◎网络、通信、安全◎

# MD4 杂凑函数的近似碰撞

张 栋[1,2],李梦东[2],沈 薇[1,2]
ZHANG Dong[1,2],LI Meng-dong[2],SHEN Wei[1,2]

1.西安电子科技大学 通信工程学院,西安 710071
2.北京电子科技学院 信息安全系,北京 100070
1.Department of Communication Engineering,Xidian University,Xi'an 710071,China
2.Department of Information Security,Beijing Electronic Science and Technology Institute,Beijing 100070,China
E-mail:zhangd@besti.cn

**ZHANG Dong,LI Meng-dong,SHEN Wei.Near-collision of MD4 Hash function.Computer Engineering and Applications,2009,45(4):89-92.**

**Abstract**: Hash functions play an important role in modern cryptography,while MD4 algorithm is the basis of the Hash functions during the development of Hash functions.Using the relevant knowledge of the differential cryptanalysis theories,the MD4 algorithm and X.Y.Wang bit flipping differential cryptanalysis are reviewed in this paper.Finally one near-collisions of MD4 is found.Meanwhile,the differential path of the collisions and sufficient conditions that satisfy the differential path are shown.
**Key words**: Hash function;MD4 algorithm;differential cryptanalysis;near-collision

**摘 要**:在现代密码学中,Hash 函数扮演着重要的角色。而在 Hash 函数发展过程中,MD4 算法又起着基石的作用。通过对 MD4 算法和王小云逐比特差分分析的介绍,利用相关差分分析的理论知识,对 MD4 算法产生了一对近似碰撞。找出了该碰撞的差分路径,并确定出满足其差分路径的充分条件。
**关键词**:Hash 函数;MD4 算法;差分分析;近似碰撞
**DOI**:10.3778/j.issn.1002-8331.2009.04.025 **文章编号**:1002-8331(2009)04-0089-04 **文献标识码**:A **中图分类号**:TP309

## 1 引言

Hash 函数是密码学和信息安全领域中一个非常重要的基本组成部分,同时 Hash 函数在计算机和网络安全等领域中的应用也屡见不鲜,因而它的安全性显得尤为重要。从 1990 年 Ron.Rivest 提出了 MD4[1]开始,有关 Hash 函数的研究在密码学中始终是一个热点问题。由于计算机等技术的飞速发展和人们对 MD4 算法更加深入的研究,使得人们找出了 MD4 的许多漏洞,于此同时又在 MD4 的基础上逐渐设计出 MD5[2]、HAVAL[3]、RIPEMD[4]、RIPEMD-160[5]、SHA-0[6]、SHA-1[7]、SHA-2[8]以及目前所征集的 SHA-3[9]系列。

从 MD4 设计至今,人们对其攻击就没有停止过。下面是一些对 MD4 算法的重要分析结果:在 1996 年,H.Dobbertin 针对 MD4 给出了一对碰撞[10],并以 $2^{-22}$ 的概率找到了这次碰撞。同时,他也指出了如何找到一对有意义的消息碰撞。在 1998 年,H.Dobbertin 表明在 MD4 的三圈函数中,前两圈不是单向的,这就意味着对于 MD4 可以有效地来找出原像攻击和第二原像攻击[11]。在 2004 年 8 月的国际密码学会议上,我国著名学者王小云教授等证明了她们能以低于 $2^8$ 的复杂度找到 MD4 的一个碰撞[12],这意味着仅仅通过手工计算就可实现攻击。2007 年 4 月黎琳在《MD4 算法分析》一文中指出[13],可以以 $2^{-56}$ 的概率找到一对近似碰撞,而作者能找到另外一条差分路径,并以更高的概率($2^{-48}$)产生一对碰撞。本文基于王小云的逐比特差分分析方法对 MD4 算法进行了差分分析,通过手动方式找出了另一条近似差分路径。这也进一步说明了 MD4 在结构设计中,对非线性布尔函数的设计存在着明显的漏洞。

