

SM3 设计及实现

SM3 算法步骤

1. 填充过程

设消息 M 的长度为 L bit, 首先将 $\text{bit}'1$ 添加到消息末尾, 再加 k 个“0”, k 是满足 $L+1+k=448 \bmod 512$ 的最小非负整数。然后再添加一个 64bit 串, 该串是 L 的二进制表示, 填充后的消息 M' 长度为 512 的整数倍。

例如: 对消息 01100001 01100010 01100011, 其长度 $L=24$, 经填充得到比特串: 01100001 01100010 01100011 1 00...00(423 比特) 00...011000(64 比特 L 的二进制表示)

2. 迭代压缩过程

(1) 迭代过程:

M' 按照 512bit 进行分组: $M'=B(0)B(1)...B(n-1)$

$n=(l+k+65)/512$.

迭代过程如下:

FOR $i=0$ to $n-1$

$V(i+1) = CF(V(i), B(i))$

ENDFOR

CF 为压缩函数, $V(0)$ 为初始值 IV , 迭代压缩的结果为 $V(n)$

(2) 消息扩展:

消息分组 $B(i)$ 扩展生成 132 个字 $W_0, W_1, \dots, W_{67}, W'_0, W'_1, \dots, W'_{63}$ 。

a) 消息分组 $B(i)$ 划分为 16 个字 W_0, W_1, \dots, W_{15} .

b) FOR $j=16$ to 67

$W_j = P_1(W_{j-16} \wedge W_{j-9} \wedge (W_{j-3} \lll 15) \wedge (W_{j-13} \lll 7) \wedge W_{j-6})$

ENDFOR

c) FOR $j=0$ to 63

$W'_j = W_j \wedge W_{j+4}$

ENDFOR

(3) 压缩函数:

$A/B/C/D/E/F/G/H$ 为字寄存器, $SS1/SS2/TT1/TT2$ 为中间变量, 压缩函数 $V(i+1)$

$= CF(V(i), B(i))$ 计算过程如下:

$ABCDEFGH = V(i)$

FOR $j=0$ to 63

$SS1 = ((A \lll 12) + E + (T_j \lll 9))$

$SS2 = SS1 \wedge (A \lll 12)$

$TT1 = FF_j(A, B, C) + D + SS2 + W'_j$

$TT2 = GG_j(E, F, G) + H + SS1 + W_j$

$D = C$

$C = B \lll 9$

$B = A$

$A = TT1$

$H = G$

$G = F \lll 19$

$F = E$

$E = P_0(TT^2)$

ENDFOR

$V(i+1) = ABCDEFGH^V(i)$

运算过程中，字按照大端格式存储。

3. 杂凑结果

杂凑结果为 256bit 值 $y=ABCDEFGH=V(n)$

实验结果：

输入数据，带入计算函数，得出结果

```
Message:
qwe
Hash:
  6b013bad a3749793 f0d7605f 8d6fe45e ab4c1a6f c82c0b06 1178a9e1 fae08397
Message:
abcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd
Hash:
  debe9ff9 2275b8a1 38604889 c18e5a4d 6fdb70e5 387e5765 293dcba3 9c0c5732

Process returned 0 (0x0)   execution time : 0.378 s
Press any key to continue.
```