第 35 卷 第 6 期 计 算 机 工 程 2009 年 3 月
Vol.35 No.6 Computer Engineering March 2009
·安全技术· 文章编号：1000—3428(2009)06—0181—02 文献标识码：A 中图分类号：TP309

一类 SHA-x 改进杂凑算法的设计及分析

戴慧珺[1]，董文瀚[2]，钟世刚[2]
(1. 西北工业大学软件与微电子学院，西安 710065；2. 空军工程大学工程学院，西安 710038)

摘 要：在 SHA-1 和 SHA-2 标准算法的基础上，提出一类 SHA-x 改进杂凑算法的设计。该算法重新设计了杂凑函数 Hash 值的生成方法，将输出消息摘要的长度从 SHA-1 的 160 bit 提高到 192 bit，同时保留了 SHA-1 模 2^{32} 加法的计算特性，以保证整个算法的加密速度。安全性分析表明，新设计的杂凑算法在不过分减慢加密速度的前提下，具有较 SHA-1 更好的抗攻击能力。

关键词：密码；杂凑算法；消息摘要；碰撞

Design and Analysis of Modified SHA-x Hash Algorithm

DAI Hui-jun[1], DONG Wen-han[2], ZHONG Shi-gang[2]
(1. College of Software and Microelectronics, Northwestern Polytechnical University, Xi'an 710065;
2. Engineering Institute, Air Force Engineering University, Xi'an 710038)

【Abstract】This paper proposes a modified SHA-x Hash algorithm on the basis of the Secure Hash Algorithms(SHA-1 and SHA-2) defined in FIPS PUB 180-2. The new algorithm increases the message digest length from 160 bit of SHA-1 to 192 bit. As a result, in contrast to the standard SHA-1 algorithm, the main advantage of the new scheme is that the number of bits of security provided for the data being hashed is improved. Secure analysis shows that since the computation method of addition modulo 2^{32} is kept, the computation speed of the modified algorithm will not be slowed down excessively.

【Key words】 cryptography; Hash algorithm; message digest; collision

1 概述

文献[1]提出了一类 MDx[2-3]系列杂凑算法的快速碰撞方法，该文同时指出对 SHA-0 算法[4]的碰撞大约能在 2^{40} 次运算后找到。这一结果表明，虽然完整的破解方案暂时尚未被提出，但替代 SHA-0 成为联邦信息处理标准的 SHA-1[5]的安全性问题正逐步地显露出来。尽管在另一方面，其他更长更安全的算法，如 SHA-256, -384, -512(称为 SHA-2)，已在 2002 年由 NIST 制定标准[5]，但这些基于复杂的大数运算和多重迭代的算法使得加密过程过于复杂和漫长，对于普通的个人或小型商业用户，这种过于安全的加密无疑是一种浪费。因此，该领域目前存在的一类问题在于：如何寻找一类较 SHA-1 更为安全，同时不过分减慢加密速度的杂凑算法，使之适用于个人和小型用户。

本文在 SHA-1 和 SHA-2 的基础上，提出一类 SHA-x 改进杂凑算法。

2 算法设计

SHA 系列算法是 NIST 公布的标准安全杂凑算法[5]，其可以对长度为 l 位的消息产生一个称之为消息摘要压缩的表达。该算法具有迭代、单向的特点，从而能够用以保证消息的完整性：即消息的任何改变，将导致消息摘要的巨大改变。杂凑算法的这一特点使得其被广泛应用于数字签名、消息认证码、随机数产生等领域。

从文献[5-7]可以看到，杂凑算法的安全性和其生成的消息摘要的长度有很大的关系，一般来说，生成的摘要越长，算法的安全性越高[7]。标准的 SHA-1 杂凑算法最终产生了一个长度为 5 个 32 bit 字的 Hash 值，即 160 bit 的消息摘要，因为摘要长度比 MD5 长 32 bit 而被认为具有相当高的安全性。由于 SHA-1 和 MD5 均由 MD4 算法导出，在目前 MD5 等杂凑算法面临被攻陷危机的情况下[1]，设计一种较 SHA-1 更安全的杂凑算法成为该领域的重要研究方向之一。为此，针对个人和小型商业用户，本文在 SHA-1 和 SHA-2 的基础上，提出如下一类 SHA-x 改进算法的设计思路：