## 2 MD4 算法的介绍

1990 年,Ron.Rivest 提出了 MD4 算法。随后,基于 MD4 算法衍生出了一系列的 Hash 函数。MD4 算法可以将任意比特长度的输入消息压缩为 128 bit 的消息摘要。总体上说,整个压缩过程中包含了预处理和迭代压缩两个部分。下面对 MD4 算法压缩过程做以简单介绍:

**步骤** 1 增加填充位。

对于任意长度的输入消息,填充该消息使其长度与 448 模 512 同余(长度≡448 mod 512)。即使消息长度本身已满足与 448 模 512 同余,也要进行填充。填充由一个 1 和后续若干个 0

**基金项目**:北京电子科技学院基金项目(the Research Project of Beijing Electronic Science and Technology Institute)。
**作者简介**:张栋(1984-),男,硕士研究生,主要研究领域为密码算法及其应用;李梦东(1964-),男,副教授,硕士生导师,主要研究领域为密码算法及其应用;沈薇(1984-),女,硕士研究生,主要研究领域为密码通信技术。
**收稿日期**:2008-07-29 **修回日期**:2008-10-07

组成。

**步骤** 2 填充长度。

将输入消息的长度用 64 位二进制表示，并将结果添加在填充位(步骤 1)之后。

**步骤** 3 初始化 MD 缓冲区。

MD4 算法的中间链接变量和最后的压缩结果都保存在 128 位的缓冲区中(由 4 个 32 位缓冲区组成),对 4 个缓冲区初始值定义如下:

$(a,b,c,d)$=(0x67452301,0xefcdab89,0x98badcfe,0x10325476)

**步骤** 4 迭代压缩。

此步骤是整个算法的关键,而其中的核心部分又由 3 个非线性布尔函数组成的。通过前三步已将输入消息变成了若干个 512 位的消息,随后将 512 位消息分组迭代压缩,具体每步迭代形式为:

$a\leftarrow((a+f_r(b,c,d)+W_j+U_r)<<<s)$

其中,

$a,b,c,d$:表示缓冲区的 4 个字,它按一定次序随迭代步变化;

$f_r$:表示基本逻辑函数 $F$、$G$、$H$ 之一($0\leqslant r\leqslant 3$);

$W_j$:表示一个 512 位消息由 16 个 32 位消息块组成,$W_j$ 表示其中的第 $j$ 块($0\leqslant j<16$);

$U_r$:表示每一圈使用不同的常数($0\leqslant r<3$);

$<<<s$:表示 32 位的变量循环左移 $s$ 位;

+:表示模 $2^{32}$ 加法。

一个 512 位消息压缩过程中,单步基本操作如图 1 所示。

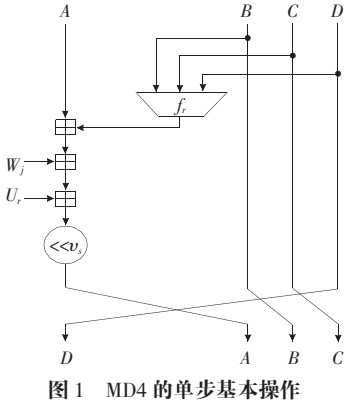


**图 1　MD4 的单步基本操作**

**步骤** 5 结果输出。

将所有的 512 位的分组处理后,最后的一个分组输出的 4 个缓冲区$(a,b,c,d)$即为 128 位的摘要。

具体 MD4 算法详见参考文献[1]。

## 3　预备知识

符号规定:以下是对各符号的解释说明。

(1)$M=(m_0,m_1,\cdots,m_{15})$与 $M=(m_0',m_1',\cdots,m_{15}')$分别表示两个 512 bit 的消息分组。

(2)$a_i,d_i,c_i,b_i$ 分别表示消息分组 $M$ 第 $4i-3,4i-2,4i-1,4i$ 步的输出,其中 $1\leqslant i\leqslant 16$。

(3)$a_i',d_i',c_i',b_i'$分别表示消息分组 $M'$第 $4i-3,4i-2,4i-1,4i$ 步的输出,其中 $1\leqslant i\leqslant 16$。

