一种基于混沌的可并行 Hash 函数*)

邓绍江[1] 廖晓峰[1] 肖 迪[1,2]

(重庆大学计算机科学与工程学院 重庆 400044)[1] (重庆大学机械工程学院 重庆 400044)[2]

摘 要 本文针对文献[4,5]中基于混沌的 Hash 构造算法的缺陷,提出一种新的算法。该算法在保证安全性的前提下,具有适合并行实现及最终 Hash 值对明文信息敏感依赖的均匀性等显著的优点。理论分析和仿真实验证明该算法可以满足 Hash 函数的各项性能要求。

关键词 混沌,Hash 函数,并行

A Parallel Hash Function Based on Chaos

DENG Shao-jiang[1] LAIO Xiao-feng[1] XIAO Di[1,2]

(College of Computer Science and Engineering, Chongqing University, Chongqing 400044, China)[1]

(College of Mechanical Engineering, Chongqing University, Chongqing 400044, China)[2]

Abstract In this paper, a novel hash algorithm based on chaos is proposed to overcome the flaws of the corresponding algorithms in [4, 5]. Its security can be guaranteed, and it can work efficiently in a parallel way. Furthermore, the algorithm structure can ensure the uniform sensitivity of final hash value to the message blocks at different positions of the whole message. Theoretical analysis and computer simulation indicate that the proposed algorithm can satisfy the performance requirements of hash function.

Keywords Chaos, Hash function, Parallel

1 引言

在曾经公认安全的传统 Hash 函数如 MD5,SHA-1 被用"模差分"方法成功攻击后,寻找新的安全的 Hash 函数的任务变得极为迫切。利用混沌构造 Hash 函数作为一个新颖的思路,已成为人们关注的热点。在文献[1]中,Yi 提出了一种基于帐篷映射的散列函数构造方法;在文献[2]中,我们提出了一个基于变参数混沌映射的单向 Hash 函数构造算法;在文献[3]中,我们基于 logistic 映射,利用动态查询表实现了一种改进的加密/散列组合方案;文献[5]针对文献[4]的 Hash 算法的一些缺陷,提出了一种构造算法。本文在对文献[4,5]的算法缺陷作进一步分析的基础上,提出一种新的混沌 Hash 函数构造算法。与现有的各种基于混沌的 Hash 算法相比,在保证安全性的前提下,该算法的显著优点是适合于并行实现以及最终的 Hash 值对明文信息敏感依赖的均匀性。该算法可以满足 Hash 函数的各项性能要求。

2 预备知识

Hash 函数 H 用于将任意长度的消息 M 映射为较短的、固定长度的一个值 $H(M)$,应满足以下条件[6]:

(1)已知 x,求 $H(x)$较为容易;而已知 h,求使得 $H(x)=h$ 的 x 在计算上不可行。

(2)已知 x,找到 $y(y\neq x)$使得 $H(y)=H(x)$在计算上不可行。

(3)找到任意两个不同的输入 x,y,使得 $H(y)=H(x)$在计算上不可行。

我们先简单回顾一下文献[4]的 Hash 函数算法,它采用了 Logistic 混沌映射 $f(X)=1-\mu * X^2$,其实现步骤如下:

(1)将待处理的明文分块,并将每个明文信息块量化。该方案是以一个字符作为一个明文信息块并转换成对应的 ASCII 码值。

(2)将量化后的值变换到某一值域,使得用该集合内的值作为参数能够使系统处于混沌状态:$\mu_i=1.746+0.001^* Asc(m_i)$。

(3)将 μ_i 带入混沌系统 $f(X)=1-\mu^* X^2$ 中,取一个固定的初始值,经过 32 次迭代可得到序列$\{X_i\}$。

(4)将每个 X_i 通过查表的方式转换成 4 位二进制序列 S_i 并级连起来,因此序列$\{X_i\}$将得到长度为 $4\times32=128$ 位的二进制序列。

