一种新的混沌映射散列函数构造方法及应用*

何婷婷 罗晓曙† 廖志贤 韦正丛

(广西师范大学电子工程学院, 桂林 541004)

(2011 年 6 月 25 日收到; 2011 年 10 月 27 日收到修改稿)

提出了一种基于混沌映射和乘同余法构建单向散列函数的算法. 该算法通过乘同余法生成伪随机序列作为系统的初始值, 把明文信息的美国标准信息交换码 (ASCII 码) 归一化后作为混沌映射的初始值, 经过可变步长的混沌数字量化后, 提取出 128 bit 的散列值. 理论分析和仿真结果表明: 该算法具有较好的不可逆性、抗碰撞性、防伪造性、初值敏感性以及较高地运行速度.

关键词: 散列值, 混沌映射, 散列函数, 敏感性

PACS: 05.45.Ac, 05.45.Pq, 05.45Vx

1 引 言

随着科技的进步, 网络通信安全问题日趋严峻. 在日常网络通信环境中, 可能受到泄密、传输分析、伪装、内容修改、顺序修改和计时修改等六种攻击. 抗前两种攻击的方法属于信息保密范畴. 对付第 3 种至第 6 种攻击的方法一般称为消息认证. 归纳起来, 消息认证就是验证所收到的消息确实是来自真正的发送方且未被修改的消息, 它也可验证消息的顺序和及时性 [1]. 单向散列函数是消息认证码的一种变体, 近年来已经成为信息领域中的热点研究问题.

传统的散列函数包括消息摘要算法第五版 (MD5)、安全散列标准修订版 (SHA-1) 等, 都已经被证实为不安全的, 因此人们迫切需要寻找新的散列算法. 由于混沌系统本身具有很好的初值敏感性和混乱扩散能力, 利用混沌系统来构造单向散列函数已经成为一个有研究价值的问题. 文献 [2—8] 分别采用混沌动态参数、混沌映射、广义混沌映射切换、二维超混沌映射、双混沌系统、超混沌 Chen 系统和混沌神经网络来构建散列函数并做出了重要成果. 但是这些算法迭代次数较多, 实现比较复杂, 不适合用来加密字符较少的通信协议.

本文基于 Logistic 映射和乘同余法构建了一个单向散列函数. 通过乘同余法生成伪随机序列作为系统的初始值可使输入更具随机性. 每处理完一个分组后经过可变步长的混沌数字量化, 可以消除数据的规则性. 研究结果表明: 该算法具有较好的抗碰撞性、单向性、快速性和易实现等特点, 满足在通信系统中的应用要求.

2 单向散列函数和 Logistic 混沌映射

2.1 Logistic 混沌映射

本文算法采用 Logistic 混沌映射, 定义式 [9] 为

$$x_{n+1} = ux_n(1-x_n), \tag{1}$$

其中, 状态量 $x_n \in [0,1]$, 系统控制参数 $u \in [0,4]$. 当 $u \in [3.9974495, 3.9999995]$, Lyapunov 指数大于零时, 系统进入混沌状态 [10].

2.2 单向散列函数定义

散列值 h 由下述形式的函数 H 生成:

$$H = H(M), \tag{2}$$

* 国家自然科学基金 (批准号: 10862001, 10947011) 和广西研究生教育创新计划 (批准号: 2010106020809M50, 2011106020809M50) 资助的课题.

† E-mail: lxs@mailbox.gxnu.edu.cn

其中 M 是一个变长消息, $H(M)$ 是定长的散列值. 当消息正确时, 将散列值附于发送方的消息后, 接收方通过重新计算散列值可认证该消息. 由于散列函数本身不具有保密性, 因此需要采用某些方法来保护散列值. 散列函数要求具有单向性、初值敏感性、防伪造性和易于实现四个特性.

3 单向散列函数的构造算法

该算法的输入是最大长度小于 2^{128} bit 的消息, 考虑到该算法要应用在实际的通信系统中, 且通信协议的字符长度较短, 对算法的执行速度要求较高, 所以该算法对输入消息以 $N = 8$ bit 为单位进行分组处理, 输出是 128 bit 的消息摘要, 图 1 为散列算法结构图.

步骤 1 消息分组. 首先对消息 M 进行分组, 若最后一个消息分组的长度为 L, 且 $L < N$, 则需要对该分组附加填充位. 填充数据的位数在 1 到 N 之间. 填充数据的内容由 1、后续的 0 以及消息分组的长度 L 组成.