(4)$\Delta m_i=m_i'-m_i$ 代表两个消息字 $m_i$ 和 $m_i'$的差分。这些差分可正可负,用于描述带符号的差分特征。

(5)$a_{i,j},b_{i,j},c_{i,j},d_{i,j}$ 分别代表 $a_i,b_i,c_i,d_i$ 的第 $j$ 比特。其中最低有效位是第 1 比特,而最高有效位是第 32 比特。

(6)$m_i[j]$表示仅把 $m_i$ 的第 $j$ 比特从 0 变到 1,其余比特保持不变;$m_i[-j]$表示仅把 $x_i$ 的第 $j$ 比特从 1 变到 0,其余比特保持不变。

(7)$m_i[\pm j_1,\pm j_2,\cdots,\pm j_l]$表示同时改变 $m_i$ 的第 $j_1,j_2,\cdots$和 $j_l$ 比特所得到的值。“+”代表比特从 0 到 1,“−”代表比特从 1 变到 0。

### 3.1　符号差分

简单的数学分析证明出:一个模差分可能会产生许多不同的异或差分，同样一个异或差分中也能形成许多不同的模差分。这些关系表明:在差分分析中,无论是模差分还是异或差分,它们都不能准确地确定出一对消息字。因而,比特符号差分(也叫符号差分)被定义,它能够逐比特地表示一对消息字的差分特性。也可以简单地理解为,符号差分完全是由模差分与异或差分组成的。

符号差分的定义：符号差分 $\Delta m$ 表示在一对 32 位比特中逐比特的差分:

$\Delta m=m'-m=(\Delta m_{31},\cdots,\Delta m_0)$

其中

$$\Delta m_i[j]=\begin{cases}-1 & m_i'[j]>m_i[j]\\ 0 & m_i'[j]=m_i[j]\\ 1 & m_i'[j]<m_i[j]\end{cases}\quad(0\leqslant i<16,0\leqslant j<32)$$

**例** 1 假设给出 10 bit 的消息对 $m'$=1011001001 与 $m$=0010110100,分别表述出模差分、异或差分以及符号差分:

模差分=$(m'-m)\bmod(2^{10})$=1000010101

异或差分=$m'\oplus m$=1001111101

符号差分=1001−1−11−101

### 3.2　非线性布尔函数的性质

MD4 压缩函数中共使用了 3 个布尔函数$(F,G,H)$,利用这些非线性布尔函数的性质可以帮助确定满足差分路径所需要的充分条件。下面列出 3 个非线性布尔函数的有关性质:

(1)第一圈中使用的 $F$ 函数:$F(X,Y,Z)=(X\wedge Y)\vee(\neg X\wedge Z)$

通过 $F$ 函数真值表 1 可推出:

$$\Rightarrow\begin{cases}F(X,Y,Z)=F(\neg X,Y,Z)\Rightarrow Y=Z\\ F(X,Y,Z)=F(X,\neg Y,Z)\Rightarrow X=0\\ F(X,Y,Z)=F(X,Y,\neg Z)\Rightarrow X=1\end{cases}$$

(2)第二圈中使用的 $G$ 函数:$G(X,Y,Z)=(X\wedge Y)\vee(X\wedge Z)\vee(Y\wedge Z)$

通过 $G$ 函数真值表 2 可推出:

$$\Rightarrow\begin{cases}G(X,Y,Z)=G(\neg X,Y,Z)\Rightarrow Y=Z\\ G(X,Y,Z)=G(X,\neg Y,Z)\Rightarrow X=Z\\ G(X,Y,Z)=G(X,Y,\neg Z)\Rightarrow X=Y\end{cases}$$

(3)第三圈中使用的 $H$ 函数:$H(X,Y,Z)=X\oplus Y\oplus Z$

通过 $H$ 函数真值表 3 可推出:

$$\Rightarrow\begin{cases}H(X,Y,Z)=\neg H(\neg X,Y,Z)=\neg H(X,\neg Y,Z)=\neg H(X,Y,\neg Z)\\ H(X,Y,Z)=H(\neg X,\neg Y,Z)=H(X,\neg Y,\neg Z)=H(\neg X,Y,\neg Z)\end{cases}$$

