一种基于分组密码的 hash 函数的安全性分析及构造

郭 伟 钱 进 王新军
(山东大学网络信息中心 济南 250100)

摘 要 利用已有的分组密码构造 hash 函数是一种非常方便的构造方法。早在 1993 年 Preneel 等人就对使用分组密码构造的 64 种 hash 函数进行了安全分类,这些 hash 函数统称为 PGV 体制,它们都是单倍分组长度的,即输出长度和分组长度相同。2002 年 Black 在他的论文中对这 64 种 hash 函数的安全性进行了严格的证明,证明其中的 20 种是安全的,其他是不安全的。随着计算技术的发展,人们感到单倍分组长度的 hash 函数的安全性不足,于是一些双倍分组长度的基于分组密码的 hash 函数被提了出来。但是其中的很多是不安全的。在 AsiaCrypt2006 上,一种使用了 5 个分组密码的双倍分组长度的 hash 函数被提了出来。作者声明这种构造方式是安全的,但没有给出安全性证明。本文对该体制进行了分析,发现其安全性并不理想,并针对本文的攻击提出了一种新的基于分组密码的 hash 函数,同时和 SHA-256 等 hash 函数的性能进行了对比。

关键词 hash 函数,分组密码,生日攻击

Construction and Analysis of Hash Function Based on Block Cipher

GUO Wei QIAN Jin WANG Xin-jun
(Network Center, Shandong University, Jinan 250100, China)

Abstract It is convenient to build hash functions on block ciphers. In 1993, Preneel etc. analyzed the security of 64 hash functions based on block ciphers which are single-block-length and named PGV schemes. In 2002, Black etc. formally proved the security of 64 PGV schemes. It is shown that 20 of them are secure and the others are not. With the development of computation technique, the security of single-block-length hash functions is not enough, therefore, some double-block-length schemes are proposed, however, many of them are not secure. In AsiaCrypt2006, a kind of hash function based on five block ciphers was proposed and it was claimed secure without security proofs. It is shown that the security of the scheme based on five block ciphers is not ideal. In this paper, a new hash function based on block ciphers is proposed and its efficiency is compared with SHA-26's.

Keywords Hash function, Block cipher, Birthday attack

1 hash 函数简介

随着计算机网络的不断普及,网络安全问题已经成为大家关注的焦点。多年来,网站运营者一直使用"hash 函数"这项技术把在网路上传递的数据(如信用卡信息、社保信息等等)打乱。hash 函数是一种非常重要的密码学工具,广泛地应用于数字签名、生成 MAC、伪随机函数、分组密码和模拟 random oracle 等领域。简单来说,hash 是这样一种函数,它接受任意长度的输入,然后输出特定长度的结果。一个安全的 hash 函数要满足抗碰撞攻击(collision attack resistant)、抗前像攻击(pre-image attack resistant)和抗第二前像攻击(second pre-image attack resistant)[6]。假设一个 hash 函数的输出长度为 n 比特,那么安全 hash 函数的抗碰撞攻击(collision attack resistant)、抗前像攻击(pre-image attack resistant)和抗第二前像攻击(second pre-image attack resistant)的复杂度分别为 $O(2^{\frac{n}{2}})$,$O(2^n)$ 和 $O(2^n)$。大多数实际使用的 hash 函数都是基于 Merkle 和 Damgård 提出的 Merkle-Damgård 方法(简称 MD 方法)[1,4]。这是一种使用固定输入输出长度的函数来构造任意长度输入和固定长度输出的 hash 函数的方法,这个固定的输入输出长度的函数叫做 hash 函数的压缩函数。MD 方法如下所示:

