计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2016.20148155
53(3):674-681,2016

一种基于变参级联混沌的 Hash 函数算法

吴 涛[1] 金建国[2] 魏明军[3]
[1](华北理工大学现代技术教育中心 河北唐山 063009)
[2](华北理工大学理学院 河北唐山 063009)
[3](华北理工大学信息工程学院 河北唐山 063009)
(happynat@163.com)

A Hash Function Algorithm Based on Variable Parameter Cascade Chaos

Wu Tao[1], Jin Jianguo[2], and Wei Mingjun[3]
[1](*Modem Technology Education Center, North China University of Science and Technology, Tangshan, Hebei* 063009)
[2](*College of Sciences, North China University of Science and Technology, Tangshan, Hebei* 063009)
[3](*College of Information Engineering, North China University of Science and Technology, Tangshan, Hebei* 063009)

Abstract A Hash function algorithm based on variable parameter cascade chaos is put forward aiming at the possible risk on the letting out of cascade chaos key and the deficiency of present Hash function. That is the status variable of another chaos system as the parameter perturbation is introduced to a Hash function cascade driving system, and the safe variable parameter cascade chaos system is realized with the control of turbulence intensity. The Hash function composed in this way not only obeys the variable parameter characteristic of chaos rules, but also possesses the feature of crosstalk step by step between the cascade subsystems. It can effectively reduce the risk of short period behavior caused by the finite computer precision and digital quantization possible, and it has great significance to improve the complexity and strong collision resistance of the compression function's internal structure. The experimental results show that compared with other chaotic Hash algorithm and SHA-3 algorithm, this algorithm has high sensitivity to initial conditions, nice chaos and diffusion ability, strong collision resistance, simple and flexible algorithm, and strong controllability of variable parameter system; and it has a favorable prospect in the field of chaos secure communication, digital signature, etc.

Key words cascade chaos; Hash function; parameter perturbation; Lyapunov index; anti collision

摘 要 针对级联混沌可能存在的密钥泄漏风险以及当前 Hash 函数的不足,提出了一种基于变参级联混沌的 Hash 函数算法,即在构成 Hash 函数的级联驱动系统中,引入了另一混沌系统的状态变量作为参数扰动,并在扰动强度的控制下实现安全的变参级联系统.由此构成的 Hash 函数不仅具有符合混沌规律的变参特性,同时还具有级联子系统间逐级串扰的性质,能有效降低由计算机有限精度和数字量化可能造成的短周期行为风险,对提高压缩函数内部结构的复杂度和抗碰撞性有着显著意义.实验结果表明:与其他混沌 Hash 算法和 SHA-3 算法相比,该算法具有高度的初值敏感性和良好的混乱与扩散性

收稿日期:2014-10-30;**修回日期**:2015-04-16
基金项目:河北省自然科学基金项目(F2014209108);河北省科技支撑计划基金项目(13210706);唐山市科技计划基金项目(13130208z)
This work was supported by the Natural Science Foundation of Hebei Province of China (F2014209108), the Key Technology Research and Development Program of Hebei Province of China (13210706), and the Science and Technology Project of Tangshan of Hebei Province of China (13130208z).
通信作者:金建国(jjg1956@126.com)

能,抗碰撞能力强,算法实现简单灵活,变参系统可控性强,在混沌保密通信、数字签名等领域具有良好的推广前景.

关键词　级联混沌;Hash 函数;参数扰动;Lyapunov 指数;抗碰撞性

中图法分类号　TP309

Hash 函数作为数字签名等安全技术中的一个关键环节,其安全性历来广受关注.特别是 2004 年以来,对传统 Hash 函数的攻击取得了突破性的进展,MD5,SHA-1,DHA-256 等 Hash 函数的安全缺陷被陆续公布,设计新的 Hash 函数成为各界广泛关注的问题.由于混沌具有高度的敏感性、遍历性和内随机性等密码学特点,因此结合混沌系统构造出高可靠性、高安全性的 Hash 函数,成为近年来新的研究方向和热点,并取得了一些研究成果.

