doi:10.3969/j.issn.1000-436x.2013.06.002

对缩减轮数 DHA-256 的原像与伪碰撞攻击

邹剑[1,2]，吴文玲[1]，吴双[1]，董乐[1,2]
(1. 中国科学院 软件研究所 可信计算与信息保障实验室，北京 100190；2. 中国科学院 研究生院，北京 100190)

摘 要：提出了对 DHA-256 散列函数 37 轮的原像攻击以及 39 轮的伪碰撞攻击。基于中间相遇攻击，利用 Biclique 方法可以改进之前对 DHA-256 的原像分析结果，将攻击轮数从原来的 35 轮提高到了 37 轮。通过上述方法还可以构造对 DHA-256 的 39 轮伪碰撞。最终，以 $2^{255.5}$ 的时间复杂度以及 2^3 的空间复杂度构造了对 DHA-256 的 37 轮原像，并以 $2^{127.5}$ 的时间复杂度以及常数 2 的空间复杂度构造了对 DHA-256 的 39 轮伪碰撞。这是目前对 DHA-256 最好的原像与碰撞攻击结果。

关键词：DHA-256 散列函数；原像攻击；伪碰撞攻击；中间相遇攻击
中图分类号：TN918 **文献标识码**：A **文章编号**：1000-436X(2013)06-0008-08

Preimage and pseudo collision attacks on round-reduced DHA-256 hash function

ZOU Jian[1,2], WU Wen-ling[1], WU Shuang[1], DONG Le[1,2]
(1. TCA, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China;
2. Graduate University, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: A preimage attack on DHA-256 hash function reduced to 37-round and a pseudo collision attack on the function reduced to 39-round were proposed respectively. Based on the meet-in-the-middle attack, the Biclique technique was used to improve the preimage attack from 35-round to 37-round. A 39-round pseudo collision was achieved using the Biclique technique. Overall, a preimage of DHA-256 was constructed with a complexity of $2^{255.5}$ and a memory of 2^3. Besides, a pseudo collision of DHA-256 was proposed with a complexity of $2^{127.5}$. These are the best results of preimage and collision attack on DHA-256 hash function.

Key words: DHA-256 hash function; preimage attack; pseudo collision attack; meet-in-the-middle

1 引言

近些年，随着散列函数被广泛地应用于消息认证以及数字签名等方面，其在密码学中也扮演着越来越重要的角色。2005 年，王小云教授先后破解了 MD5 和 SHA-0 等散列算法。美国 NIST 于 2008 年底开始了 SHA-3 散列函数的征集工作。随着各种新算法的提出，各种新型的攻击方法也孕育而生。

总体而言，散列函数应该满足如下 3 种安全需求：抗第一原像、抗第二原像和抗碰撞性。相比于碰撞，原像对于散列函数显得更加有威胁。目前使用中间相遇方法来求解散列函数的第一原像已经成为了一个热门的研究领域。利用中间相遇方法已对多个散列函数取得了比较好的分析结果，如 MD4[1,2]、HAVAL[3]、Tiger[4]、MD5[5]、SHA-0 和 SHA-1[6]。在 FSE2012 上，Ji Li 等人提出了利用中间相遇原像攻击来构造对散列函数的伪碰撞攻击[7]，使得中间相遇方法可以进一步被用来构造散列函数的伪碰撞。Dmitry 等人也提出了 Biclique 方法来

收稿日期：2012-08-30；**修回日期**：2013-03-25
基金项目：国家重点基础研究发展计划（“973”计划）基金资助项目（2013CB338002）；国家自然科学基金资助项目（61272476，61232009）
Foundation Items: The National Basic Research Program of China (973 Program) (2013CB338002)； The National Natural Science Foundation of China (61272476, 61232009)

改进传统的中间相遇攻击，并给出了目前对于 SHA-2 等散列算法最好的第一原像攻击[8]。

DHA-256[9]是 Jesang Lee 等韩国学者提出的散列函数，其设计者认为 DHA-256 是 SHA-256 的改进版。DHA-256 与 SHA-256 的区别在于压缩函数的每一轮用同一块消息来更新 2 个内部状态。由于此改变，DHA-256 比 SHA-256 更能抵抗目前已知的攻击。IAIK 韩国密码分析小组给出了对 DHA-256 的初步分析结果[10]。

本文利用目前最新的 Biclique 方法来改进对 DHA-256 的原像攻击，把攻击轮数由原来的 35 轮[11]提高到 37 轮，并利用对 DHA-256 的伪原像攻击构造了 39 轮的伪碰撞，如表 1 所示。

表 1　　攻击结果对比

攻击目标及轮数	攻击类型	时间复杂度	存储复杂度	来源
35 轮 DHA-256	第一原像	$2^{248.82}$	2^{16}	文献[11]
37 轮 DHA-256	第一原像	$2^{255.5}$	2^{3}	本文第 5 节
39 轮 DHA-256	伪碰撞	$2^{127.5}$	2	本文第 6 节

2　DHA-256 散列函数介绍

下面介绍 DHA-256 算法。

输入：256 bit 的初始链值，待处理消息 M。

输出：256 bit 的散列值。

DHA-256 按如下方式生成散列值。

$$\begin{cases} V_0 \leftarrow IV \\ V_{i+1} \leftarrow CF(V_i, M_i), \ i = 0,1,\cdots,n-1 \end{cases}$$

在进行压缩函数操作前，首先要进行消息填充。DHA-256 的消息填充规则与 SHA-256 一样，先在消息后面添加一个比特“1”，再添加若干的比特“0”，使得 $len_0 + len_M + 1 \equiv 448 \bmod 512$（其中，$len_0$ 表示添加 0 的比特数，len_M 表示消息 M 的比特长度），然后再添加用来表示消息长度的 64 bit，使得添加完的消息长度是 512 bit 的整数倍。最后将添加后的消息 M^* 分割成 512 bit 的消息块 $M_i (i = 0,1,\cdots, n-1)$，作为压缩函数的一个输入。

