第 55 卷 第 9 期 2006 年 9 月
1000-3290/2006/55(09)/4442-08

物 理 学 报
ACTA PHYSICA SINICA

Vol.55,No.9,September,2006
©2006 Chin. Phys. Soc.

基于混沌动态 S-Box 的 Hash 函数*

郭现峰[1)2)] 张家树[1)]

1)(西南交通大学信号与信息处理四川省重点实验室,成都 610031)

2)(西南民族大学计算机科学与技术学院,成都 610041)

(2005 年 11 月 4 日收到;2005 年 12 月 26 日收到修改稿)

结合混沌系统与传统单向 Hash 函数设计方法的优点,提出了一种基于混沌动态 S-Box 的带秘密密钥的单向 Hash 函数构造方法.该方法用混沌 S-Box 替换和函数查找表来生成具有混沌特性的 Hash 摘要.与现有混沌 Hash 算法相比,新方案没有将原始数据直接参与混沌迭代,而是采用混沌动态 S-Box 替换来提高系统的实时性能.研究结果表明:该方法不仅有很好单向性、初值和密钥敏感性,且有较大的密钥空间,易于实现.

关键词:Hash 函数,混沌,S-Box,函数查找表

PACC:0545

1. 引 言

单向 Hash 函数在现代信息安全领域得到广泛应用,根据使用过程中是否使用密钥来划分,单向 Hash 函数可分为带密钥的 Hash 函数(K-HF)和不带密钥的 Hash 函数[1].K-HF 多用于认证、密钥共享和软件保护等方面[2],而不带密钥的 Hash 函数一般用于完整性验证、数字签名等.传统单向 Hash 函数一般是基于复杂度假设构造的,需要进行繁杂的异或、模加等逻辑运算或多次分组迭代,如 MD 系列和 SHA[3] 等,即使被处理的消息很短时,运算量都很大.

近年来,利用混沌系统的确定性和对初值的敏感性来构造密码算法[4—10]已成为国内外的研究热点,其中基于混沌的 Hash 函数确实能很好地解决传统 Hash 函数运算量问题,但由于混沌系统数字化实现中的有限字长效应,使得这些基于混沌构造的单向 Hash 函数存在以下不足:1)混沌系统的鲁棒性降低了保密性[11];2)一些混沌系统自身存在问题,利用混沌预测技术可以破译、提取出信息信号[12—15],得不到预定的保密效果;3)计算机有效字长精度效应,使得混沌映射退化为周期序列[11].为了克服这些不足,文献[5,7,10]分别用广义混沌映射切换、时空混沌和超混沌的方法来构造单向 Hash 函数,它们均通过原始数据的混沌迭代来完成 Hash 运算,从而增加了运算复杂度,且安全性也并没有明显提高.因此,如何利用离散混沌系统对初值和参数的极端敏感性、混沌序列的白噪声统计性和遍历性等优良特征并结合传统密码学中的经典理论来构造兼顾安全性和复杂度的混沌单向 Hash 算法就成为当前和今后研究的重点.

针对现有混沌单向 Hash 算法的不足,本文提出了一个基于混沌动态 S-Box 的 K-HF 构造算法.该算法首先选择一个混沌映射 $G_{(\alpha)}(x_i)$,将混沌参数 α、初值 x_0 和初始迭代次数 m 作为秘密密钥 $K=(\alpha,x_0,m)$,经混沌迭代、线性映射构造一个基于密钥的混沌动态 S-Box;然后将原始数据的线性映射进行混沌动态 S-Box 替换;最后通过函数查找表动态选取转换函数 $F(*)$ 生成 $4n$ Byte 的 Hash 值.在不增加任何附加运算的情况下,动态调整 n 的大小可以生成不同长度的 Hash 摘要,从而增加了系统的灵活性.另外,该算法没有将原始数据直接参与混沌迭代,而是借助于混沌动态 S-Box 替换来实现系统的混沌性,提高了执行效率.理论分析和仿真结果证明该算法具有很好的单向性、置乱性、初值与密钥敏感性,且兼顾了安全性与复杂度,是一个较易于实现的 K-HF 算法.

* 国家自然科学基金(批准号:60572027)、教育部新世纪优秀人才计划项目(批准号:NCET-05-0794)、四川省青年基金(批准号:03ZQ026-033),四川省应用基础研究项目(批准号:2006J13-110)和国防科技重点实验室基金(批准号:51435080104QT2201)资助的课题.

2. K-HF 与混沌系统

Berson 等人在 1993 年提出 K-HF[16]，后由 Bakhtiari 等给出了较完整的定义[2].