如文献[1]利用斜帐篷映射在特定参数下的混沌行为构建 Hash 函数算法,通过多次迭代混沌方程来扩散消息变化对 Hash 值的影响;文献[2-3]则利用高维混沌映射的不变测度、遍历性和无倍周期分岔等特性构建安全的 Hash 函数,通过非线性耦合作用显著增强了消息块值对混沌状态值的扩散效应,但因算法复杂导致运算效率较低[4].除了将消息映射到相空间的方法外,文献[5]提出了一种简单的可变参混沌 Hash 算法,通过 $\mu=3.9999995-k_i\times 0.00001$ 将明文消息调制到 Logistic 映射的参数空间,利用混沌系统对参数极度敏感的特性达到很好的扩散效应;文献[6]则进一步基于切比雪夫多项式算法提出了一种变参数的并行混沌 Hash 算法,该算法从相应消息块中动态获取可变参数,最终通过一维 Logistic 映射的状态值来构建 Hash 值;为了克服单一迭代系统可能存在的碰撞缺陷,文献[7]提出了基于复合混沌动力系统的 Hash 函数算法,利用二维 Logistic 映射的输出作为分段线性映射的分段参数,最终通过分段线性映射的多次迭代构造 Hash 值.文献[8-10]结合时空混沌的特性,利用其格点间耦合项的扰动作用,使系统具有更好的随机性态和扩散能力.相对于简单混沌映射的 Hash 函数,基于时空混沌构造的 Hash 函数是一个格子之间存在相互耦合作用的复杂高维混沌系统,在带来安全性的同时,也提高了算法的复杂性.上述研究对 Hash 函数的改进作出了重大的贡献,但仍存在一些局限.例如基于离散混沌映射的方案,存在着 Lyapunov 指数小、初值条件和系统参数有限等缺陷,而基于连续混沌和高维混沌映射的方案,由于其算法复杂导致产生的序列码率较低.为此,文献[4]提出了离散混沌的一种级联方案,详细研究了级联对动力学性能的改善,证明了混沌级联是改善系统随机性和复杂性的一种简单实用的方法.级联驱动的方法虽具有序列产生率高、安全性能好的一面,但也存在着潜在的安全威胁.文献[11]就指出由于构成级联系统的子系统间存在着逐级串扰,即使构成级联的各个子系统均是混沌的,但由它们构成的级联系统也有不是混沌的情况,而且级联后的系统是否混沌与构成它的子系统的序有关.

为了克服级联混沌系统可能存在的安全风险,本文提出了一种利用参数扰动项[12]改进级联混沌性能的方案,并以此构建安全可靠的 Hash 函数,即在构成 Hash 函数的级联驱动系统中引入另一混沌系统的状态变量作为参数扰动,并在扰动强度的控制下实现安全的变参级联方案.该方案的优点在于:1)构成 Hash 函数的级联混沌系统为混沌变参系统,Hash 函数内部结构更加难以预测;2)利用逐级串扰特性增强了压缩函数各轮计算之间位模式的安全性,提高了相空间重构、回归映射及碰撞攻击的难度;3)通过参数扰动强度控制变参系统,改善了系统参数的可用空间.

1　参数扰动下的变参级联混沌

1.1　参数扰动下的变参级联混沌方程

以 Logistic 映射构成的级联混沌为驱动系统,一维 ICMIC 映射[13]为扰动系统构造变参级联混沌系统,系统状态方程如下:

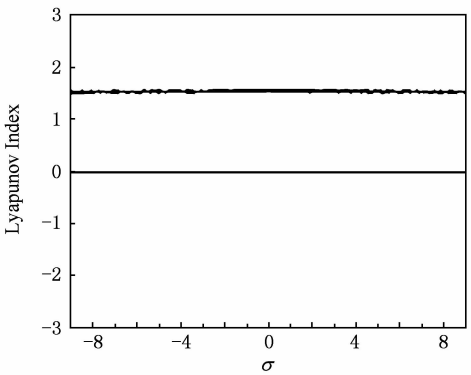
$$\begin{cases} x_1=1-(\mu_1+\sigma z_1)x_0^2;\\ x_2=1-(\mu_2+\sigma z_2)x_1^2;\\ \vdots\\ x_j=1-(\mu_j+\sigma z_j)x_{j-1}^2;\\ x_{j+1}=1-(\mu_{j+1}+\sigma z_{j+1})x_j^2;\\ \vdots\\ x_m=1-(\mu_m+\sigma z_m)x_{m-1}^2. \end{cases} \tag{1}$$

$$z_{n+1}=\frac{1}{1\,000}\sin\left(\frac{a}{z_n}\right),a\in(0,\infty). \tag{2}$$

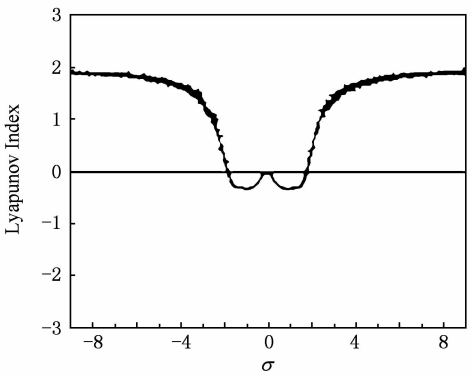
其中,x 为式(1)的变量,$\mu_1,\mu_2,\mu_3,\cdots,\mu_m$ 为式(1)的参量,σz_j 为参数扰动项,z_j 的值由式(2)获得,σ 值作为扰动强度.当 $j=1,2,\cdots,m$ 时,式(1)中上一级(第 $j-1$ 级)子系统的输出 x_{j-1} 将作为下一级(第 j 级)子系统的输入,并通过式(2)的输出值 z_j 和扰动强度 σ 的乘积对参数进行扰动;直到 $j=m$ 时,最后一级子系统的输出 x_m 将作为系统下一次循环的初始值,从而实现 m 级子系统的闭环级联驱动.