DHA-256 的压缩函数 $V_{i+1} \leftarrow CF(V_i, M_i)$ 是按如下步骤进行操作的。

1) 将消息 M_i 分割为 16 块 32 bit 的消息字 $W_j (j = 0,1,\cdots,15)$，并把它们按下式扩展成 64 块 32 bit 的消息字。

2) $$W_j = \begin{cases} M_j, & 0 \leqslant j \leqslant 15 \\ \sigma_1(W_{j-1}) + W_{j-9} + \sigma_2(W_{j-15}) + W_{j-16}, & 16 \leqslant j \leqslant 63 \end{cases}$$

其中，$\sigma_1(x) = x \oplus (x <<< 7) \oplus (x <<< 22)$，$\sigma_2(x) = x \oplus (x <<< 13) \oplus (x <<< 27)$。

3) $S_0 \leftarrow V_i$，其中，$S_j = (A_j, \cdots, H_j)$。

4) 用步函数 f（如图 1 所示）来更新状态 S_j，压缩函数包含 64 轮步函数操作，输出 $V_{i+1} \leftarrow S_0 + S_{64}$，此处加法是模加运算而不是异或操作。

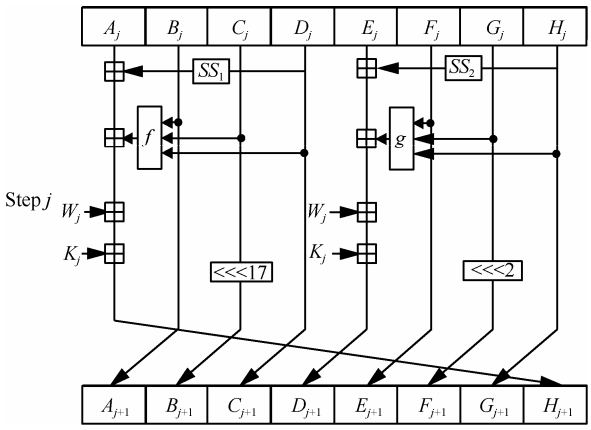


图 1　DHA-256 步函数

关于 DHA-256 更详细的信息请参阅文献[9]。

在图 1 中，DHA-256 所采用的函数分别为

$$f(x,y,z) = (x \wedge y) \vee (\neg x \wedge z)$$

$$g(x,y,x) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$$

$$SS_1(x) = x \oplus (x <<< 11) \oplus (x <<< 25)$$

$$SS_2(x) = x \oplus (x <<< 19) \oplus (x <<< 29)$$

3　预备知识

3.1　将伪原像转化为原像

伪原像攻击是要找到 (x, M) 满足 $CF(x, M) = y$，其中，y 是预先给定的值。伪原像攻击无需限制 x 等于给定的初始链值。目前已经有通用算法可以将伪原像转化为原像，此算法的细节可以参阅文献[12]。假设攻击者能以 2^k 的时间复杂度来找到目标算法的伪原像，则根据转化算法，攻击者能以 $2^{\frac{n+k}{2}+1}$ 的时间复杂度得到对应的原像。

3.2　中间相遇原像攻击

中间相遇（如图 2 所示）的攻击过程如下。

1) 敌手分别选取中立字 W_a 和 W_b，并根据这 2 个中立字将压缩函数分成 2 块（本文将子函数称为

块），并分别称为前向块和后向块，其中，前向块与消息 W_a 相互独立，后向块与消息 W_b 相互独立。

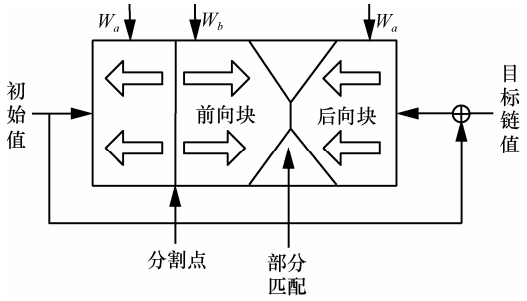


图 2　中间相遇伪原像攻击

2) 对于分割点处的状态值以及除了中立字外的消息进行随机赋值。对于中立字 W_a 所有可能取值，敌手从分割点向后计算得到在匹配点的状态值，并将此状态值存在表 L_a 中。

3) 对于中立字 W_b 的所有可能取值，敌手从分割点向前计算得到在匹配点的状态值，并检查表 L_a 中是否存在一个匹配。

4) 利用不同的初始赋值，不断重复步骤 2)和步骤 3)，直到找个一个全状态的匹配。上述 4 个步骤提供了一个求给定算法伪原像的方法。

5) 敌手可以再利用 3.1 节中的方法将伪原像攻击转化为原像攻击。

3.3　将部分目标原像转化为碰撞攻击的一般方法

假设有一个 Oracle Λ，能够以 2^s 的时间复杂度来找到目标散列函数 t bit 的部分原像，并有 Λ 对于 2 次询问能返回不同的消息 M'。显然敌手可以按如下方式不断地调用 Λ，并以时间复杂度 $2^s \times 2^{(n-t)/2}$ 来构造此目标散列函数的一个碰撞。

1) 设 t bit 的随机数据为 d'。

2) 以初始链值 IV 和 d' 为变量，调用 Λ $2^{(n-t)/2}$ 次。

经过上述步骤可以得到 $2^{(n-t)/2}$ 个 $(n-t)$ bit 的随机数据，由生日攻击可知，这些数据中存在一个碰撞的概率很大。由于已经固定 t bit 数据为 d'，则只要找到剩余 $(n-t)$ bit 的碰撞，敌手就找到了整体 n bit 的碰撞。在上述攻击中，敌手可以使用免存储的生日攻击[13,14]，所以整个攻击就只需要 $2^s \times 2^{(n-t)/2}$ 的时间复杂度。如果 $2^{\frac{(n-t)}{2}+s} < 2^{\frac{n}{2}}$，即 $s < \frac{t}{2}$，则上述攻击是一个有效的攻击。

