第 55 卷 第 11 期 2006 年 11 月
1000-3290/2006/55(11)/5700-07

物 理 学 报
ACTA PHYSICA SINICA

Vol.55,No.11,November,2006

基于切延迟椭圆反射腔映射系统的单向 Hash 函数构造

盛利元　李更强　李志炜

(中南大学物理科学与计算技术学院，长沙　410083)

(2006 年 1 月 24 日收到；2006 年 3 月 21 日收到修改稿)

提出了一种基于切延迟椭圆反射腔映射系统(TD-ERCS)的单向 Hash 函数算法.该算法只需将明文信息线性变换为系统的参数序列,然后让 TD-ERCS 依次迭代,再依照一定的规则提取 Hash 值,勿需要增加任何冗余计算.该算法的密钥在[2^{64},2^{160}]内任意取值,输出 Hash 值长 160bit.基础的安全性测试表明,这种 Hash 函数具有很好的单向性、弱碰撞性、初值敏感性,较其他混沌系统构造的 Hash 函数具有更强的安全性,且实现简单,运行速度快,是传统 Hash 函数的一种理想的替代算法.还导出了评估碰撞性的一个客观标准.

关键词：混沌，Hash 函数，TD-ERCS

PACC：0545

1. 引　　言

在公认足够安全的传统 Hash 函数如 MD5,SHA-1 等被 Wang 小组[1]用“模差分”方法攻击成功后,传统 Hash 函数的安全性受到了严重威胁,密码学界反响强烈,甚至已经波及商业的公共信息安全,寻找新的安全的 Hash 函数的任务已经摆在密码学界的面前[2].基于复杂度假设的传统 Hash 函数,需要大量复杂的异或等逻辑运算,由于状态空间有限以及不可分析性,潜伏着某些设计者难以发现的安全漏洞.近年来,为获得更安全的 Hash 函数,利用混沌系统的特性构造单向 Hash 函数取得不小进展[3—10],理论渐趋成熟.用混沌系统构造综合性能优异的 Hash 函数,关键取决于混沌系统自身对应算法安全性的特质,这些特质称为混沌的安全性条件.我们已经归纳出了五项安全性条件[11]：离散映射、全域混沌、全域零相关、巨大的参数和初值空间、极强的抗退化能力.现有的混沌系统还不能同时满足以上的安全性条件,相关的 Hash 函数设计方案也存在不少缺陷,如基于混沌同步的鲁棒性实现摘要提取,势必降低系统的敏感性,攻击者可利用这一特征进行攻击[3]；又如基于一维混沌系统的 Hash 函数,发现利用自适应同步预测相空间重构等各种混沌预测技术可以成功分析预测[12—15]；有限精度效应使混沌序列退化为周期序列,若混沌系统抗退化能力弱,构造 Hash 函数是不安全的,没有实际应用价值.基于以上原因,我们试图用切延迟椭圆反射腔映射系统(tangent-delay ellipse reflecting cavity-map system，TD-ERCS)[16]构造一种新的 Hash 函数.

TD-ERCS 是专门为混沌加密理论而设计的混沌系统,具有良好的安全性特质.已经证明该系统满足前四项安全性条件；至于第五项安全性条件,我们发现了该系统在非双曲不动点处存在一种截断误差诱导阵发混沌的新机理[17],也发现该系统存在着截断误差在非双曲不动点局域拓扑地扩展维度现象,这就是说,若混沌系统处处都可能存在非双曲不动点,则该系统就可能具有极强的抗退化能力；另外还证明了 TD-ERCS 不存在由短周期引起的弱密钥[18—20].因此,用 TD-ERCS 混沌系统构造 Hash 函数具有的安全性上的优势.

2. 单向 Hash 函数的构造方法

2.1. TD-ERCS 混沌系统模型

该模型的定义及工作原理参考文献[16,21],这里仅给出与算法有关的公式.