1.2 扰动强度 σ 与 Lyapunov 指数分析

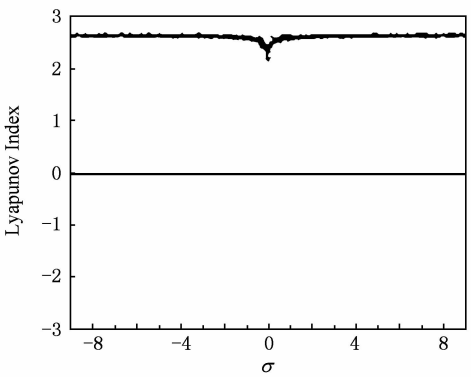
为保证式(1)不会产生溢出,规定 $\mu\in(1.5,1.99]$,$\sigma\in[-9,9]$,并选取适当的 σ 值使式(1)在式(2)的扰动下依然保持混沌状态.以文献[11]表3,4,7 提供的级联系统为例,设式(1)的初值 $x_0=0.8$,式(2)的初值 $z_0=0.6$,参数 $a=2$,采用文献[14-15]提出的方法计算式(1)在式(2)的扰动下,其 Lyapunov 指数随扰动强度 σ 值的变化情况,如图 1 所示.



(a) $m=4,\mu_1=1.501,\mu_2=1.601,\mu_3=1.701,\mu_4=1.801$



(b) $m=5,\mu_1=1.501,\mu_2=1.601,\mu_3=1.701,$
$\mu_4=1.801,\mu_5=1.901$



(c) $m=6,\mu_1=1.9862,\mu_2=1.901,\mu_3=1.801,$
$\mu_4=1.701,\mu_5=1.601,\mu_6=1.501$

Fig. 1 The relationship between σ and Lyapunov index.

图 1 σ-Lyapunov 指数关系图($a=2$)

当 $\sigma=0$ 时,式(1)退化为常参数级联混沌系统,其 Lyapunov 指数约等于 $-0.020\,834\,249\,7$,与文献[11]的结果一致.但随着扰动强度 σ 的增强,当 $|\sigma|>2$ 后,式(1)的 Lyapunov 指数持续大于 0,并最终稳定到 1.902 368 348 附近.经研究发现,只要 σ 的取值范围适当,经扰动后的级联系统正 Lyapunov 指数通常会高于或近似于未经扰动的原始级联系统(即 $\sigma=0$ 时),如图 1 所示;此外,在其他参数不变的情况下,改变扰动系统的参数 a 对 σ-Lyapunov 指数图的基本形态影响不大,但会影响正负 Lyapunov 指数出现时 σ 的取值范围,如图 2~3 所示:

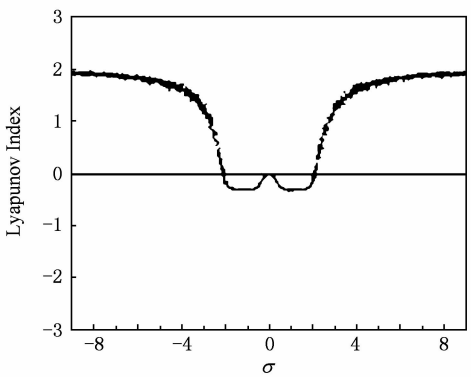


Fig. 2 Effects of positive and negative regions of the Lyapunov index ($a=0.05$).

图 2 σ 对 Lyapunov 指数正负区域的影响($a=0.05$)

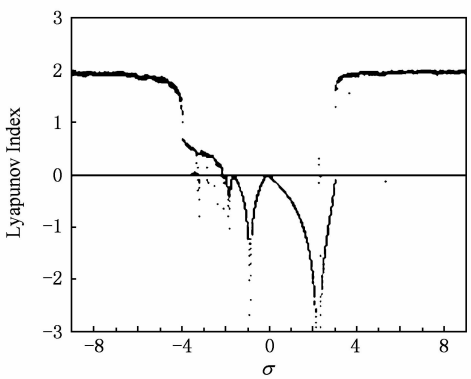


Fig. 3 Effects of positive and negative regions of the Lyapunov index ($a=0.001$).

图 3 σ 对 Lyapunov 指数正负区域的影响($a=0.001$)

可见,参数扰动项 σz 对级联混沌系统的动力学特性具有显著的影响.由于级联混沌存在显著的子系统间串扰特性,即使各独立子系统的 Lyapunov 指数均为正,整个系统最终的 Lyapunov 指数仍有可能出现负值[11].如果此时只对级联系统的原始参数 $\mu_1,\mu_2,\mu_3,\cdots,\mu_m$ 做混沌值 z_j 的线性叠加(即 $\sigma=1$),如图 2,3 所示,系统的 Lyapunov 指数仍会是负值,从而易造成密钥窗口的泄漏.因此,在对级联混沌系统施以变参时,应充分考虑扰动强度 σ 对系统动力特性的影响,选择合适范围内的值,使系统始终保持在混沌状态;同时,上述研究表明可用 σ 值的选取范围远比一般级联系统的原始参数 μ_m 要大得多,因此将消息明文调制到扰动强度 σ 上远比其调制到系统的原始参数上更有优势,相对于文献[5-7]具有更好的变参级联效果.