将部分原像攻击转化为碰撞攻击，需要注意以下 2 点。

1) 保证匹配点在最后。

2) 要使得一个 t bit 的部分原像的复杂度 $2^s < 2^{\frac{t}{2}}$。

在本文中间相遇只能求解伪原像，求原像需转换算法，而转化算法的时间复杂度要高于求碰撞的复杂度，因此目前只能找到伪碰撞。

3.4　Biclique 攻击方法

初始化结构（如图 3 所示）最早是由日本学者 Aoki 和 Sasaki 在攻击 MD4 时提出的[1]。利用此技术可以交换分割点附近的中立字。通过交换分割点处的中立字，敌手可以攻击更长的轮数。需要注意的是上述交换必须不改变轮函数的值。

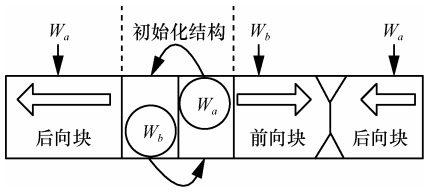


图 3　初始化结构

Biclique 方法[8]（如图 4 所示）是对初始化结构的一种改进。记后向块的起始状态为 Q，当以中立字 $W_a[i]$ 向后计算时，就将状态值记为 Q_i。同理，当以中立字 $W_b[j]$ 向前计算时，记前向块的状态值为 P_j。对于所有的 Q_i 和 P_j 必须满足如下关系。

$$\forall i,j: Q_i \xrightarrow[\text{Biclique}]{W_a=i \,\|\, W_b=j} P_j$$

如果对于中立字 W_a 和 W_b 都有 2^d 候选值，则敌手通过上式可得到 2^d 个 Q_i 和 P_j 的值。笔者称其为构造了一个 d 维的 Biclique。构造 Biclique 的方法

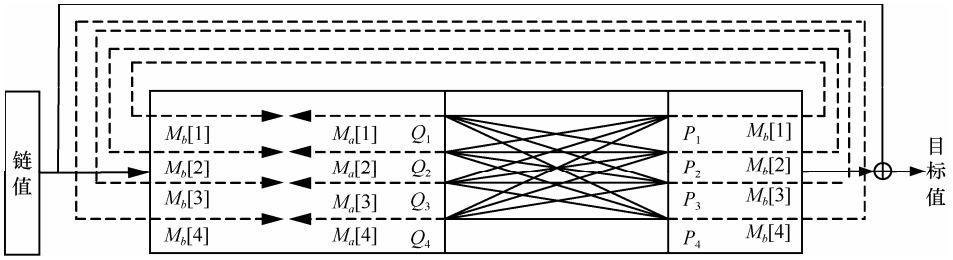


图 4　二维 Biclique

有很多，本文是利用定理 1 中模运算的性质来构造。

定理 1　对于模加运算 $A+B$ 以及模减运算 $A-B \pmod{2^n}$，有如下性质：

$$[A+B]_i = A_i \oplus B_i \oplus C_i^+ \quad \text{with } C^+ = MAJ(A,B,C^+)^{\ll 1}$$

$$[A-B]_i = A_i \oplus B_i \oplus C_i^- \quad \text{with } C^- = MAJ(\overline{A},B,C^-)^{\ll 1}$$

证明　对于 $[A+B]_i$，如果可以同时限制进位 C_i^+ 和 B_i 的值均为 0，则 $[A+B]_i$ 只受 A_i 的影响，且不会向前进位。同理，如果可以同时限制进位 C_i^+ 和 B_i 的值均为 1，则 $[A+B]_i$ 也只受 A_i 的影响，且一定会向前进位。对于 $[A-B]_i$，若同时限制借位 C_i^- 与 B_i 的值均为 0，则 $[A-B]_i$ 只受 A_i 影响，且一定不会向前借位，同理若 C_i^- 与 B_i 均为 1，则 $[A-B]_i$ 也只受 A_i 影响，且一定会向前借位。证毕。

4　对缩减轮 DHA-256 散列函数的伪原像攻击

由于 DHA-256 在每一轮的步函数中都要用同一消息处理 2 个状态块，其相比于 SHA-256 而言有着更强的消息依赖性，相比于 SHA-256，敌手将更难选取中立字。在本节中，笔者将提出一些技术改进攻击结果。

4.1　DHA-256 的消息扩展算法

对于 DHA-256，敌手可以从 $\{W_0,\cdots,W_{15}\}$ 中任意挑选中立字，又根据 DHA-256 的消息扩展算法是可逆的特点，任意 16 个连续的消息都能唯一决定剩下的所有消息，这使得敌手可以任意选取中立字的位置，假设敌手从 $\{W_z,\cdots,W_{z+15}\}$ 开始，其中，z 为任意整数，为了考察最优的选取方式，笔者将消息字向 2 个方向进行扩展。

