

# SM4

一种分组密码算法。原名 SMS4，是中国国家密码管理局于 2012 年 3 月 21 日公布的商用密码标准，用于无线局域网产品的安全加密。

算法的分组长度为 128 比特，密钥长度为 128 比特。其加密算法和密钥扩展算法都采用 32 轮的非线性迭代结构，解密算法与加密算法的结构完全相同，只是轮子密钥的使用顺序相反。

## 1.1 符号及术语

- ①  $F_2^e$  表示  $e$  比特二进制串，则  $F_2^{32}$  中的元素表示字， $F_2^8$  中的元素表示字节；
- ② S 盒为  $F_2^8 \rightarrow F_2^8$  上的置换；
- ③  $X_i$ 、 $Y_i$  分别表示 32 比特的字；
- ④  $MK$ ：种子密钥，表示算法的 128 比特的初始密钥；
- ⑤  $rk_i$ ：第  $i$  轮 32 比特的轮子密钥；
- ⑥  $\oplus$ ：按位异或操作；
- ⑦  $\lll$ ：循环左移操作；
- ⑧  $F$  表示轮加密函数。

## 1.2 加密算法

SM4 以字为单位进行加密处理。记 128 比特的明文为  $(X_0, X_1, X_2, X_3)$ ，密文为  $(Y_0, Y_1, Y_2, Y_3)$ ，图 1 所示为加密算法结构。加密算法先将轮函数  $F$  迭代 32 次，再通过置换操作  $R$  得到密文。设第  $i$  次轮函数的输入为  $(X_i, X_{i+1}, X_{i+2}, X_{i+3})$ ，使用的轮子密钥为  $rk_i$ ，输出为  $X_{i+4}$ ，则一轮加密变换可以表示为：

$$X_{i+4} = F(X_i, X_{i+1}, X_{i+2}, X_{i+3}, rk_i) \quad (1)$$

轮函数  $F$  的结构如图 1 右侧所示，具体操作步骤如下：

- ① 输入  $X_{i+1}$ 、 $X_{i+2}$  和  $X_{i+3}$  进行异或操作得到 32 比特输出，设为  $t$ ；
- ②  $t$  与轮子密钥  $rk_i$  异或；
- ③ 经过非线性变换  $\tau$ ：32 比特输入分成 4 个字节，每个字节分别经过 S 盒，最终得到 32 比特输出；其中 S 盒真值表如表 1 所示；
- ④ 经过线性变换  $L$ ：设 32 比特输入为  $I$ ，输出为  $O$ ，则
$$O = L(I) = I \oplus (I \lll 2) \oplus (I \lll 10) \oplus (I \lll 18) \oplus (I \lll 24) \quad (2)$$
- ⑤  $O$  与输入  $X_i$  进行异或操作得到一轮的输出  $X_{i+4}$ 。

