# 现代密码学

## SM3密码杂凑算法

信息与软件工程学院

# SM3密码杂凑算法

- SM3是中国国家密码管理局颁布的中国商用密码标准算法，它是一类密码杂凑函数，可用于数字签名及验证、消息认证码生成及验证、随机数生成。
- 标准起草人：王小云、李峥、于红波、张超、罗鹏、吕述望
- 2012年3月，成为中国商用密码标准（GM/T 0004-2012）
- 2016年8月，成为中国国家密码标准（GB/T 32905-2016）
- 2018年11月22日，含有我国SM3杂凑密码算法的ISO/IEC 10118 – 3 : 2018《信息安全技术杂凑函数第3部分:专用杂凑函数》最新一版(第4版)由国际标准化组织(ISO)发布，SM3算法正式成为国际标准。

# SM3密码杂凑算法的描述

算法的输入数据长度为*l*比特，$l < 2^{64}$ 输出哈希值长度为256比特。

**1.** 常数与函数

**(1)** 常数

初始值

$$IV = 7380166F4914B2B9$$
$$172442D7DA8A0600$$
$$A96F30BC163138AA$$
$$E38DEE4DB0FB0E4E$$

常量

$$T_j = \begin{cases} 79CC4519, & 0 \le j \le 15 \\ 7A879D8A, & 16 \le j \le 63 \end{cases}$$

# SM3密码杂凑算法的描述

**(2) 函数**

布尔函数：

式中 $X,Y,Z$ 为**32**位字，$\wedge,\vee,-,\oplus$ 分别是逻辑与、逻辑或、逻辑非和逐比特异或运算

$$FF_j(X,Y,Z)=\begin{cases}X\oplus Y\oplus Z, & 0\le j\le 15\\ (X\wedge Y)\vee(X\wedge Z)\vee(Y\wedge Z), & 16\le j\le 63\end{cases}$$

$$GG_j(X,Y,Z)=\begin{cases}X\oplus Y\oplus Z, & 0\le j\le 15\\ (X\wedge Y)\vee(\bar{X}\wedge Z), & 16\le j\le 63\end{cases}$$

混淆

置换函数：

$$P_0(X)=X\oplus(X<<<9)\oplus(X<<<17);$$
$$P_1(X)=X\oplus(X<<<15)\oplus(X<<<23).$$

扩散

式中 $X$ 为**32**位字，符号 $a<<<n$ 表示把 $a$ 循环左移 $n$ 位。

# SM3密码杂凑算法的描述

2. 算法描述

算法对数据首先进行填充，再进行迭代压缩后生成哈希值。

(1) 填充并附加消息的长度

- 对消息填充的目的是使填充后的数据长度为512的整数倍。
- 设消息$m$的长度为$l$比特。首先将比特“1”添加到m的末尾，再添加$k$个“0”，其中$k$满足

$$l+1+k=448 \bmod 512$$

- 然后再添加一个64位比特串，该比特串是长度$l$的二进制表示。

举例：消息为01100001011000100110001 1，其长度为$l$=24，填充后的比特串为

423个0　　64比特

$$01100001011000100110001\underbrace{1\,00\cdots00}_{423\text{个}0}\underbrace{00\cdots011000}_{64\text{比特}}$$