(1)将消息摘要的长度增加到 192 bit，以获得较 SHA-1 更高的抗碰撞性。

(2)保留 SHA-1 模 2^{32} 加法的计算特性，从而不过度减慢整个算法的加密速度。

(3)改进算法和 SHA-1 类似，也分为预处理和消息摘要的 Hash 计算 2 个部分。

2.1 预处理

预处理的目的是使填充后的消息总长为 512 的倍数，本文采用的预处理方法和 SHA-1 完全一致，为保持完整性，简单介绍如下：

(1) 填充消息：将长度为 l (同 SHA-1, $l < 2^{64}$)的消息 M，在其结尾依次加上一个“1”，k 个“0”和一个 64 bit 的整数，填充成一个长度为 $512 \times N$ 位的消息。这里，正整数 k 为

$$(l+1+k) \bmod 512 = 448 \quad (1)$$

的最小解；64 bit 整数用以表示消息的初始长度。

(2)将填充后的消息按顺序切割成 N 个块 $M^{(i)}$的序列，每

作者简介：戴慧珺(1979－)，女，讲师、硕士，主研方向：软件技术，数据库，信息安全技术；董文瀚，讲师、博士后；钟世刚，讲师、硕士

收稿日期：2008-08-12 **E-mail：**xiaodai@nwpu.edu.cn

个块 $M^{(i)}$包括 16 个字，即 512 bit。

2.2 消息摘要的 Hash 计算

对于 i = 1 到 N，做如下的循环：

(1)准备消息列表 W_t：

$$W_t=\begin{cases}M_t^{(i)} & 0\le t\le 15\\ R_{OTL}^{\ 1}(W_{t-3}\oplus W_{t-8}\oplus W_{t-14}\oplus W_{t-16}) & 16\le t\le 79\end{cases} \tag{2}$$

其中，$M_t^{(i)}$, $0\le t\le 15$ 表示第 i 个数据块的第 t 个字；$R_{OTL}^{\ 1}$表示按位左移 1 bit；⊕表示按位“异或”逻辑运算。

(2)工作变量初始化：

$$\begin{cases}A=H_0^{(i-1)} & B=H_1^{(i-1)}\\ C=H_2^{(i-1)} & D=H_3^{(i-1)}\\ E=H_4^{(i-1)} & F=H_5^{(i-1)}\end{cases} \tag{3}$$

其中，A-F 表示 6 个长度为 32 bit 字的工作变量；$H_0^{(i-1)}$, $H_2^{(i-1)}$,…, $H_5^{(i-1)}$表示上一轮计算的中间 Hash 值，当 i = 1 时，$H_0^{(0)}$, $H_1^{(0)}$,…,$H_5^{(0)}$为常初始值，参考 SHA-256，不妨取为

$$\begin{cases}H_0^{(0)}=0x6a09e667 & H_1^{(0)}=0xbb67ae85\\ H_2^{(0)}=0x3c6ef372 & H_3^{(0)}=0xa54ff53a\\ H_4^{(0)}=0x510e527f & H_5^{(0)}=0x9b05688c\end{cases} \tag{4}$$

初始值 $H_i^{(0)}$的选取可以是任意的，其在具体实现中类似于 SHA-1 以小数在前格式存储，即字的最低位字节放在低地址字节上。

(3)对于 t = 0 到 79，做如下变换：

$$\begin{cases}T=R_{OTL}^{\ 5}(A)+f_t(B,C,D)+E+F+W_t+K_t\\ F=E\\ E=D\\ D=C\\ C=R_{OTL}^{\ 30}(B)\\ B=A\\ A=T\end{cases} \tag{5}$$

其中，T 表示一个中间变量；K_t为一个 80 个 32 bit 字的常数序列，定义为

$$K_t=\begin{cases}0x5a827999 & 0\le t\le 19\\ 0x6ed9eba1 & 20\le t\le 39\\ 0x8f1bbcdc & 30\le t\le 59\\ 0xca62c1d6 & 40\le t\le 79\end{cases} \tag{6}$$

逻辑函数 f_t 定义为

$$f_t=\begin{cases}Ch(x,y,z)=(x\wedge y)\oplus(\neg x\wedge z) & 0\le t\le 19\\ P_{arity}(x,y,z)=x\oplus y\oplus z & 20\le t\le 39\\ M_{aj}(x,y,z)=(x\wedge y)\oplus(x\wedge z)\oplus(y\wedge z) & 40\le t\le 59\\ P_{arity}(x,y,z)=x\oplus y\oplus z & 60\le t\le 39\end{cases} \tag{7}$$

其中，符号“¬”表示“补码”逻辑运算；“∧”表示“与”逻辑运算。

(4)计算第 i 个中间 Hash 值：

$$\begin{cases}H_0^{(i)}=A+H_0^{(i-1)}\\ H_1^{(i)}=B+H_1^{(i-1)}\\ H_2^{(i)}=C+H_2^{(i-1)}\\ H_3^{(i)}=D+H_3^{(i-1)}\\ H_4^{(i)}=E+H_4^{(i-1)}\\ H_5^{(i)}=F+H_5^{(i-1)}\end{cases} \tag{8}$$

