

ICS 35.040
L 80
备案号:36825—2012



中华人民共和国密码行业标准

GM/T 0002—2012

SM4 分组密码算法

SM4 block cipher algorithm

2012-03-21 发布

2012-03-21 实施

国家密码管理局 发布



目 次

前言	Ⅲ
1 范围	1
2 术语和定义	1
3 符号和缩略语	1
4 算法结构	1
5 密钥及密钥参量	2
6 轮函数 F	2
6.1 轮函数结构	2
6.2 合成置换 T	2
7 算法描述	3
7.1 加密算法	3
7.2 解密算法	3
7.3 密钥扩展算法	3
附录 A (资料性附录) 运算示例	4
A.1 示例 1	4
A.2 示例 2	5

前 言

本标准依据 GB/T 1.1—2009 给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准的附录 A 为资料性附录。

本标准由国家密码管理局提出并归口。

本标准起草单位：中国科学院数据与通信保护研究教育中心、国家密码管理局商用密码检测中心。

本标准主要起草人：吕述望、李大为、张超、张众、董芳、毛颖颖、刘振华。

SM4 分组密码算法

1 范围

本标准规定了 SM4 分组密码算法的算法结构和算法描述,并给出了运算示例。
本标准适用于密码应用中使用分组密码的需求。

2 术语和定义

下列术语和定义适用于本文件。

2.1

分组长度 block length

一个信息分组的比特位数。

2.2

密钥长度 key length

密钥的比特位数。

2.3

密钥扩展算法 key expansion algorithm

将密钥变换为轮密钥的运算单元。

2.4

轮数 rounds

轮函数的迭代次数。

2.5

字 word

长度为 32 比特的组(串)。

2.6

S 盒 S-box

S 盒为固定的 8 比特输入 8 比特输出的置换,记为 Sbox(.)。

3 符号和缩略语

下列符号和缩略语适用于本文件:

\oplus 32 位异或

$\lll i$ 32 位循环左移 i 位

4 算法结构

SM4 密码算法是一个分组算法。该算法的分组长度为 128 比特,密钥长度为 128 比特。加密算法与密钥扩展算法都采用 32 轮非线性迭代结构。数据解密和数据加密的算法结构相同,只是轮密钥的使用顺序相反,解密轮密钥是加密轮密钥的逆序。

5 密钥及密钥参量

加密密钥长度为 128 比特,表示为 $MK=(MK_0, MK_1, MK_2, MK_3)$,其中 $MK_i(i=0,1,2,3)$ 为字。

轮密钥表示为 $(rk_0, rk_1, \dots, rk_{31})$,其中 $rk_i(i=0, \dots, 31)$ 为 32 比特字。轮密钥由加密密钥生成。

$FK=(FK_0, FK_1, FK_2, FK_3)$ 为系统参数, $CK=(CK_0, CK_1, \dots, CK_{31})$ 为固定参数,用于密钥扩展算法,其中 $FK_i(i=0, \dots, 3)$ 、 $CK_i(i=0, \dots, 31)$ 为字。

6 轮函数 F

6.1 轮函数结构

设输入为 $(X_0, X_1, X_2, X_3) \in (Z_2^{32})^4$, 轮密钥为 $rk \in Z_2^{32}$, 则轮函数 F 为:

$$F(X_0, X_1, X_2, X_3, rk) = X_0 \oplus T(X_1 \oplus X_2 \oplus X_3 \oplus rk)$$

6.2 合成置换 T

$T: Z_2^{32} \rightarrow Z_2^{32}$ 是一个可逆变换,由非线性变换 τ 和线性变换 L 复合而成,即 $T(\cdot) = L(\tau(\cdot))$ 。

(1) 非线性变换 τ

τ 由 4 个并行的 S 盒构成。

设输入为 $A=(a_0, a_1, a_2, a_3) \in (Z_2^8)^4$, 输出为 $B=(b_0, b_1, b_2, b_3) \in (Z_2^8)^4$, 则

$$(b_0, b_1, b_2, b_3) = \tau(A) = (Sbox(a_0), Sbox(a_1), Sbox(a_2), Sbox(a_3))$$