(5)最后将每个明文信息块所得的各个 128 位二进制序列逐位模 2 相加,即得到全部明文的 128 位 Hash 值。

该算法存在一个严重的隐患,其根源在于对每个明文信息块的处理是完全独立的,彼此缺乏关联。假定原来的明文段是"ABCDEFG",现在被篡改为"GFEDCBA",最终的 Hash 值仍将是一样的。事实上,任意改变字符的排列顺序,都不会影响最终的 Hash 结果。

文献[5]注意到了文献[4]的这个缺陷,将原算法第(3)步改为:将处理上一个明文信息块后的最终迭代值作为下一个明文信息块的迭代初始值,开始进行新的 32 次迭代。然而,这种串型处理结构,每处理一个明文信息块(一个字符)就需要 32 次迭代/64 次乘法运算,当一段消息字符数较多时,整个算法的效率将极其低下。

文献[5]针对文献[4]的算法在对 ASCII 码值相近的字符进行 Hash 时前面几位可能相同的情况,提出的改进是以 Random(-1,1)随机分配[-1,1]之间的随机数作为 Hash 整段消息时最初的迭代初始值。然而,这种改进没有注意到混沌的一个著名特性——蝴蝶效应,当两次迭代的初始值有极细微的不同时,所得到的迭代轨道将完全不同,因此,文献[5]的改进算法在不同的时间对同一段消息进行 Hash 处理,所得到的最终 Hash 值将会不同。

*)国家自然科学基金资助项目(60703035),中国博士后科学基金资助项目(20070410204,20060400714),重庆市科委自然科学基金资助项目(CSTC2007BB2130,2006BB2227)。邓绍江 副教授,博士后,主要研究方向为混沌密码学;肖 迪 副教授,博士后,主要研究方向为混沌密码学。

3 基于混沌的可并行 HASH 函数

针对以上算法的缺陷，本文提出一种新的混沌 Hash 函数构造算法，算法采用分段线性混沌映射[7]：

$$X(t+1)=F_Q(X(t))$$

$$=\begin{cases} X(t/Q,) & 0\leqslant X(t)<0 \\ (X(t)-Q)/(0.5-Q), & Q\leqslant X(t)<0.5 \\ (1-X(t)-Q)/(0.5-Q), & 0.5\leqslant X(t)<1-Q \\ (1-X(t))/Q, & 1-Q\leqslant X(t)\leqslant 1 \end{cases}$$

其中，$X\in[0,1]$，$Q\in(0,0.5)$。该迭代系统是混沌的，其输出序列$\{X(t)\}$在$[0,1]$上遍历且均匀分布，具有良好的自相关特性且呈δ形态。

3.1 算法描述

(1)以一个字符作为一个明文信息块将待处理的明文分块。

(2)将第 i 个明文信息块 $M_i=(p_{i1}\ p_{i2}\cdots p_{i8})$（其中，$i=0,1,\cdots,s$，$s$ 为明文所包含的信息块数）转化为一对二进制小数$(m_i,\widetilde{m}_i)$，其中 $m_i=0.\ p_{i1}\ p_{i2}\cdots p_{i8}$，$\widetilde{m}_i=0.\ p_{i8}\ p_{i7}\cdots p_{i1}$。计算 $z_i=i/s$，将当前序号“i”规格化处理为 $z_i\in(0,1]$；计算 $uu_0=(\widetilde{m}_i+z_i)/4\in(0,0.5)$。

(3)分别将 uu_0，m_i 作为处理第 i 个消息块 M_i 时分段线性混沌映射的初始参数和初始值，经过 34 次迭代，顺次取后 32 次迭代值可得到序列$\{X_j\}(j=1,2,\cdots,32)$。

