第 55 卷 第 10 期 2006 年 10 月
1000-3290/2006/55(10)/5048-07

物 理 学 报
ACTA PHYSICA SINICA

Vol.55,No.10,October,2006

一类基于混沌映射构造 Hash 函数方法的碰撞缺陷*

王继志† 王英龙 王美琴

(山东省计算中心,济南 250014)

(2005 年 12 月 22 日收到;2006 年 3 月 8 日收到修改稿)

混沌动力学系统在一定的参数范围内出现混沌运动,且它所产生的混沌序列具有良好的伪随机特性.以混沌映射为基础,已有文献提出了一些构造单向散列函数的方法.对这些方法进行分析,证明它们均存在碰撞缺陷,并总结了采用混沌映射构造单向散列函数时应注意的问题.

关键词:混沌,单向散列函数,碰撞

PACC:0545

1. 引 言

随着计算机网络技术,特别是电子商务的发展,使得电子邮件、Web 浏览的用户越来越多,人们在网络上传递着各种信息,其中包括敏感的政治和商业信息.认证用户身份和保证用户使用时的安全正日益受到各方面的挑战,同时也是制约电子商务得到更进一步发展的主要因素之一.如何提高数据安全性已成为日益重要的研究课题.

数据加密技术是提高网络中数据传输安全性的关键所在,而在安全认证和数据完整性中常用到的是单向散列函数.关于单向散列函数的概念最早是在 1968 年由 Wilkes 提出的,单向散列函数的特点是正向计算简单、反向计算复杂,而且很难找到两个不同的输入之对应于同一个输出值的一种函数.它一般分为两类:无密钥单向散列函数和有密钥单向散列函数.常见的单向散列函数有:SHA1,MD5 等.

单向散列函数的原理如下:单向散列函数 $H(M)$ 作用于任意长度的消息 M,它返回一个固定长度的散列值

$$h = H(M),$$

其中 h 的长度为 m.

输入为任意长度但输出为固定长度的函数有很多种,但单向散列函数还具有其单向的如下特性:

(1)给定 M,很容易计算 h.

(2)给定 h,根据 $h = H(M)$ 计算 M 很难.

(3)给定 M,要找另一个消息 M' 并满足 $H(M) = H(M')$ 很难.

自 2004 年王小云发现 MD5 的碰撞问题以来,关于 Hash 函数的研究又成为一个热点.采用混沌映射构造单向散列函数是其中的一个研究方向.

混沌是由确定性动力学系统产生的一种看似随机的非线性现象.混沌信号具有的非周期性、连续宽带频谱、类似噪声的特性,使它具有天然的隐蔽性,对初始条件和微小扰动的高度敏感性,又使混沌信号具有长期不可预测性.混沌信号的隐蔽性和不可预测性,使其适用于保密通信.另外,混沌系统本身是非线性确定性系统,因而方便于保密通信系统的构造与研究.

混沌用于保密方面的工作主要有以下方面:一个是利用混沌产生伪随机序列并利用伪随机序列加密文本文件、图像文件;另一种是利用混沌同步理论进行同步保密通讯.

本文总结了目前所提出的基于混沌映射的单向散列函数构造方法,对方法进行了分析,指出其中存在的碰撞问题,最后总结了在构造单向散列函数时应注意的问题.

* 山东省自然科学基金(批准号:Z2003G01)资助的课题.

† E-mail:wangjzh@keylab.net

2. 一类构造方法的缺陷

文献[1—6]分别给出了一种基于混沌映射构造单向散列函数的方法，下面分别对这 6 篇文献进行分析，指出它们各自方法的缺陷.

(1)文献[1]利用二维超混沌映射构造单向散列函数.

