计算机科学 1997Vol. 24№. 4

一种新型HASH函数的设计

A New Design of Hash Function

李 文 郭宝安
(清华大学计算机系 北京 100084)

摘 要 some design techniques of SHA are showed in this paper, which is a famous Hash Function and has some good features. Two problems are discussed about it, then a new design of hash function based on them is proposed.

关键词 Cryptography, Digital Signature and Authentication, Hash Function

一、Hash函数的设计

Hash函数经常使用在密码技术中，用以进行数据签名和身份验证。它通常有三种实现方式：(1)使用数学上的单向函数。例如：基于因子分解或离散对数问题的Hash函数，往往具有很好的密码学性质，且满足Hash函数的单向、无碰撞基本要求。但是由于其巨大的计算量，速度太慢，不能实用。(2)使用分组密码系统。例如：通过DES或IDEA等高强度密码系统的级联，可以实现性能较好的Hash算法。但这类算法依赖于密钥，如果加密和Hash压缩使用同样的密码体制，会带来严重的安全问题。(3)基于软件的Hash算法。利用计算机软件系统的简单变换和函数，通过圈函数迭代，同样可以得到安全的Hash函数，例如著名的MD4，MD5[2-5]，SHA[1]，这类算法不依赖于密码系统，可以和各种密码系统联合使用，便于软件、硬件实现。我们主要以SHA算法为例，对第三类Hash进行研究。

SHA称为安全散列算法，由美国国家标准与技术局与美国安全局一道设计且与数字签名标准DSA一起使用。它的基本步骤如下：

step1：填充消息，使消息长度为512的倍数，最后32个比特是原消息的长度。

step2：初始化A,B,C,D,E五个双字节(32比特)的无符号整数，进入一个循环，每次处理512比特的消息块。

```
BEGIN
  AA=A,BB=B,CC=C,DD=D,EE=E;
  将16个双字节字扩展到80个双字节
  ROUND1;
  ROUND2;
  ROUND3;
  ROUND4;
  A=A+AA,B=B+BB,C=C+CC,D=D+DD,E=E
  +EE;
END
```

step3：将A,B,C,D,E级联输出，即得到长为160比特的报文摘要。其中：

1)数据扩展算法：

$W_t=M_t$, for $t=0$ to 15

$W_t=W_{t-3}\oplus W_{t-8}\oplus W_{t-14}\oplus W_{t-16}$, $t=16$ to 79

2)每个ROUND顺次操作20个消息字，每轮的运算所用公式形式相同，只是所用参数不同。公式如下：

```
TEMP=(A<<<5)+f_t(B,C,D)+E+W_t+K_t
E=D
D=C
C=(B<<<30)
B=A
A=TEMP
```

其中K_t是常数，每轮变化一次，f_t每轮对应一个三元函数，函数值仍然是一个双字节的整数。每轮对应的函数表达式如下：

$f_1(X,Y,Z)=XY+\sim XZ$
$f_2(X,Y,Z)=X\oplus Y\oplus Z$
$f_3(X,Y,Z)=XY+XZ+YZ$
$f_4(X,Y,Z)=X\oplus Y\oplus Z$

上式中的+,⊕,~都是按位进行运算。

关于SHA安全性的一些已有讨论[1]是：SHA与MD4非常相似，但作了一些改变，增加了安全强度。(1)增加了第四轮。(2)每一步加上了上一步的结果，这将引起更快的雪崩效应，而且SHA增加了第五个变量，使得用于攻击MD5的den Boer-Bosselaers[6]的攻击无效。而且，增加了摘要的长度，增大了生日攻击一类攻击方法的难度。(3)数据扩展采用循环纠错码的形式使每一轮中的输入字W_t都不同。可以认为：SHA=MD4+"扩展的"转换+附加轮+更好的雪崩

二、SAC性质与明文扩展

1. 每轮布尔函数f_t的性质

首先引入几个基本概念与定义。用V_n表示元素

取自GF(2)中的n维向量空间。向量α的Hamming重量用W(α)表示，也就是α中1的个数。

定义1　V_n上的一个函数f，如果向量$(f(\alpha_0),f(\alpha_1),\cdots,f(\alpha_{2^n-1})$的Hamming重量为$2^{n-1}$，那么函数f是平衡的。其中$\alpha_0=(0,0,\cdots,0),\alpha_1=(0,0,\cdots,1),\cdots,\alpha_{2^n-1}=(1,1,\cdots,1)$。

Webster和Tavars在研究密码学上强S盒的设计时，首次引入了严格雪崩准则，又称SAC。

定义2　V_n上的一个函数f，如果对一切$\alpha\in V_n$且$W(\alpha)=1$，$f(x)\oplus f(x+\alpha)$是平衡的，其中$x=(x_1,x_2,\cdots,x_n)$，则称f满足严格雪崩准则(SAC)。