其中, Sbox 数据如下:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	D6	90	E9	FE	CC	E1	3D	B7	16	B6	14	C2	28	FB	2C	05
1	2B	67	9A	76	2A	BE	04	C3	AA	44	13	26	49	86	06	99
2	9C	42	50	F4	91	EF	98	7A	33	54	0B	43	ED	CF	AC	62
3	E4	B3	1C	A9	C9	08	E8	95	80	DF	94	FA	75	8F	3F	A6
4	47	07	A7	FC	F3	73	17	BA	83	59	3C	19	E6	85	4F	A8
5	68	6B	81	B2	71	64	DA	8B	F8	EB	0F	4B	70	56	9D	35
6	1E	24	0E	5E	63	58	D1	A2	25	22	7C	3B	01	21	78	87
7	D4	00	46	57	9F	D3	27	52	4C	36	02	E7	A0	C4	C8	9E
8	EA	BF	8A	D2	40	C7	38	B5	A3	F7	F2	CE	F9	61	15	A1
9	E0	AE	5D	A4	9B	34	1A	55	AD	93	32	30	F5	8C	B1	E3
A	1D	F6	E2	2E	82	66	CA	60	C0	29	23	AB	0D	53	4E	6F
B	D5	DB	37	45	DE	FD	8E	2F	03	FF	6A	72	6D	6C	5B	51
C	8D	1B	AF	92	BB	DD	BC	7F	11	D9	5C	41	1F	10	5A	D8
D	0A	C1	31	88	A5	CD	7B	BD	2D	74	D0	12	B8	E5	B4	B0
E	89	69	97	4A	0C	96	77	7E	65	B9	F1	09	C5	6E	C6	84
F	18	F0	7D	EC	3A	DC	4D	20	79	EE	5F	3E	D7	CB	39	48

注: 输入 'EF', 则经 S 盒后的值为表中第 E 行和第 F 列的值, $Sbox(EF)=84$ 。

(2) 线性变换 L

非线性变换 τ 的输出是线性变换 L 的输入。设输入为 $B \in Z_2^{32}$, 输出为 $C \in Z_2^{32}$, 则:

$$C = L(B) = B \oplus (B \lll 2) \oplus (B \lll 10) \oplus (B \lll 18) \oplus (B \lll 24)$$

7 算法描述

7.1 加密算法

本加密算法由 32 次迭代运算和 1 次反序变换 R 组成。

设明文输入为 $(X_0, X_1, X_2, X_3) \in (Z_2^{32})^4$, 密文输出为 $(Y_0, Y_1, Y_2, Y_3) \in (Z_2^{32})^4$, 轮密钥为 $rk_i \in Z_2^{32}$, $i=0, 1, 2, \dots, 31$ 。加密算法的运算过程如下:

- (1) 32 次迭代运算: $X_{i+4} = F(X_i, X_{i+1}, X_{i+2}, X_{i+3}, rk_i)$, $i=0, 1, \dots, 31$;
- (2) 反序变换: $(Y_0, Y_1, Y_2, Y_3) = R(X_{32}, X_{33}, X_{34}, X_{35}) = (X_{35}, X_{34}, X_{33}, X_{32})$ 。

7.2 解密算法

本算法的解密变换与加密变换结构相同, 不同的仅是轮密钥的使用顺序。解密时, 使用轮密钥序 $(rk_{31}, rk_{30}, \dots, rk_0)$ 。

7.3 密钥扩展算法

本算法轮密钥由加密密钥通过密钥扩展算法生成。

加密密钥 $MK = (MK_0, MK_1, MK_2, MK_3) \in (Z_2^{32})^4$, 轮密钥生成方法为:

$$(K_0, K_1, K_2, K_3) = (MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_3 \oplus FK_3),$$

$$rk_i = K_{i+4} = K_i \oplus T'(K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i), i=0, 1, \dots, 31。$$

其中:

- (1) T' 是将 6.2 中合成置换 T 的线性变换 L 替换为 L' ;

$$L'(B) = B \oplus (B \lll 13) \oplus (B \lll 23);$$

- (2) 系统参数 FK 的取值为:

$$FK_0 = (A3B1BAC6), FK_1 = (56AA3350), FK_2 = (677D9197), FK_3 = (B27022DC);$$

- (3) 固定参数 CK 取值方法为:

设 $ck_{i,j}$ 为 CK_i 的第 j 字节 ($i=0, 1, \dots, 31; j=0, 1, 2, 3$), 即 $CK_i = (ck_{i,0}, ck_{i,1}, ck_{i,2}, ck_{i,3}) \in (Z_2^8)^4$, 则 $ck_{i,j} = (4i+j) \times 7 \pmod{256}$ 。

固定参数 CK_i ($i=0, 1, \dots, 31$) 具体值为:

00070E15, 1C232A31, 383F464D, 545B6269,
70777E85, 8C939AA1, A8AFB6BD, C4CBD2D9,
E0E7EEF5, FC030A11, 181F262D, 343B4249,
50575E65, 6C737A81, 888F969D, A4ABB2B9,
C0C7CED5, DCE3EAF1, F8FF060D, 141B2229,
30373E45, 4C535A61, 686F767D, 848B9299,
A0A7AEB5, BCC3CAD1, D8DFE6ED, F4FB0209,
10171E25, 2C333A41, 484F565D, 646B7279。

附 录 A
(资料性附录)
运 算 示 例

A.1 示例 1

本部分为 SM4 分组密码算法对一组明文进行加密的运算示例。

输入明文: 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10

输入密钥: 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10

轮密钥与每轮输出状态:

$rk[0]=F12186F9$ $X[4]=27FAD345$
 $rk[1]=41662B61$ $X[5]=A18B4CB2$
 $rk[2]=5A6AB19A$ $X[6]=11C1E22A$
 $rk[3]=7BA92077$ $X[7]=CC13E2EE$
 $rk[4]=367360F4$ $X[8]=F87C5BD5$
 $rk[5]=776A0C61$ $X[9]=33220757$
 $rk[6]=B6BB89B3$ $X[10]=77F4C297$
 $rk[7]=24763151$ $X[11]=7A96F2EB$
 $rk[8]=A520307C$ $X[12]=27DAC07F$
 $rk[9]=B7584DBD$ $X[13]=42DD0F19$
 $rk[10]=C30753ED$ $X[14]=B8A5DA02$
 $rk[11]=7EE55B57$ $X[15]=907127FA$
 $rk[12]=6988608C$ $X[16]=8B952B83$
 $rk[13]=30D895B7$ $X[17]=D42B7C59$
 $rk[14]=44BA14AF$ $X[18]=2FFC5831$
 $rk[15]=104495A1$ $X[19]=F69E6888$
 $rk[16]=D120B428$ $X[20]=AF2432C4$
 $rk[17]=73B55FA3$ $X[21]=ED1EC85E$
 $rk[18]=CC874966$ $X[22]=55A3BA22$
 $rk[19]=92244439$ $X[23]=124B18AA$
 $rk[20]=E89E641F$ $X[24]=6AE7725F$
 $rk[21]=98CA015A$ $X[25]=F4CBA1F9$
 $rk[22]=C7159060$ $X[26]=1DCDFA10$
 $rk[23]=99E1FD2E$ $X[27]=2FF60603$
 $rk[24]=B79BD80C$ $X[28]=EFF24FDC$
 $rk[25]=1D2115B0$ $X[29]=6FE46B75$
 $rk[26]=0E228AEB$ $X[30]=893450AD$
 $rk[27]=F1780C81$ $X[31]=7B938F4C$
 $rk[28]=428D3654$ $X[32]=536E4246$
 $rk[29]=62293496$ $X[33]=86B3E94F$
 $rk[30]=01CF72E5$ $X[34]=D206965E$

$rk[31]=9124A012$ $X[35]=681EDF34$

输出密文: 68 1E DF 34 D2 06 96 5E 86 B3 E9 4F 53 6E 42 46

A.2 示例 2

本部分为 SM4 分组密码算法使用固定的加密密钥,对一组明文反复加密 1 000 000 次的运算示例。

输入明文: 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10

输入密钥: 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10

输出密文: 59 52 98 C7 C6 FD 27 1F 04 02 F8 04 C3 3D 3F 66