对于向后的方向：

$$W_{z-1} = W_{z+15} - \sigma_1(W_{z+14}) - W_{z+6} - \sigma_2(W_z)$$
$$W_{z-2} = W_{z+14} - \sigma_1(W_{z+13}) - W_{z+5} - \sigma_2(W_{z-1})$$
$$W_{z-3} = W_{z+13} - \sigma_1(W_{z+12}) - W_{z+4} - \sigma_2(W_{z-2})$$
$$W_{z-4} = W_{z+12} - \sigma_1(W_{z+11}) - W_{z+3} - \sigma_2(W_{z-3})$$
$$W_{z-5} = W_{z+11} - \sigma_1(W_{z+10}) - W_{z+2} - \sigma_2(W_{z-4})$$
$$W_{z-6} = W_{z+10} - \sigma_1(W_{z+9}) - W_{z+1} - \sigma_2(W_{z-5})$$
$$W_{z-7} = W_{z+9} - \sigma_1(W_{z+8}) - W_z - \sigma_2(W_{z-6})$$
$$W_{z-8} = W_{z+8} - \sigma_1(W_{z+7}) - W_{z-1} - \sigma_2(W_{z-7})$$
$$W_{z-9} = W_{z+7} - \sigma_1(W_{z+6}) - W_{z-2} - \sigma_2(W_{z-8})$$
$$W_{z-10} = W_{z+6} - \sigma_1(W_{z+5}) - W_{z-3} - \sigma_2(W_{z-9})$$
$$W_{z-11} = W_{z+5} - \sigma_1(W_{z+4}) - W_{z-4} - \sigma_2(W_{z-10})$$
$$W_{z-12} = W_{z+4} - \sigma_1(W_{z+3}) - W_{z-5} - \sigma_2(W_{z-11})$$

对于向前的方向：

$$W_{z+16} = \sigma_1(W_{z+15}) + W_{z+7} + \sigma_2(W_{z+1}) + W_z$$
$$W_{z+17} = \sigma_1(W_{z+16}) + W_{z+8} + \sigma_2(W_{z+2}) + W_{z+1}$$
$$W_{z+18} = \sigma_1(W_{z+17}) + W_{z+9} + \sigma_2(W_{z+3}) + W_{z+2}$$
$$W_{z+19} = \sigma_1(W_{z+18}) + W_{z+10} + \sigma_2(W_{z+4}) + W_{z+3}$$
$$W_{z+20} = \sigma_1(W_{z+19}) + W_{z+11} + \sigma_2(W_{z+5}) + W_{z+4}$$
$$W_{z+21} = \sigma_1(W_{z+20}) + W_{z+12} + \sigma_2(W_{z+6}) + W_{z+5}$$
$$W_{z+22} = \sigma_1(W_{z+21}) + W_{z+13} + \sigma_2(W_{z+7}) + W_{z+6}$$
$$W_{z+23} = \sigma_1(W_{z+22}) + W_{z+14} + \sigma_2(W_{z+8}) + W_{z+7}$$

经过测试，笔者发现选取 W_{z+4} 和 W_{z+8} 作为中立字可以攻击到更多的轮数。

4.2　Biclique 构造

为了构造对 DHA-256 的三维 Biclique，笔者要将消息 W_{z+4} 和 W_{z+8} 的位置进行交换（如图 5 所示）。为此，需要利用起始点的自由度来满足消息的位置交换。在此令起始状态为 S_{z+5}，并先对其进行赋值。由于状态 S_{z+5} 包括 8 块 32 bit 的状态块，分别为 A_{z+5}、B_{z+5}、C_{z+5}、D_{z+5}、E_{z+5}、F_{z+5}、G_{z+5} 和 H_{z+5}。令其中 D_{z+5}、F_{z+5}、G_{z+5} 和 H_{z+5} 的状态值全为 0，B_{z+5} 的第 31、28、18、17、10、0 比特取 1 而其余比特取 0，C_{z+5} 的第 30、20、1 比特取 1 而其余比特取 0。对于上述赋值的原因，笔者会在具体构造三维 Biclique 时进行说明，下同。

令消息 W_{z+4} 第 31、28、18 比特为中立比特，剩下的比特全取 0，消息 W_{z+8} 第 27、17、2 比特为中立比特，剩下的比特全取 0。这里先没有给出状态 A_{z+5} 和 E_{z+5} 的赋值，是因为笔者需要利用它们的自由度来消去 W_{z+8} 的影响，笔者会在具体构造三维 Biclique 时加以说明。

如表 2 和表 3 所示，可以按如下步骤构造三维的 Biclique。

表 2　　消息 W_{Z+4} 向下传播

轮数	A	B	C	D	E	F	G	H	W
Z+4									a
Z+5				a				a	
Z+6			a	θ			a	θ'	
Z+7		a'	θ	λ		a''	θ'	λ'	
Z+8	a'				a''				

注：表格中的字母是被消息 W_{z+4} 影响的比特位置，如下所示：

$a=\{31,28,18\}$, $a'=\{16,13,3\}$, $a''=\{30,20,1\}$,
$\theta=\{31,28,25,18,15,5\}$, $\theta'=\{31,29,28,24,21,18,11,10,7\}$
$\lambda=\{31,30,29,28,26,25,24,21,19,18,16,15,12,11,10,8,7,5,4\}$
$\lambda'=\{31,30,29,28,26,25,24,21,20,19,18,16,15,11,10,8,7,5,4\}$

表 3 消息 W_{Z+8} 向下传播

轮数	A	B	C	D	E	F	G	H	W
Z+4									
Z+5				b'				b''	
Z+6			b'				b''		
Z+7		b				b			
Z+8	b				b				

注：表格中的字母是受到消息 W_{z+8} 影响的比特位置,如下所示：

$b=\{27,17,2\}$, $b'=\{17,10,0\}$, $b''=\{25,15,0\}$

首先是消息 W_{z+4} 如何向下传播（如表 2 所示），其具体步骤如下。

1) $D_{z+5}+W_{z+4}=D_{z+5}$, $H_{z+5}+W_{z+4}=H_{z+5}$。由 D_{z+5}、H_{z+5} 和消息 W_{z+4} 的初始赋值，由定理 1 可得新生成的 D_{z+5} 和 H_{z+5} 只在第 31、28、18 比特受到消息 W_{z+4} 的影响，其余比特均为 0。