$H(h_0, m_1 \| \cdots \| m_k)$

For i=1 to k

$h_i = f(h_{i-1}, m_i)$

Return h_k

$H_0 \in \{0,1\}^n$ 是一个固定的常数,是 hash 函数的初始值(initial value);f 是 hash 函数的压缩函数;$M = m_1 \| \cdots m_k \in (\{0,1\}^m)^*$ 是输入的消息,分成 l 块,每块长度固定,如果长度不足则进行填充,最后一块需要保存原始消息的长度;h_i ($i = 1 \cdots k-1$) 是 hash 函数的中间变量(chain value);最后输出 h_k。MD 方法也叫迭代构造方法,Merkle 和 Damgård 证明,如果压缩函数是安全的,那么根据 MD 构造方法构造的 hash 函数也是安全的。所以 MD 方法的关键是压缩函数的构造。根据压缩函数构造方式的不同可以把基于 MD 方法构造的 hash 函数分为专门设计的 hash 函数(dedicated design)和基于分组密码的 hash 函数两大类。前者是使用专门为 hash 函数设计的压缩函数进行迭代,速度非常快,但是设计复杂而且没有一个可以利用的安全模型来证明 hash 函数的安全性,因为这一类 hash 函数的压缩函数都是基于以往安全分析的经验构造的;而后者利用已有分组密码作为压缩函数进行操作,由于现在分组密码很成熟,有很多可以选择的分组密码算法,因此构造 hash 函数非常简单方便,而且可以在证明 hash 函数安全性的时候使用分组密码已经存在的安全性假设,证明方便,但是分组密码毕竟不是为 hash 函数专门设计的,所以

郭 伟 硕士,研究方向为工作流、网络安全;钱 进 硕士,研究方向为面向对象数据库、数据仓库;王新军 教授。

这一类的 hash 函数一般来说效率都不如专门设计的 hash 函数高。

简单来说，基于分组密码的 hash 函数就是把 MD 构造方法里的压缩函数使用分组密码代替以后产生的 hash 函数。1993 年，Preneel 等人提出了 64 种单倍分组长度的基于分组密码的 hash 函数体制[2]（简称 PGV 体制），并对这些体制进行了分类，他们认为其中的 12 种体制是安全的，但是没有给出严格的证明。2002 年，Black 等人在黑盒子模型下（Black box model）对 PGV 体制的各个函数进行了安全性证明[3]，他们发现其中的 12 种体制的安全性是理想的，另外有 8 种抗碰撞攻击的复杂度是理想的，但是抗前像攻击的复杂度为 $O(2^{\frac{n}{2}})$，其余的 44 种体制是完全不安全的。这两篇文章完美地解决了使用分组密码构造单倍分组长度的 hash 函数的问题。由于计算能力的发展，单倍分组长度的 hash 函数的输出长度可能不能满足计算安全的需要，双倍分组长度的 hash 函数应运而生。双倍分组长度的 hash 函数采用两个分组密码，输出的长度为分组长度的两倍。这类 hash 函数分为两大类，*rate* 为 1 和 *rate* 小于 1。所谓 *rate*，就是调用一次分组密码处理的消息块数。Knudsen 证明了所有 *rate* 为 1 的双倍分组长度的 hash 函数的抗碰撞攻击的复杂度为 $O(2^{\frac{n}{2}})$，抗前像和第二前像攻击的复杂度为 $O(2^n)$，而安全的双倍分组长度的 hash 函数分别应该为 $O(2^n)$ 和 $O(2^n)$[6]（n 为 hash 函数的输出长度），这说明所有 *rate* 为 1 的双倍分组长度的 hash 函数是不安全的。*rate* 小于 1 的例如 MDC-2[10] 的 *rate* 为 $\frac{1}{2}$ 且已被采纳为国际标准。后来 Nandi 等人提出了 *rate* 为 $\frac{1}{3}$ 和 $\frac{2}{3}$ 的体制[7]，但是被证明是不安全的[6]。

在对基于分组密码的 hash 函数进行安全性分析时采用黑盒子模型（black box model），这种模型最早由 Shannon 在文献[8] 中提出并在以后的密码学分析中得到了广泛的应用。在分析基于分组密码的安全性时所使用的分组密码如定义 1 所示。

定义 1（分组密码） 分组密码是一个如下所示的映射 $E:\{0,1\}^k\times\{0,1\}^n\to\{0,1\}^n$，对不同的 K，$K\in\{0,1\}^k$，$E(K,m)$（$m\in\{0,1\}^n$）是一个在 $\{0,1\}^n$ 上的置换。

本文研究一类基于分组密码的 hash 函数的压缩函数在碰撞攻击（collision attack）和自由碰撞攻击（free-start collision attack）下的安全性。它们的定义如下。