步骤 2 设置系统初始值. 通过乘同余法生成伪随机序列的方法来生成初始值, 令所得初始值为 t. 乘同余法的定义式为

$$x_n = ax_{n-1} \,(\mathrm{mod}\ B)\,,$$
$$(x_0,\ B) = 1, \quad r_n = x_n/B, \tag{3}$$

其中 B 为模数, a 为乘子, x_0 为初始值, x_n, B, a 均为非负整数. 由 (3) 式产生的 x_n $(n = 1, 2, \cdots)$ 满足: $0 \leqslant x_n \leqslant B$, 使 $r_n \in [0, 1]$, 且 r_n 的值是均匀分布的随机数. 再通过一个数字量化过程, 得到一个只包含 0—9, A—F 数码的 128 bit 的随机数列. 参数 a, B, x_n 具有敏感性, 当它们的值发生微小的变化时, 随机序列会发生巨大的改变.

步骤 3 将每个分组转换为各自对应的美国标准信息交换码 (ASCII) 码值 k_i, 把第一个分组转换所得数值通过 $y_0 = k_0/2^8$ 线性映射到 (0, 1) 开区间. 以 y_0 作为混沌映射的迭代初始点, 系数 $u = 3.9999995 - k_i \times 0.00001$, 通过 1000 次的迭代运算, 记录第 $k_0 + 100$ 次至 $k_0 + 131$ 次的迭代结果. 将表 1 循环移位 W 次 ($W = k_0 \bmod 16$), 并将这些结果通过表 1 的对应关系转换成 128 bit 二进制序列 x_0, 计算 $x_0 \oplus t$ 即为第一个分组的散列值 H_0, 同时把第 $k_0 + 131$ 次的迭代结果作为下一个消息分组的迭代初始值 y_1.

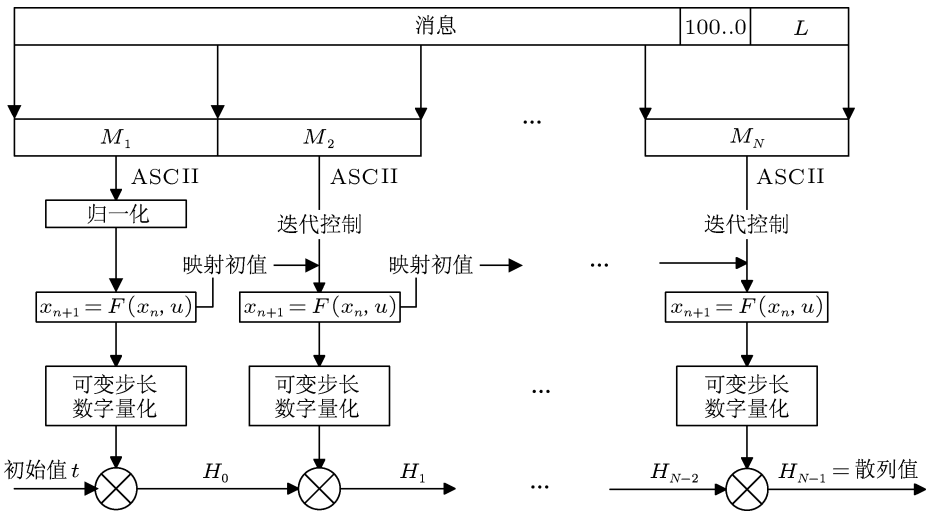

图 1 散列算法结构图

表 1 混沌序列值与二进制序列的对应关系

混沌序列值	二进制序列	混沌序列值	二进制序列	混沌序列值	二进制序列	混沌序列值	二进制序列
(0,1/16)	000(0)	[1/4, 5/16)	0100 (4)	[1/2, 9/16)	1000 (8)	[3/4, 13/16)	1100(C)
[1/16, 1/8)	0001(1)	[5/16, 3/8)	0101(5)	[9/16, 5/8)	1001(9)	[13/16, 7/8)	1101(D)
[1/8, 3/16)	0010(2)	[3/8, 7/16)	0110(6)	[5/8, 11/16)	1010 (A)	[7/8, 15/16)	1110 (E)
[3/16, 1/4)	0011 (3)	[7/16, 1/2)	0111 (7)	[11/16, 3/4)	1011 (B)	[15/16, 1)	1111 (F)