2　Hash 函数的构造

2.1　压缩函数方程

设 Hash 值长度为 $L_{\mathrm{H}}(b)$,式(1)是一个 6 级级联混沌,其初始参数为 $\mu_1=1.9862,\mu_2=1.901,\mu_3=1.801,\mu_4=1.701,\mu_5=1.601,\mu_6=1.501$;式(2)的初值和参数设为 $z_0=0.6,a=2$;由图 2 可知,扰动强度 σ 在区间[-9,9]内均可满足式(1)的 Lyapunov 指数大于 0.以式(1)为驱动系统,式(2)为扰动系统构造基于变参级联混沌的 Hash 函数,其压缩函数方程为

$$\begin{cases} CV_0 = IV; \\ CV_p = H_L(CV_{p-1},\sigma_p,j_p,\chi_{p,0}), \\ \quad 1\leqslant p\leqslant t; \\ H(M) = CV_t. \end{cases} \tag{3}$$

输入:4 个输入分量,依次为 $CV_{p-1},\sigma_p,\chi_{p,0},j_p$,其代表的物理含义分别为上一轮输出的链接变量、本轮迭代的扰动强度、式(1)的迭代初值以及迭代起始位置;

输出:长度为 $L_{\mathrm{H}}(b)$的中间链接变量 CV_p;最后一轮输出的链接变量 CV_t 即为消息 M 的最终散列值 $H(M)$.

2.2　算法描述

算法 1.变参扰动级联的混沌 Hash 函数算法.

Step1. 消息预处理.

将任意长度的消息明文 M 分割成长度为 $L(b)$ 的消息块 $M_0,M_1,M_2,\cdots,M_t$,分割前按文献[16]的方法对明文 M 进行填充,填充后 $M=*\cdots*100\cdots0Length(M)$,其中$*\cdots*$表示原始明文部分,$Length(M)$表示 64 b 的原始明文长度,在二者之间可填充长度为 r 的"100…0"位串,r 的值为满足 M 被整数分割的最小正整数解.

Step2. 初始化赋值.

以 $L=8$ 为例,将第 1 个消息块 M_0 转换为对应的十进制数值 D_0,则 $\chi_{0,0}=D_0/2^8,\chi_{0,0}\in[0,1)$.以 $\chi_{0,0}$作为式(1)的第 1 轮迭代初值,令 $\sigma_0=9(D_0+1)/2^8$ 作为本轮迭代的扰动强度,式(1)在式(2)的扰动下开始迭代 Y 次,由于明文消息已调制到级联混沌的参数空间,此处 $Y=N+N_k$ 为一常数,其中 $N\geqslant300$ 用于满足变参后的轨道分离,$N_k=L_{\mathrm{H}}/4$ 用于产生链接变量的 4 b 二进制码元.此时得到消息块 M_0 对应的 Y 个迭代值为

$$x_{0,0}\rightarrow x_{0,1},x_{0,2},\cdots,x_{0,Y-N_k+1},x_{0,Y-N_k+2},\cdots,x_{0,Y}.$$

将最后 N_k 个输出值依次代入函数 G 得到对应的 4 b 二进制码元,即:

$$\begin{cases} h_1^{(0)} = G(\chi_{0,Y-N_k+1}); \\ h_2^{(0)} = G(\chi_{0,Y-N_k+2}); \\ \vdots \\ h_{N_k}^{(0)} = G(\chi_{0,Y}). \end{cases} \tag{4}$$

函数 G 的定义如下:

定义 1.设函数 $G:A\rightarrow H,A=\{a_j\mid j=1,2,\cdots,N_k\}$,其中 a_j 是属于[-1,1]区间的实小数;$H=\{h_j\mid j=1,2,\cdots,N_k\}$,其中 h_j 是 $4b$ 码元对应的十进制数值.若 $|a_j|=1$,则 $h_j=15$;否则易知,总能找到一个正整数 $z\in[1,16]$,使得 $(z-1)/16\leqslant|a_j|<z/16$,则 $h_j=z-1$.

将式(4)中各码元合并成一个二进制序列 CV_0,即得到压缩函数的初值 IV:

$$\begin{cases} CV_0 = h_1^{(0)}h_2^{(0)}\cdots h_k^{(0)}h_{k+1}^{(0)}\cdots h_{N_k}^{(0)}; \\ IV = CV. \end{cases}$$

Step3. 生成 Hash 值.

