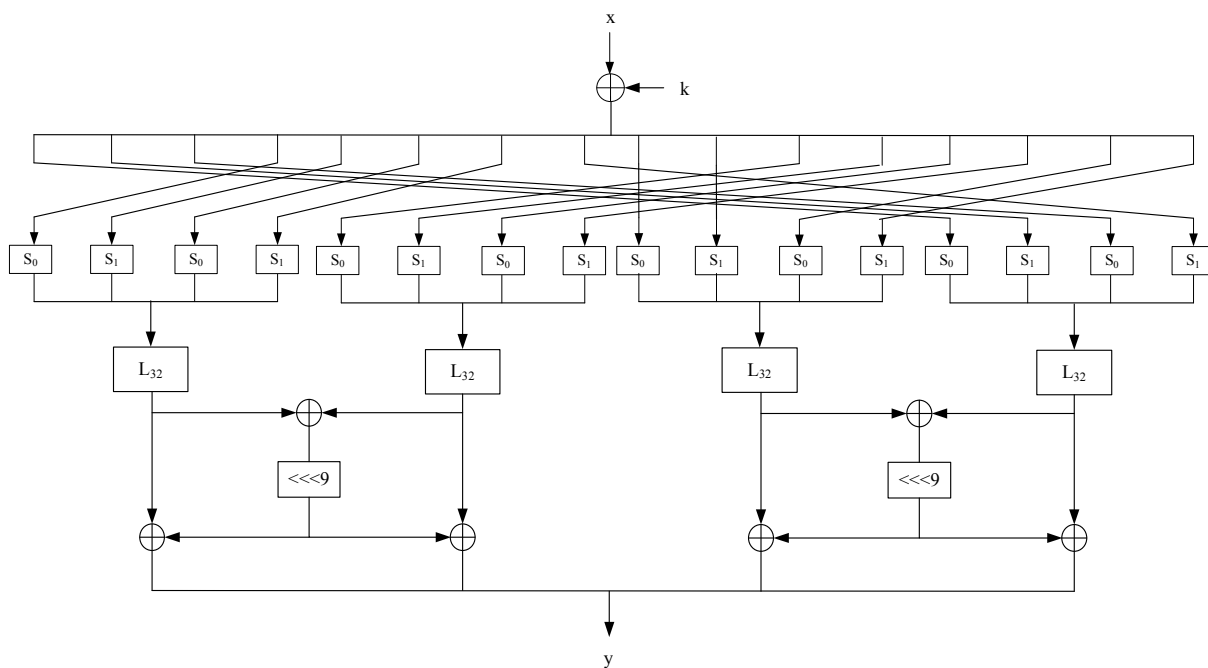


图 4 SMBA-256 轮函数图

8.5 SMBA-128-128 和 SMBA-256-256 密钥扩展轮变换图

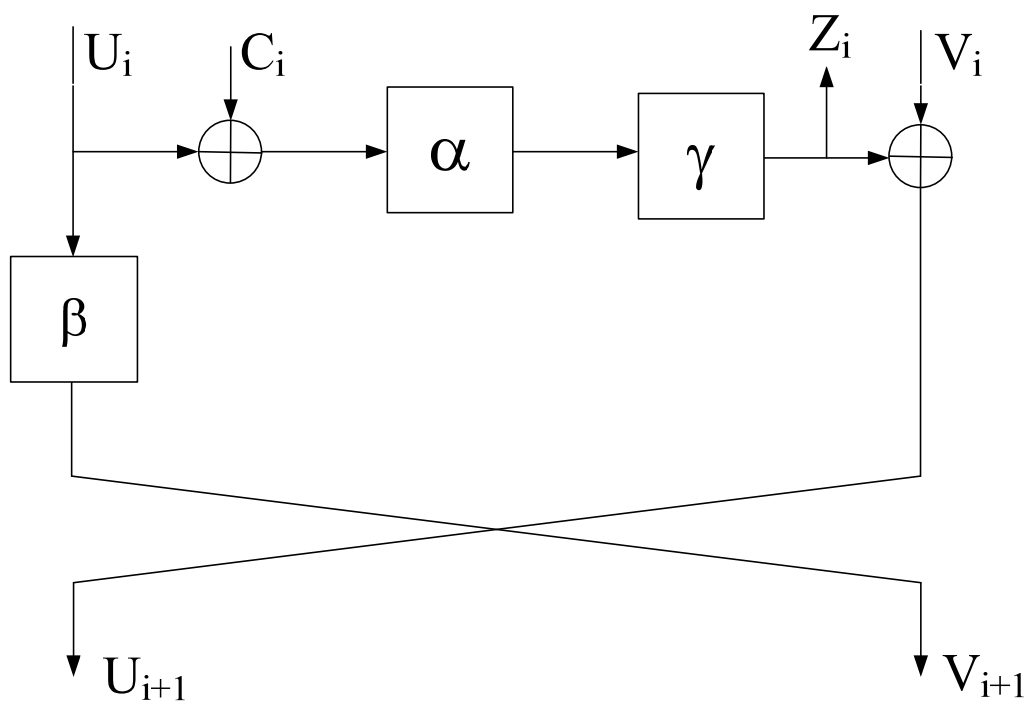


图 5 SMBA-128-128 和 SMBA-256-256 密钥扩展轮变换图

8.6 SMBA-128-256 密钥扩展轮变换图

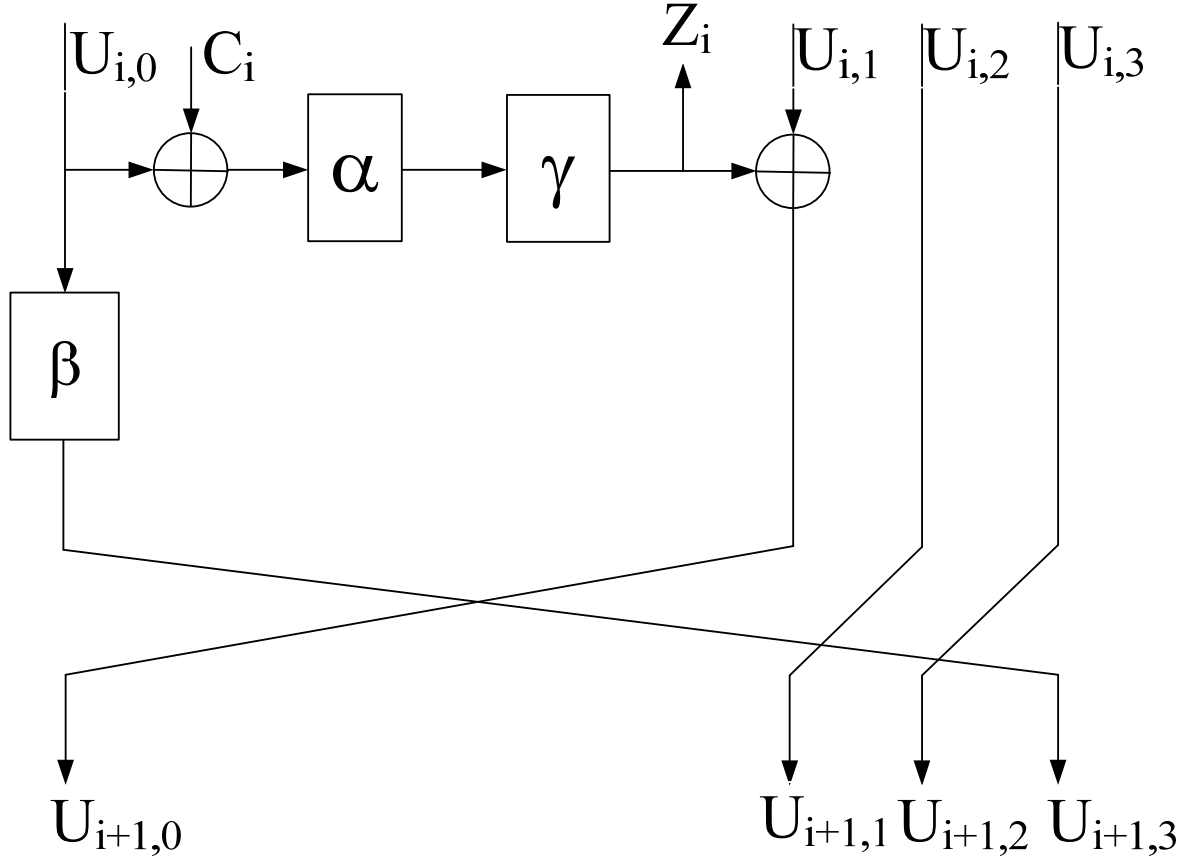


图 6 SMBA-128-256 密钥扩展轮变换图

9 附录 S 盒 S_0 、 S_1 构造过程

9.1 S_0 构造过程

9.1.1 有限域多项式基上的计算

设 Z 是 $GF(2)$ 中的不可约多项式 z^2+z+1 的根，则 $(Z,1)$ 是 $GF(2^2)$ 在 $GF(2)$ 上的一组多项式基。