具体方法如下：

(a)将待处理的报文按对应字节的 ASCⅡ码转换为数字，线性变换为[－1，1]范围内的数，这样整个报文得到一个大数组，记为 T，记数组长度即报文字节数为 N；

(b)令 $x_1 = 2T_1/255 - 0.8$，$x_2 = 2T_2/255 - 0.8$，进行迭代运算，令迭代轮数为 $r = R([N/R] + K)$，这里，[]表示取整运算，K，R 均为大于 0 的整数；

j 从 3[①] 到 r 按如下混沌映射计算：

$a_j = ((1 + 0.3(a_{j-2} - c) + 2917a_{j-1}^2) \bmod 2) - 1$，

当 $j \leqslant N$ 时，令 $c = 2\ T_j/255 - 0.8$；

当 $j > N$ 时，令 $c = a_{j-3}$[②]；

(c)将 a_{r-1}，a_r 分别进行线性变换：$a_k' = 1.045a_k - 0.55$，$a_l' = 1.245a_l + 0.615$ 得到 a_{r-1}' 和 a_r'；

令 $x_1 = a_{r-1}'$，$x_2 = a_r'$，m 从 2 到 $3R$ 按

$$x_{m+1} = 1.66\ y_m - 1.3y_m^2,$$
$$y_{m+1} = -1.1\ x_m + 0.3\ y_m$$

进行计算；

(d)将 x_R，x_{2R}，x_{3R}，y_R，y_{2R}，y_{3R} 六个迭代结果分别通过线性变换映射成 40bit，40bit，48bit，40bit，40bit，48bit 的二进制数，然后按照从 40bit 中选出序号第 15 位到第 34 位，从 48bit 中选出序号第 13 位到 36 位的规则，分别从中选择出 20bit，20bit，24bit，20bit，20bit，24bit，合起来组成 128bit 的 Hash 值.

下面来说明上述方法的缺陷：

设三字节明文 $S = s_1 s_2 s_3$，则字节长度 $N = 3$. 取

$$f(s) = 2s/255 - 0.8, \tag{1}$$

则迭代初值 $a_1 = f(s_1)$，$a_2 = f(s_2)$；令

$$a_j = g(a_{j-1}, a_{j-2}, c)$$
$$= ((1 + 0.3(a_{j-2} - c) + 2917\ a_{j-1}^2) \bmod 2) - 1, \tag{2}$$

则迭代序列如下：

$$\begin{cases} a_3 = g(a_2, a_1, c) = g(a_2, a_1, f(s_3)), \\ a_4 = g(a_3, a_2, a_1), \\ \cdots \\ a_r = g(a_{r-1}, a_{r-2}, a_{r-3}), \end{cases} \tag{3}$$

可以构造四字节明文 $S = s_1 s_2 s_3 s_4 = s_1 s_2 s_3 s_1$，即取第四个字节 $s_4 = s_1$，则字节长度 $N = 4$.

则根据(1)式，可得迭代初值

$$a_1 = f(s_1), a_2 = f(s_2)；$$

根据(2)式，可得迭代序列

$$\begin{cases} a_3 = g(a_2, a_1, c) = g(a_2, a_1, f(s_3)), \\ a_4 = g(a_3, a_2, f(s_4)) = g(a_3, a_2, a_1), \\ a_5 = g(a_4, a_3, a_2), \\ \cdots \\ a_r = g(a_{r-1}, a_{r-2}, a_{r-3}). \end{cases} \tag{4}$$

比较(3)，(4)式可得，两个明文迭代序列的 a_1，a_2，a_3，a_4 相等，可用数学归纳法证明迭代序列的 a_5，…，a_r 也相等. 因此两个明文得到的迭代序列完全一致，若 $R([3/R] + K) = R([4/R] + K)$(该条件显然很容易满足)，即迭代次数一样，根据前述步骤(c)，(d)，最后算得的散列值是相等的，即发生了碰撞.

该结论可以推广到一般情况.

命题 1　对于任意明文 $S = s_1 s_2 \cdots s_{N-2} s_{N-1} s_N$，可以构造 $S' = s_1 s_2 \cdots s_{N-2} s_{N-1} s_N s_{N-2}$，只要 $R([N/R] + K) = R([(N+1)/R] + K)$，则 $H(S) = H(S')$.

该命题的证明与上述分析类似，证明略.

根据命题 1 可以得到如下推论：

推论 1　对于任意明文 $S = s_1 s_2 \cdots s_{N-2} s_{N-1} s_N$，可以构造 $S' = s_1 s_2 \cdots s_{N-2} s_{N-1} s_N s_{N+1} s_{N+2} s_{N+3} \cdots s_{N+k}$，取

$$S_{N+i} = \begin{cases} s_{N-2}, \text{当 } i \bmod 3 = 1, \\ s_{N-1}, \text{当 } i \bmod 3 = 2, i = 1, 2, \cdots, k, \\ s_N, \text{当 } i \bmod 3 = 0, \end{cases}$$

当 $R([N/R] + K) = R([(N+k)/R] + K)$ 时，则 $H(S) = H(S')$.