### 3.3　符号差分的进位扩展

符号差分的表示方法比较多,每一位非零元素 $\Delta m_i[j]$都可以按照下面的进位扩展来表示:

**表 1**　*F* 函数真值表

| *X* | *Y* | *Z* | *F* |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

**表 2**　*G* 函数真值表

| *X* | *Y* | *Z* | *F* |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

**表 3**　*H* 函数真值表

| *X* | *Y* | *Z* | *F* |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

$$\Delta m_i[j]=\begin{cases}\Delta m_i[j+1]+\Delta m_i[-j] & \text{如果符号 } j>0\\ \Delta m_i[j-1]+\Delta m_i[-j] & \text{如果符号 } j<0\end{cases}\quad(0\leqslant i<16,0\leqslant j<32)$$

**例** 2　假设差分 $\Delta m$,则进位扩展如下所示(其中 $\hat{j}$ 表示第 $j$ 位的进位扩展):

$$\Delta m=\Delta[-21,\hat{19}]=\Delta[-21,\hat{20},-19]=\Delta[-20,-19]=$$
$$\Delta[-\hat{21},20,-19]=\Delta[-22,21,20,-19]=$$
$$\Delta[-\hat{21},19]=\Delta[-\hat{22},21,19]=\Delta[-23,22,21,19]=$$
$$\Delta[-22,\hat{21},19]=\Delta[-22,21,20,-19]$$

## 4　MD4 算法的差分分析

早在 1990 年，差分分析的思想就已经由 E.Biham 和 A.Shamir 提出[14],针对对称密码体制的选择明文(或选择密文)攻击,它是最有效的方法之一。其主要思想是通过分析特定明文差分对结果密文差分的影响来获得可能性最大的密钥。王小云等基于差分分析的思想，采用不同于传统异或差分的模减差分,提出了一系列针对标准 Hash 函数算法的攻击。在王小云等人的差分分析方法中，通过对每轮圈函数的整数模减差分和异或差分的分析得到差分特征，从而得到更多的信息来寻找碰撞。

碰撞的产生也被认为是一种特殊的差分,它使得输入的非零差分产生一个输出的零差分。也就是说消息差分 $\Delta m=m'-m\neq 0$,但是它们最后产生的 $\Delta H=h(m')-h(m)=0$。而近似碰撞是指:在差分分析过程中没有完全抵消掉输入消息引起的非零差分,但最后结果产生的输出差分近似为零。产生近似碰撞也有着十分重要的意义,它不但能使人们更好地理解运用差分分析方法,而且是寻找完全碰撞的基础。

### 4.1　选择适当的消息输入差分

选择适当的消息输入差分也就是选择 $\Delta m=m'-m$ 且 $\Delta m\neq 0$，正确地选择消息差分是产生高概率碰撞差分路线的关键一步。

定义两个明文 $m$ 和 $m'$ 的差分 $\Delta m$:$\begin{cases}m=(m_0,m_1,\cdots,m_{15})\\ m'=(m_0',m_1',\cdots,m_{15}')\end{cases}$

令 $\Delta m=m'-m=(\Delta m_0,\Delta m_1,\cdots,\Delta m_{15})=(m_0',m_1',\cdots,m_{15}')-(m_0,m_1,\cdots,m_{15})$,经试验运算分析,选择输入明文差分满足:$\Delta m_3=2^2$。

### 4.2　寻找差分路径

差分分析的关键步骤就是寻找最优的差分路径,寻找最优差分路径就是为了很好地抵消掉由于输入明文所引起的差分。具体在寻找差分路径的过程中,需要使用轮函数的性质以及比特进位来产生一些想要的比特差分,同时也可以抵消一些不想要的非零比特差分或消息差分。在本文中,通过手动方式找出 MD4 的近似碰撞差分路径,具体差分特征见表 4。