从第 2 个消息分组 M_1 开始,到最后 1 个消息分组 M_t 作同样的处理.以消息分组 M_p 为例,首先将其转换为对应的十进制数值 D_p,然后需要确定本轮压缩函数的 4 个输入分量:

CV_{p-1}:上一轮压缩函数的输出;

σ_p:作为扰动强度,$\sigma_p=9(D_p+1)/2^8,\sigma_p\in(0,9]$;

$\chi_{p,0}$:式(1)上一轮的迭代终值,即 $\chi_{p,0}=\chi_{p-1,Y}$;

j_p:式(1)上一轮迭代终止位置的下一级.

将 $CV_{p-1},\sigma_p,\chi_{p,0},j_p$ 代入式(3)后,以 $\chi_{p,0}$ 为初值、σ_p 为扰动强度从式(1)的第 j_p 个子系统处开始

迭代,在式(2)的扰动下迭代 Y 次,得到消息块 M_p 对应的 Y 个迭代值如下:

$$x_{p,0} \rightarrow x_{p,1},\cdots,x_{p,Y-N_k+1},\cdots,x_{p,Y}.$$

同理,将最后 N_k 个输出值依次代入函数 G 得到对应的 4 b 二进制码元,即:

$$h_k^{(p)} = G(\chi_{p,Y-N_k+k+1}) \begin{cases} h_1^{(p)} = G(\chi_{p,Y-N_k+1}); \\ h_2^{(p)} = G(\chi_{p,Y-N_k+2}); \\ \vdots \\ h_k^{(p)} = G(\chi_{p,Y-N_k+k}); \\ h_{k+1}^{(p)} = G(\chi_{p,Y-N_k+k+1}); \\ \vdots \\ h_{N_k}^{(p)} = G(\chi_{p,Y}). \end{cases}$$

其中,$h_k^{(p)}$ 为构成 CV_p 的第 k 个 4 b 二进制码元(对应 4 b 的二进制域),$k=1,2,\cdots,N_k$,将上述 N_k 个位域进行合并,得到:

$$CV_p = h_1^{(p)} h_2^{(p)} \cdots h_k^{(p)} h_{k+1}^{(p)} \cdots h_{N_k}^{(p)}. \tag{5}$$

将式(5)代入式(6),得到本轮输出的链接变量 CV_p.

$$CV_p = CV_p \text{ XOR } CV_{p-1}. \tag{6}$$

其中 XOR 代表按位异或运算. 至此,本轮的压缩函数计算完成.

循环重复上述过程,直到处理完最后 1 个分组:

$$\begin{cases} CV_1 = H_L(CV_0,\sigma_1,j_1,\chi_{1,0}); \\ CV_2 = H_L(CV_1,\sigma_2,j_2,\chi_{2,0}); \\ \vdots \\ CV_t = H_L(CV_{t-1},\sigma_t,j_t,\chi_{t,0}); \end{cases} \tag{7}$$

则 CV_t 即为消息最终的单向散列值.

3 实验结果与分析

3.1 Hash 函数算例及敏感性分析

以英文版《双城记》开篇第 1 段为文本范例,取 $L_H=128$ 计算其 Hash 值,则 $N_k=32$. 实验通过微小改变消息内容产生新的 Hash 值,并与原 Hash 值进行比较来验证算法的敏感性,结果如表 1 所示. 经过计算发现 Hash 值二进制位上 1 的个数在 128 b 散列值中占到的位数平均值为 52.66%,均超过总位数的一半. 同时相对于每次条件的改变,所得 Hash 值平均变化位数为 65.5 个,可见明文的每位微小变化都可以导致散列值 50%以上的位值发生变化,说明算法具有高度的初值敏感性.

Table 1 The Simulation Results of Hash Function Example

表 1 Hash 函数算例仿真结果

Text Alteration	Hash Values Under Different Conditions
Original Text	C6355784474239A7A9F9136D19F518DB
The first character of the ASCⅡ code plus 1	1F02158EFC14BF9384EB819CD79F59E8
Intermediate character 'e' of the ASCⅡ code minus 1	5E8616EFD798FA235C5CAB9F739F2896
The character '.' in the end is changed into '!'	9BAB7595BAE9B06E565699E7C5159534
Adds a blank space to the end of the file	9A2AF144BE2E36DEF5F66A31168C45C4

3.2 混乱与扩散性质统计分析

使用文献[6]中定义的统计测试方法,通过 6 个统计指标 $Q=(H,\Delta H,B,\Delta B,P,\Delta P)$ 来衡量算法的混乱与扩散性质,计算方法如下:

$$H = \overline{H}_i = \frac{1}{N}\sum_{i=1}^{N} H_i = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\sum_j \varepsilon_j}{L_H}\right),$$

$$\Delta H = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(H_i-\overline{H})^2},$$

$$B = \overline{B}_i = \frac{1}{N}\sum_{i=1}^{N} B_i,$$

$$\Delta B = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i-\overline{B})^2},$$

$$P = \frac{\overline{B}}{L_H}\times 100\%,$$

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

其中,H 和 ΔH 表示平均位分布及其均方差,B 和 ΔB 表示平均变化位数及其均方差,P 和 ΔP 表示平均变化概率及其均方差,N 为统计次数. 分别对 128 b,256 b,512 b 的 Hash 值进行混乱与扩散性质实验,每项实验又分别经过 $N=128,256,512,1024,2048$ 次统计后,得到的统计结果如表 2~4 所示.