给定 TD-ERCS 系统参数 $\mu\in(0,1]$,初值 $x_0\in[-1,1]$和 $\alpha\in(0,\pi)$,切延迟 $m(m=2,4,5,6,\cdots)$,TD-ERCS 通过迭代关系

$$x_n=-\frac{2k_{n-1}y_{n-1}+x_{n-1}(\mu^2-k_{n-1}^2)}{\mu^2+k_{n-1}^2},\qquad(1)$$

$$k_n=\frac{2k'_{n-m}-k_{n-1}+k_{n-1}k'^2_{n-m}}{1+2k_{n-1}k'_{n-m}-k'^2_{n-m}},\qquad(2)$$

$$n = 1,2,3,\cdots$$

将产生两个独立的实值序列:

$$x_{m+0}, x_{m+1}, x_{m+2}, \cdots, \tag{3}$$

$$k_{m+0}, k_{m+1}, k_{m+2}, \cdots, \tag{4}$$

其中

$$k'_{n-m} = \begin{cases} -\dfrac{x_{n-1}}{y_{n-1}}\mu^2 & n < m, \\ -\dfrac{x_{n-m}}{y_{n-m}}\mu^2 & n \geqslant m, \end{cases} \tag{5}$$

$$y_n = k_{n-1}(x_n - x_{n-1}) + y_{n-1}, \tag{6}$$

$$y_0 = \mu\sqrt{1 - x_0^2}, \tag{7}$$

$$k'_0 = -\frac{x_0}{y_0}\mu^2, \tag{8}$$

$$k_0 = \frac{\tan\alpha + k'_0}{1 - k'_0\tan\alpha}. \tag{9}$$

(μ, x_0, α, m)称为 TD-ERCS 种子参数,构成 Hash 函数的密钥. $n < m$ 的状态称为过渡态,$n \geqslant m$ 的状态称为正常态.序列(3)和(4)都是正常态序列.迭代中所有实数均采用双精度浮点数.

2.2. Hash 函数

2.2.1. 基本结构

基本结构如图 1 所示.一个 TD-ERCS 框为对应(1)式和(2)式的一步迭代.整体上分成两部分:过渡态和正常态.首先在用户密钥(μ, x_0, α, m)下完成过渡态的迭代,共 $m-1$ 步,通过(5)式分别得到 m 个不同的初始斜率.然后将待处理报文线性变换为系统的参数序列 M:$\mu_0, \mu_1, \mu_2, \cdots$,并在正常态中依次替换系统参数 μ 进行迭代,得到新的序列(3)和(4),再按照一定的规则从序列(3)和(4)中抽取 Hash 值.该算法本质上就是 TD-ERCS 混沌系统的一种简单动力学迭代过程,不附加任何其他为改善系统性能的"padding".

2.2.2. 用户密钥

类似文献[21]中的处理方法,用户密钥用 K 表示,16 进制数字符集

$$H = \{0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f\} \tag{10}$$

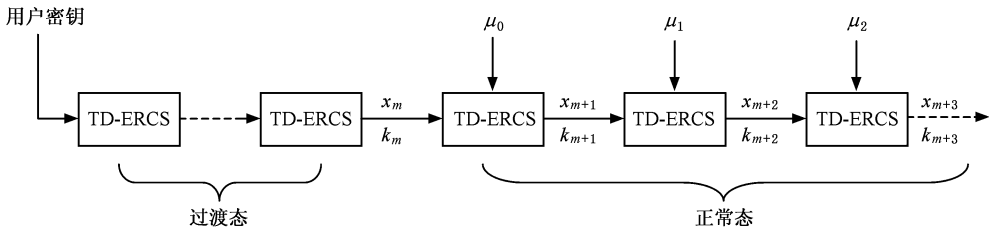

图 1　基于 TD-ERCS 的 Hash 函数基本结构

为用户密钥的有效字符,简称 H 字符.

K 的定义域为$[2^{64}, 2^{160}]$.在该区间,K 可取任意一个整数,故 K 不是固定长度的,而是一个长为 16—40 个 H 符号串,或 64—160 bit 的二进制数符号串.