表 4 中各列分别表示(从左至右):步数、消息变量、选择消息输入、移位值、明文差分、第 $i$ 步输出差分和第 $i$ 步输出变量值。

**表 4**　MD4 碰撞的差分特征

| Step | Chaining Value for M | $m_i$ | Shift | $\Delta m_i$ | The $i$-th step difference | The $i$-th output for M′ |
|---|---|---|---|---|---|---|
| 1 | $a_1$ | $m_0$ | 3 | | | |
| 2 | $d_1$ | $m_1$ | 7 | | | |
| 3 | $c_1$ | $m_2$ | 11 | | | |
| 4 | $b_1$ | $m_3$ | 19 | $2^2$ | $2^{21}$ | $b_1$[22] |
| 5 | $a_2$ | $m_4$ | 3 | | | |
| 6 | $d_2$ | $m_5$ | 7 | | | |
| 7 | $c_2$ | $m_6$ | 11 | | | |
| 8 | $b_2$ | $m_7$ | 19 | | $2^8$ | $b_2$[9] |
| 9 | $a_3$ | $m_8$ | 3 | | | |
| 10 | $d_3$ | $m_9$ | 7 | | | |
| 11 | $c_3$ | $m_{10}$ | 11 | | $2^{19}$ | $c_3$[−20,−21,−22,−23,24] |
| 12 | $b_3$ | $m_{11}$ | 19 | | | |
| 13 | $a_4$ | $m_{12}$ | 3 | | $-2^{25}$ | $a_4$[−26] |
| 14 | $d_4$ | $m_{13}$ | 7 | | | |
| 15 | $c_4$ | $m_{14}$ | 11 | | | |
| 16 | $b_4$ | $m_{15}$ | 19 | | | |
| 17 | $a_5$ | $m_0$ | 3 | | $-2^{28}$ | $a_5$[−29] |
| 18 | $d_5$ | $m_4$ | 5 | | | |
| 19 | $c_5$ | $m_8$ | 9 | | | |
| 20 | $b_5$ | $m_{12}$ | 13 | | | |
| 21 | $a_6$ | $m_1$ | 3 | | $-2^{31}$ | $a_6$[−32] |
| 22 | $d_6$ | $m_5$ | 5 | | | |
| 23 | $c_6$ | $m_9$ | 9 | | | |
| 24 | $b_6$ | $m_{13}$ | 13 | | | |
| 25 | $a_7$ | $m_2$ | 3 | | $-2^2$ | $a_7$[−3] |
| 26 | $d_7$ | $m_6$ | 5 | | | |
| 27 | $c_7$ | $m_{10}$ | 9 | | | |
| 28 | $b_7$ | $m_{14}$ | 13 | | | |
| 29 | $a_8$ | $m_3$ | 3 | $2^2$ | | |
| 30 | $d_8$ | $m_7$ | 5 | | | |
| … | … | … | … | … | … | … |
| 44 | $b_{11}$ | $m_{13}$ | 15 | | | |
| 45 | $a_{12}$ | $m_3$ | 3 | $2^2$ | $2^5$ | $a_{12}$[6] |
| 46 | $d_{12}$ | $m_{11}$ | 9 | | $2^{14}$ | $d_{12}$[15] |
| 47 | $c_{12}$ | $m_7$ | 11 | | | |
| 48 | $b_{12}$ | $m_{15}$ | 15 | | | |

### 4.3　确定满足差分路径的充分条件

通过确定链接变量的一些条件,才能保证 4.2 节所给出的差分路径。一个可行的差分路径意味着从该路径中推导出来的充分条件不是相互矛盾的。通过分析,利用 MD4 中 3 个非线性布尔函数的性质,得出以下充分条件(见表 5)。

下面简单说明表 5 中 Step4~Step7 充分条件的推导，其他各个充分条件同样可以根据差分特性和非线性布尔函数的性质推出:

Step4:$b_1\leftarrow((b_0+F(c_1,d_1,a_1)+m_3)<<<19)$

由于 $\Delta m_3=2^2$,而在上式中只有 $m_3$ 产生了差分,所以经过移位后 $\Delta b_1=2^{21}$。也就是说 $b_1$[22]由 0→1,故得 $b_{1,22}=0$。