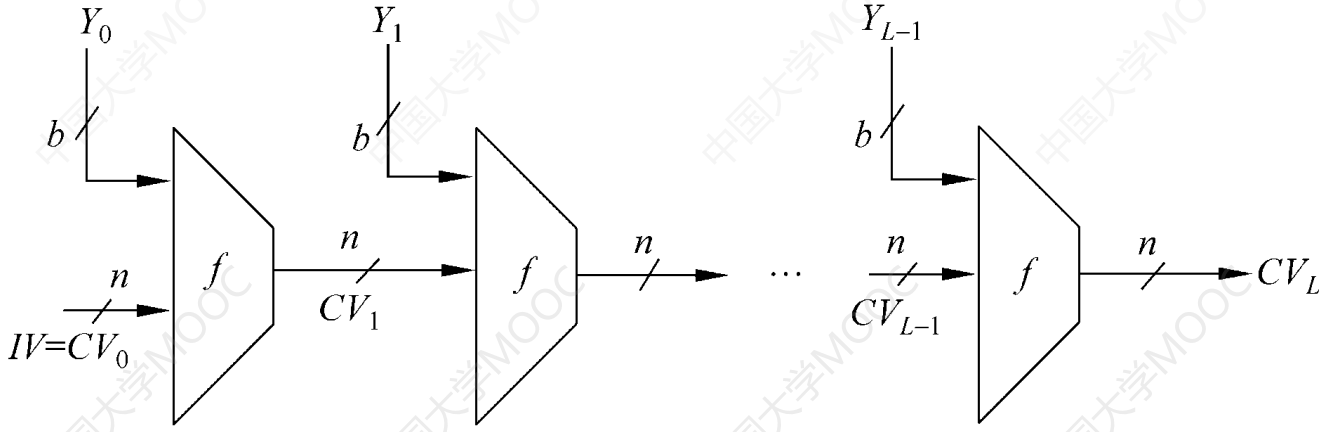
# SM3密码杂凑算法的描述

(2) 迭代压缩

将填充后的消息 $m'$ 按512比特进行分组得 $m' = B^{(0)}B^{(1)}\cdots B^{(L-1)}$ 对 $m'$按下列方式迭代压缩：

$$\text{FOR } i=0 \text{ to } L-1 \quad V^{(i+1)} = CF(V^{(i)}, B^{(i)})$$

其中 $CF$ 是压缩函数，$V^{(0)}$ 为 256 比特初始值 IV ，$B^{(i)}$ 为填充后的消息分组，迭代压缩的结果为 $V^{(L)}$，$V^{(L)}$ 即为消息 $m$ 的哈希值。



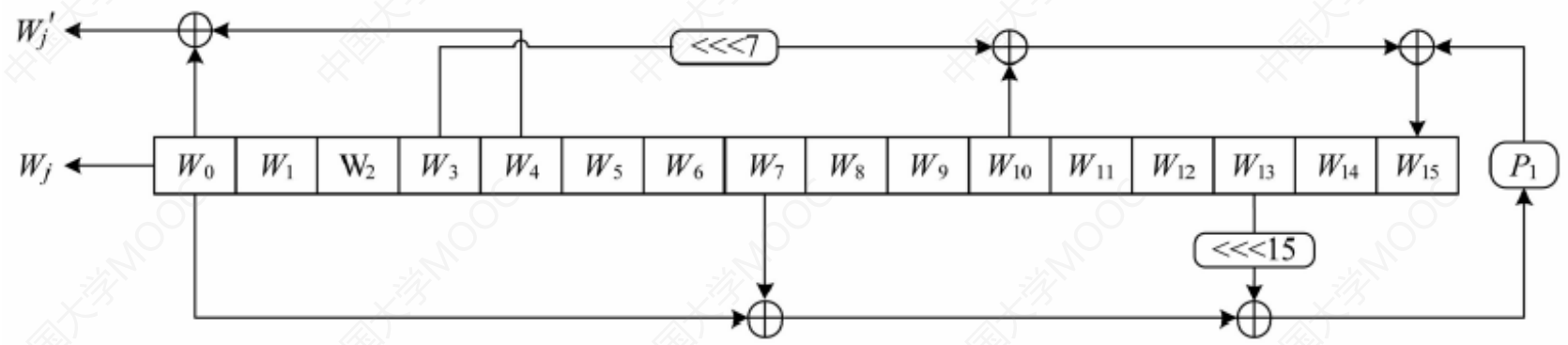



图 1　SM3 消息扩展过程

，步骤如下：

① 消息分组$B^{(i)}$划分为**16**个字$W_0, W_1, \cdots, W_{15}$。

② FOR $j$=16 to 67

$$W_j = P_1\left(W_{j-16} \oplus W_{j-9} \oplus \left(W_{j-3} <<< 15\right)\right) \oplus \left(W_{j-13} <<< 7\right) \oplus W_{j-6}$$

③ FOR $j$=0 to 63

$$W_j' = W_j \oplus W_{j+4}$$

$B^{(i)}$ 经消息扩展后得到**132**个字 $W_0, W_1, \cdots, W_{67}, W_0', W_1', \cdots, W_{63}'$ 。

# SM3密码杂凑算法的描述

(4) 压缩函数

设 $A,B,C,D,E,F,G,H$ 为字寄存器，$SS_1,SS_2,TT_1,TT_2$ 为中间变量，压缩函数 $V^{(i+1)}=CF(V^{(i)},B^{(i)})\ (0\le i\le n-1)$ 的计算过程如下：

