现代密码学作业——第六讲

一、第一节

1、预留了 64 比特，用于填充消息的长度。

2、否则可以用中间相遇攻击求出二次原像，复杂度仅为 $O(2^{n/2})$。

3、SM3 杂凑算法。

SM3密码杂凑算法

1 范围

本文本规定了SM3密码杂凑算法的计算方法和计算步骤，并给出了运算示例。

本文本适用于商用密码应用中的数字签名和验证、消息认证码的生成与验证以及随机数的生成，可满足多种密码应用的安全需求。同时，本文本还可为安全产品生产商提供产品和技术的标准定位以及标准化的参考，提高安全产品的可信性与互操作性。

2 术语和定义

1.1

比特串 bit string

由0和1组成的二进制数字序列。

1.2

大端 big-endian

数据在内存中的一种表示格式，规定左边为高有效位，右边为低有效位。数的高阶字节放在存储器的低地址，数的低阶字节放在存储器的高地址。

1.3

消息 message

任意有限长度的比特串。本文本中消息作为杂凑算法的输入数据。

1.4

杂凑值 hash value

杂凑算法作用于消息后输出的特定长度的比特串。本文本中的杂凑值长度为256比特。

1.5

字 word

长度为32的比特串。

3 符号

下列符号适用于本文本。

$ABCDEFGH$：8个字寄存器或它们的值的串联

$B^{(i)}$：第i个消息分组

CF：压缩函数

FF_j：布尔函数，随j的变化取不同的表达式

GG_j：布尔函数，随j的变化取不同的表达式

IV：初始值，用于确定压缩函数寄存器的初态

P_0：压缩函数中的置换函数

P_1：消息扩展中的置换函数

T_j：常量，随j的变化取不同的值

m：消息

m'：填充后的消息

mod：模运算

∧：32比特与运算

∨：32比特或运算

$\oplus$：32比特异或运算
$\neg$：32比特非运算
+：mod2^{32}算术加运算
$\lll k$：循环左移k比特运算
$\leftarrow$：左向赋值运算符

4 常数与函数

4.1 初始值

IV=7380166f 4914b2b9 172442d7 da8a0600 a96f30bc 163138aa e38dee4d b0fb0e4e

4.2 常量

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

4.3 布尔函数

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

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

式中X,Y,Z为字。

4.4 置换函数

$P_0(X) = X \oplus (X \lll 9) \oplus (X \lll 17)$
$P_1(X) = X \oplus (X \lll 15) \oplus (X \lll 23)$
式中X为字。

5 算法描述

5.1 概述

对长度为$l(l < 2^{64})$比特的消息m，SM3杂凑算法经过填充和迭代压缩，生成杂凑值，杂凑值长度为256比特。

5.2 填充

假设消息m的长度为l比特。首先将比特“1”添加到消息的末尾，再添加k个“0”，k是满足$l+1+k \equiv 448 mod 512$的最小的非负整数。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。填充后的消息m'的比特长度为512的倍数。

例如：对消息01100001 01100010 01100011，其长度l=24，经填充得到比特串：

$$01100001\ 01100010\ 01100011\ 1\ \overbrace{00\cdots00}^{423\text{比特}}\ \overbrace{\underbrace{00\cdots011000}_{1\text{的二进制表示}}}^{64\text{比特}}$$

5.3 迭代压缩

5.3.1 迭代过程

将填充后的消息m'按512比特进行分组：$m' = B^{(0)}B^{(1)}\cdots B^{(n-1)}$

其中$n=(l+k+65)/512$。

对m'按下列方式迭代：

FOR i=0 TO n-1

$V^{(i+1)} = CF(V^{(i)}, B^{(i)})$

ENDFOR

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$V^{(n)}$。

5.3.2 消息扩展

将消息分组$B^{(i)}$按以下方法扩展生成132个字$W_0, W_1, \cdots, W_{67}, W'_0, W'_1, \cdots, W'_{63}$，用于压缩函数$CF$：

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

b)FOR j=16 TO 67

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

ENDFOR

c)FOR j=0 TO 63

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

ENDFOR

5.3.3 压缩函数

令*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$V^{i+1} = CF(V^{(i)}, B^{(i)}), 0 \leq i \leq n-1$。计算过程描述如下：

$ABCDEFGH \leftarrow V^{(i)}$

FOR j=0 TO 63

$SS1 \leftarrow ((A \lll 12) + E + (T_j \lll j)) \lll 7$

$SS2 \leftarrow SS1 \oplus (A \lll 12)$

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

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

$D \leftarrow C$

$C \leftarrow B \lll 9$

$B \leftarrow A$

$A \leftarrow TT1$

$H \leftarrow G$

$G \leftarrow F \lll 19$

$F \leftarrow E$

$E \leftarrow P_0(TT2)$

ENDFOR

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

其中，字的存储为大端(big-endian)格式。

5.4 杂凑值

$ABCDEFGH \leftarrow V^{(n)}$

输出256比特的杂凑值$y = ABCDEFGH$。

一、第二节

1、第一、二种可以攻击成功，第三种和选择处理不行。

2、

①假设已知（C1||C2）j 为 IV 加密后的值

②构造一个 C2’=C2’ ⊕r

③将(C1||C2’)输入 oracle 得到(m1||m2’)

④推导出 m1||m2=m1||m2’ ⊕r