设 Y 是 $GF(2^2)$ 中的不可约多项式 y^2+y+Z (因为 $\text{tr}_{4/2}(Z)=Z+Z^2=1$) 的根，则 $(Y,1)$ 是 $GF(2^4)$ 在 $GF(2^2)$ 上的一组多项式基， $(ZY,Y,Z,1)$ 是 $GF(2^4)$ 在 $GF(2)$ 上的一组基。

有 $z^2+z+1=(z+Z)(z+Z^2)$, $y^2+y+Z=(y+Y)(y+Y^4)$ 。

$GF(2^2)$ 在基 $(Z,1)$ 上的乘法：

$$\begin{aligned} (b_0Z+b_1)(c_0Z+c_1) &= (b_0c_0+b_0c_1+b_1c_0)Z+(b_0c_0+b_1c_1) \\ &= ((b_0+b_1)(c_0+c_1)+b_1c_1)Z+(b_0c_0+b_1c_1) \end{aligned}$$

GF(2²)在基(Z,1)上的乘法:

$$\begin{aligned} Z(b_0Z+b_1)(c_0Z+c_1) &= ((b_0c_1+b_1c_0)+b_1c_1)Z+(b_0c_0+(b_0c_1+b_1c_0)) \\ &= ((b_0+b_1)(c_0+c_1)+b_0c_0)Z+((b_0+b_1)(c_0+c_1)+b_1c_1) \end{aligned}$$

GF(2⁴)在基(Y,1)上的乘法:

$$(b_0Y+b_1)(c_0Y+c_1) = ((b_0+b_1)(c_0+c_1)+b_1c_1)Y+(Zb_0c_0+b_1c_1)$$

9.1.2 S₀ 构造图示

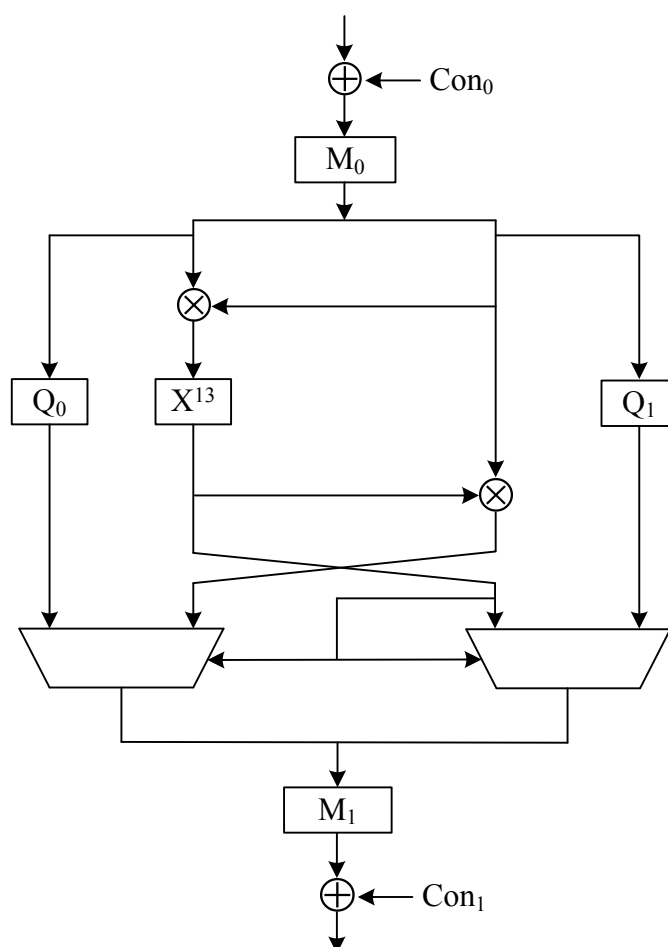


图 7 S₀ 结构图

9.1.3 S₀ 构造过程描述

设 S₀ 的 8 比特输入为 x，则如下计算 S₀ 的 8 比特输出 S₀[x]:

- (1) $a = M_0(x \oplus \text{Con}_0)$ ，将 a 分为 2 个 4 比特 a_0 、 a_1 ，即 $a = a_0 || a_1$;
- (2) $b = a_0 \otimes a_1$;
- (3) $c = b^{13}$;
- (4) $d = c \otimes a_1$;

(5) $e_0 = Q_0[a_0]$, $e_1 = Q_1[a_1]$;

(6) 若 c 为 0, 则 $f = e_0 || e_1$; 否则 $f = d || c$;

(7) $S_0[x] = M_1(f) \oplus \text{Con}_1$

其中, 符号 \otimes 表示 $GF(2^4)$ 在基 $(ZY, Y, Z, 1)$ 上的乘法。 b^{13} 表示 $GF(2^4)$ 在基 $(ZY, Y, Z, 1)$ 上 b 的 13 次幂, 该变换用 16 进制表示为:

0x0, 0x1, 0x2, 0x3, 0x9, 0xb, 0xc, 0xf, 0xe, 0x5, 0xd, 0x4, 0x7, 0x8, 0xa, 0x6.

两个 4 比特 S 盒 Q_0 、 Q_1 分别为:

0x0, 0x8, 0x4, 0x2, 0xa, 0xd, 0x9, 0x1, 0xb, 0x6, 0xf, 0xe, 0x5, 0xc, 0x3, 0x7.

0x0, 0x2, 0x8, 0x1, 0x6, 0x9, 0x7, 0x5, 0x4, 0xa, 0xc, 0xd, 0x3, 0xf, 0xe, 0xb.

两个矩阵 M_0 和 M_1 都是置换矩阵, M_0 的第 i 行第 $\tau_0(i)$ 列为 1, 其余为 0; M_1 的第 i 行第 $\tau_1(i)$ 列为 1, 其余为 0; $0 \leq i \leq 7$ 。 τ_0 和 τ_1 分别为:

0, 4, 2, 6, 5, 3, 1, 7.

1, 4, 7, 0, 2, 5, 6, 3.

两个 8 比特常数分别为: $\text{Con}_0 = 0x52$, $\text{Con}_1 = 0x4d$ 。

上述步骤 (7) 的“若 c 为 0, 则 $f = e_0 || e_1$; 否则 $f = d || c$;”也可如下实现:

设 $c = c_0 || c_1 || c_2 || c_3$, 令 $g = c_0 \vee c_1 \vee c_2 \vee c_3$, $h = g || g || g || g || g || g || g || g$, $f = ((e_0 || e_1) \oplus (d || c)) \& h \oplus (e_0 || e_1)$;

9.2 S_1 构造过程

9.2.1 有限域正规基上的计算

当 $q = 2^n$ 时, (α, α^q) 是 $GF(q^2)$ 在 $GF(q)$ 上的一组正规基, 当且仅当 $\text{tr}_{q^2/q}(\alpha) = \alpha + \alpha^q \neq 0$ 。

设 Z 是 $GF(2)$ 中的不可约多项式 $z^2 + z + 1$ 的根, 则 (Z^2, Z) 是 $GF(2^2)$ 在 $GF(2)$ 上的一组正规基。

设 Y 是 $GF(2^2)$ 中的不可约多项式 $y^2 + y + Z$ (因为 $\text{tr}_{4/2}(Z) = 1$) 的根, 则 (Y^4, Y) 是 $GF(2^4)$ 在 $GF(2^2)$ 上的一组正规基, $(Z^2 Y^4, Z Y^4, Z^2 Y, Z Y)$ 是 $GF(2^4)$ 在 $GF(2)$ 上的一组基。

设 X 是 $GF(2^4)$ 中的不可约多项式 $x^2 + x + Z^2 Y^4$ (因为 $\text{tr}_{16/4}(Z^2 Y^4) = 1$) 的根, 则 (X^{16}, X) 是 $GF(2^8)$ 在 $GF(2^4)$ 上的一组正规基。

有 $z^2 + z + 1 = (z + Z)(z + Z^2)$, $y^2 + y + Z = (y + Y)(y + Y^4)$, $x^2 + x + Z^2 Y^4 = (x + X)(x + X^{16})$ 。约定 $0^{-1} = 0$ 。