该推论可由命题 1 用数学归纳法证得，证明略.

文献[2]采用的是相同的构造方法，结论也是相同的. 只需将分析过程中的(1)，(2)式分别换成如下式子：

$$f(s) = (2.4s)/255 - 1.2,$$

①文献[1]中此处为 2，显然为笔误，因为若从 2 开始计算，则迭代初值应从 0 开始计数

②文献[1]中此处为 a_{j-2}，显然为笔误，否则式中第二项恒为 0，没有意义

$g(a_{j-1},a_{j-2},c)=1+0.3(a_{j-2}-c)-1.08a_{j-1}^2$，结论完全一致.

(2)文献[3]构造了一个广义混沌映射模型，利用此模型来构造 Hash 函数，在方法上与文献[1,2]的方法基本思想一致，但稍有不同.

具体方法如下：

a)将待处理报文按对应字节的 ASCⅡ码转换为数字，线性变换为 -1—1 范围内的数，这样整个报文得到一个大数组，记为 S，记数组长度即报文字节数为 N；

b)令 $x_1=2S_1/(k-1)-0.8$，$x_2=2S_2/(k-1)-0.8$，其中 $k=256$；

c)迭代令迭代轮数为 $r=R([N/R]+1)$，其中[]为取整运算，$R=32$，

j 从 3 到 r 计算

当 $j\leqslant r/5$ 时，按

$$x_j=(1+0.3(x_{j-2}-c)+2917x_{j-1}^2)\bmod 2-1$$

进行迭代；

当 $j\leqslant N$ 时，$c=2S_j/(k-1)-0.8$；

当 $j>N$ 时，$c=x_N$.

同理，当 $r/5<j\leqslant 2r/5$，$2r/5<j\leqslant 3r/5$，…，$4r/5<j\leqslant r$ 时，分别用不同的混沌映射按上述方法进行迭代；

进一步迭代，j 从 3 到 $3R$ 计算，按当前迭代模型计算，其中 $c=x_{r-2}$，$x_1=x_{r-1}$，$x_2=x_r$.

d)从迭代结果序列中取出 x_R，x_{2R}，x_{3R}，将它们经线性变换和取整运算映射为两个 40bit，一个 48bit 的二进制数，合起来作为最后 128bit 的 Hash 结果.

下面来说明上述方法的缺陷：

设两字节明文 $S=s_1s_2$，则字节长度 $N=2$.取

$$f(s)=2s/255-0.8, \tag{5}$$

则迭代初值 $a_1=f(s_1)$，$a_2=f(s_2)$，$a_N=f(s_2)$；令

$$a_j=g(a_{j-1},a_{j-2},c), \tag{6}$$

其中 $g()$为当前迭代混沌映射函数(由于原文中没有给出 5 个分段的具体映射函数，所以这里用 $g()$统一表示).则迭代序列如下：

$$\begin{cases} a_3=g(a_2,a_1,c)=g(a_2,a_1,a_N), \\ a_4=g(a_3,a_2,a_N), \\ \quad\cdots \\ a_r=g(a_{r-1},a_{r-2},a_N). \end{cases} \tag{7}$$

可以构造三字节明文 $S=s_1s_2s_3=s_1s_2s_2$，即取第三个字节 $s_3=s_2$，则字节长度 $N=3$.则根据(5)式，可得迭代初值

$$a_1=f(s_1),a_2=f(s_2)\ a_N=f(s_3)=f(s_2)；$$

根据(6)式，可得迭代序列

$$\begin{cases} a_3=g(a_2,a_1,f(s_3))=g(a_2,a_1,f(s_2)), \\ a_4=g(a_3,a_2,a_N)=g(a_3,a_2,f(s_2)), \\ \quad\cdots \\ a_r=g(a_{r-1},a_{r-2},a_N), \end{cases} \tag{8}$$

由于 $r=R([2/R]+1)=R([3/R]+1)$，步骤 c)中的迭代分段只与 r 有关，因此两种情况下迭代次数分段是一致的，且每个分段采用的映射函数是一致的，比较(7)，(8)式可知迭代序列完全一样.又由于两种情况下，当前迭代函数是相同的，因此根据进一步迭代和步骤 d)，得到的散列值是相同的.