步骤 4 从第二个消息分组开始, 使用前一分组的第 k_0+131 次的迭代结果作为本次迭代的初始值. Logistic 映射的系数 $u=3.9999995-k_i\times 0.00001$, 通过 1000 次的迭代运算, 记录第 k_1+100 次至 k_1+131 次的迭代结果. 将表 1 循环可变步长移位 W 次 ($W=k_1 \bmod 16$), 并将这些结果通过表 1 的对应关系转换成 128 bit 二进制序列 x_1, 计算 $H_0\oplus x_1$, 即为第二个分组的散列值 H_1. 同理可求得其他分组的散列值.

步骤 5 当所有的分组处理完毕后, 最后一个分组的散列值 H_{N-1} 就是最终长度为 128 bit 的 Hash 函数值.

消息 M 有着不可逆的非线性关系. 但是在该算法中通过使用上一个消息分组的迭代终值作为下一个分组的迭代初始值, 其目的是为了使散列值同明文的每比特密切相关, 提高算法的并行度. 当消息 M 发生微小的改变时, 必将引起散列值 H_i 极大地改变. 在算法中对不同的消息分组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, 但是由于迭代值选择不一样, 也可以保证最后的散列值不一样 [11]. 传统的散列函数的构造只是将每个分组相应位异或 (XOR), 这种算法用于随机数的数据完整性检查是比较有效的. 但是若数据格式不是随机的, 则会降低函数的有效性. 本文对该算法进行改进, 每处理完一个分组后, 使表 1 循环移位 W 次. 通过查询表 1 把迭代结果转换成 128 bit 的二进制序列, 并将该值与上一分组的散列值进行异或. 这样可使输入更具随机性, 从而消除输入数据的规则性.

4 算法安全性分析

4.1 敏感性分析

对一个 128 bit 的散列值而言, 每一位可取的值只有 0 和 1, 理想的雪崩效应是明文发生微小的变化将导致散列值以每比特 50%概率发生变化. 对于一条确定的明文, 每次改变其 1 bit 位上的值, 即将第 i 个位的 “0”(“1”) 改为 “1”(“0”), 计算改变后明文的散列值 h_i, 然后将其和原始明文的散列值 h_0 进行比较, 并计算 h_i 和 h_0 二进制表示的 Hamming 距离 $D(H_o,H_i)$, 获得的散列值比特变化率如下式所示 [8]:

$$r(i)=\frac{D(h_0,h_i)}{128}\times 100\%. \tag{4}$$

本文对所构造的单向散列函数用以上的方法进行了敏感性测试. 分别取明文的长度为 1024 和 4096 bit, 每次仅改变 1 个比特值, 计算改变后明文的散列值, 将其与原始明文的散列值进行比较, 其比较仿真结果如图 2 和图 3 所示. 从图 2 和图 3 中我们可以清楚地看到, 当明文任一比特发生变化时, 散列值比特变化率在 50%左右浮动. 仿真结果表明, 该算法具有理想的雪崩效应和敏感性, 当明文发生微小改变时都会引起散列值发生巨大变化.

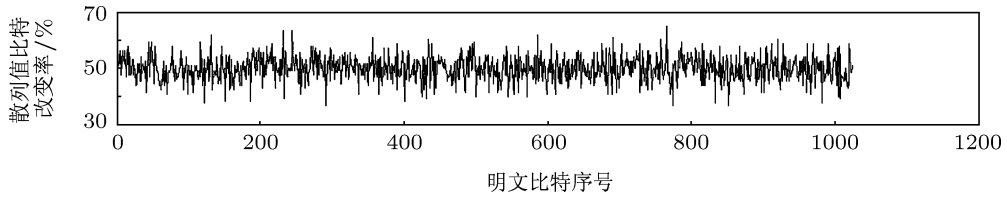

图 2 明文为 1024 bit 的散列值改变率

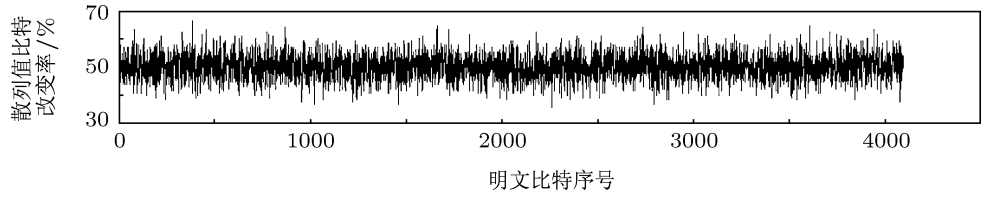

图 3 明文为 4096 bit 的散列值改变率

4.2 文本散列值分布分析

散列函数是否具有安全性很大程度上取决于散列值的分布. 若散列值分布不均匀, 密码盗窃者就可以通过各种技术手段对信息进行破译, 从而达到窃取、篡改密文的目的.