定义 2（碰撞攻击） 已知压缩函数 C 和中间变量 H'，找到两个不同的消息 M 和 M'，使得 $C(H',M)=C(H',M')$。

定义 3（自由碰撞攻击） 已知压缩函数 C，找到两个不同的对 (H',M) 和 (H'',M')，使得 $C(H',M)=C(H'',M')$。

如果压缩函数是安全的，则必须要满足抗碰撞攻击和抗前像攻击，后来 Lai 等人指出，如果一个压缩函数不能抗自由碰撞攻击或自由前像攻击，那么由这种压缩函数迭代构造的 hash 函数可能存在安全缺陷[5]。

2 对一类 hash 函数的攻击和分析

2.1 体制描述

在 Asiacrypt2006 上，Peyrin 等人提出了一个 *rate* 为 $\frac{1}{5}$ 和 $\frac{2}{5}$ 的基于分组密码的 hash 函数体制[9]。*rate* 为 $\frac{1}{5}$ 的体制如图 1 所示。

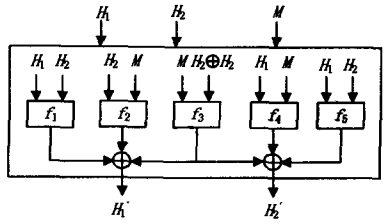

图 1 rate 为 $\frac{1}{5}$ 的体制示意图

$F_1(H_1,H_2,M)=f_1(H_1,H_2)\oplus f_2(H_2,M)\oplus f_3(M,H_1\oplus H_2)$

$F_2(H_1,H_2,M)=f_3(M,H_1\oplus H_2)\oplus f_4(H_1,M)\oplus f_5(H_1,H_2)$

$F(H_1,H_2,M)=F_1(H_1,H_2,M)\parallel F_2(H_1,H_2,M)$

记为体制 1，H_1，H_2 是中间变量（chain value），M 为所处理的消息。*rate* 为 $\frac{2}{5}$ 的体制如图 2 所示。

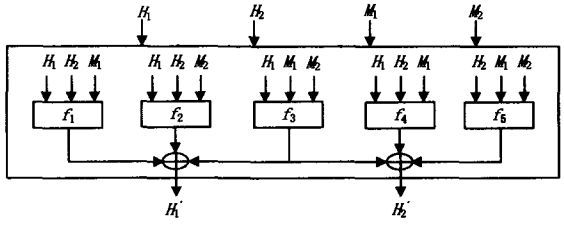

图 2 rate 为 $\frac{2}{5}$ 的体制示意图

$F_1(H_1,H_2,M_1,M_2)=f_1(H_1,H_2,M_1)\oplus f_2(H_1,H_2,M_2)\oplus f_3(H_1,M_1,M_2)$

$F_2(H_1,H_2,M_1,M_2)=f_3(H_1,M_1,M_2)\oplus f_4(H_1,H_2,M_1)\oplus f_5(H_2,M_1,M_2)$

$F(H_1,H_2,M_1,M_2)=F_1(H_1,H_2,M_1,M_2)\parallel F_2(H_1,H_2,M_1,M_2)$

记为体制 2，与体制 1 不同的是，体制 2 的压缩函数采用了两个消息块。Peyrin 等声称这两个体制是安全的，但是没有给出安全性证明。下面我们给出这两个体制碰撞攻击的复杂度。

在对这两个体制进行攻击之前，我们对将要用到的符号进行一下定义。$\leftarrow$ 表示赋值操作，$\overset{\$}{\leftarrow}$ 表示从右边所给出的范围里随机选取一个值赋给左边的变量，n 表示分组长度，$Add(T,H)$ 表示向表 T 中添加一个元素 h，$Sort(T)$ 表示对表 T 中的元素进行排序（降序或升序）。

2.2 对体制 1 的自由碰撞攻击

我们对体制 1 构造一个自由碰撞攻击，攻击描述如下。

1. $H_2\leftarrow$Constant
2. For i=1 to $2^{\frac{5n}{6}}$
3. $H_1\overset{\$}{\leftarrow}\{0,1\}^n$; $M\overset{\$}{\leftarrow}\{0,1\}^n$;
4. $Add(T_H,H_1)$; $Add(T_M,M)$;
5. $A\leftarrow f_1(H_1,H_2)$; $B\leftarrow f_2(H_2,M)$;
6. $Add(T_A,A)$; $Add(T_B,B)$;
7. End For
8. $D\leftarrow X\parallel\alpha(|D|=n,\alpha\overset{\$}{\leftarrow}\{0,1\}^{\frac{5n}{6}},X\overset{\$}{\leftarrow}\{0,1\}^{\frac{n}{6}})$;
9. For every $A\in T_A$
10. $A=A\oplus D$;
11. End For