表 2~4 的测试结果表明:该算法在产生不同长度 Hash 值时,位分布均匀,H 值均接近理想状态的 50%;平均变化位数 B 分别为 64.058,128.396,256.194,均超过 Hash 值长度的一半,说明消息的每位变化均可导致 Hash 值一半以上的位值发生变化;此外,每位平均变化概率 P 分别为 50.054,50.156,

50.038,非常接近雪崩效应的理想值 50%;同时随着 Hash 值长度的增加,可以看到 ΔH 和 ΔP 的平均值逐渐减少,说明这 2 个统计量的稳定性逐渐增强,ΔB 的值虽略有升高,但升幅比例较低,ΔH,ΔB,ΔP 统计量的上述特征反映了 Hash 值性质的稳定性.

Table 2　Statistics of Hash Value (L_H=128)

表 2　Hash 结果统计表(L_H=128)

N	H/%	ΔH/%	B	ΔB	P/%	ΔP/%
256	49.47	4.33	63.71	5.63	49.78	4.40
512	49.64	4.19	64.46	5.52	50.36	4.31
1024	49.90	4.30	64.01	5.49	50.01	4.29
2048	49.88	4.30	64.08	5.65	50.07	4.42
10000	49.94	4.46	64.03	5.67	50.05	4.43
Mean	49.766	4.316	64.058	5.67	50.054	4.37

Table 3　Statistics of Hash Value (L_H=256)

表 3　Hash 结果统计表(L_H=256)

N	H/%	ΔH/%	B	ΔB	P/%	ΔP/%
256	50.06	3.22	128.91	8.04	50.36	3.14
512	50.04	3.13	128.46	8.41	50.18	3.29
1024	50.06	3.18	128.36	8.29	50.14	3.24
2048	49.98	3.17	128.33	8.10	50.13	3.17
10000	50.03	3.15	127.92	7.97	49.97	3.11
Mean	50.034	3.17	128.396	8.162	50.156	3.19

Table 4 Statistics of Hash Value (L_H=512)

表 4　Hash 结果统计表(L_H=512)

N	H/%	ΔH/%	B	ΔB	P/%	ΔP/%
256	49.99	2.49	256	10.88	50.00	2.12
512	50.00	2.35	256.61	10.93	50.12	2.14
1024	50.02	2.27	256.34	10.99	50.07	2.15
2048	50.01	2.21	256.07	11.31	50.01	2.21
10000	50.00	2.27	255.95	11.28	49.99	2.20
Mean	50.004	2.318	256.194	11.078	50.038	2.164

3.3　抗碰撞性测试

抗碰撞能力是评价 Hash 函数安全性的另一个重要指标[3].采用文献[3]的测试方法,通过式(8)~(10)计算 N 次实验中相邻 Hash 值的单位字符平均差异度:

$$d=\sum_{i=1}^{n}\mid t(e_i)-t(e'_i)\mid, \tag{8}$$

其中,e 和 e' 分别对应变化前后 Hash 值第 i 个位置的字节码,函数 $t(e)$ 将该字节码 e 转换为对应的十进制数值.

$$D=\frac{2\sum_{i=1}^{N}d_i}{(N\times N_k)}, \tag{9}$$

$$\Delta D=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(D_i-\overline{D})^2}. \tag{10}$$

式(9)(10)中,N 为统计次数,d_i 为第 i 次实验结果与第 $i-1$ 次实验结果比较的绝对差异度,D 为相邻 2 个 Hash 值的单位字符平均差异度.实验分别对 128 b,256 b,512 b 的 Hash 值进行 N=10 000 次碰撞测试,得到的统计结果如表 5 所示:

Table 5　The Test of Anti Collision Performance(N=10 000)

表 5　算法抗碰撞性能测试结果(N=10 000)

L_H/b	d			D	ΔD
	Maximum	Minimum	Mean		
128	2 440	606	1 365.01	85.31	15.00
256	4 235	1 543	2 729.61	85.30	10.60
512	7 396	3 834	5 459.84	85.31	7.48

表 5 的结果表明:在该算法下不同长度 Hash 值的单位字符平均差异度 D 分别为 85.31,85.30,85.31,均接近理想状态[10]下的理论值 85.333,说明算法对不同长度 Hash 值都具有较强的抗碰撞性.同时,其均方差值较小且彼此间相差不大,说明算法的抗碰撞性比较稳定[3].

3.4　与其他典型 Hash 算法的比较

首先,选取近年来一些典型的混沌 Hash 算法作对比,表 6 描述了该算法与其他文献算法在同一统计量方面的性能对比.