针对以下 6 种条件, 分别用本算法进行文本的散列值仿真.

初始明文: Guilin is regarded as the most picturesque city in China. Two crystal-clear rivers meander through the city, which are encircled by hills with unusual & bizarre rock formations and caves inside. Two crusted movements of earth took place about 200 and 180 million years age thrusting the limestone sediments out of the sea bottom. They were forced upwards more than 200 meters to the surface. This Karst formation was molded through many years of erosion by the wind and rain to become the hills and rocks with bizarre shapes. There are numerous complete Karst (limestone sites), which are of high scientific value and tour value. Guilin is named after the fragrance of osmanthus tree and saw its first inhabitants in Qin Dynasty over 2000 years ago. Guilin experienced a prosperous period during Tang, Song, Yuan, Ming and Qing Dynasty under the patronage from succeeding Emperors.

条件 1 初始明文;

条件 2 将初始文本中的首字符 G 改为 g;

条件 3 将初始文本中的数值 180 改为 181;

条件 4 将初始文本中的字符 forced 改为 enforced;

条件 5 将初始文本中末尾的句号改为逗号;

条件 6 在初始文本的末尾加上一个空格.

仿真得到的散列值用十六进制数表示如下:

条件 1 35F3DAF1339C53474E82BF59DE3A2B9F;

条件 2 0FEB4A8B8096F57F8191D8C914CFBE30;

条件 3 B09D5273D7F0624579901A51A8FC134C;

条件 4 5C665EDD631A12549CEC50AA98451399;

条件 5 DB32E228CE870BD3F60024D61E962722;

条件 6 C4E39E47B00DC0B959D42253E6135C8F.

将明文和散列值用二维图形形象展示它们之间的区别, 如图 4、图 5 所示. 从图 4 中我们可以很清楚地看到, 初始明文的 ASCII 码值都集中在一个比较小的范围之内. 从图 5 可以看出经过该算法散列后的十六进制 Hash 值, 可以较均匀地呈现出分散态. 因此, 通过扰乱、扩散等作用下的 Hash 值已经没有包含任何的统计信息, 达到了构建散列函数的各项指标.

对不同条件下的散列值分别用 0, 1 序列的图形化表示, 如图 6 所示. 从图 6 可以看出, 当明文发生微小的改变时, 会使散列值发生巨大地变化. 假若我们把该散列值用在通信系统中, 当明文被攻击者篡改、攻击, 造成数据丢失时, 都会使散列值发生改变, 接收方就可以通过校验散列值而丢弃该报文的接收, 从而保证了通信的安全性.