2) $D_{z+6}=W_{z+5}+K_{z+5}+g(F_{z+5},G_{z+5},H_{z+5})+E_{z+5}+SS_2(H_{z+5})$, $H_{z+6}=W_{z+5}+K_{z+5}+f(B_{z+5},C_{z+5},A_{z+5})+SS_1(D_{z+5})$ 对 D_{z+6}，因为 F_{z+5} 和 G_{z+5} 全为 0，所以 g 函数不会受到 H_{z+5} 的影响，并且输出也全为 0。可令 $W_{z+5}+K_{z+5}$ 的值为 0,利用 W_{z+5} 的自由度,又由 W_{z+8} 向上传播的路径（如表 3 所示），可得 W_{z+8} 会影响 $SS_2(H_{z+5})$，但此处可以利用 E_{z+5} 的自由度消除消息 W_{z+8} 对 $SS_2(H_{z+5})$ 的影响，则由 $D_{z+6}=W_{z+5}+K_{z+5}+g(F_{z+5},G_{z+5},H_{z+5})+E_{z+5}+SS_2(H_{z+5})$ 可得：D_{z+6} 除了第 31、28、25、18、15、5 bit 受到消息 W_{z+4} 的影响外（受到 $SS_2(H_{z+5})$ 的影响），其余比特为 0。对 H_{z+6}，本文有类似的结论，由前面的赋值，B_{z+5} 的第 31、28、18、17、10、0 比特为 1 而其余比特为 0，C_{z+5} 也只在第 30、20、1 比特处取 1,则 $f(B_{z+5},C_{z+5},D_{z+5})$ 全为 0,即 f 不受 D_{z+5} 的影响。又由 W_{z+8} 向上传播的路径可得：W_{z+8} 对 $SS_1(D_{z+5})$ 有影响（如表 3 示，下面具体介绍），但可利用 A_{z+5} 的自由度去除消息 W_{z+8} 对 $SS_1(D_{z+5})$ 的影响，所以由 H_{z+6} 的定义 $H_{z+6}=W_{z+5}+K_{z+5}+f(B_{z+5},C_{z+5},D_{z+5})+A_{z+5}+SS_1(D_{z+5})$ 可得：H_{z+6} 除了第 31、29、28、24、21、18、11、10、7 比特位置受 W_{z+4} 影响外，其余比特全是 0。

3) $D_{z+7}=W_{z+6}+K_{z+6}+g(F_{z+6},H_{z+5},H_{z+6})+E_{z+6}+SS_2(H_{z+6})$, $H_{z+7}=W_{z+6}+K_{z+6}+f(B_{z+6},D_{z+5},D_{z+6})+A_{z+6}+SS_1(D_{z+6})$, 对于 D_{z+7}，在 g 函数中，由于 H_{z+5} 和 H_{z+6} 在第 31、28、18 比特处均受到 W_{z+4} 的影响，所以 $g(F_{z+6},H_{z+5},H_{z+6})$ 在第 31、28、18 比特处一定会受到 W_{z+4} 的影响，而其余比特均为 0，再令 $W_{z+6}+K_{z+6}$ 为 0，这里需要利用 W_{z+6} 的自由度，又根据初始的赋值有 $E_{z+6}=F_{z+5}=0$，可得 $W_{z+6}+K_{z+6}+g(F_{z+6},H_{z+5},H_{z+6})+E_{z+6}$ 的第 31、28、18 比特处受到 W_{z+4} 的影响，而其余比特均为 0。所以 D_{z+7} 只会在第 31、30、29、28、26、25、24、21、19、18、16、15、11、10、8、7、5、4 比特处受到 W_{z+4} 的影响，而其余比特均为 0，注意到这里的第 19 比特是因为不知道第 18 比特是否会进位而不知道其具体取值。对于 H_{z+7}，笔者有类似的结论，因为 f 函数的输入 D_{z+5} 和 D_{z+6} 在第 31、28、18 比特处均受到 W_{z+4} 的影响，又因为 $B_{z+6}=C_{z+5}<<<17$，所以由初始赋值可得 B_{z+6} 只在第 25、15、5 比特处取值为 1，所以 $f(B_{z+6},D_{z+5},D_{z+6})$ 只在第 31、28、18 比特处受到 W_{z+4} 的影响，而其余比特均为 0。又因为 $W_{z+6}+K_{z+6}$ 的比特取值全为 0，所以 $W_{z+6}+K_{z+6}+f(B_{z+6},D_{z+5},D_{z+6})+A_{z+6}$ 只会在第 31、29、28、19、18 比特被 W_{z+4} 影响，并在第 17、10、0 比特取 1（因为 B_{z+5} 的赋值），所以 $W_{z+6}+K_{z+6}+f(B_{z+6},D_{z+5},D_{z+6})+A_{z+6}+SS_1(D_{z+6})$ 只会在第 31、30、29、28、26、25、24、21、20、19、18、16、15、12、11、10、8、7、5、4 比特处会受到 W_{z+4} 影响，并在第 17、0 比特处取 1，而其余比特均为 0。

其次是消息 W_{z+8} 如何向上传播（如表 3 所示），其具体过程如下。

1) 对于 $A_{z+8}-W_{z+8}$，已知消息 W_{z+8} 除了第 27、17、2 比特是中立比特外，其余都是 0，而 A_{z+8} 除了第 16、13、3 比特处受 W_{z+4} 的影响外，其余比特取值全为 0，则由定理 1 可得：W_{z+8} 只会影响 $A_{z+8}-W_{z+8}$ 的第 27、17、2 比特，同理，对于 $E_{z+8}-W_{z+8}$ 也有类似的结论。

2) 考虑 $A_{z+8}-W_{z+8}$ 向上传播到状态 B_{z+7}，因为状态 C_{z+7} 和 D_{z+7} 的第 27、17、2 比特处全为 0，所以 $B_{z+7}=A_{z+8}-W_{z+8}$ 就不会通过 f 函数影响到 D_{z+8}。类似地,对于 $E_{z+8}-W_{z+8}$ 向上传播,状态 G_{z+7} 和 H_{z+7} 的第 27、17、2 比特处也全为 0，这样 $F_{z+7}=E_{z+8}-W_{z+8}$ 就不会通过 g 函数影响到状态 H_{z+8}。