定义 1　K-HF 函数 $H(*)$ 是一个 Hash 函数族 $\{h_k : k \in K\}$，对任意 $h \in K$，$h_k(M) : \Sigma \to V_m$ 将信息集合 Σ 中任意长度的信息集合 M 映射为长度为 m 的信息摘要 $h_k(M)$，并满足：

1) $h_k(M)$ 是密钥单向函数，即

(a) 给定 $k \in K$ 和 $M \in \Sigma$，计算 $h_k(M)$ 是容易的；

(b) 在没有 k 的情况下，给定 $h_k(M)$，求 M 是困难的；

(c) 在没有 k 的情况下，给定 M，求 $h_k(M)$ 是困难的.

2) $h_k(M)$ 是密钥碰撞自由函数，即若没有密钥 k，求 $M, M' \in \Sigma$，且满足 $M \neq M'$，$h_k(M) = h_k(M')$ 是困难的.

3) 给定一组 $M \in \Sigma$ 及对应的 $h_k(M)$，求其他信息 M' 的 $h_k(M')$ 或 $h_k(M')$ 的信息 M' 是困难的.

根据上述定义，Bakhtiari 等人指出为了抵御穷举搜索攻击、生日攻击和统计分析等，在实际应用中最好选择秘密密钥和信息摘要的长度都不小于 128 bit、摘要在摘要空间中均匀分布的单向 Hash 函数.

混沌[17] 是指确定性非线性系统普遍具有的一种复杂动力学行为，它对系统初始状态或系统参数异常敏感. 混沌序列具有遍历性和类噪声统计特性，因此，混沌映射本身是不可逆和难以长期预测的.

与不带密钥的 Hash 函数相比，K-HF 会随密钥的改变而生成不同的摘要，这样就可以在完整性验证的同时实现源认证，但也要求 K-HF 有充分的密钥敏感性和足够大的密钥空间来抵御统计分析和暴力搜索攻击等. 而混沌系统对初值或参数极端敏感，初值或参数的微小扰动都可以产生差别很大的混沌序列，并且其初值或参数取值空间在理想状态下可以无穷大，因此选用具有良好特性的混沌系统来构造混沌 K-HF 算法，并将其初值或参数作为秘密密钥可以取得很好的效果. 对于混沌 Hash 中将原始数据的线性变换直接参与混沌迭代引起的计算量问题，本文通过混沌动态 S-Box 替换来解决.

文献[18]引入了一个混沌特性较好的扩展 Tent 映射，定义如下：

$$G_{(\alpha)} : x_i = \begin{cases} F_\alpha(x_{i-1}), & 0 < x_{i-1} < 1, \\ \beta, & x_{i-1} = 0 \text{ 或 } 1, \end{cases} \tag{1}$$

其中 $0 < \alpha, \beta < 1$，$\beta \neq \alpha$，$F_\alpha(\beta) \neq \alpha$，且 $F_\alpha(x_i)$ 为满足下式的 Tent 映射：

$$F_\alpha : x_i = \begin{cases} \dfrac{x_{i-1}}{\alpha}, & 0 \leqslant x_{i-1} \leqslant \alpha, \\ \dfrac{1 - x_{i-1}}{1 - \alpha}, & \alpha < x_{i-1} \leqslant 1. \end{cases} \tag{2}$$

在(1)式中 x_{i-1} 取 0 或 1 的情况较少，故 β 一般不作为混沌参数使用，随机选取混沌控制参数 α 和初值 x_0 对混沌映射(1)进行循环迭代 5000 次的混沌序列分布如图 1 所示. 可见，混沌映射(1)对初值和参数非常敏感，且在(0,1)中服从均匀分布[18]，有很好的混沌特性，具备了构造 K-HF 的条件.

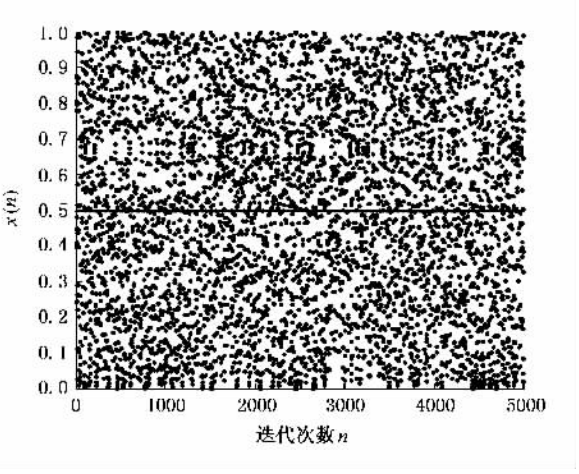

图 1　扩展 Tent 映射混沌序列分布图

3. 基于混沌动态 S-Box 的 K-HF 构造

3.1. 混沌动态 S-Box 的设计

传统密码学中使用的固定 S-Box 可以很好地实现非线性转换，有较高的执行效率，但 S-Box 内容是公开且永久不变的，易被敌手所利用. 为此，本文选用初值和参数极端敏感且混沌序列在(0,1)中服从均匀分布的混沌映射(1)来构造混沌动态 S-Box，构造方法如下：