Table 6　Confusion and Diffusion of the Contrast of the Hash Algorithms (N=10 000,L_H=128)

表 6　Hash 算法的混乱与扩散统计性能对比(N=10 000,L_H=128)

Reference	B	ΔB	P/%	ΔP/%	D
Kanso[1]	64.150	5.740	50.120	4.480	93.375
Akhavan[2]	63.919	5.647	49.937	4.412	85.373
Kanso[3]	63.940	5.640	49.950	4.410	85.250
Nouri[6]	63.979	2.380	49.980	1.860	98.688
Liao Dong[10]	63.890	5.628	49.850	4.470	84.155
This Paper	64.058	5.670	50.054	4.370	85.310

从表 6 可以看出,在 10 000 次测试中,除文献[1]外,本文算法的 B 值最大,即对于每位明文变化,

本文算法产生的 Hash 值具有更好的敏感性;且每位平均变化概率为 50.054%,非常接近理想状态下的 50%,与文献[2-3,6]的结果比较接近,优于文献[1]和文献[10]的结果;在抗碰撞性方面,本文算法的平均差异度均值 $D=85.31$,最接近理论值 85.333.通过比较发现,本文算法体现出了较好的混乱与扩散性能,且具有很强的抗碰撞性.

其次,选取新一代 Hash 函数标准 SHA3-512 算法作为对比研究对象,表 7 描述了本文算法与 SHA3 算法的性能对比数据.

Table 7 Comparison with the Performance of SHA3 Algorithm ($N=10000, L_H=512$)

表 7 与 SHA3 算法的性能对比 ($N=10000, L_H=512$)

Reference	B	ΔB	P/%	ΔP /%	D	Run-bits (Left/Right)
SHA-3[17]	255.86	11.38	49.97	2.22	85.29	1.0097
This Paper	255.95	11.28	49.99	2.20	85.31	0.9745

从表 7 可以看出,本文算法在每位明文变化、平均变化概率和抗碰撞性方面比 SHA-3 算法更接近理想值.在运行位测试[17]方面,SHA-3 算法的位分布更加平衡,略向左倾斜;而本文算法略向右倾斜,二者相差不大.在算法结构上,SHA-3 算法采用了创新的"海绵引擎"[17],计算方法上仅使用了 AND, XOR,NOT 等位操作模式,因此更加快速,适合于各种硬件实现;而本文算法引入了浮点运算,因此在移动终端上的计算性能略逊于 SHA3 算法,但本文算法充分利用了初始敏感性、内在随机性等混沌系统的固有性质,算法朴素简单,抗碰撞性更强,此外由于引入了混沌参数扰动的方法,增加了系统的复杂性,减少了由计算机有限精度而产生短周期行为的风险[18].

4 结束语

本文将级联混沌技术与参数扰动方法相结合,提出了一种基于变参级联混沌的 Hash 函数设计方案.通过对其 Lyapunov 指数的分析发现,经过参数扰动后的级联混沌系统,只要扰动强度 σ 的取值范围适当,总可以使系统保持混沌状态,从而避免了级联混沌可能带来的密钥窗口泄漏问题[11];同时相对于原有级联混沌的复杂参数组合,扰动强度 σ 不仅具有更大的取值范围,也更加易于控制.因此,将消息明文调制到扰动强度 σ 上远比其调制到系统的原始参数上更有优势.此外,与现有典型混沌 Hash 函数[1-10]相比,由于构成 Hash 函数的级联系统不但存在着子系统间的逐级串扰,且其参数的变化也符合混沌特性,从而使 Hash 函数的内部结构更加难以预测,压缩函数各轮计算之间位模式的安全性得以显著增强,为抵御现有的暴力攻击和统计分析攻击提供了保障.实验与分析结果表明,算法具有高度的单向性和初值敏感性、良好的混乱与扩散性能,抗碰撞能力强;同时算法实现简单灵活、扩展性强,具有良好的推广前景.

如何将参数扰动方法与高维级联混沌技术相结合是下一步值得深入研究的课题,此外该方法在实际应用中还有若干问题有待解决,如扰动系统本身的安全性和同步性、扰动后系统相空间的概率密度分布规律等都有待进一步研究.

参 考 文 献

[1] Kanso A, Yahyaoui H, Almulla M. Keyed hash function based on a chaotic map [J]. Information Sciences, 2012, 186 (1): 249-264

[2] Akhavan A, Samsudin A, Akhshani A. A novel parallel hash function based on 3D chaotic map [J]. EURASIP Journal on Advances in Signal Processing, 2013, 2013(1): 1-12

[3] Kanso A, Ghebleh M. A fast and efficient chaos-based keyed hash function [J]. Communications in Nonlinear Science and Numerical Simulation, 2013, 18(1): 109-123

[4] Wang Guangyi, Yuan Fang. Cascade chaos and its dynamic characteristics [J]. Acta Physica Sinica, 2013, 62(2): 111-120 (in Chinese)
(王光义, 袁方. 级联混沌及其动力学特性研究[J]. 物理学报, 2013, 62(2): 111-120)