(4)将每个 X_j 通过查表 1 转换成 5 位二进制序列 S_j，级连起来得到长度为 $5\times32=160$ 位的二进制序列。

(5)最后将每个明文信息块所得的各个 160 位二进制序列逐位模 2 相加，即得到全部明文的 160 位 Hash 值。

表 1

[31/32, 1]	[30/32,31/32)	[29/32,30/32)	[28/32,29/32)	[27/32,28/32)	[26/32,27/32)	
11111	11110	11101	11100	11011	11010	
[25/32,26/32)	[24/32,25/32)	[23/32,24/32)	[22/32,23/32)	[21/32,22/32)	[20/32,21/32)	
11001	11000	10111	10110	10101	10100	
[19/32,20/32)	[18/32,19/32)	[17/32,18/32)	[16/32,17/32)	[15/32,16/32)	[14/32,15/32)	
10011	10010	10001	10000	01111	01110	
[13/32,14/32)	[12/32,13/32)	[11/32,12/32)	[10/32,11/32)	[9/32,10/32)	[8/32,9/32)	[7/32,8/32)
01101	01100	01011	01010	01001	01000	00111
[6/32,7/32)	[5/32,6/32)	[4/32,5/32)	[3/32,4/32)	[2/32,3/32)	[1/32,2/32)	[0/32,1/32)
00110	00101	00100	00011	00010	00001	00000

3.2 算法的构造特点

(1)安全性

在处理每个信息块 $M_i(i=1,2,\cdots,s)$时，分段线性混沌映射的参数和初值都将根据不同位置的信息块而动态变化。这一方面是以一种简单可行的方式引入了扰动，可防止混沌的动力学特性退化；另一方面使得所产生的序列与信息块 M_i $(i=1,2,\cdots,s)$自身的构成及其顺序号“i”密切相关。这种机制是算法安全的重要基础：因为不同的信息块 M_i 所对应的序列互不相同；而同一个消息块，当其顺序号“i”取不同值时，产生的序列也是不同的。虽然本算法对每个明文信息块的处理仍然是独立的，但由于处理过程和每个消息块顺序建立了关联关系，从而巧妙地消除了文献[4]算法的隐患。

(2)并行性

从算法描述可以清楚地看出，步骤(2)-(4)对每个明文信息块的处理是相互独立的，完全可以并行进行，从而保证了整个算法运算的高效性。此外，在处理第 i 个消息块 M_i 时，经过 34 次迭代，舍去前两次的迭代值，顺次取后 32 次迭代值得到序列$\{X_j\}\{j=1,2,\cdots,32\}$，可以解决文献[4]的算法在对 ASCII 码值相近的字符进行 Hash 时前面几位可能相同的问题。这是因为每个信息块与分段线性混沌映射的参数和初值都相关，其相关性比原算法每个信息块仅仅和混沌映射的参数相关大大增强了。

(3)敏感依赖的均匀性

最终 Hash 值是将每个明文信息块所得的各个 160 位二进制序列逐位模 2 相加而得出，由模 2 相加运算的特点可以看出，每个明文信息块 $M_i(i=1,2,\cdots,s)$的改变，必将导致相应的二进制序列的改变，其对最终 Hash 值的影响作用是等价的。这保证了最终 Hash 值对消息明文敏感依赖的均匀性。

4 性能分析

4.1 Hash 值对明文信息的敏感依赖

用本算法分别在 6 种条件下进行明文信息的 Hash 仿真：

条件 1：初始文本 1“Unique merits of chaos bring much promise of application in the information security field.”；条件 2：文本 2 将文本 1 的首字母 U 改为 V；条件 3：文本 3 将文本 1 中的 much 改为 mach；条件 4：文本 4 将文本 1 末尾的句号改成逗号；条件 5：文本 5 在文本 1 的最后再添加一个空格；条件 6：将初始文本第 1 个明文信息块“U”和第 2 个明文信息块“n”交换。仿真得到的 Hash 结果用 16 进制数表示如下：