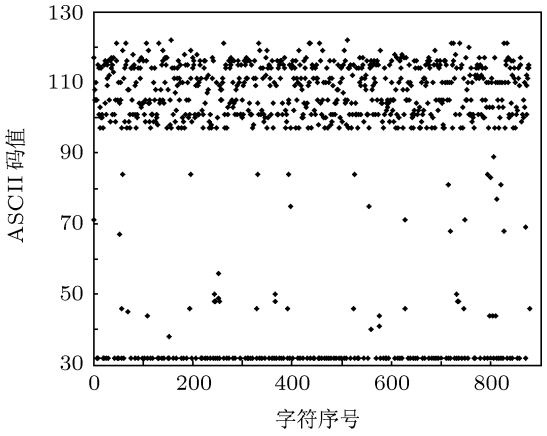

图 4 初始明文 ASCII 码值分布图

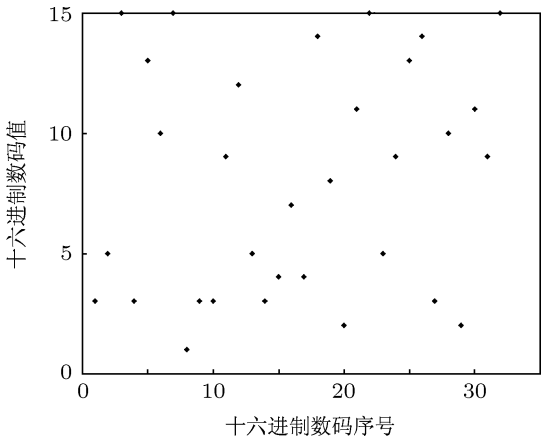

图 5 初始明文十六进制散列值分布图

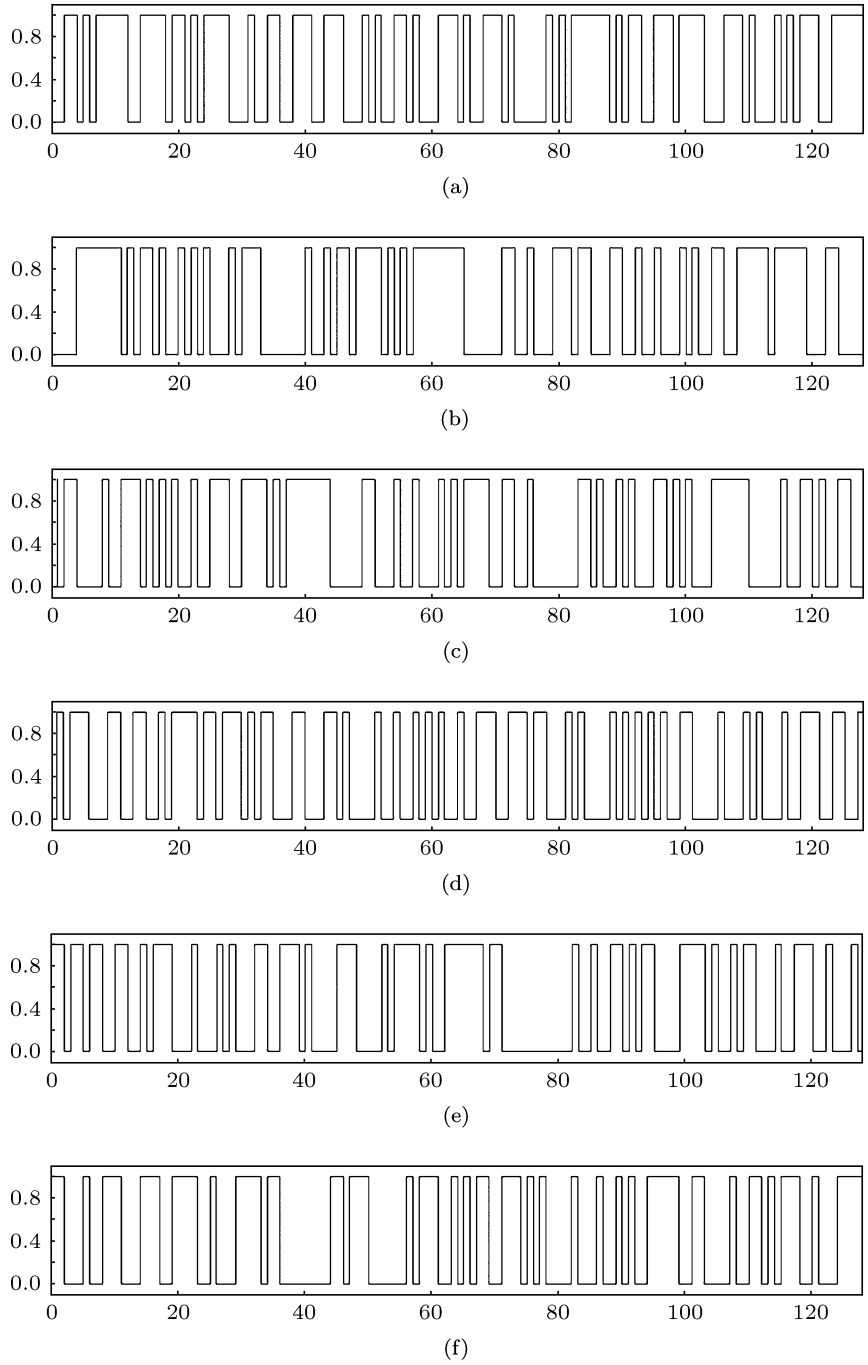

图 6 初始明文在六种条件下的散列值比较 (a) 条件 1; (b) 条件 2; (c) 条件 3; (d) 条件 4; (e) 条件 5; (f) 条件 6

4.3 散列值扩散与混乱特性分析

混乱与扩散这两个概念是在 Shannon 的信息论中提出的. 它是加密体质中的两个重要性质. 它要求每位明文的影响尽可能地扩散至更多位的散列值中, 同时也希望明文与散列值具有更复杂的关系, 其目的是为了掩盖明文的概率特性. 我们将明文映射至参数空间中, 并使得不同的明文分组具有不同的迭代次数, 这样可以更有效地抵抗已知明文攻击和差分攻击. 因此, 散列值的理想情况是明文的细微变化引起散列值每比特以 50%的概率变化. 定义以下四个统计量 [7] 对该算法进行特性分析:

$$\bar{B} = \frac{1}{N}\sum_{i=1}^{N} B_i, \tag{5}$$

$$\Delta B = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(B_i - \bar{B}\right)^2}, \tag{6}$$

$$P = (\bar{B}/128) \times 100\%, \tag{7}$$

$$\Delta P = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(B_i/128 - P\right)^2} \times 100\%. \tag{8}$$

以上 4 个公式中, N 为统计次数, B_i 为第 i 次试验结果的变化比特数, P 为平均变化概率, $\bar{B}$ 为平均变化比特数, ΔB 为 B 的均方差, ΔP 为 P 的均方差. 试验方法为, 从明文空间中随机抽取一段明文进行测试, 改变任一比特位, 与原始文本进行比较, 统计出不相等的比特位. 经过 $N = 128, 200, 256, 512, 1024, 2048$ 次测试, 得到的统计结果如表 2 所示.