K 可表示为 4 个密码子块的首尾连接,定义为

$$K = K_1 K_2 K_3 K_4. \tag{11}$$

根据计算机中实数的双精度表示法(见图 2),K_i 与种子参数的对应关系与赋值如下:

K_1 取 13 个 H 符号,长 52 bit,作为二进制码赋给参数 μ 的尾数.

K_2 取 13 个 H 符号,长 52 bit,作为二进制码赋给初值 α 的尾数.

K_3 取 13 个 H 符号,长 52 bit,作为二进制码赋给初值 x_0 的尾数.

K_4 取 1 个 H 符号,长 4 bit,以整数形式按照下式赋给切延迟 m:

$$m = \begin{cases} 2 & K_4 = 0 \\ 4 & K_4 = 1, \\ K_4 + 3 & 2 \leqslant K_4 \leqslant 15. \end{cases} \tag{12}$$

2.2.3. 参数序列 M

待处理报文采用标准的 ASCII 码的字符,即 C:$C_0 C_1 C_2 C_3 \cdots$,每个字符 C_i 对应数 0—255 之间的一个 8 bit 二进制码.将 C 按照每 6 个字符依次分块,得字符块序列 B:$B_0 B_1 B_2 B_3 \cdots B_N$. B 与 C 的对应关系为

$$B_i = C_{6i}C_{6i+1}C_{6i+2}C_{6i+3}C_{6i+4}C_{6i+5}, \quad i = 0,1,2,\cdots,N. \tag{13}$$

若最后一个字符块 B_N 不足 6 个字符,补"0"到 6 个字符.这样,每个字符块可表示成长为 48 bit 的二进

制码,即

$$B_i = b_{i0}\,b_{i1}\,b_{i2}\,b_{i3}\cdots b_{i46}\,b_{i47},\quad i = 0,1,2,\cdots,N. \tag{14}$$

定义 1 将(14)式中 B_i 的二进制码位前后对称互换构成新的 48 bit 的二进制码块,即

$$B_i' = b_{i47}\,b_{i46}\,b_{i45}\,b_{i44}\cdots b_{i1}\,b_{i0},\quad i = 0,1,2,\cdots,N, \tag{15}$$

称这种操作为倒序操作,B_i' 为 B_i 的倒序块.

如同 K_1 赋值给参数 μ 一样,将 B_i 后补"1111"赋值给 μ_i,B_i' 后补"1111"赋值给 μ_i',即得 μ_i 和 μ_i' 的二进制数

$$\mu_i = 0.b_{i0}\,b_{i1}\,b_{i2}\,b_{i3}\cdots b_{i46}\,b_{i47}1111, \tag{16}$$

$$\mu_i' = 0.b_{i47}\,b_{i46}\,b_{i45}\,b_{i44}\cdots b_{i1}\,b_{i0}1111, \tag{17}$$

$$i = 0,1,2,\cdots,N.$$

由此得到参数序列 M 和参数序列 M'.

2.2.4. 正常态迭代

正常态迭代 $2(N+1)+38$ 次,具体分配如下:

1)前 $N+1$ 次迭代按参数序列 M:$\mu_0,\mu_1,\mu_2,\cdots,\mu_N$ 进行;

2)接下来的 $N+1$ 次迭代按参数序列 M':$\mu_N',\mu_{N-1}',\mu_{N-2}',\cdots,\mu_0'$ 进行;

3)最后 38 次迭代用参数 μ 进行.