Step5:$a_2\leftarrow((a_1+F(b_1,c_1,d_1)+m_4)<<<3)$

上式中只有 $\Delta b_1=2^{21}$ 产生了差分，又根据差分路径 $a_2$ 此时没有产生差分,所以必须通过 $F$ 函数的性质($F(X,Y,Z)=$

表 5 MD4 碰撞的充分条件

| Step | Chaining Value for $M$ | Sufficient conditions |
|---|---|---|
| 1 | $a_1$ | |
| 2 | $d_1$ | |
| 3 | $c_1$ | $c_{1,22}=d_{1,22}$ |
| 4 | $b_1$ | $b_{1,22}=0$ |
| 5 | $a_2$ | $a_{2,22}=0$ |
| 6 | $d_2$ | $d_{2,22}=1$ |
| 7 | $c_2$ | $c_{2,9}=d_{2,9}$ |
| 8 | $b_2$ | $b_{2,9}=0$ |
| 9 | $a_3$ | $a_{3,9}=0$ |
| 10 | $d_3$ | $d_{3,9}=0;d_{3,20}=a_{3,20};d_{3,21}=a_{3,21};d_{3,22}=a_{3,22};d_{3,23}=a_{3,23};d_{3,24}=a_{3,24}$ |
| 11 | $c_3$ | $c_{3,20}=1;c_{3,21}=1;c_{3,22}=1;c_{3,23}=1;c_{3,24}=0;c_{3,9}=d_{3,9}=1$ |
| 12 | $b_3$ | $b_{3,20}=0;b_{3,21}=0;b_{3,22}=0;b_{3,23}=1;b_{3,24}=0;b_{3,26}=c_{3,26}$ |
| 13 | $a_4$ | $a_{3,20}=1;a_{3,21}=1;a_{3,22}=1;a_{3,23}=1;a_{3,24}=1;a_{4,26}=1$ |
| 14 | $d_4$ | $d_{4,26}=0$ |
| 15 | $c_4$ | $c_{4,26}=1$ |
| 16 | $b_4$ | $c_{4,29}=b_{4,29}$ |
| 17 | $a_5$ | $a_{5,29}=1$ |
| 18 | $d_5$ | $d_{5,29}=b_{4,29}$ |
| 19 | $c_5$ | $c_{5,29}=d_{5,29}$ |
| 20 | $b_5$ | $b_{5,32}=c_{5,32}$ |
| 21 | $a_6$ | $a_{6,32}=1$ |
| 22 | $d_6$ | $d_{6,32}=b_{5,32}$ |
| 23 | $c_6$ | $c_{6,32}=d_{6,32}$ |
| 24 | $b_6$ | $b_{6,3}=c_{6,3}$ |
| 25 | $a_7$ | $a_{7,3}=1$ |
| 26 | $d_7$ | $d_{7,3}=b_{6,3}$ |
| 27 | $c_7$ | $c_{7,3}=d_{7,3}$ |
| 28 | $b_7$ | |
| … | … | … |
| 43 | $c_{11}$ | |
| 44 | $b_{11}$ | |
| 45 | $a_{12}$ | $a_{12,6}=0$ |
| 46 | $d_{12}$ | $d_{12,15}=0$ |
| 47 | $c_{12}$ | |
| 48 | $b_{12}$ | |

$F(\neg X,Y,Z)=Y=Z$)来抵消掉 $\Delta b_1$ 引起的差分,故得 $c_{1,22}=d_{1,22}$。

Step6:$d_2\leftarrow((d_1+F(a_2,b_1,c_1)+m_5)<<<7)$

上式中同样只有 $\Delta b_1=2^{21}$ 产生了差分,根据差分路径 $d_2$ 此时没有产生差分,所以必须通过 $F$ 函数的性质($F(X,Y,Z)=F(X,\neg Y,Z)=X=0$)来抵消掉 $\Delta b_1$ 引起的差分,故得 $a_{2,22}=0$。