表 2 明文每比特变化时的 4 个统计量值

统计次数 N	$\bar{B}$	ΔB	$P/\%$	$\Delta P/\%$
128	63.333	5.832	49.479	4.684
200	64.015	5.784	50.012	4.803
256	63.570	5.817	49.664	4.568
512	63.792	5.902	49.838	4.824
1024	64.801	5.893	50.626	4.743
2048	63.473	5.924	49.588	4.597
平均值	63.831	5.859	49.868	4.703

从表 2 可以看出, 散列值的平均变化比特数已经非常接近 64, 基本上达到理想状态. ΔB, ΔP 标志着整个算法的稳定性和混乱性, 试验所得两个统计值很小且很接近. 因此, 该算法具有稳定性, 并具有很强的混乱、扩散能力.

4.4 抗碰撞性分析

散列函数抗穷举分析的能力仅仅依赖于算法所产生的散列码长度. 一般来说散列码的长度要大于等于 128 bit 就能满足一般实际应用的需求, 并能抗生日攻击.

抗碰撞性是指找到具有相同散列值的两个不同明文在计算上是不可行的. 对本文算法进行仿真分析, 首先取初始明文一字节, 即为 8 bit, ASCII 码对应的值为 0—255; 散列结果同样也取 8 bit, 值也为 0—255 之间的数, 这样做的目的是使明文空间与散列值空间相等. 记散列值空间即像空间中任一值对应明文空间中原像的个数为 k, 散列空间中具有 k 个原像的点的个数为 $n(k)$. $n(1)$ 越大, 其他各项的值越小, 则碰撞越少, 该算法的混乱能力越强. 因此, $n(1)$ 的大小可作为衡量抗碰撞性能的指标, 从 $n(k)$ 的分布情况可统计出该算法的抗碰撞性能 [7]. 定义性能指标 $T(k)$ 为

$$T(k) = \frac{n(k)}{\sum_{k=0}^{k} n(k)}. \tag{9}$$

通过试验, 本文散列算法的 $n(0)$—$n(4)$ 值依次为 75, 116, 49, 12, 4. 当 $k > 4$ 时 $n(k) = 0$, 图 7 为 k-$n(k)$ 分布图. $T(k) = n(1)/256 = 116/256 = 0.453125$, 可见该算法具有很好的抗碰撞性能.