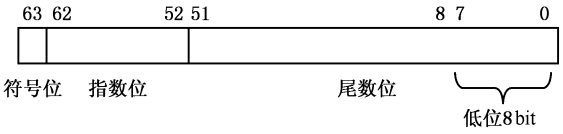

图 2 IEEE754 实数双精度表示中低位 8 bit

2.2.5. Hash 值提取

在最后的 38 次迭代中,分别取 $x_{m+2(N+1)+18+i}$,$k_{m+2(N+1)+18+i}$($i = 1,2,3,\cdots,20$)的双精度浮点数中尾数的低位 8 bit(见图 2)作异或运算,所得二进制码作为 Hash 值的一部分,记为 h_i.最后将所有的 h_i 首尾联接得输出的 Hash 值

$$h = h_1\,h_2\,h_3\cdots h_{19}\,h_{20}. \tag{18}$$

3. 文本 Hash 值的测试实验

任取密钥

$$K = fedcba9876543210012345678 9abcdeffedcba98, \tag{19}$$

即

$$K_1 = fedcba9876543,$$

$$K_2 = 2100123456789,$$

$$K_3 = abcdeffedcba9,$$

$$K_4 = 8.$$

转换为种子参数

$$\mu = 0.9955555555555555,$$

$$\alpha = 0.1289073350694443,$$

$$x_0 = 0.6711111066076489,$$

$$m = 11.$$

取 5 个相差细微的测试文本.

文本 1:Youth means a temperamental predominance of courage over timidity of the appetite, for adventure over the love of ease. This often exists in a man of 60 more than a boy of 20. Nobody grows old merely by a numbers of years. We grow old by deserting our ideals.

文本 2:将文本 1 中的首字母 Y 改为小写 y;

文本 3:将文本 1 中的 60 改为 61;

文本 4:将文本 1 中的 adventure 改为 adventures;

文本 5:将文本 1 中的 ideals 改为 ideal.

对应的 Hash 值见图 3,十六进制表示为

文本 1:F4DA01142584F48E1B0F6BF6AAECB529C0F1CEF4;

文本 2:DFDE3E596773B50515BD8E4F767B3C06028AA3F8;

文本 3:EDE05F2F85795205FE5E979436EB51DCF47F75B8;

文本 4:6BCFBC57BC9A8862D171CC04CEC8478220ECA2A4;

文本 5:1B16D6C0E73F095BC0DA8C2A084CFCE03EC98B66.

由以上 5 个 Hash 值可看出,文本微小变化将导致其 Hash 值巨大变化.可见该算法的单向 Hash 性能好,对信息变化具有高度敏感性.

4. 混乱与散布性质统计分析

Hash 函数的混乱与散布性质测试是 Hash 函数安全性分析的基本内容之一.性能良好的 Hash 函数要求明文与所对应的 Hash 值之间相关性很小,其特征是对信息的任何改动都应导致 Hash 值 bit 位以 50% 的概率变化.测试方法如下:在明文空间中随机选取一段明文,计算其 Hash 值,然后任意改变明文 1bit,得一新的 Hash 值,比较两个 Hash 值,统计 bit

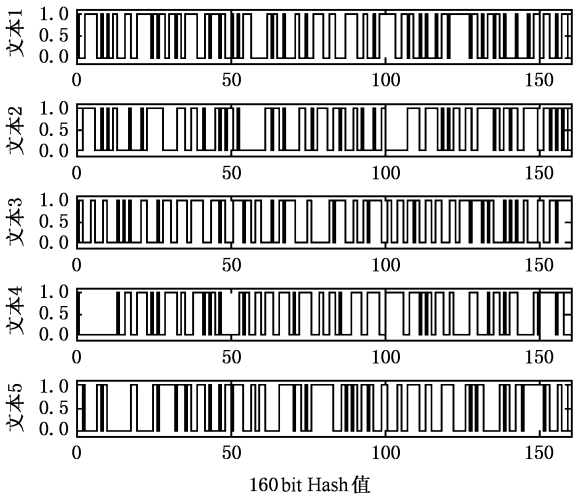

图 3　文本 1—5 的 Hash 值比较

位变化数(称为置乱数),如此重复若干次试验.