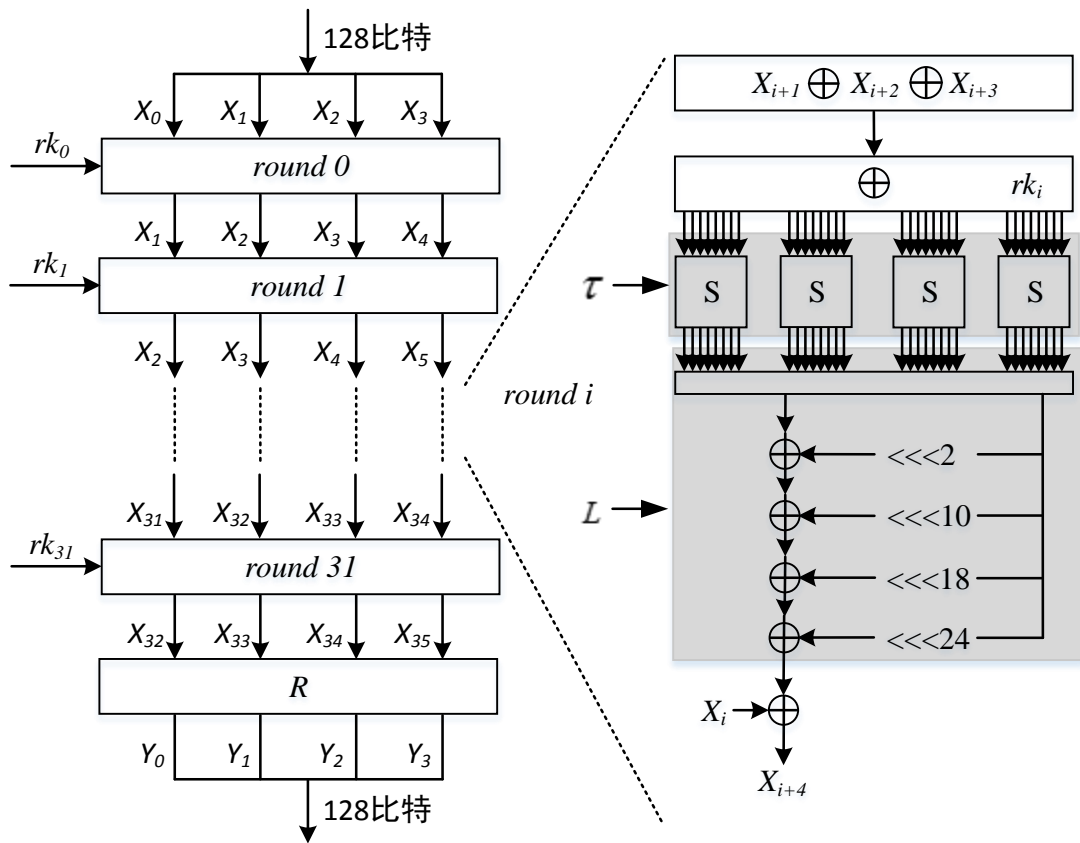


图1 加密算法

表1 S 盒真值表

|   | 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 | 20 | 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 |

\* 表中数据均是十六进制格式

\*\* 输入高 4 比特对应行坐标，低 4 比特对应列坐标，例如输入为 EF，输出则为 84

轮函数一共迭代 32 次，每次迭代得到一个 32 比特输出。最后四轮的输出分别为  $X_{32}$ 、 $X_{33}$ 、 $X_{34}$  和  $X_{35}$ ，再经过一个  $R$  变换后得到密文，其中：

$$\begin{aligned} R(X_{32}, X_{33}, X_{34}, X_{35}) &= (X_{35}, X_{34}, X_{33}, X_{32}) \\ &= (Y_0, Y_1, Y_2, Y_3) \end{aligned} \tag{3}$$

### 1.3 解密算法

解密算法与加密算法结构完全一致，密文  $(Y_0, Y_1, Y_2, Y_3)$  首先经过 32 次轮函数  $F$ ，再经过置换  $R$  即可得到明文，只是轮子密钥的使用顺序与加密算法相反，依次是  $rk_{31}$ 、 $rk_{30}$ 、 $\cdots$ 、 $rk_1$ 、 $rk_0$ 。由于解密算法和解密算法结构完全相同，因此可以复用加密算法的电路，在硬件实现中可以大大地减少电路面积。

### 1.4 密钥扩展算法

设 128 比特的种子密钥  $MK = (MK_0, MK_1, MK_2, MK_3)$ ，其中  $MK_0$ 、 $MK_1$ 、 $MK_2$ 、 $MK_3$  均为 32 比特的字。密钥扩展算法的流程如下：

- ①  $MK_0$ 、 $MK_1$ 、 $MK_2$ 、 $MK_3$  分别与系统参数  $FK_0$ 、 $FK_1$ 、 $FK_2$ 、 $FK_3$  进行异或操作，设得到的值分别为  $k_0$ 、 $k_1$ 、 $k_2$ 、 $k_3$ 。系统参数的取值如表 2 所示。

| 表2 SM4 算法系统参数 |            |
|---------------|------------|
| 参数            | 取值         |
| $FK_0$        | 0xA3B1BAC6 |
| $FK_1$        | 0x56AA3350 |
| $FK_2$        | 0x677D9197 |
| $FK_3$        | 0xB27022DC |

- ② 循环执行如下操作：

$$rk_i = k_{i+4} = k_i \oplus T'(k_{i+1} \oplus k_{i+2} \oplus k_{i+3} \oplus CK_i) \tag{4}$$