该结论可以推广到一般情况.

命题 2 对于任意明文 $S=s_1s_2\cdots s_{N-2}s_{N-1}s_N$，可以构造 $S'=s_1s_2\cdots s_{N-2}s_{N-1}s_Ns_N$，只要 $R([N/R]+1)=R([(N+1)/R]+1)$，则 $H(S)=H(S')$.

该命题的证明与上述分析类似，证明略.

根据命题 2 可以得到如下推论：

推论 2 对于任意明文 $S=s_1s_2\cdots s_{N-2}s_{N-1}s_N$，可以构造 $S'=s_1s_2\cdots s_{N-2}s_{N-1}s_Ns_{N+1}s_{N+2}s_{N+3}\cdots s_{N+k}$，取

$$s_{N+i}=s_N(i=1,2,\cdots,k),$$

当 $R([N/R]+1)=R([(N+k)/R]+1)$ 时，则 $H(S)=H(S')$.

该推论可由命题 2 用数学归纳法证得，证明略.

(3)文献[4]利用时空混沌的单向耦合映像格子模型设计单向散列函数.

方法如下：

a)待处理消息按对应字节 $C_1C_2C_3\cdots C_N$(C 为消息的 ASCⅡ码)线性变换为[0,1]范围内的数，整个消息变为一个数列，记为 $M_1M_2M_3\cdots M_N$，其中数列个数即消息字节数为 N，计算公式如下：

$$M_1=C_1/256,M_2=C_2/256,$$
$$M_3=C_3/256,\cdots,M_N=C_N/256；$$

b)令 $M_1M_2\cdots M_N$ 分别为 N 个格子的初值：

$$X_0(1)=M_1,X_0(2)=M_2,$$
$$X_0(3)=M_3,\cdots,X_0(N)=M_N；$$

c)应用下述单向耦合映像格子模型迭代初值：

$$x_{n+1}(i)=(1\text{-}e)f(x_n(i))+ef(x_n(i-1)),$$

其中 $f(x)=4.0x(1-x)$，$e=0.8$，有 N 个格子，周期边界条件 $x_n(N+i)=x_n(i)$.

迭代生成时空混沌序列 N 组：

$$X_n(1), X_n(2), X_n(3), \cdots, X_n(N)；$$

d) 从迭代结果序列中取出最后一组序列的 $X_R(N), X_{2R}(N), X_{3R}(N)$，这里的 R 要远大于 N，将他们经线性变换和取整运算映射为两个 40bit，一个 48bit 的二进制数，合起来作为最后的 128bit 的 Hash 结果.

下面说明该方法的缺陷：

取单字节明文 $C = c_1$，$N = 1$. 则 $m_1 = c_1/256$，$x_0(1) = m_1$，根据

$$\begin{aligned} x_{n+1}(i) &= g(x_n(i), x_n(i-1)) \\ &= (1-e)f(x_n(i)) + ef(x_n(i-1)), \end{aligned} \tag{9}$$

$$x_n(N+i) = x_n(i), \tag{10}$$

则迭代序列如下：

$$\begin{cases} x_1(1) = g(x_0(1), x_0(1)), \\ x_2(1) = g(x_1(1), x_1(1)), \\ \quad \cdots \\ x_n(1) = g(x_{n-1}(1), x_{n-1}(1)). \end{cases} \tag{11}$$

取两字节明文 $C = c_1c_2 = c_1c_1$，$N = 2$. 则 $m_1 = m_2 = c_1/256$，$x_0(1) = x_0(2) = m_1$，根据(9)，(10)式可得

$$\begin{cases} x_1(1) = g(x_0(1), x_0(2)), \\ x_1(2) = g(x_0(2), x_0(1)) \\ \quad\ = g(x_0(1), x_0(1)), \\ x_2(1) = g(x_1(1), x_1(2)), \\ x_2(2) = g(x_1(2), x_1(1)) \\ \quad\ = g(x_1(1), x_1(1)), \\ \quad \cdots \\ x_n(1) = g(x_{n-1}(1), x_{n-1}(2)), \\ x_n(2) = g(x_{n-1}(2), x_{n-1}(1)) \\ \quad\ = g(x_{n-1}(1), x_{n-1}(1)). \end{cases} \tag{12}$$

比较(11)，(12)式，(11)式中的迭代序列和(12)式中的第二列是完全相同的，因此根据步骤 d)得到的散列值是相等的.