条件 1：DBA3E1B6736F6B7933C316CD415D5EDD078B08B0

条件 2：C0DAE8934C7DA9E5D1741D10E984B25ED7BF118C

条件 3：90468475A2C29AD78B40462CB05C8B94C4DABFF7

条件 4：D8A1C31CE4E8EB79323CCCB7040705993BBB801D

条件 5：78E9A3B9A9CF0C1461682565D1FB1A5CC8525472

条件 6：CA6CF67EAB9DBE3AE0ACB47542B668D0F298F3F4

用 0,1 序列的图形化表示如图 1 所示。

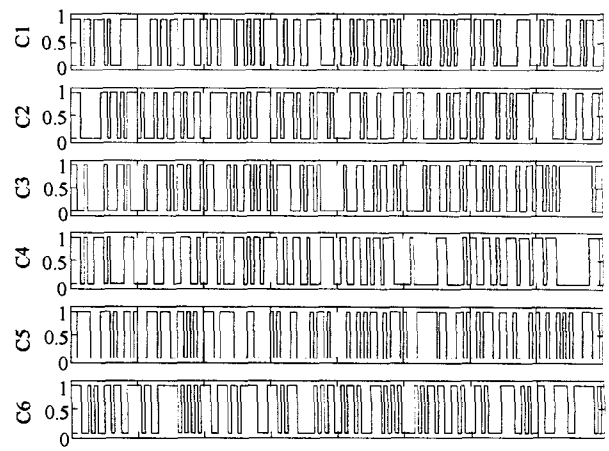

图 1 不同条件下的 Hash 值比较图

从仿真结果可以看出，本算法的单向 Hash 性能和敏感性很好，明文信息的细微变化都将给最终结果带来巨大变化，得到的 Hash 值截然不同。

4.2 混乱与扩散性质统计分析

Hash函数要尽量做到相关明文对应的Hash值不相关，而对于结果的二进制表示，每位只有1或0两种可能，因此理想Hash的扩散效果应该是初值的细微变化将导致结果的每位都以50%的概率变化。我们的测试方法是：每次测试时，在明文信息空间中随机地选取一段明文求其Hash值，然后改变消息明文1位的值得到另一Hash值，比较两个Hash值求出变化的位数 B_i。定义以下4个统计量：

平均变化位数 $\bar{B}=\frac{1}{N}\sum_{i=1}^{N}B_i$，平均变化概率 $P=(\bar{B}/160)\times 100\%$，B的均方差 $\Delta B=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i-\bar{B})^2}$，$P$ 的均方差 $\Delta P=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i/160-P)^2}\times 100\%$。

对文本“Unique merits of chaos bring much promise of application in the information security field.”进行182次测试，得到本算法的实验数据记入表2，相应的位变化数分布图如图2所示。从图2中可以看出，消息明文1位改变所引起的Hash值实际变化的位数非常集中地分布在理想状态下的变化数80附近，表明该算法具备很强的混乱与扩散能力。

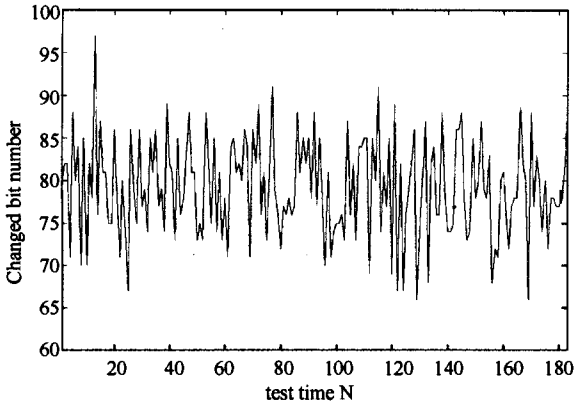

图2 位变化数分布图

表2 1位明文信息变化，Hash值的 $\bar{B}$，P，ΔB 和 ΔP 值

$\bar{B}$	P/%	ΔB	ΔP/%
79.2514	49.53	5.8321	3.65

从表2中的数据可以看出，本算法的平均变化位数和每位平均变化概率都接近理想状态下的80位和50%，相当充分和均匀地利用了密文空间，从统计效果上保证了攻击者无法在已知一些明文密文对的情况下伪造或反推出其它的明文密文。另外，ΔB 和 ΔP 很小，表明本算法Hash的混乱和扩散性能相当稳定。