3) 考虑 B_{z+7} 再向上传播到 C_{z+6}，因为 B_{z+6} 的第 17、10、0 比特（因为轮函数要向左循环 17 bit）取 0，所以 C_{z+6} 不会影响到 D_{z+7}。类似地，有 F_{z+6} 和 H_{z+6} 的第 25、15、0（因为轮函数要向左循环移

位 2 bit）比特位置全为 0，从而保证 G_{z+6}=F_{z+7} 不会通过 g 函数影响到 H_{z+7}。

4) 当 C_{z+6} 继续向上传播到 D_{z+5}，由于初始赋值 B_{z+5} 只在第 17、10、0 比特处取 1，使得 D_{z+5} 不会经过函数 f 影响到 D_{z+6}。此处注意到，D_{z+5} 还要经过一个 $SS_1(D_{z+5})$ 变化，所以这里需要利用状态 A_{z+5} 的自由度将受到消息 W_{z+8} 影响的 $SS_1(D_{z+5})$ 全变为 0（注意到此处 $SS_1(D_{z+5})$ 还受到消息 W_{z+4} 的影响，但笔者只需要用状态 A_{z+5} 的自由度消去其中被 W_{z+8} 影响的那一部分，使其变为 0），这样 W_{z+8} 就不会影响到 D_{z+6}。类似地，当 G_{z+6} 继续向上传播到 H_{z+5} 时，除了不能使其由 g 函数影响 H_{z+6} 之外（这个由初始赋值来保证），还要利用 E_{z+5} 的自由度将 $SS_2(H_{z+5})$，也即受到消息 W_{z+8} 影响的比特都变为 0。

从上面的构造过程可以看出，2 条路径赋值没有矛盾，所以可以将 2 条路径合并而没有矛盾，即已经成功地构造 5 轮三维的 Biclique。

4.3　消息补偿

在前向块中，W_{z+17} 受到了中立字 W_{z+8} 的影响（$W_{z+17}=\sigma_1(W_{z+16})+W_{z+8}+\sigma_2(W_{z+2})+W_{z+1}$），所以需要利用 W_{z+1} 来补偿 W_{z+8}。这里补偿的意思是令 W_{z+17} 不受 W_{z+8} 的影响，为此需要利用 W_{z+1} 的自由度令下式成立：$W_{z+1}+W_{z+8}=C_1$（其中，C_1 是任意常数），所以上式就变为 $W_{z+17}=\sigma_1(W_{z+16})+\sigma_2\ (W_{z+2})+C_1$，则 W_{z+17} 不受 W_{z+8} 的影响。但 W_{z+1} 受到 W_{z+8} 的影响，又会影响 W_{z+16}，笔者就利用 W_z 的自由度来消去 W_{z+1}，即令 $\sigma_2(W_{z+1})+W_z=C_2$（其中，$C_2$ 是任意常数），则 $W_{z+16}=\sigma_1(W_{z+15})+W_{z+7}+C_2$，即 W_{z+16} 不受消息 W_{z+8} 的影响。这里并不限制 C_1 和 C_2 的取值。所以依据 DHA-256 的消息扩展和此处的消息补偿，对于前向块，直到 W_{z+22} 都不会被受到 W_{z+8} 的影响。

类似地，对于后向块，可以利用 W_{z+13} 来消掉 W_{z+4} 对 W_{z-3} 的影响，再利用 W_{z+14} 和 W_{z+15} 分别消掉 W_{z+13} 对 W_{z-2} 以及 W_{z+14} 对 W_{z-1} 的影响。可以表示为 $W_{z+13}-W_{z+4}=C_3$ 、$W_{z+14}-\sigma_1(W_{z+13})=C_4$ 和 $W_{z+15}-\sigma_1(W_{z+14})=C_5$。（其中，$C_3$、$C_4$ 和 C_5 都是任意的常数）。类比于前向块，对于后向块，通过上述赋值就可以保证直到 W_{z-10} 都不会受到 W_{z+4} 的影响。用图 5 来表示消息补偿后的消息概况。

4.4　非直接部分匹配

由于 DHA-256 每一轮步函数加入的消息会影响下一轮中的 2 个状态块，若用传统的部分匹配只能增加 3 轮。本文中利用非直接部分匹配改进传统的部分匹配，达到能增加 4 轮的攻击效果。在这里，令 z=12，从后面的分析会了解，此赋值可以让敌手从 DHA-256 的起始轮开始攻击。

观察匹配点 A_{35}，其由前向块计算所得的公式为 $A_{35}=\psi(W_{16})$，而从后向块计算所得的公式为 $A_{35}=\varphi(W_{20})-W_{16}$，因此可以将寻找满足状态 A_{35} 的消息 W_{16} 和 W_{20} 的任务变为寻找消息 W_{16} 和 W_{20}，使得等式 $\psi(W_{16})+W_{16}=\varphi(W_{20})$ 成立。

5　对 37 轮 DHA-256 的原像攻击

若要构造对 DHA-256 的原像攻击，则必须要满足 DHA-256 的消息填充规则。由于令 z=12，从之前的分析过程可知，对消息 W_{14}、W_{15} 以及 W_{13} 的最后一个比特没有要求，这样就可以满足 DHA-256 的消息填充规则。攻击算法及复杂度计算如下。

1) 按 4.2 节的方法构造三维的 Biclique，再对剩下的没有限制的消息比特进行随机赋值。

2) 对于 W_{16} 所有可能的值，计算消息 W_{25}、W_{26}、W_{27} 相对应的值（消息补偿部分），并向前计算得到 $\psi(W_{16})$ 的值，将 $(W_{16},W_{16}+\psi(W_{16}))$ 对存储在表 L_a 中。