$GF(2^2)$ 在基 (Z^2, Z) 上的乘法:

$$(b_0 Z^2 + b_1 Z)(c_0 Z^2 + c_1 Z) = ((b_0 + b_1)(c_0 + c_1) + b_0 c_0) Z^2 + ((b_0 + b_1)(c_0 + c_1) + b_1 c_1) Z$$

GF(2²)在基(Z²,Z)上的乘法:

$$Z(b_0Z^2+b_1Z)(c_0Z^2+c_1Z)=(b_0c_0+ b_1c_1)Z^2+((b_0+b_1)(c_0+c_1)+ b_0c_0)Z$$

GF(2⁴)在基(Y⁴,Y)上的乘法:

$$(b_0Y^4+b_1Y)(c_0Y^4+c_1Y)=(Z(b_0+b_1)(c_0+c_1)+b_0c_0)Y^4+(Z(b_0+b_1)(c_0+c_1)+b_1c_1)Y$$

GF(2⁴)在基(Y⁴,Y)上的乘法:

$$Z^2Y^4(b_0Z^2Y^4+b_1ZY^4+b_2Z^2Y+b_3ZY)^2=b_0Z^2Y^4+(b_0+b_1)ZY^4+(b_1+b_3)Z^2Y+(b_0+b_2)ZY$$

GF(2⁸)在基(X¹⁶,X)上的求逆:

$$(b_0X^{16}+b_1X)^{-1}=(b_0b_1+Z^2Y^4(b_0+b_1)^2)^{-1}(b_1X^{16}+b_0X)$$

9.2.2 S₁ 构造图示

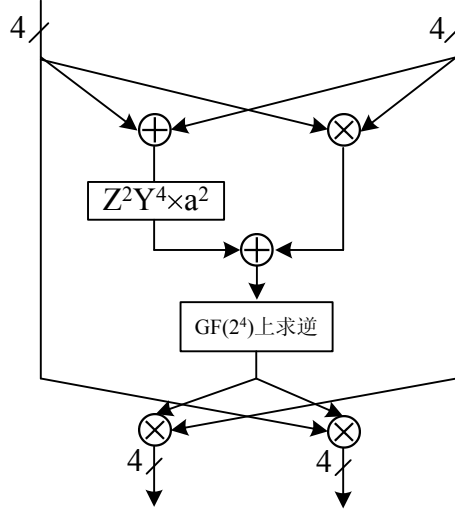


图 8 GF(2⁸)上求逆

9.2.3 S₁ 构造过程描述

设 S₁ 的 8 比特输入为 x，则如下计算 S₁ 的 8 比特输出 S₁[x]:

- (1) $a = M_2(x \oplus \text{Con}_2)$ ，将 a 分为 2 个 4 比特 a_0 、 a_1 ，即 $a = a_0 || a_1$;
- (2) $b_0 = a_0 \oplus a_1$ ， $b_1 = a_0 \otimes a_1$;
- (3) $c = Z^2Y^4 \otimes b_0^2$;
- (4) $d = c \oplus b_1$;
- (5) $e = d^{-1}$;
- (6) $f_0 = e \otimes a_1$ ， $f_1 = e \otimes a_0$;
- (7) $S_1[x] = M_3(f_0 || f_1) \oplus \text{Con}_3$

其中,符号 \otimes 表示 GF(2⁴)在基(Z²Y⁴,ZY⁴,Z²Y,ZY)上的乘法。 d^{-1} 表示 GF(2⁴)在基(Z²Y⁴,ZY⁴,

Z^2Y, ZY)上 d 的逆, 该变换用 16 进制表示为:

0x0,0x4,0xc,0x8,0x1,0xa,0xe,0xd,0x3,0xb,0x5,0x9,0x2,0x7,0x6,0xf.

两个矩阵 M_2 和 M_3 都是置换矩阵, M_2 的第 i 行第 $\tau_2(i)$ 列为 1, 其余为 0; M_3 的第 i 行第 $\tau_3(i)$ 列为 1, 其余为 0; $0 \leq i \leq 7$ 。 τ_2 和 τ_3 分别为:

4,6,5,1,0,7,2,3.

5,1,3,6,2,4,0,7.

两个 8 比特常数分别为: $Con_0=0xa4$, $Con_1=0x5f$ 。