$ABCDEFGH=V^{(i)}$

FOR $j$=0 to 63

$SS_1\leftarrow\left((A<<<12)+E+\left(T_j<<<j\right)\right)<<<7;$

$SS_2=SS_1\oplus(A<<<12);$

$TT_1=FF_j(A,B,C)+D+SS_2+W_j';$

$TT_2=GG_j(E,F,G)+H+SS_1+W_j;$

$D=C;$

$C=B<<<9;$

$B=A;$

$A=TT_1;$

$H=G;$

$G=F<<<19;$

$F=E;$

$E=P_0(TT_2)$

ENDFOR

$V^{(i+1)}=ABCDEFGH\oplus V^{(i)}$

其中 + 为模 $2^{32}$ 加运算，字的存储为大端格式

# SM3密码杂

$ABCDEFGH = V^{(i)}$

FOR $j$=0 to 63

$SS_1 \leftarrow \left((A <<< 12) + E + \left(T_j <<< j\right)\right) <<< 7;$

$SS_2 = SS_1 \oplus (A <<< 12);$

$TT_1 = FF_j(A,B,C) + D + SS_2 + W'_j;$

$TT_2 = GG_j(E,F,G) + H + SS_1 + W_j;$

$D = C;$

$C = B <<< 9;$

$B = A;$

$A = TT_1;$

$H = G;$

$G = F <<< 19;$

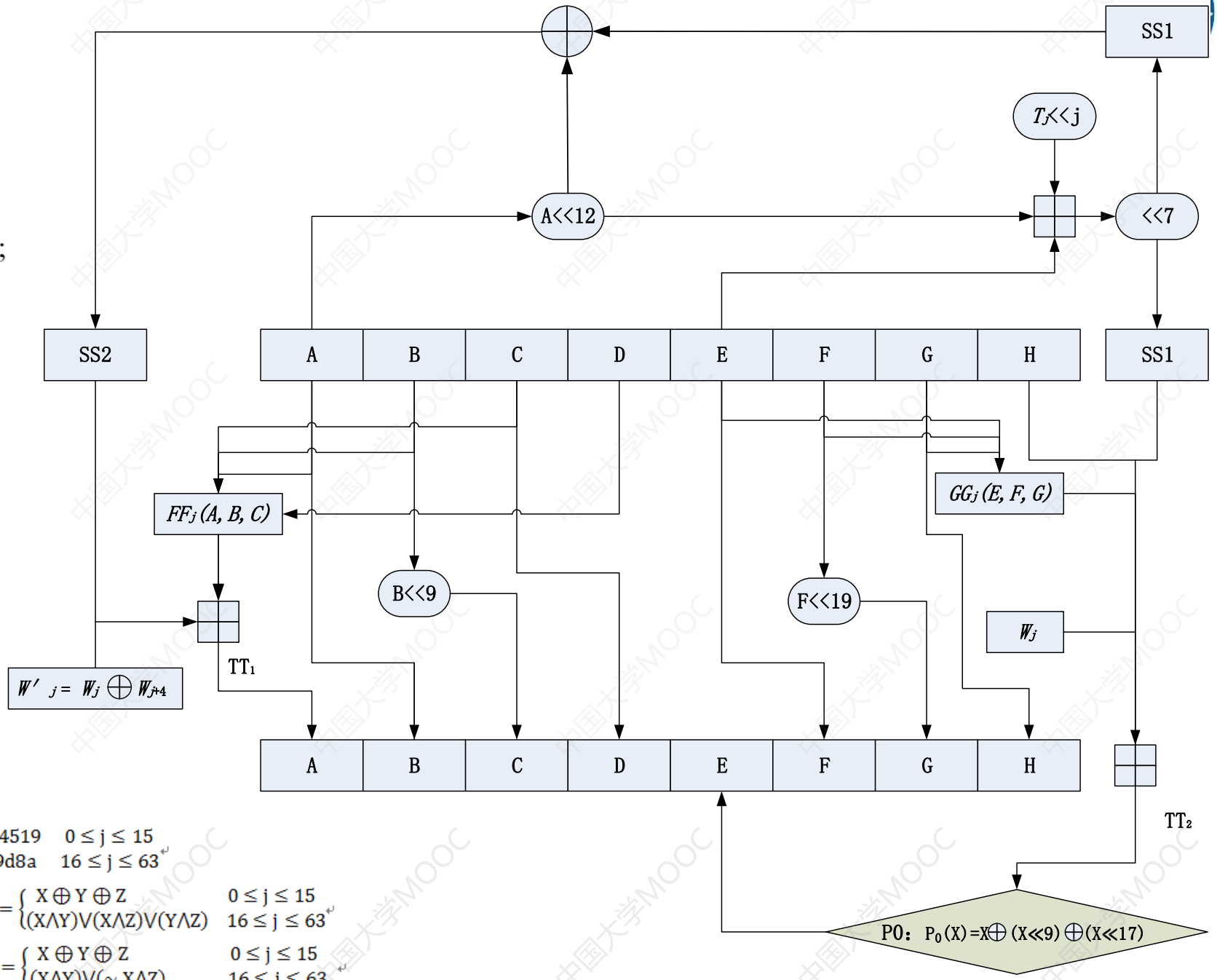
$F = E;$

$E = P_0(TT_2)$

$ENDFOR$

$V^{(i+1)} = ABCDEFGH \oplus V^{(i)}$

SM3算法第j轮迭代压缩流程图




其中：

$$Tj = \begin{cases} 79cc4519 & 0 \le j \le 15 \\ 7a879d8a & 16 \le j \le 63 \end{cases}$$

$$FF_j(X,Y,Z) = \begin{cases} X \oplus Y \oplus Z & 0 \le j \le 15 \\ (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z) & 16 \le j \le 63 \end{cases}$$

$$GG_j(X,Y,Z) = \begin{cases} X \oplus Y \oplus Z & 0 \le j \le 15 \\ (X \wedge Y) \vee (\sim X \wedge Z) & 16 \le j \le 63 \end{cases}$$