在重复步骤(1)~步骤(4) N 次后，即处理完 $M^{(N)}$后，生成的 192 bit 消息摘要 M 为

$$H_0^{(N)}\,\|\,H_1^{(N)}\,\|\,H_2^{(N)}\,\|\,H_3^{(N)}\,\|\,H_4^{(N)}\,\|\,H_5^{(N)} \tag{9}$$

3 算法分析

将改进算法和标准的 SHA-1 和 SHA-256 做如表 1 所示的对比。

表 1 改进算法和 SHA-1, SHA-256 的对比

算法	消息最大长度/bit	基本处理单元/bit	摘要长度/bit	算法步骤/轮	安全位/bit
SHA-1	2^{64} - 1	512	160	80	80
SHA-256	2^{64} - 1	512	256	64	128
改进算法	2^{64} - 1	512	192	80	96

表 1 中安全位的定义见文献[5]，指一个长度为 n 的消息摘要对强抗冲突性的生日攻击[8]大约需要 $2^{n/2}$ 次的计算量。

3.1 抗攻击性质

由上述的设计过程和表 1 不难得出新设计的改进算法具有以下对强行攻击的性质：

(1)单向性：对于任何给定的码 h，寻找 x 使得 $H(x)=h$ 的计算复杂度为 $O(2^{192})$。

(2)弱抗冲突性质：对任何给定的分组 x，寻找不等于 x 的 y，寻找使得 $H(y)=H(x)$的计算复杂度为 $O(2^{192})$。

(3)强抗冲突性质：寻找任何的(x, y)对，寻找使得 $H(x)=H(y)$的计算复杂度为 $O(2^{96})$。

从以上 3 个性质可以看出，由于改进算法与 SHA-1 最重要的区别在于前者的摘要比后者长 32 bit，用强行技术产生任何一个消息使其摘要等于给定的消息摘要，使用改进算法将提高 2^{32} 数量级的操作。此外，使用强行技术，产生具有相同消息摘要的 2 个消息的难度，改进算法将操作的数量级从 SHA-1 的 2^{80} 提高到 2^{96}。可见，新设计的算法对强行攻击具有更大的强度。

另一方面，从密码分析攻击的角度来看，一般讲来，一个理想的 Hash 函数应该具有以下的特性：使用密码分析攻击的难度应该大于或等于强行攻击的难度[9]。虽然 SHA-1, SHA-2 对此的有关设计标准几乎没有公开，其强度难以判断，但可以肯定的是，SHA-1, SHA-2 算法应该具有 Hash 函数的这一理想特性[7]。因此，作为其改进算法，新算法能够秉承该特性。

3.2 计算速度

不能回避的事实是，尽管新算法和 SHA-1 有相同的计算步骤(如表１)，但其每一步都需要处理 192 bit 的缓存，相比之下，SHA-1 仅处理 160 bit，因此，新算法实际上牺牲了计算速度来获得更高的安全性，但是，由设计步骤可以看出，改进算法继承了 SHA-1 模 2^{32} 加法的计算特性，在相同的计算环境下，其计算速度与 SHA-1 应具有相同数量级，参照 2.1 节中所得到的安全性能，可以得到如下的结论：新设计的改进算法能够以较小的速度牺牲换取较大的安全特性。

4 算例

本节给出改进算法的一个算例。令消息 M 为一个长度 l=24 bit 的 ASCII 字符串“abc”，即二进制字符串

"01100001 01100010 01100011" (10)

将其按式(1)~式(9)生成消息摘要的过程如下：

(1)预处理

将消息 M 填充 1 个“1”和 k 个“0”。这里，k=423 如式 (1) 取值；附加的 64 bit 16 进制码为“00000000 00000018”，表示原始消息的的长度为 24 bit；此时，填充后的消息包括 N=1 个块 $M^{(1)}$，其包含的 16 个字如下：

$$\begin{cases}W0=61626380 & W1=00000000 & W2=00000000 & W3=00000000\\ W4=00000000 & W5=00000000 & W6=00000000 & W7=00000000\\ W8=00000000 & W9=00000000 & W10=00000000 & W11=00000000\\ W12=00000000 & W13=00000000 & W14=00000000 & W15=00000018\end{cases} \tag{11}$$

(2)按式(3)、式(4)对寄存器进行初始化。

(下转第 185 页)