4.3 抗碰撞分析

我们通过以下的实验来定量测试本算法的抗碰撞能力[2,3]：在消息明文空间中随机地选取一段明文求出并以ASCII码形式存储其Hash值，然后随机地选择并改变消息明文中1位的值得到另一新的Hash结果，同样以ASCII码形式存储其Hash值。比较两个Hash结果，若两个Hash值中相同位置上ASCII码字符相同，则称为被击中一次，统计被击中的次数，并且用公式 $d=\sum_{i=1}^{N}|t(e_i)-t(e'_i)|$ 计算两个Hash值的绝对差异度，其中，e_i 和 e'_i 分别是原始Hash值和新的Hash值的第 i 个ASCII码字符，而函数 $t(*)$ 将ASCII码字符转化为它们相应的十进制数值。做了182次这样的实验，其中仅有8次击中一次，174次没有发生击中，在相同位置上有相同值的ASCII码字符的分布图如图3所示，最大的相同字符数才仅为1，碰撞的程度很低。d 的最大、最小、平均值和平均每个字符的差异度列在表3中。

表3 两个Hash值的绝对差异度

最大值	最小值	平均值	平均值/字符
2542	1119	1911.5	95.575

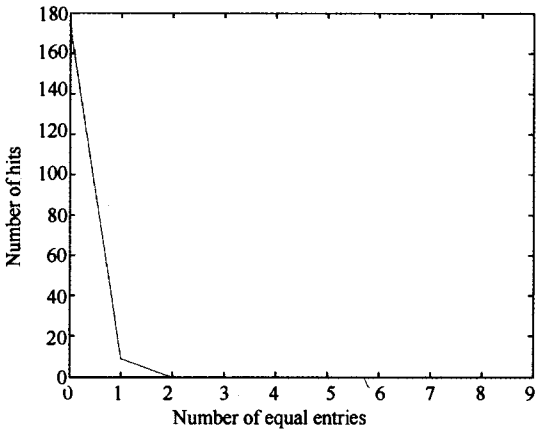

图3 Hash值在相同位置上有相同值的ASCII码字符的分布图

4.4 运算效率分析

由于本算法整体上具有并行的特点，将明文消息按字符分块后，各消息块的处理可以完全并行，因此，与通常的串行模式的Hash算法相比，其在运算效率上的优势是非常显著的。

结束语 在本文中，我们提出一种适合并行实现的Hash函数构造算法，算法从结构上保证了最终Hash值对消息明文敏感依赖的均匀性。该算法符合散列函数的各项性能要求。整个算法简单高效，可操作性强，可靠性较好，是一个在并行计算平台上构造散列函数的优秀候选者。

参 考 文 献

[1] Yi X. Hash function based on chaotic tent maps. IEEE Transactions on Circuits and Systems-II: 2005,52(6):354-357

[2] Xiao D, Liao X F, Deng S J. One-way hash function construction based on the chaotic map with changeable-parameter. Chaos, Solitons & Fractals, 2005,24(1):65-71

[3] Xiao D, Liao X F, Wong K W. Improving the security of a dynamic look-up table based chaotic cryptosystem. IEEE Transactions on Circuits and Systems-Ⅱ, 2006,53(6):502-506

[4] 陈志德，黄元石. 混沌型单向散列函数. 通信技术，2001 (7):96-98

[5] 游中胜，刘锋. 构造基于Logistic映射的Hash函数. 计算机科学，2006,33(4):106-107

[6] 杨波. 现代密码学，第二版. 清华大学出版社，2007

[7] Baranousky A, Daems D. Design of one-dimensional chaotic maps with prescribed statistical properties. Int. J. Bifurcation and Chaos, 1995,5(6):1585-1598