该结论可推广到一般情况.

命题 3　对于任意的明文 $C = c_1c_2\cdots c_N$，可以构造明文

$$C' = c_1c_2\cdots c_N \underbrace{c_1c_2\cdots c_N\cdots c_1c_2\cdots c_N}_{i\text{ 个 }c_1c_2\cdots c_N}, i = 1,2,\cdots$$

只要迭代次数 n 一样，则 $H(C) = H(C')$.

该命题的证明与上面的分析略有不同，可以用数学归纳法进行证明.

证明　设明文 C 得到的迭代序列为

$$\begin{cases} x_1(1) = g(x_0(1), x_0(N)), \\ \quad \cdots \\ x_1(N) = g(x_0(N), x_0(N-1)), \\ x_2(1) = g(x_1(1), x_1(N)), \\ \quad \cdots \\ x_2(N) = g(x_1(N), x_1(N-1)), \\ \quad \cdots \\ x_n(1) = g(x_{n-1}(1), x_{n-1}(N)), \\ \quad \cdots \\ x_n(N) = g(x_{n-1}(N), x_{n-1}(N-1)), \end{cases} \tag{13}$$

第 1 步：证明明文 C' 得到的迭代序列

$$x_m(k) = x_m(iN+k), m = 1,2,\cdots,n；$$
$$k = 1,2,\cdots N；i = 1,2,\cdots$$

即 $i+1$ 个 $c_1c_2\cdots c_N$ 相应的迭代值是相等的. 可用数学归纳法证明.

当 $m = 1$ 时，

$$\begin{aligned} x_1(k) &= g(x_0(k), \\ x_0(k-1)) &= g(m_k, m_{k-1}), \\ x_1(i*N+k) &= g(x_0(i*N+k), \\ x_0(i*N+k-1)) &= g(m_{i*N+k}, m_{i*N+k-1}) \\ &= g(m_k, m_{k-1}), \end{aligned}$$

命题成立.

假设 $m = l$ 时，命题成立.

当 $m = l+1$ 时，

$$x_{l+1}(k) = g(x_l(k), x_l(k-1)),$$
$$x_{l+1}(iN+k) = g(x_l(i*N+k),$$
$$x_l(iN+k-1)) = g(x_l(k), x_l(k-1)),$$

命题成立. 所以第 1 步命题得证.

因此设

$$f = \begin{cases} x_1(1) = g(x_0(1), x_0(N)), \\ \quad \cdots \\ x_1(N) = g(x_0(N), x_0(N-1)), \\ x_2(1) = g(x_1(1), x_1(N)), \\ \quad \cdots \\ x_2(N) = g(x_1(N), x_1(N-1)), \\ x_n(1) = \cdots g(x_{n-1}(1), x_{n-1}(N)), \\ \quad \cdots \\ x_n(N) = g(x_{n-1}(N), x_{n-1}(N-1)), \end{cases} \tag{14}$$

则明文 C' 得到的迭代序列可表示为

$$\underbrace{f,f,\cdots,f}_{\text{共 } i+1 \text{ 个}}$$

第 2 步 :证明(13),(14)式中相应项是相等的.这一步也可用数学归纳法证明,与第 1 步的证明是类似的,不再累述.

由此可得,两个明文得到的最后一列的迭代序列是相同的,根据步骤 d)得到的散列值也是相等的.

因此,$H(C)=H(C')$,证毕.

(4)文献[5]利用 Logistic 映射来设计单向散列函数.

具体方法如下:

a)将明文分块,在文献的实现方法中是以一个字符作为一个明文信息块.

b)将消息的每一块量化,在文献的实现方法中是采用求每个字符的 ASCⅡ码值.

c)将量化后的值映射到某一值域,使得用该集合内的值作为参数能够使混沌系统处于混沌状态,文献的实现采用的映射是:$u=1.746+0.001\,\mathrm{Asc}(m_i)$,Asc()表示取字符的 ASCⅡ码值.

d)将 u 代入混沌系统 $f(X)=1-uXX$ 中,经 32 次迭代可得到序列 X_i.

e)将 X_i 通过一定方式转化为二进制序列.

f)将每个 X_i 所得的二进制序列逐位模 2 相加可得该消息的单向散列.