用 G_i 表示 N 次测试中第 i 次测试 Hash 值的置乱数,对应 Hash 值位的变化率为 $P_i = \frac{G_i}{160}$, G_i 和 P_i 的平均值分别为

$$\bar{G} = \frac{1}{N}\sum_{i=1}^{N} G_i, \tag{20}$$

$$\bar{P} = \frac{\bar{G}}{160} \times 100\%, \tag{21}$$

对应的标准差分别为

$$\Delta G = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(G_i - \bar{G})^2}, \tag{22}$$

$$\Delta P = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(P_i - \bar{P})^2}. \tag{23}$$

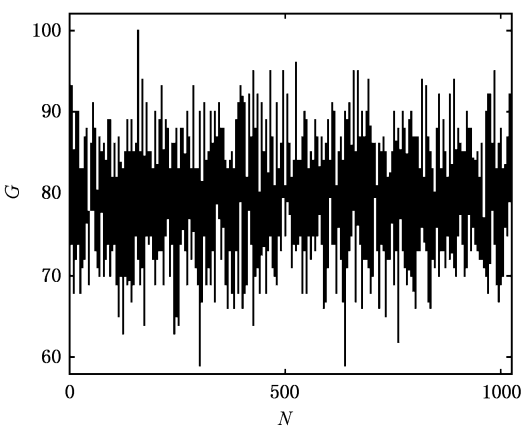

图 4　置乱数分布图

分别取 $N = 256, 512, 1024, 2048$ 进行 4 轮测试,用 G_{max} 和 G_{min} 分别表示每一轮的最大置乱数和最小置乱数.以文本 1 为测试对象,每次改变 1 bit,测得 $\bar{G}$, $\bar{P}$, ΔG, ΔP, G_{max}, G_{min} 如表 1 所示.实验表明,平均置乱数为 79.55,Hash 值位的平均变化率为 49.72%,非常接近理想数 80 和 50% 的变化概率,上下波动幅度小,在 6—7 之间,最大置乱数和最小置乱数分别为 100 和 55,图 4 给出了 $N = 1024$ 的置乱数分布,没有出现波动幅度非常剧烈的现象,这些结果表明该算法对明文的置乱能力强而稳定.表 2 给出了文献[5—9]提出的几种 Hash 函数的 ΔP 值.ΔP 值越小,算法的置乱能力与稳定性越好,可见本文提供的 Hash 函数性能是最好的.

表 1　4 轮置乱数的测试结果

N	256	512	1024	2048	总平均
$\bar{G}$	79.95	79.28	79.47	79.57	79.55
ΔG	6.908	6.375	6.303	6.291	6.408
$\bar{P}$/%	49.97	49.55	49.67	49.73	49.72
ΔP/%	4.318	3.984	3.939	3.932	4.043
G_{max}	100	97	100	99	100
G_{min}	62	55	59	55	55

表 2　文献[5—9]提出的 Hash 函数的 ΔP 值

N	256	512	1024	2048	总平均
TD-ERCS	4.318	3.984	3.939	3.932	4.043
[5]	4.772	4.555	4.461	4.396	4.546
[6]	4.66	4.38	4.40	4.45	4.473
[7]	4.102	4.104	4.105	4.014	4.081
[8]	4.31	4.46	4.46	4.52	4.438
[9]	5.013	4.927	4.684	4.506	4.783

5. 碰撞分析

碰撞分析是 Hash 函数安全性基础分析之一.所谓碰撞,是指不同的明文具有相同的 Hash 值,即发生了多对一的 Hash 映射.常见的一种测试方法[5]是取一字节初始文本,即 8bit,其 Hash 值也只取 8bit,均对应 0—255 的整数,这样初值空间与终值空间相同.记终值空间中任一值对应初值空间中原像的个数记为 k,记终值空间中具有 k 个原像的点的个数为 $n(k)$, $n(1)$ 越大, $n(0)$ 和其他各项越小,说明碰撞越少,混沌函数的散乱能力越强,用终值空间与初值空间的测度之比来定量衡量碰撞发生程度,令

$$P = \frac{256 - n(0)}{256}, \tag{24}$$

P 的值越接近 1,碰撞程度越低,等于 1 时,完全没有碰撞发生.