其中  $0 \leq i \leq 31$ ， $CK_i$  为 32 比特的轮常量，其取值如表 3 所示。每循环一次得到一个轮子密钥。

| 表3 SM4 算法轮常量 |            |            |            |
|--------------|------------|------------|------------|
| 0x00070E15   | 0x1C232A31 | 0x383F464D | 0x545B6269 |
| 0x70777E85   | 0x8C939AA1 | 0xA8AFB6BD | 0xC4CBD2D9 |
| 0xE0E7EEF5   | 0xFC030A11 | 0x181F262D | 0x343B4249 |

|            |            |            |            |
|------------|------------|------------|------------|
| 0x50575E65 | 0x6C737A81 | 0x888F969D | 0xA4ABB2B9 |
| 0xC0C7CED5 | 0xDCE3EAF1 | 0xF8FF060D | 0x141B2229 |
| 0x30373E45 | 0x4C535A61 | 0x686F767D | 0x848B9299 |
| 0xA0A7AEB5 | 0xBCC3CAD1 | 0xD8DFE6ED | 0xF4FB0209 |
| 0x10171E25 | 0x2C333A41 | 0x484F565D | 0x646B7279 |

\* 第 0 轮常量是 0x00070E15，第 1 轮常量是 0x1C232A31，依此类推

其中  $T'$  包括非线性变换和线性变换两部分，非线性变换与加密算法中的  $\tau$  一致，32 比特输入分成 4 个字节分别经过 S 盒；线性变换与加密算法的不同，其中：

$$L'(I) = I \oplus (I \lll 13) \oplus (I \lll 23) \quad (5)$$

# DES

- 中文名称：数据加密标准
- 拼    音：shuju jiami biao zhun
- 英文名称：DES

数据加密标准 DES(Data Encryption Standard)是一种分组 Feistel 结构的密码算法，其分组长度为 64 比特，密钥长度为 56 比特。DES 的加密算法先经过初始置换  $IP$ ，再将轮函数  $F$  迭代 16 轮，最后经过一次互换及初始逆置换  $IP^{-1}$  操作得到密文。轮函数则由扩展置换  $E$ 、异或操作、 $F_2^6 \rightarrow F_2^4$  的  $S$  盒以及置换  $P$  组成。

DES 的发展最早要上溯到二十世纪六十年代后期，IBM 成立了一个项目用于研究计算机中的加密技术，该项目由 Horst Feistel 带领。至 1971 项目结束时，成功设计了 Lucifer 算法，其分组长度为 64 比特，密钥长度为 128 比特。1973 年 3 月 15 日，美国国家标准局(NBS，即现在的美国国家标准和技术协会 NIST)发表申明公开征集数据加密标准，用于计算机网络通信中数据的安全加密。但最初提交的算法均不符合要求，因此美国国家标准局于次年 8 月 27 日开始重新征集算法，此次 IBM 提交一个基于 Lucifer 修改的算法。1976 年，美国国家标准局在与美国国家安全局(NSA)讨论后，对提交的算法进行了修改，将密钥长度削减到 56 比特，同时对  $S$  盒的设计做了修改，使其抵抗差分分析(此时分析方法尚未公布)的能力更强，但抗穷搜能力下降，并将修改后的算法定为数据加密标准。1977 年 1 月 15 日，DES 被正式批准为美国联邦信息处理标准 FIPS-46。

之后十多年的时间内，DES 在工业界得到了非常广泛的应用，学术界也有很多学者开始对其安全性进行研究。由于 NSA 将其密钥长度降低到 56 比特，同时对  $S$  盒设计的相关细节并未披露，DES 的安全性一直饱受怀疑。1991 年，以色列密码学家 Bihan 和 Shamir 提出的差分分析方法能从理论上破解 DES；1992 年，日本密码学家 Matsui 提出线性分析，并利用该方法于 1994 首次通过实验的方法分析 DES；这些使得 DES 的安全性受到很大影响。另外，二十世纪八九十年代计算机的飞速发展使得穷搜密钥成为可能，1977 年 Verser 等人利用互联网首次成功地穷搜 DES 密钥。因此美国国家标准和技术协会于 1997 年宣布开始征集高级加密标准 AES(Advanced Encryption Standard)。现如今，DES 已逐步被 AES 取代，但其对于现代密码学的发展仍有着举足轻重的意义。