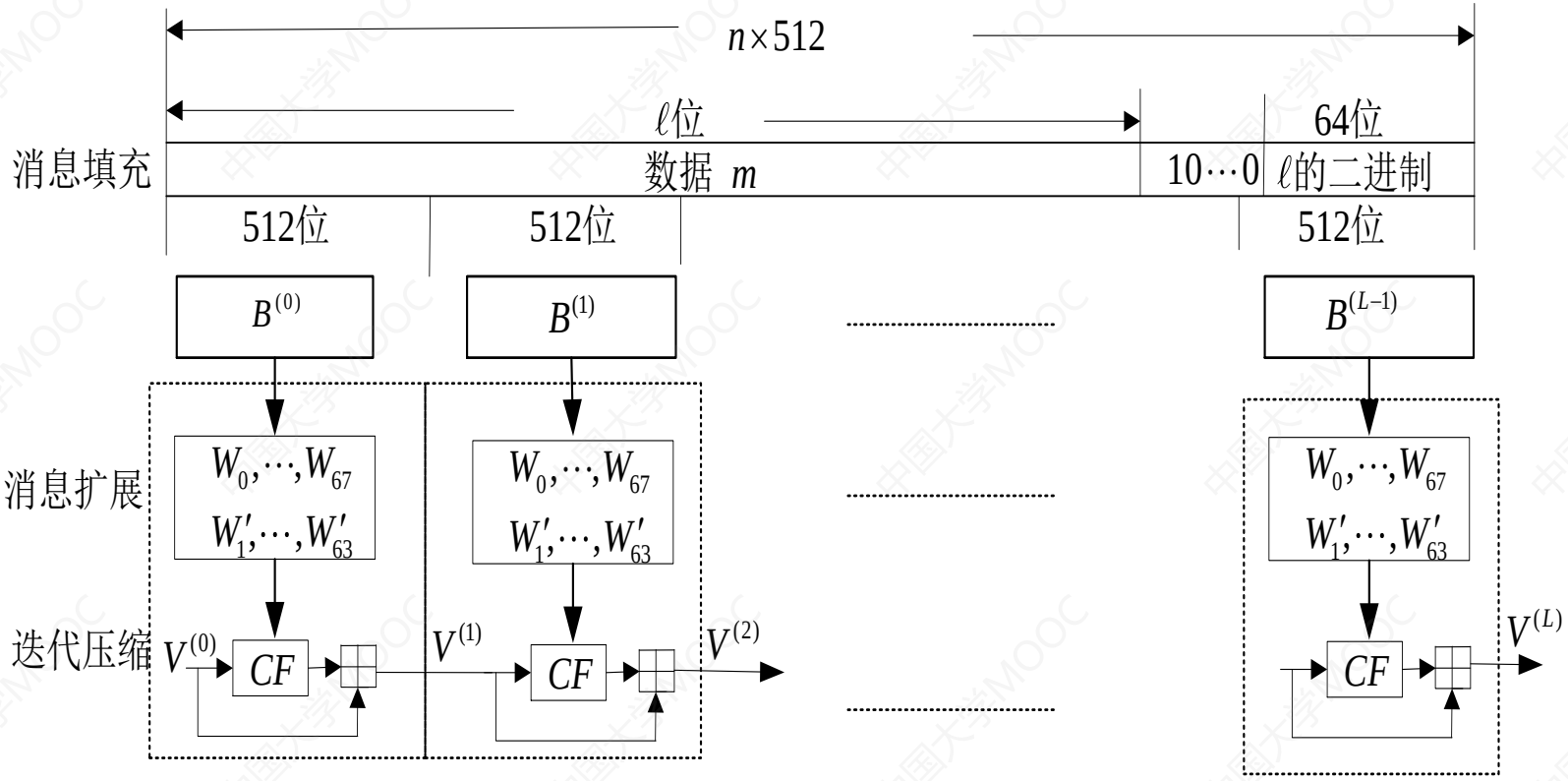
# SM3密码杂凑算法的描述

(5) 输出哈希值

$$ABCDEFGH = V^{(L)}$$

输出 256 比特的哈希值 $y = ABCDEFGH$。

# SM3密码杂凑算法的描述




**SM3** 产生消息哈希值的处理过程

# SM3哈希算法的安全性

- 压缩函数是哈希函数安全的关键
  - SM3的压缩函数 $CF$ 中的布尔函数 $FF_j(X,Y,Z)$ 和 $GG_j(X,Y,Z)$ 是非线性函数，经过循环迭代后提供混淆作用
  - 置换函数 $P_0(X)$ 和 $P_1(X)$ 是线性函数，经过循环迭代后提供扩散作用。
  - 再加上 $CF$ 中的其他运算的共同作用，压缩函数 $CF$ 具有很高的安全性，从而确保SM3具有很高的安全性。

# 算法比较

王小云, 于红波. SM3密码杂凑算法[J]. 信息安全研究, 2016(11).

表 2　SM3 密码杂凑算法和其他标准的 ASIC 实现

| 算法 | 面积 (gates) | 时钟 /MHz | 吞吐量 /Mbps | 吞吐量面积比 /(Kbps/gate) |
|---|---|---|---|---|
| SM3[10] | 11 068 | 216.00 | 1 619 | 146.28 |
| SHA-256[11] | 15 400 | 189.75 | 1 349 | 87.60 |
| SHA-512[11] | 30 747 | 169.20 | 1 969 | 64.04 |
| Whirlpool[11] | 38 911 | 101.94 | 2 485 | 63.86 |
| SHA-3[12] | 56 320 | 487.80 | 21 229 | 376.94 |

表 5　SM3 密码杂凑算法和其他杂凑标准的最好分析结果