测试的碰撞分布如图 5 所示,$n(0)$至 $n(7)$依次为 96,98,38,17,4,1,1,0,$k>7$ 亦均为 0,$P=0.625$. 可见,该算法碰撞程度较低.

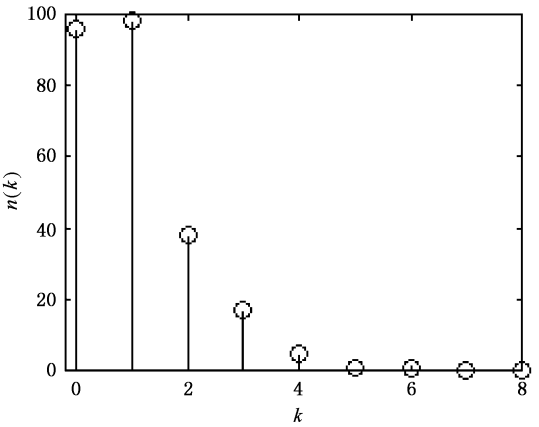

图 5 k-$n(k)$分布图

这种方法有一定的局限性,由于缩小了终值空间,故不能说明实际意义上的初值空间与终值空间的对应关系,难以表达 Hash 函数的碰撞性能.

另一种碰撞测试方法[8,10]如下:在明文信息中随机取一段,求出其 Hash 值并转换成 ASCII 字符储存.然后随机改变该明文段中 1bit,计算其 Hash 值也转换成 ASCII 字符储存.比较这两个用 ASCII 字符表示的 Hash 值,若两个 Hash 值中相同位置上 ASCⅡ字符相同,则称为被击中 1 次,统计被击中次数.同时,定义两个 Hash 值之间的绝对距离为

$$d = \sum_{i=1}^{S} | t(e_i) - t(e_i') |, \tag{25}$$

其中 e_i 和 e_i' 分别是最初的和新的 Hash 值的第 i 个 ASCⅡ字符,S 为 Hash 值对应 ASCII 字符的个数,函数 $t(\cdot)$将 e_i 和 e_i' 转换成对应的十进制数.

这种碰撞测试方法可抽象为以下概率模型:设 $\xi_1 \in (0,1)$和 $\xi_2 \in (0,1)$是已知分布的两个相互独立的随机变量,又设随机变量 $\eta = |\xi_2 - \xi_1|$,求数学期望 $E\{\eta\}$.若 ξ_1 和 ξ_2 是均匀分布的,可以证明,$E\{\eta\} = \frac{1}{3}$.于是,若两个 Hash 值分别由两个独立的均匀分布的随机序列所组成,也就是将 e_i 和 e_i' 视为相互独立的且以等概率取 ASCII 字符,即 e_i 和 e_i' 等概率取[0,1,2,3,…,254,255]上的整数,则可以证明,Hash 值的单位字符的平均绝对距离 $E\left\{\frac{d}{S}\right\} = \frac{1}{3} \times 256 = 85.33$.该概率模型的意义在于:若扰动前后的两个明文所得到的两个 Hash 值统计上由相互独立的两个均匀随机序列构成,则从 Hash 值一端来看,哪怕只有 1bit 不一样,这两个明文也是相互独立的,因此,除了穷举法,任何基于统计方法的碰撞分析都将失效.(25)式及理论值 85.33 提供了一个评估 Hash 函数碰撞性能的客观标准.