3) 对于 W_{20} 所有可能的值，计算消息 W_{12} 和 W_{13} 相对应的值（消息补偿部分），并向后计算得到 $\varphi(W_{20})$ 的值，并将其存在表 L_b 中。

4) 将表 L_a 与表 L_b 的值进行比较，若找到一个匹配，则继续计算剩下的状态值，若剩下的状态也全部匹配，则就找到了一个伪原像。

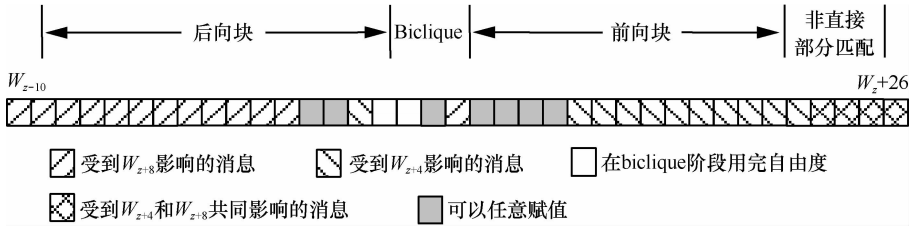


图 5　经过消息补偿后的消息

5) 利用不同的初始赋值，重复上述 4 个步骤，直到找到一个全匹配。

6) 重复步骤 5)多次，找到足够多的伪原像，再利用将伪原像转化为原像的算法找到一个原像。

对于上述过程，步骤 2)和步骤 3)的时间复杂度和存储复杂度均为 2^3，它们产生了 2^6 对。一个对在状态全匹配的概率为 2^{-256}，则步骤 1)~步骤 3)至少需要重复 $2^{256-6}=2^{250}$ 次，因此找一个伪原像所需的时间复杂度为 $2^{250+3}=2^{253}$，存储复杂度为 2^3。再根据 3.1 节的转换算法找到一个原像的时间复杂度为 $2^{\frac{253+256}{2}+1}=2^{255.5}$，存储复杂度为 2^3。

在原像攻击中，开始阶段总共有 256 bit 状态的自由度以及 512 bit 消息的自由度。在 Biclique 阶段，先需要用到状态的 256 bit 自由度，而后又需要用到消息 W_{17} 和 W_{18} 的自由度，即需要 64 bit 的自由度。其次在消息补偿阶段需要用掉 100 bit 的自由度。最后还要满足消息填充的要求，又要去掉 65 bit 的自由度，所以此处剩余的自由度为:256+512−256−100−64−65=283 bit，因此有足够的自由度完成上述攻击过程。整个攻击过程如图 6 所示。

在上述构造原像的攻击中，所需构造的原像块至少需要两块，其中第二块用来构造所需的伪原像攻击，而第一块用来连接第二块的链值与初始链值。

6 利用中间相遇伪原像来构造伪碰撞

在 FSE 2012 上，Ji Li 等人提出了利用中间相遇来构造伪碰撞。这种方法不仅为求伪碰撞提供了新的方法，而且还给出了目前对于 SHA-2 等散列函数最好的伪碰撞攻击。其基本思想是通过部分目标原像来构造碰撞攻击。

上述伪原像攻击结果可以构造对 DHA-256 的 37 轮伪碰撞。如果只需构造伪碰撞，就不用考虑将伪原像转换为原像，即不要求 $2^{\frac{n+k}{2}+1}$ 中有 $k<n-2$ 成立。因此可以构造一维的 Biclique，详见表 4 和表 5。其中，消息 W_3 向下传播，消息 W_9 向上传播。通过第 4 节中的方法，可以构造对 DHA-256 的 39 轮伪原像，其时间复杂度是 2^{255}，因为 $k=255$，则 $k>n-2$，虽然无法构造对 DHA-256 的原像攻击，但却可以将其转化为伪碰撞攻击。此时，前向块中立字为 1 bit，后向块中立字也为 1 bit，令部分匹配的大小为 1 bit，则就以 2^1 的时间复杂度得到 2 个对，这些对在给定的 1 bit 处都匹配，接下来重复上述攻击过程 $2^{(256-1)/2-1}=2^{126.5}$，则可得到 $2^{126.5+1}=2^{127.5}$ 数据。由生日攻击可知，这些数据以高概率在剩下的 255 bit 处存在一个碰撞。综上，笔者以 $2^{126.5+1}=2^{127.5}$ 的复杂度得到了对 DHA-256 的一个 39 轮的伪碰撞。

表 4 伪碰撞的 Biclique，消息 W_3 向下传播

R	A	B	C	D	E	F	G	H	W
3									31
4				31				31	
5			31	γ			31	γ_1	
6		16	γ	ω		1	γ_1	ω_1	
7	16	a_1	ω	ω_2	1	γ_2	ω_1	ω_3	
8	a_1	ω_4	ω_2	ω_6	γ_2	ω_5	ω_3	ω_7	
9	ω_4				ω_5				