下面说明该方法的缺陷:

令每个分块得到的二进制序列 $X_i=L(s_i)$.取明文 $S=s_1s_2$,则 $H(S)=(X_1+X_2)\bmod 2$,可以构造 $S'=s_2s_1$,则 $H(S')=(X_2+X_1)\bmod 2=H(S)$.

该缺陷可以推广到一般情况:

命题 4 对于任意明文,将其字符顺序改变后,得到的散列值不变.

该命题的证明是显然的.

很明显,由于加法满足交换律,明文中字符的顺序不影响最后的散列值.由此可见,虽然该算法对每个分组单独进行计算,使得算法很容易并行化,提高了算法的运行速度,但最后仅仅是采用加法对各个分组得到的迭代值进行处理显然是不行的.

(5)文献[6]利用 Logistic 映射和 Lorenz 映射设计单向散列函数.

方法如下:

Step1 将明文 M 分为 n 个组 $m_1m_2\cdots m_n$,若最后一组明文的长度不足 24 个字符,则用‘ * ’号填充.每个 $m_k(k=1,2,\cdots,n)$ 由 $m_{k1}m_{k2}m_{k3}$ 组成,其中每个 $m_{ki}(i=1,2,3)$ 又由 8 个字符组成.

Step2 :设 $k=1$,计算

$$\begin{aligned}U_x=u(m_{k1})=&1.872\\&+(\mathrm{asc}(\mathrm{substr}(m_{k1},1,1))/256*2\hat{}\,0\\&+\mathrm{asc}(\mathrm{substr}(m_{k1},2,1))/256*2\hat{}(-8)\\&+\mathrm{asc}(\mathrm{substr}(m_{k1},3,1))/256*2\hat{}(-16)\\&+\mathrm{asc}(\mathrm{substr}(m_{k1},4,1))/256*2\hat{}(-24)\\&+\mathrm{asc}(\mathrm{substr}(m_{k1},5,1))/256*2\hat{}(-32)\\&+\mathrm{asc}(\mathrm{substr}(m_{k1},6,1))/256*2\hat{}(-40)\\&+\mathrm{asc}(\mathrm{substr}(m_{k1},7,1))/256*2\hat{}(-48)\\&+\mathrm{asc}(\mathrm{substr}(m_{k1},8,1))/256*2\hat{}(-56))/4,\end{aligned}$$

$U_y=u(m_{k2})$, $U_z=u(m_{k3})$.

Step3 :计算 $X_1=1-u_x*X_0*X_0$ $Y_1=1-u_y*Y_0*Y_0$ $Z_1=1-u_z*Z_0*Z_0$.

Step4 :令 $X_0=X_1$, $Y_0=Y_1$, $Z_0=Z_1$ 重复 Step3 31 次.

Step5 :令 $k=k+1$,重复 Step2,Step3,Step4,直到 $k=n$.

Step6:

…

Step24:

原文中给出了共 24 步的方法,本文省略.

下面说明该方法的缺陷:

该方法的缺陷是显然的,从 Step1 中可以看到,对于最后一个分组不足部分只是简单的填充‘ * ’,那么很显然如果有一明文是以‘ * ’结尾的,则它与去掉‘ * ’后的明文的散列值是相等的.

这一问题实质上就是如何对明文分组不足的部分进行处理的问题.在 MD5[7] 中,对于这一问题,除了填充字符外,还要添加原始明文的长度,从而避免了这一缺陷.但在文献[6]没有进行这一处理.

3. 分 析

从文献[1—6]给出的构造方法步骤可以看出,其设计思想是一致的,即

1)将明文映射成某个范围中的数值;

2)选择初值,利用混沌映射进行迭代；

3)选择某几次的迭代结果,映射成散列值；

然而它们的研究重点是在混沌映射的选择上,对于步骤 1,3 没有进行关注,而问题恰恰出在这两个方面.

对于混沌映射来说存在相空间和参数空间,那么在进行步骤 1 的时候就面临一个选择,即把明文映射到相空间还是参数空间.而上面的分析可以看出,映射到相空间的都存在问题,如文献[1—4];而映射到参数空间的,对于步骤 1 目前没有发现问题.对于这一现象,我们可以从混沌映射的性质出发来进行初步的解释.