| 算法 | 攻击类型 | 步(轮)数 | 百分比/% | 文献 |
|---|---|---|---|---|
| SM3 | 碰撞攻击 | 20 | 31 | [18] |
| | 原像攻击 | 30 | 47 | [24-25] |
| | 区分器攻击 | 37 | 58 | [27] |
| SHA-1 | 碰撞攻击 | 80 | 100 | [4,28-29] |
| | 原像攻击 | 62 | 77.5 | [30] |
| RIPEMD-128 | 碰撞攻击 | 40 | 62.5 | [31] |
| | 原像攻击 | 36 | 56.25 | [32] |
| | 区分器攻击 | 64 | 100 | [33] |
| RIPEMD-160 | 原像攻击 | 34 | 53.12 | [34] |
| | 区分器攻击 | 51 | 79.68 | [35] |
| SHA-256 | 碰撞攻击 | 31 | 48.4 | [36] |
| | 原像攻击 | 45 | 70.3 | [23] |
| | 区分器攻击 | 47 | 73.4 | [37] |
| Whirlpool | 碰撞攻击 | 8 | 80 | [38] |
| | 原像攻击 | 6 | 60 | [38] |
| | 区分器攻击 | 10 | 100 | [39] |
| Stribog | 碰撞攻击 | 7.5 | 62.5 | [40] |
| | 原像攻击 | 6 | 50 | [41] |
| KECCAK-256 | 碰撞攻击 | 5 | 20.8 | [42] |
| | 原像攻击 | 2 | 8 | [43] |
| | 区分器攻击 | 24 | 100 | [44] |
| KECCAK-512 | 碰撞攻击 | 3 | 12.5 | [42] |
| | 区分器攻击 | 24 | 100 | [44] |

# 感 谢 聆 听!

xynie@uestc.edu.cn