12. Sort (T_A);[同时对 T_{H_1} 中相应位置的元素进行排序,使得 T_{H_1} 和 T_A 中元素保持对应]
13. For every $B \in T_B$
14. If ($B \in T_A$) then
15. Add(T_C,(H_1,M))[这里(H_1,M)满足 $H_1 \in T_{H_1}$, $M \in T_M$]
16. End If
17. End For
18. Find $2^{\frac{2n}{3}}$- way collisions for the least significant $\frac{2n}{3}$ bits of f_3 with the elements in T_C, then the $\frac{2n}{3}$ bits of output is fixed and store these elements in T_{Mul}.
19. Use the $\frac{2n}{3}$ elements in T_{Mul} to find the collision for the remaining $\frac{4n}{3}$ bits of the output.

下面来分析这个攻击的复杂度。第 1 步到第 7 步的复杂度为 $O(2^{\frac{5n}{6}})$,空间复杂度为 $O(2^{\frac{5n}{6}})$;9—11 步的复杂度为 $O(2^{\frac{5n}{6}})$;12 步是排序,快速排序的时间复杂度为 $O(2^{\frac{5n}{6}}\log 2^{\frac{5n}{6}})=O(n2^{\frac{5n}{6}})$;13—16 步如果采用折半查找方法时复杂度为 $O(n2^{\frac{5n}{6}})$;18 步根据 Joux 的分析[11]和定理 1 的证明其复杂度为 $O(2^{\frac{2n}{3}})$,那么表 T_{Mul} 中有 $2^{\frac{2n}{3}}$ 个元素使得 $f_1 \oplus f_2 \oplus f_3$ 的最低 $\frac{2n}{3}$ 比特是确定的;19 的复杂度为 $O(2^{\frac{2n}{3}})$(生日攻击)。综上所述,这个碰撞攻击的复杂度为 $O(n2^{\frac{5n}{6}})$,如果 $n=256$,这个复杂度大约为 $O(2^{\frac{5n}{6}})$。

定理 1 假设 $f(H,M)$ 是 hash 函数的压缩函数,输出长度为 n 比特,如果 f 是理想的,则对其寻找 r-way 碰撞即 $f(H_1,M_1)=f(H_2,M_2)=\cdots=f(H_r,M_r)$ 的复杂度为 $O(2^{\frac{(r-1)n}{r}})$。

证明:$r=2$ 时,随机选取二元组(H,M),计算 $f(H,M)$,记第 i 次计算的结果为 f_i,记事件 C_i 为 f_i 等于 $f_1,\cdots,f_{i-1}$ 中的一个。$P(C_i)=\frac{i-1}{2^n}$,则经过 q 次计算产生 2 个碰撞个概率为 $P(C_1 \vee C_2 \vee \cdots \vee C_q)=\sum_{i=1}^{q}\frac{i-1}{2^n}\leqslant\frac{q^2}{2^{n+1}}$,如果 $q=O(2^{\frac{n}{2}})$,产生碰撞的概率很高。

$r=3$ 时,记事件 C_i 为第 i 次计算产生 3 个碰撞($i\geqslant 3$)。C_i 由两个事件组成,即前面 $i-1$ 次计算产生两个碰撞的事件 C'_i 和 f_i 的值等于这两个碰撞的事件 C''_i。由上述 $r=2$ 的证明可知,$P(C'_i)=\sum_{k=1}^{i-1}\frac{k-1}{2^n}$,由于压缩函数是理想的 $P(C''_i)=\frac{1}{2^n}$。$P(C_i)=(C'_i \wedge C''_i)=\frac{1}{2^n}\sum_{k=1}^{i-1}\frac{k-1}{2^n}$。则经过 q 次计算产生 3 个碰撞个概率为 $P(C_3 \vee C_4 \vee \cdots \vee C_q)=\sum_{i=3}^{q}\frac{1}{2^n}\sum_{k=1}^{i-1}\frac{k-1}{2^n}\leqslant\sum_{i=3}^{q}\frac{i^2}{2^{2n+1}}\leqslant\frac{q^3}{2^{2n+1}}$,如果 $q=O(2^{\frac{2n}{3}})$,产生碰撞的概率很高。当 $r=4,5,6\cdots$ 时依次类推。