从步骤 3 可以知道,最后的散列值是选择某几次的迭代结果经过变换而得到的,而这些迭代结果都是在相空间中的,也就是说当我们构造明文初值通过混沌映射进行迭代时,得到了一条相空间中的轨迹,但如果取相轨迹中的其他点作为初值,通过同样的混沌映射迭代,仍然可以得到同样的一条相轨迹.这就是为什么将明文映射到相空间,比较容易构造另外的明文,使得他们的迭代结果相同,最后的散列值也是相同的原因.

另外一个问题就是明文分组的处理,这包括两个方面:明文分组不足如何处理;分组得到的迭代值如何处理.对于前一个问题,不能仅仅是填充一些字符,还应填充一些原始明文的信息,例如 MD5 中是填充原始明文的长度.对于后一个问题,可以将前一分组的迭代结果作为后一分组迭代的初值,或者每个分组单独进行迭代计算,得到结果后再进行一次迭代计算,即要保证运算的结果是与输入的顺序是有关的.

4. 总　　结

从上面的分析可以看出,基于混沌映射的单向散列函数构造方法需要注意以下问题：

1)最好将明文映射到参数空间.如果要映射到相空间,要特别注意不同的明文要有不同的迭代次数.

2)分组得到的迭代值应该作为下一分组迭代过程的初值,也就是说不同分组的迭代应该是相关的,而不应完全分离,虽然这样做可以提高算法的并行度.

3)对于分组不足的处理,不能仅仅是单纯的添加某一字符,还需要添加原始明文的信息.

4)对于最后生成散列值的迭代值的选择,应尽量对不同的明文选择不同迭代次数的值,这样即使最后的迭代序列完全一致,由于迭代值选择不一样,也可以保证最后的散列值不一样.

5)若混沌映射函数所需的迭代初值大于 1 个,那么需要注意如果明文映射得到的初值个数小于混沌映射函数所需的迭代初值的个数时,如何进行处理.例如,文献[1,2]中都存在这个问题,构造方法中至少需要 3 个初值,那么对于单字节、双字节明文如何处理,没有明确说明.

[1] Peng F, Qiu S S, Long M 2005 *Acta Phys. Sin.* **54** 4562 (in Chinese) [彭　飞、丘水生、龙　敏 2005 物理学报 **54** 4562]

[2] Liu J N, Xie J C, Wang P 2000 *J. Tsinghua Univ.* (Sci & Tech) **40** 55 (in Chinese) [刘军宁、谢杰成、王　普 2000 清华大学学报(自然科学版) **40** 55]

[3] Wang X M, Zhang J S, Zhang W F 2003 *Acta Phys. Sin.* **52** 2737 (in Chinese) [王小敏、张家树、张文芳 2003 物理学报 **52** 2737]

[4] Zhang H, Wang X F, Li Z H, Liu D H 2005 *Acta Phys. Sin.* **54** 4006 (in Chinese) [张　瀚、王秀峰、李朝晖、刘大海 2005 物理学报 **54** 4006]

[5] Chen Z D, Huang Y S 2001 *J. Commun. Tech.* **118** 96 [陈志德、黄元石 2001 通信技术 **118** 96]

[6] Chen Z D 2001 Fuzhou Univ. Master Degree Thesis (in Chinese) [陈志德 2001 福州大学硕士学位论文]

[7] Rivest R 1992 *The MD5 Message-Digest Algorithm* http://www.ietf.org/rfc/rfc1321.txt

The collision problem of one kind of methods for constructing one-way Hash function based on chaotic map *

Wang Ji-Zhi† Wang Ying-Long Wang Mei-Qin
(*Shandong Computer Science Center*, *Jinan* 250014, *China*)
(Received 22 December 2005; revised manuscript received 8 March 2006)

Abstract

Chaos would happen within certain ranges of parameters of chaotic system. The chaos sequence has good pseud random character. Now some papers have provided some methods to construct one-way hash function based on chaotic map. Through analyzing these methods, the appearance of collision is proved. So some key problems are pointed out which should be taken care of while constructing one-way function based on chaotic map.

Keywords: chaos, one-way Hash function, collision
PACC: 0545

* Project supported by the Natural Science Foundation of Shandong Province, China (Grant No. Z2003G01)
† E-mail: wangjzh@keylab.net