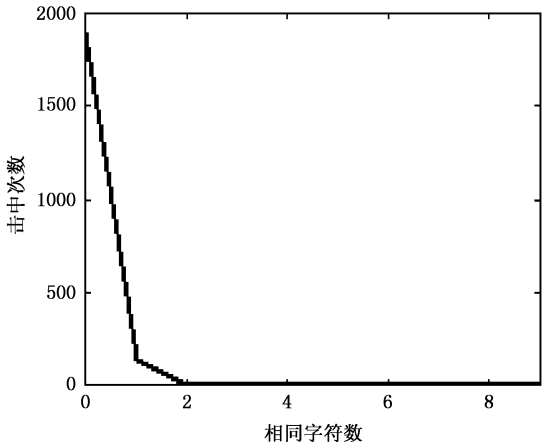

图 6 Hash 值中相同位置上 ASCII 字符相同的数目分布

采用与(19)式相同的密钥连续测试 2048 次.Hash 值中同一位置具有相同字符的数目分布情况见图 6.2048 次测试中有 140 次测试击中 1 次,有 7 次测试击中 2 次,1901 次测试没有发生击中,可见碰撞是很低的.表 3 给出了 d 的最大值,最小值,平均值和平均值/字符.从表 3 可知,Hash 值的平均距离/字符为 85.96,与理论值 85.33 仅差 0.74%,可见本文提供的 Hash 函数具有极高的抗碰撞分析能力.文献[8]提供的 Hash 函数相应实验值为 90.125,与理论值 85.33 差 5.62%.文献[10]提供了混沌查表式 Hash 函数的两个实验数据,分别为$\frac{21860}{256} = 85.39$和$\frac{21840}{256} = 85.31$,可见非常接近理论值,但算法结构被分析是不安全的[22].

表 3 两个 Hash 值之间的绝对距离 d

最大距离	最小距离	平均距离	平均距离/字符
2520	960	1719	85.96

6. 运算速度分析

算法的运算速度也是重要指标之一.基于混沌的 Hash 函数都需采用浮点运算,因此判断一个算法的速度只需计算算法中平均 Hash 一个 ASCII 字符所需的乘法与加法运算次数,与迭代步数、计算机性能以及使用的语言无直接关系.由于加(减)法运算消耗机时远少于乘(除)法运算,乘(除)法运算消耗机时又远少于函数运算,故一个好的算法应尽量避免函数运算,减少乘(除)法运算.

为简化起见,我们假定可以略去算法中前期的过渡运算和后期不消耗 ASCII 字符的收尾运算,只考虑 Hash ASCII 字符的核心运算.表 4 列出了 6 种 Hash 函数算法[4,5,7—9]中单位字符所需要的乘(除)法次数和加(减)法次数,可见本文提供的 Hash 函数算法运算速度是最快的.我们注意到文献[4,5,7]均声称可以将 3—5 个字符线性变换成 -1—1 或 0—1 之间的实数进行迭代将成倍提高运算速度.实际上,这样的声称是不成立的.首先,这些算法采用一维混沌系统,敏感性低,由于计算机截断误差,这些系统存在严重的退化现象,需要附加“padding”;其次,数的尾数 bit 位变化的敏感性随 bit 位趋于低位而减弱,因而算法对明文中 bit 位变化的敏感性将是起伏性的,不均匀的.为了消除这种现象发生,如同文献[8]的算法一样必需让所有的迭代至少重复 3 次.这样一来,运算速度不会有显著提高.

表 4　几种 Hash 函数中单位字符所需的乘法与加法数

	TD-ERCS	[4]	[5]	[7]	[8]	[9]
乘法	4	6	6	6	6	27
加法	3	5	5	5	9	9

7. 结　论

充分的数据分析表明,由于 TD-ERCS 混沌系统具有良好的动力学特性和统计特性,所构造的 Hash 函数较之其他混沌系统构造的 Hash 函数而言不仅简单,运算速度快,而且还具有更高的安全性,灵活方便,因而是目前为止能够被选为替代传统 Hash 函数的一种较理想的算法.同时,本文导出的用于评估 Hash 函数碰撞特性的客观标准具有重要的理论意义和实用价值.由于过去没有这个标准,文献[8]和文献[10]的作者无法说明他们测试的数据是好还是坏,文献[4—7,9]也没有拿出这个重要的数据.这个标准同样可以用于传统 Hash 函数的碰撞性能评估.