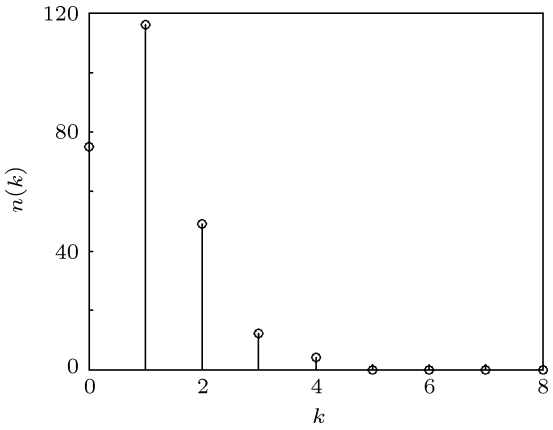

图 7 k-$n(k)$ 分布图

5 结 论

本文提出了一种新的混沌映射单向散列函数构造算法并讨论了其在通信系统中的应用. 通过试验表明, 该算法具有很好的初值敏感性和迭代过程的单向性, 从结构上保证了散列值与明文息息相关. 当明文有微小的改变, 散列值以接近每比特 50%的概率变化, 具有理想的雪崩效应. 这就保证了该散列算法具有较强的混乱和扩散能力, 在保证数据安全性的同时也保证了较高的执行速度, 达到了散列函数的各项要求.

[1] William S (translated by Meng Q S, Wang L N, Fu J M) 2007 *Cryptography and Network Security Principles and Practices* (4th Ed.) (Beijing: Electronic Industry Press) pp310–413 (in Chinese) [威廉 S 著 (孟庆树, 王丽娜, 傅建明译) 2007 密码编码学与网络安全: 原理与实践 (第 4 版) (北京: 电子工业出版社) 第 310—413 页]

[2] Guo W, Cao Y, Wang X M, He D K 2008 *J. Commun.* **29** 93 (in Chinese) [郭伟, 曹杨, 王小敏, 何大可 2008 通信学报 **29** 93]

[3] Liu J N, Xie Q C, Wang P 2000 *J. Tsinghua Univ.* (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]

[4] 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]

[5] 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]

[6] Wei P C, Zhang W, Liao X F, Yang H Q 2006 *J. Commun.* **27** 27 (in Chinese) [韦鹏程, 张伟, 廖晓峰, 杨华千 2006 通信学报 **27** 27]

[7] Ren H P, Zhuang Y 2009 *J. Commun.* **30** 100 (in Chinese) [任海鹏, 庄元 2009 通信学报 **30** 100]

[8] Liu G J, Dan L, Dai Y W, Sun J S, Wang Z Q 2006 *Acta Phys. Sin.* **55** 5688 (in Chinese) [刘光杰, 单梁, 戴跃伟, 孙金生, 王执铨 2006 物理学报 **55** 5688]

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

[10] Liu Y Z, Lin C S, Li X C, Liu H P, Wang Z L 2011 *Acta Phys. Sin.* **60** 030502 (in Chinese) [刘扬正, 林长圣, 李心朝, 刘海鹏, 王忠林 2011 物理学报 **60** 030502]

[11] Wang J Z, Wang Y L, Wang M Q 2006 *Acta Phys. Sin.* **55** 5048 (in Chinese) [王继志, 王英龙, 王美琴 2006 物理学报 **55** 5048]

A new chaos mapping hash function structural method and its application*

He Ting-Ting Luo Xiao-Shu† Liao Zhi-Xian Wei Zheng-Cong

(*College of Electronic Engineering, Guangxi Normal University, Guilin* 541004*, China*)

(Received 25 June 2011; revised manuscript received 27 October 2011)

Abstract

A one-way hash function algorithm is proposed based on the chaos mapping and multiplicative congruential method. The initial value of the system is generated by the pseudo-random sequence which is obtained through the multiplicative congruential method. the normalized ASCII of the plaintext is used as the initial value of the chaos mapping. After a variable-step chaotic digital quantification, 128 bit hash value is extracted from the systems. Theoretical analysis and simulation results show that the proposed method has better characteristics of irreversibility, collision resistance, anti-forgery, initial sensitivity and higher operation speed.

Keywords: hash value, chaos mapping, hash function, sensitivity

PACS: 05.45.Ac, 05.45.Pq, 05.45Vx

* Project supported by the National Natural Science Foudation of China (Grant Nos. 10862001, 10947011) and the Innovation Project of Guangxi Graduate Education, China (Grant Nos. 2010106020809M50, 2011106020809M50).

† E-mail: lxs@mailbox.gxnu.edu.cn