2.3 对体制 2 的碰撞攻击

体制 2 和体制 1 的区别是使用了两个消息块,这样在构造碰撞攻击时就具有更大的灵活性。其攻击如下所示。

1. $H_1 \longleftarrow Constant_1$; $H_2 \longleftarrow Constant_2$;
2. For i=1 to $2^{\frac{5n}{6}}$
3. $M_1 \xleftarrow{\$} \{0,1\}^n$; $M_2 \xleftarrow{\$} \{0,1\}^n$;
4. Add(T_{M_1},M_1); Add(T_{M_2},M_2);
5. $A \longleftarrow f_1(H_1,H_2,M_1)$; $B \longleftarrow f_2(H_1,H_2,M_2)$;
6. Add(T_A,A); Add(T_B,B);
7. End For
8. $D \longleftarrow XP\alpha(|D|=n, \alpha \xleftarrow{\$} \{0,1\}^{\frac{5n}{6}}, X \xleftarrow{\$} \{0,1\}^{\frac{n}{6}})$;
9. For every $A \in T_A$
10. $A=A \oplus D$;
11. End For
12. Sort (T_A);[同时对 T_{M_1} 中相应位置的元素进行排序,使得 T_{M_1} 和 T_A 中元素保持对应]
13. For every $B \in T_B$
14. If ($B \in T_A$) then
15. Add(T_C,(M_1,M_2))[这里(M_1,M_2)满足 $M_1 \in T_{M_1}$, $M_2 \in T_{M_2}$]
16. End If
17. End For
18. Find $2^{\frac{2n}{3}}$- way collisions for the least significant $\frac{2n}{3}$ bits of f_3 with the elements in T_C, then the $\frac{2n}{3}$ bits of output is fixed and store these elements in T_{Mul}.
19. Use the $2^{\frac{2n}{3}}$ elements in T_{Mul} to find the collision for the remaining $\frac{4n}{3}$ bits of the output

复杂度分析同上,大约为 $O(2^{\frac{5n}{6}})$。

综上所述,体制 1 和体制 2 都不是理想的压缩函数,它们在碰撞攻击下的复杂度都没有达到 $2n$(n 为分组长度)这个理想的界。Nandi 在文献[7]使用与上述两种体制类似的方式构造了两个基于分组密码的压缩函数,它使用了 3 个分组密码,Knudsen 证明了这两个体制是不安全的[6],本文使用的方法和文献[6]中的类似。可以看到这两类体制都采用了对分组密码的输出进行异或得到输出的方法,这种方法如果不能保证参加异或的分组密码相互独立,则不可避免地会存在上述的攻击。

3 构造新的基于分组密码的 hash 函数

3.1 新体制描述

针对异或的缺点,我们提出一种新的基于分组密码的 hash 函数,记为体制 3,如下所示(示意图见图 3)。

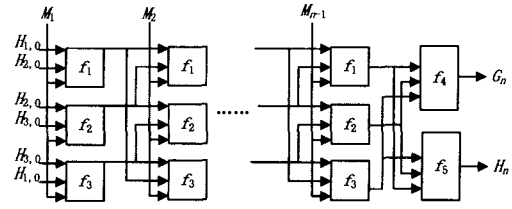

图 3 新的基于分组密码的 hash 函数示意图

For i=1 to n do

$H_{1,i}=f_1(H_{1,i-1},H_{2,i-1},M_i)$

$H_{2,i}=f_2(H_{2,i-1},H_{3,i-1},M_i)$

$H_{3,i}=f_3(H_{3,i-1},H_{1,i-1},M_i)$

End For

$G_n=f_4(H_{1,n-1},H_{2,n-1},H_{3,n-1})$

$H_n=f_5(H_{3,n-1},H_{2,n-1},H_{1,n-1})$

$H=G_n \| H_n$

这里 f_1, f_2, f_3, f_4, f_5 是分组密码，n 代表消息块数，hash 函数有三个初始值 $H_{1,0}, H_{2,0}$ 和 $H_{3,0}$，最终的输出结果为两个分组长度，在最终结果之前输出长度为三个分组长度。