现在对f_i的性质进行研究。实际上每轮中的函数是一个$V_{32}\times V_{32}\times V_{32}\to V_{32}$的函数，但由于所有计算都是基于位的，所以简化对函数f_i的考虑，仅考虑$f_i:V_3\to V_1$的情形，即是一个三元布尔函数。先写出对应于所有8个输入值(0,0,0)，(0,0,1)，…，(1,1,1)的四个函数的布尔值：

f_1:(0,0,0,1,1,0,1,1)，可以验证该函数是平衡且符合SAC。

f_2:(0,1,0,1,0,1,0,1)，可以验证该函数平衡但并不符合SAC，实际上该函数仅是个奇偶判别函数，指示输入值中1的个数。

f_3:(0,0,0,1,0,1,1,1)，可以验证该函数是平衡且符合SAC。

f_4与f_3的性质完全相同。

可以发现所选择的f_1，f_3的表示形式分别是有三项和两项的布尔函数中最简单的，计算所费步数最少。这也表明SHA的设计者确实考虑到了计算速度的问题。

每轮中使用的函数是平衡且SAC是较容易理解的，这种性质使得输入的每一比特的变化，在输出端有一个随机的变化，随机性正是HASH函数所要求的。然而，在第二轮和第四轮使用的函数显然是一个线性函数，在GF(2)可以表示成A+B+C，这无疑将给基于线性的HASH函数分析如差分分析及线性分析以机会。

2. 对数据扩展公式性质的研究

由于对位进行单独操作，所以可以把16个双字节消息扩展至80个双字节消息看成是一个编码过程，即是一个(80,16)的码。扩展所采用的公式显然是使这个码具有循环纠错码的性质，但我们应该看到，由于循环纠错码的规律性有可能减弱数据扩展所带来的安全性方面的好处。

下列形式的扩展公式(对于其中a,b,c,d的选择是需要作一番考虑的)：

$W_t=W_{t-a}\oplus W_{t-b}\oplus W_{t-c}\oplus W_{t-d}\quad a<b<c<d\leqslant 16\ t=16\ to\ 79$

由于所有的W_t都可以表示成W_0到W_{15}这16个双字字消息的线性表示，所以出于随机性的目的，在所有W_t的表示中，W_0到W_{15}的出现次数应该基本上相同而且使用次数不能太少。使用均值和方差两个标准来衡量，经过详细考察，发现SHA所作的选择较好地符合了上述的条件，但却不是最优的，较好的选择有(4,5,13,16)等。

三、改进的新型Hash算法

我们仍然遵循SHA的框架，只是将消息摘要规模减小到4个双字，即128bit，以利于加密和数字签名。我们的Hash算法如下：

1)数据扩展算法：

$W_t=M_t$, for　t=0 to 15
$W_t=W_{t-4}\oplus W_{t-5}\oplus W_{t-13}\oplus W_{t-16}$, t=16 to 79

2)每个ROUND顺次操作20个消息字，每轮的运算所用公式形式相同，只是所用参数不同。公式如下：

$TEMP=(A<<<5)+f_t(B,C,W_t)+D<<<7+K_t$
$D=C$
$C=(B<<<30)$
$B=A$
$A=TEMP$

然而，每轮的K_t不再是常数，而是随t而变，K_t是$2^{32}\times\sin(2\times t)$的整数部分，且$f_t$也有改变：

$f_1(X,Y,Z)=XY+\sim XZ$
$f_2(X,Y,Z)=XY+X\sim Z+Y\sim Z$
$f_3(X,Y,Z)=X\sim Z+YZ$
$f_4(X,Y,Z)=XY+XZ+YZ$

我们所使用的所有f_t都是从所有满足平衡且SAC条件的三元布尔函数中精心挑选出来的平衡的SAC函数，在表示形式上做到尽量简单，不会增加太多的计算量。

欢迎对此Hash函数进行攻击和分析。

参考文献

[1] Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, John Wiley & Sons, Inc

[2] R. Rivest, The MD4 Message Digest Algorithm, RFC1186, Oct. 1990

[3] R. Rivest, The MD4 Message Digest Algorithm, Advances in Cryptology-CROPTO' 90 Proceedings, Berlin: Springer-Verlag, 1991

[4] R. Rivest, The MD4 Message Digest Algorithm, RFC 1320, Apr. 1992

[5] R. Rivest, The MD5 Message Digest Algorithm, RFC 1321, Apr. 1992

[6] B. den Boer and A. Bosselaers, An Attack on the Last Two Rounds of MD4, Advances in Cryptology-EUROCRYPT' 93 Proceedings, Berlin: Springer-Verlag