Step7:$c_2\leftarrow((c_1+F(d_2,a_2,b_1)+m_6)<<<11)$

上式中只有 $\Delta b_1=2^{21}$ 产生了差分,根据差分路径 $c_2$ 此时没有产生差分,所以必须通过 $F$ 函数的性质($F(X,Y,Z)=F(X,Y,\neg Z)=X=1$)来抵消掉 $\Delta b_1$ 引起的差分,故得 $d_{2,22}=1$。

表 5 中各列分别表示(从左至右):步数、消息变量和充分条件。

## 5 结束语

在本文中,根据王小云的逐比特差分分析方法,利用非线性布尔函数等理论知识,针对 MD4 找出了一对近似碰撞。对于一个消息 $M$,能以 $2^{-48}$ 较高的概率找到 $M'$,使得 $M$ 与 $M'$ 产生一对近似碰撞。文中描述了具体差分分析的过程,并给出了该差分路径和满足该路径的充分条件,这些工作对于研究自动搜索差分路径和寻找完全碰撞都有着十分重要的意义。

## 参考文献:


[1] Rivest R L.The MD4 message digest algorithm[C]//Crypo'90 Proceedings,1991.

[2] Rivest R L.The MD5 message digest algorithm[R].Request for Comments (RFC 1320),Internet Activities Board,Internet Privacy Task Force,April 1992.

[3] Zheng Y,Pieprzyk J,Seberry J.HAVAL-A one-way hashing algorithm with variable length of output[C]//Auscrypto'92 Proceedings,1992:83-104.

[4] RIPE,Integrity primitives for secure information systems,final report of RACE integrity primitives evalutiobn(RIPE-RACE 1040)[C]//LNCS 1007,1995.

[5] Dobbertin H,Bosselaers A,Preneel B.RIPEMD-160:A strengthened version of RIPEMD[C]//LNCS 1039,Fast Software Encryption,1996.

[6] FIPS 180-0 Secure Hash standard[S].NIST,US Department of Commerce,Washington D C,May 1993.

[7] FIPS 180-1 Secure Hash standard[S].NIST,US Department of Commerce,Washington D C,April 1995.[S.l.]:Sringer-Verlag,1996.

[8] FIPS 180-2 Secure Hash standard[S].[2002].http://csrc.nist.gov/publications/.

[9] Federal Register.Vol.72,No.212,Friday,Novernber 2,2007,Notices.

[10] Dobbertin H.Cryptanalysis of MD4[C]//Gollmann D.LNCS 1039:Fast Software Encryption.[S.l.]:Springer-Verlag,1996.

[11] Dobbertin H.The first two round of MD4 are not one-way[C]//Fast Software Encryption,1998.

[12] Wang X Y,Lai X J,Feng D G,et al.Cryptanalysis of the hash functions MD4 and RIPEMD [C]//Advances in Cryptology-Eurocrypt,May 2005.[S.l.]:Springer-Verlag,2005:1-18.

[13] 黎琳.MD4 算法分析[J].山东大学学报:理学版,2007,42(4):1-5.

[14] Biham E,Shamir A.Differential cryptanalysis of the data encryption standard[M].[S.l.]:Springer-Verlag,1993.

[15] Schlaffer M,Oswald E.Searching for differential paths in MD4[C]//Fast Software Encryption,2006:242-261.

[16] Yu Sasaki,Lei Wang,Kazuo Ohta,et al.New message difference for MD4[C]//LNCS 4593:Fast Software Encryption,2007:329-348.

[17] Wang X Y,Guo F D,Lai X J,et al.Collisions for Hash functions MD4,MD5,Haval-128 and RIPEMD[C]//Rump Session of Cryto'04,E-print Archive,August 2004.

[18] Wang X Y,Yu H B.How to break MD5 and other Hash functions[C]//Advances in Cryptology-Eurocrypt 2005.[S.l.]:Springer-Verlag,2005:19-35.

[19] Stallings W.密码编码学与网络安全:原理与实践[M].孟庆树,王丽娜,傅建明,等译.北京:电子工业出版社,2006.

[20] 王小云,张金清.MD5 报文摘要算法的各圈函数碰撞分析[J].计算机工程与科学,1996,18(2):15-22.