[5] He Tingting, Luo Xiaoshu, Liao Zhixian, et al. A new chaos mapping hash function structural method and its application [J]. Acta Physica Sinica, 2012, 61(11): 164-170 (in Chinese)
(何婷婷, 罗晓曙, 廖志贤, 等. 一种新的混沌映射散列函数构造方法及应用[J]. 物理学报, 2012, 61(11): 164-170)

[6] Nouri M, Safarinia M, Pourmahdi P, et al. The parallel one-way Hash function based on Chebyshev-Halley methods with variable parameter [J]. International Journal of Computers Communications & Control, 2014, 9(1): 24-36

[7] Wang Liyan, Li Zhanmei, Liu Zixin. One-way Hash function construction based on composite chaotic map [J]. Journal of Dalian University of Technology, 2013, 53(5): 742-748 (in Chinese)
(王丽燕, 李占梅, 刘自新. 一种基于复合混沌映射的单向 Hash 函数构造[J]. 大连理工大学学报, 2013, 53(5): 742-748)

[8] Liu Jiandong. One-way Hash function based on integer coupled Tent maps and its performance analysis [J]. Journal of Computer Research and Development, 2008, 45(3): 563-569 (in Chinese)
(刘建东. 基于整数耦合帐篷映射的单向 Hash 函数及其性能分析[J]. 计算机研究与发展, 2008, 45(3): 563-569)

[9] Xu Jie, Yang Dijie, Long Keping. Design and analysis of a cryptographic Hash function based on Time-delay chaotic system [J]. Journal of University of Electronic Science and Technology of China, 2011, 40(3): 451-455 (in Chinese)
(徐杰, 杨娣洁, 隆克平. 基于时滞混沌系统的带密钥 Hash 函数的设计与分析[J]. 电子科技大学学报, 2011, 40(3): 451-455)

[10] Liao Dong, Wang Xiaomin, Zhang Jiashu, et al. The chaotic Hash function based on spatial expansion construction with controllable parameters [J]. Acta Physica Sinica, 2012, 61(23): 103-112 (in Chinese)
(廖东, 王小敏, 张家树, 等. 基于空间伸缩结构的参数可控的混沌 Hash 函数[J]. 物理学报, 2012, 61(23): 103-112)

[11] Jin Jianguo, Di Zhigang, Wei Mingjun. Potential risk of variable parameter cascade chaos system [J]. Acta Physica Sinica, 2014, 63(12): 88-100 (in Chinese)
(金建国, 邸志刚, 魏明军. 变参级联混沌系统中的潜在风险[J]. 物理学报, 2014, 63(12): 88-100)

[12] Li Zhenbo, Tang Jiashi. Chaotic synchronization with parameter perturbation andits secure communication scheme [J]. Control Theory & Applications, 2014, 31(5): 592-600 (in Chinese)
(李震波, 唐驾时. 参数扰动下的混沌同步控制及其保密通信方案[J]. 控制理论与应用, 2014, 31(5): 592-600)

[13] He Chen, He Di, Jiang Lingge, et al. A chaotic map with infinite collapses [C] //Proc of IEEE TENCON 2000. Piscataway, NJ: IEEE, 2000: 95-99

[14] Wolf A, Swift J B, Swinney H L, et al. Determining Lyapunov exponents from a time series [J]. Physica D: Nonlinear Phenomena, 1985, 16(3): 285-317

[15] Briggs K. An improved method for estimating Liapunov exponents of chaotic time series [J]. Physics Letters A, 1990, 151(1/2): 27-32

[16] Wang Gaoli, Shen Yanzhao. Preimage and pseudo-collision attacks on 29-step SM3 hash function with padding [J]. Journal on Communications, 2014, 35(2): 40-45 (in Chinese)
(王高丽, 申延召. 带消息填充的 29 步 SM3 算法原根和伪碰撞攻击[J]. 通信学报, 2014, 35(2): 40-45)

[17] National Institute of Standards and Technology. Keccak and the SHA-3 standardization [EB/OL]. (2013-02-06) [2014-10-21]. http://csrc. nist. gov/groups/ST/hash/sha-3/sha-3_standardization. html

[18] Liu Jiandong, Yang Kai, Yu Youming. Improved coupled Tent map lattices model and its characteristics analysis [J]. Journal of Computer Research and Development, 2011, 48(9): 1667-1675 (in Chinese)
(刘建东, 杨凯, 余有明. 改进型耦合帐篷映像格子模型及其性能分析[J]. 计算机研究与发展, 2011, 48(9): 1667-1675)

Wu Tao, born in 1979. Master, associate professor. His main research interests include chaotic encryption, information security.

Jin Jianguo, born in 1956. Professor. His main research interests include chaotic encryption, information security.

Wei Mingjun, born in 1969. Master, associate professor. His main research interests include chaotic encryption, information security.