3.2 安全性分析

为了便于证明，我们把体制 3 里的分组密码使用双倍密钥长度的分组密码体制代替：$f_1 = E_{H_1 \| H_2}(M)$，$f_2 = E_{H_2 \| H_3}(M) \oplus M$，$f_3 = E_{H_3 \| H_1}(M) \oplus M$。可以看出，如果是单倍分组长度的压缩函数 f_1 是完全不安全的。对于不同的 (H_1, H_2) 和 (H'_1, H'_2)，计算 $M_i = E^{-1}_{H_1 \| H_2}(v)$ 和 $M'_i = E^{-1}_{H'_1 \| H'_2}(v)$（$v$ 是一个任意的常量），则 (H_1, H_2, M) 和 (H'_1, H'_2, M') 是一对碰撞。但是用这样的分组密码构造的体制 3 是安全的。这里我们使用 H_1, H_2, H_3 和 M 分别作为压缩函数的输入中间变量和消息块（下同），假设压缩函数的输出长度为 $2n$ 比特。

定理 2 体制 3 的碰撞攻击的复杂度为 $O(2^n)$。

证明：假设对 f_1 进行第 i 次上述计算得到了第 i 个产生碰撞的三元组 (H_1^i, H_2^i, M^i)，$E_{H_2^i \| H_3^i}(M^i) \oplus M^i = y_{2,i} \oplus M^i$，$E_{H_3^i \| H_1^i}(M^i) \oplus M^i = y_{3,i} \oplus M^i$，则对 f_2 和 f_3 计算 i 次后得到的输出列表分别为 $(y_{2,1} \oplus M^1, \cdots, y_{2,i} \oplus M^i)$ 和 $(y_{3,1} \oplus M^1, \cdots, y_{3,i} \oplus M^i)$。假设事件 C_i 为 $\exists j \in [1, i-1]((y_{2i} \oplus M^i = y_{2j} \oplus M^j) \wedge (y_{3i} \oplus M^i = y_{3j} \oplus M^j))$，由于分组密码 E 是如定义 1 所定义的置换，因此 $P(C_i) = \frac{i-1}{(2^n-(i-1))^2}$。则对 f_1 计算 q 次后，事件 C_i 发生的概率为 $\sum_{i=1}^{q} \frac{i-1}{(2^n-(i-1))^2} \leqslant \sum_{i=1}^{q} \frac{i}{(2^n-2^{n-1})^2} = \frac{q(q+1)}{2^{2n-1}}$，得证。

定理 3 体制 3 的前像攻击的复杂度为 $O(2^{2n})$。

证明：对于 f_1 而言，已知其前像 G_1，求 H 的前像只需令 $M = E^{-1}_{H_1 \| H_2}(G_1)$，很明显 $E_{H_1 \| H_2}(M) = G_1$。随机选择 H_1 和 H_2 就会产生不同的 M 使得 $E_{H_1 \| H_2}(M) = G_1$。假设对 f_1 进行第 i 次上述计算得到了第 i 个前像三元组 (H_1^i, H_2^i, M^i)，f_2 和 f_3 的前像分别为 G_2 和 G_3，f_2 和 f_3 的输出列表分别为 $(y_{2,1} \oplus M^1, \cdots, y_{2,i} \oplus M^i)$ 和 $(y_{3,1} \oplus M^1, \cdots, y_{3,i} \oplus M^i)$。假设事件 C_i 为 $((y_{2i} \oplus M^i = G_2) \wedge (y_{3i} \oplus M^i = G_3))$，由于分组密码 E 是如定义 1 所定义的置换，因此 $P(C_i) = \frac{1}{(2^n-(i-1))^2}$。则计算 q 次后，事件 C_i 发生的概率为 $\sum_{i=1}^{q} \frac{1}{(2^n-(i-1))^2} \leqslant \frac{q}{(2^n-q)^2}$，得证。

3.3 性能比较

假设输入消息块数为 m，则新体制的 *rate* 为 $\frac{m}{3m+2}$，如果 m 很大，这个值接近 $\frac{1}{3}$，略小于 $\frac{2}{5}$。新的体制可以用 AES-256 实现，表 1 列举了新体制和 Peyrin 体制以及 SHA-256 的速度对比。我们构造的体制在可证明安全的前提下速度比 *rate* 为 $\frac{2}{5}$ 的体制 2 要慢。