[1] Wang X Y, Feng D G, Lai X J *et al* 2004 CRYPTO '04

[2] Thompson E 2005 *Digital Investigation* **2** 36

[3] Zhang J S, Xian X C 2001 *Acta Phys. Sin.* **50** 2121 (in Chinese) [张家树、肖先赐 2001 物理学报 **50** 2121]

[4] Liu J N, Xie J C, Wang P 2000 *J. Tsinghua University* (Natural Science Edition) **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]

[5] 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]

[6] Wang X M, Zhang J S, Zhang W F 2005 *Acta Phys. Sin.* **54** 5566 (in Chinese) [王小敏、张家树、张文芳 2005 物理学报 **54** 5566]

[7] 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]

[8] Xiao D, Liao X F, Deng S J 2005 *Chaos, Solitons & Fractals* **24** 65

[9] Zhang H, Wang X F, Li Z H *et al* 2005 *Acta Phys. Sin.* **54** 4006 (in Chinese) [张　瀚、王秀峰、李朝晖等 2005 物理学报 **54** 4006]

[10] Wong K W 2003 *Phys. Lett.* A **307** 292

[11] Sheng L Y, Zhang Q, Sun K H *et al* 2005 *Journal on Communications* **26**(4) 122 (in Chinese) [盛利元、张　卿、孙克辉等 2005 通信学报 **26**(4) 122]

[12] Short K M 1994 *Int. J. Bifurc. Chaos* **4** 959

[13] Short K M 1997 *Int. J. Bifurc. Chaos* **7** 1579

[14] Zhang J S, Xian X C 2000 *China Phys.* **9** 408

[15] Zhang J S, Xian X C 2000 *Chin. Phys. Lett.* **17** 88

[16] Sheng L Y, Sun K H, Li C B 2004 *Acta Phys. Sin.* **53** 2871 (in Chinese) [盛利元、孙克辉、李传兵 2004 物理学报 **53** 2871]

[17] Sheng L Y, Jia W Y 2005 *Acta Phys. Sin.* **54** 5574 (in Chinese) [盛利元、贾伟尧 2005 物理学报 **54** 5574]

[18] Sheng L Y December 12, 2005 Scincepaper Online http://www.paper.edu.cn

[19] Sheng L Y, Jia W Y December 16, 2005 Scincepaper Online http://www.paper.edu.cn

[20] Sheng L Y, Jia W Y January 12, 2006 Scincepaper Online http://www.paper.edu.cn

[21] Sheng L Y, Cao L L, Sun K H *et al* 2005 *Acta Phys. Sin.* **54** 4031 (in Chinese) [盛利元、曹莉凌、孙克辉等 2005 物理学报 **54** 4031]

[22] Alvarez G, Montoya F, Romera M *et al* 2004 *Phys. Lett.* A **326** 211

One-way Hash function construction based on tangent-delay ellipse reflecting cavity-map system

Sheng Li-Yuan Li Geng-Qiang Li Zhi-Wei

(*School of Physics Science and Technology, Central South University, Changsha* 410083, *China*)

(Received 24 January 2006; revised manuscript received 21 March 2006)

Abstract

An algorithm for one-way Hash function construction based on tangent-delay ellipse reflecting cavity-map system (TD-ERCS) is proposed in this paper. In the algorithm, the plaintext dealt with is first transformed into a systemic parameter sequence linearly, and then TD-ERCS is iterated in order of the parameter sequence directly, the final Hash value of 160 bits is obtained by means of the nonlinear transform on the iteration sequence, and no padding of calculation is added. Users' keys of the algorithm can be chosen in the region $[2^{64}, 2^{160}]$ arbitrarily. Theoretical analysis and basic security tests indicate that our Hash function has good one-way, weak collision property, better security than other chaotic Hash functions, and it can be realized easily with great rapidity. Our algorithm of Hash function is an ideal substitution for conventional Hash function. And also, a natural criterion (theoretical value of 85.33) to evaluate collision property of Hash function is educed in this paper.

Keywords: chaos, Hash function, TD-ERCS

PACC: 0545