注：表格中的字母是被消息 W_3 影响的比特位置，如下所示：

$\gamma=\{31,28,18\}$，$\gamma_1=\{31,24,10\}$，$\gamma_2=\{26,12,1\}$，

$a_1=\{16,13,3\}$，$\omega=\{29,28,24,21,18,11,10,7\}$，

$\omega_1=\{29,24,21,19,11,10,7\}$，$\omega_2=\{30,25,19,18,11,6,4\}$，

$\omega_3=\{29,24,21,18,17,14,10,8,4\}$，

$\omega_4=\{28,27,24,14,13,9,6,3\}$，$\omega_5=\{31,26,23,21,13,12,9\}$，

$\omega_6=\{29,27,26,25,23,21,17,16,15,11,7,1\}$，

$\omega_7=\{31,30,26,23,22,20,18,17,16,15,12,11,9,7,5\}$

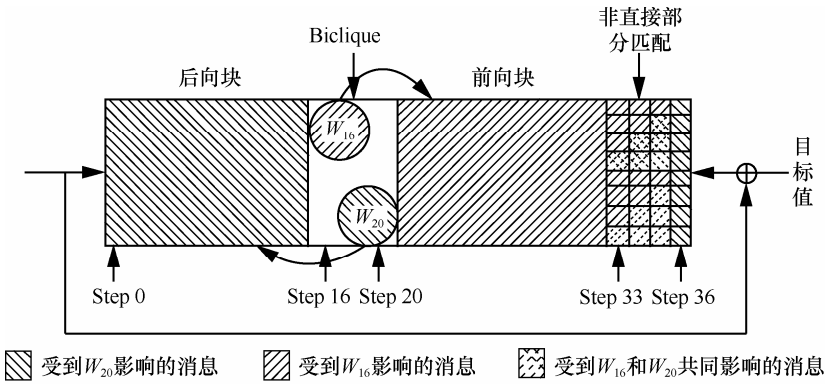


图 6 利用中间相遇伪原像来构造伪碰撞

表 5　伪碰撞的 Biclique，消息 W_9 向上传播

R	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*W*
4		17				0			31
5	17				0				
6				17				0	
7			17				0		
8		2				2			
9	2				2				2

7　结束语

本文利用 Biclique 方法提出了对 DHA-256 的 37 轮原像攻击以及 39 轮的伪碰撞。在本文之前，对 DHA-256 最好的原像攻击是 35 轮，本文结果要比其多 2 轮。据笔者所知，对于 DHA-256 伪碰撞的结果是在本文第一次被提出。

从上面的攻击，笔者发现利用中间相遇的思想仅能得到目标散列函数的伪原像伪碰撞。如果需要求原像则需要一个转换算法，但目前仍没有通用有效的方法能将中间相遇伪碰撞转化为碰撞。下一步的工作在于如何能更有效地利用中间相遇攻击来构造对目标函数的碰撞攻击。

参考文献：

[1] AOKI K, SASAKI Y. Preimage attacks on one-block MD4, 63-step MD5 and more[A]. Workshop Records of SAC 2008[C]. Sackville, Canada, 2008. 82-98.

[2] GUO J, LING S, RRCHBERGER C. Advanced meet-in-the-middle preimage attacks: first results on full tiger, and improved results on MD4 and SHA-2[A]. ASIACRYPT 2010 LNCS[C]. Singapore, 2010. 56-75.

[3] AOKI K, SASAKI Y. Preimage attacks on 3, 4, and 5-Pass HAVAL[A]. ASIACRYPT 2008 LNCS[C]. Melbourne, Australia, 2008. 253-271.

[4] ISOBE T, SHIBUTANI K. Preimage attacks on reduced tiger and SHA-2[A]. FSE 2009 LNCS[C]. Leuven, Belgium, 2009. 139-155.

[5] SASAKI Y, AOKI K. Preimage attacks on step-reduced MD5[A]. ACISP 2008 LNCS[C]. Wollongong, Australia, 2008. 282-296.

[6] CANNIERE C, RECHBERGER C. Preimages for reduced SHA-0 and SHA-1[A]. CRYPTO 2008 LNCS[C]. Santa Barbara, CA, USA, 2008. 179-202.

[7] LI J, ISOBE T, SHIBUTANI K. Converting meet-in-the-middle pre-image attack into pseudo collision attack: application to SHA-2[A]. FSE 2012 LNCS[C]. Washington DC, USA, 2012. 264-286.

[8] KHOVRATOVICH D, RECHBERGER C, SAVELIEVA A. Bicliques for preimages: attacks on skein-512 and the SHA-2 family[A]. FSE 2012 LNCS[C]. Washington DC, USA, 2012. 244-263.

[9] LEE J, CHANG D, KIM H. A new 256-bit hash function DHA-256 enhancing the security of SHA-256[EB/OL]. http://csrc.nist.gov/groups/ST/hash/documents/changD-DHA256.pdf.

[10] IAIK krypto group: preliminary analysis of DHA-256[EB/OL]. http://eprint.iacr.org/2005/398.pdf.

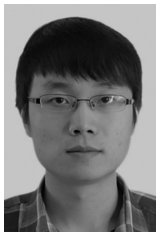
[11] ZHONG J M, LAI X J. Preimage attack on reduced DHA-256[J]. J Inf Sci Eng, 2011, 27(4):1315-1327.

[12] ALFRED J, MENEZES P C V O, VANSTONE S A. Handbook of Applied Cryptography[M]. CRC Press, 1996.

[13] DELESCAILLE J P, QUISQUATER J J. How easy is collision search? application to DES[A]. EUROCRYPT 1989 LNCS[C]. Houthalen, Belgium, 1990. 429-434.

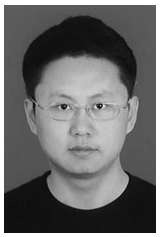
[14] SEDGEWICK R, SZYMANSKI T G, YAO A C. The complexity of finding cycles in periodic functions[J]. SIAM J Comput, 1982, 11(2): 376-390.

作者简介：

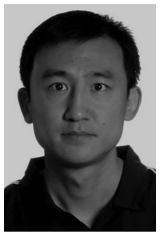


邹剑（1985-），男，福建福州人，中国科学院博士生，主要研究方向为散列函数分析。

吴文玲（1966-），女，陕西蒲城人，博士，中国科学院研究员、博士生导师，主要研究方向为私钥密码体制的设计与分析。



吴双（1983-），男，重庆人，博士，中国科学院助理研究员，主要研究方向为散列函数的分析与设计。



董乐（1980-），男，河南新乡人，中国科学院博士生，主要研究方向为散列函数和分组密码的分析。