表 1 速度对照表

体制	所采用的分组密码	每秒钟运算次数	运算 10^6 次所耗费时间(秒)	所处理的数据长度(bit)
体制 1	AES-128	90909.09	11	512
体制 2	AES-256	142857.14	7	512
体制 3	AES-256	111111.11	9	512
SHA-256	无	333333.00	3	512

所有代码为 C 语言实现，测试平台平台为：DELL OPTIPLEX 170L CPU：Celeron 2.66GHz；硬盘：80G，7200 转，2M 缓存；内存：ddr333 256M。从表 1 可以看出 SHA-256 速度最快，体制 2 次之，体制 1 最慢，体制 3 的 rate 大于体制 1 小于体制 2，速度介于两者之间。考虑到 AES-256 的密钥编排和加密速度比 AES-128 慢很多，所以如果能够使用 AES-128 来实现体制 2 和体制 3 的话，速度应该会提高很多。表 2 给出了 AES-256 和 AES-128 密钥编排和加密速度的对比。

表 2 AES-256 和 AES-128 速度对照表

分组密码	每秒钟密钥编排次数	每秒钟加密次数
AES-128	500000.00	250000.00
AES-256	250000.00	166666.66

测试平台同表 1。虽然使用分组密码构造的 hash 函数的效率不如专门设计的 hash 函数高，但是实际测试表明专门设计的 hash 函数的速度并不比一些 *rate* 比较低的基于分组密码的 hash 函数快太多。

结束语 本文分析了 Asiacrypt2006 上提出的两种基于分组密码的 hash 函数的压缩函数的抗碰撞安全性，针对存在的问题提出了一个新的基于分组密码的 hash 函数，然后使用 C 语言实现了这几种 hash 函数并把这些体制的速度和 SHA-256 的速度进行了对比。结果发现，SHA-256 的速度并没有比这些效率不高的基于分组密码的 hash 函数快太多。这主要是由于 AES 的加密速度非常快，这给构造基于分组密码的 hash 函数带来了机会。AES 的出现，使得人们不必和以往一样要构造 *rate* 高的 hash 函数，*rate* 低的 hash 函数的速度也可能合乎要求。

参 考 文 献

[1] Damgård I B. A design principle for hash functions // Advances in Cryptology-Crypto' 89, LNCS 435. Springer-Verlag, 1989: 416-427

[2] Preneel B, Govaerts R, Vandewalle J. Hash functions based on block ciphers: A synthetic approach // Advances in Cryptology-Crypto'93, LNCS 773. Springer-Verlag, 1994: 368-378

[3] Black J, Rogaway P, Shrimpton T. Black-box analysis of the block-cipher based hash function constructions from PGV // Advances in Cryptology-Crypto'02, LNCS 2442. Springer-Verlag, 2002: 320-335

[4] Merkle R. One way hash functions and DES // Advances in Cryptology-Crypto'89, LNCS 435. Springer-Verlag, 1989: 428-446

[5] Lai Xuejia, Massey J. Hash functions based on block cipehers // Advance in Cryptology-EUROCRYPT' 92 Proceedings, LNCS 658. Springer-Verlag, 1993: 55-70

[6] Knudsen L, Lai X, Preneel B. Attacks on fast double block length hash functions. Journal of Cryptology, 1998, 11(1)

[7] Nandi M, Lee W, Sakurai K, et al. Security analysis of a 2/3-rate double length compression function in the black-box model // FSE2005, LNCS 3557. ENSTA, 2005: 243-254

[8] Shannon C. Communication theory of secrecy systems. Bell Systems Technical Journal, 1949: 656-715

[9] Peyrin T, Gilbert H, Muller F, et al. Combining Compression Functions and Block Cipher-Based Hash. 已录用于 Asiacrypt2006

[10] Brachtl B, Coppersmith D, Hyden M, et al. Data authentication using modification detection codes based on a public one way encryption function. U. S. Patent Number 4,908,861, March 1990

[11] Joux A. Multicollisions in Iterated Hash Functions Application to Cascaded Constructions // CRYPTO 2004, LNCS 3152. Springer-Verlag, 2004: 306-316