1) 将混沌参数 α、初值 x_0 和初始迭代次数 m 作为秘密密钥 $K = (\alpha, x_0, m)$，经混沌映射(1)的混沌迭代，得混沌序列

$$S = \{a_i \mid i = 0,1,2,\cdots\}, \tag{3}$$

其中 $a_i = G_{(\alpha)}^{m+i}(x_0)$，即 $G_{(\alpha)}(x_0)$ 循环迭代 $m + i$ 次.

2) 将序列(3)中的 a_i 二值化为

$$b_i = \begin{cases} 0, & 0 \leqslant a_i \leqslant 0.5, \\ 1, & 0.5 < a_i \leqslant 1, \end{cases}$$

则有 0-1 序列

$$T = \{b_i \mid i = 0,1,2,\cdots\}, \tag{4}$$

由 0-1 序列(4)得噪声向量

$$U_k = \{b_{8k}, b_{8k+1}, \cdots, b_{8k+7}\}, \quad k \in Z \text{ 且 } k \geqslant 0. \tag{5}$$

3) 对整数 $i,j \in [0,15]$,利用噪声向量(5)定义二维向量 $S[i][j] = U_{16i+j}$,称 16×16 阶方阵 $S[i][j]$ 为混沌动态 S-Box.

从上述构造方法可见,这种混沌动态 S-Box 随密钥 K 的改变而改变,其构成元素的安全性是基于混沌系统的初值和参数极端敏感性.由混沌系统对初值和参数的极端敏感性以及混沌映射(1)在(0,1)中服从均匀分布可知,构造的混沌动态 S-Box 是对密钥 K 极端敏感的函数且有很好的统计特性,因此,没有密钥 K 就难以有效地重构出混沌动态 S-Box.另外,混沌动态 S-Box 内容的隐蔽性使敌手很难由替换前的数据来求替换后的数据,这样就克服了传统的固定 S-Box 易被敌手利用的弊端,从而增强了系统的安全性.

3.2. 函数查找表

虽然混沌动态 S-Box 可以随密钥 K 的改变而改变,并利用混沌特性增强了混乱度,但单一的混沌动态 S-Box 仍无法有效抵御利用大量替换前和替换后的数据所进行的统计分析攻击,为此,本文引入了具有传统加密方法优点的函数查找表来予以克服,并定义相应的转换函数如下:

$$\begin{aligned} A &= f_1(A,B,C,D) \\ &= (A + \bar{B} \oplus C \oplus D) \lll 3, \end{aligned} \tag{6}$$

$$\begin{aligned} B &= f_2(A,B,C,D) \\ &= (B + A \oplus \bar{C} \oplus D) \lll 1, \end{aligned} \tag{7}$$

$$\begin{aligned} C &= f_3(A,B,C,D) \\ &= (C + A \oplus B \oplus \bar{D}) \lll 2, \end{aligned} \tag{8}$$

$$\begin{aligned} D &= f_4(A,B,C,D) \\ &= (D + A \oplus B \oplus \bar{C}) \lll 3, \end{aligned} \tag{9}$$

其中 A,B,C 和 D 分别为 $8n$ bit 的寄存器," + "是模 2^{8n} 加,$\bar{X}$ 是"按位取反",$X \oplus Y$ 是按位"异或",$X \lll s$ 表示将 X 循环左移 s 位,各式最后函数结果分别存于相应的寄存器.

定义 2 若 f_a, f_b 是满足(6)—(9)式的转换函数,$f_a \circ f_b = f_a(f_b(\cdot))$ 为级联函数运算,则定义表 1 为函数查找表.

函数查找表提供了 16 种可选级联函数,不同的 TT_1 就有不同的变换函数 $F(*)$ 与之对应.与单一的变换函数相比,函数查找表以很小的计算代价增强了变换混乱度,加大了攻击复杂性,并具有易于软硬件实现等特点.因此,将函数查找表与混沌动态 S-Box 级联使用,可以加大统计分析攻击的难度,能进一步提高系统的安全性能.

表 1 函数查找表

TT_1	$F(*)$	TT_1	$F(*)$	TT_1	$F(*)$	TT_1	$F(*)$
0000	$f_1 \circ f_3 \circ f_2 \circ f_4$	0100	$f_2 \circ f_3 \circ f_1 \circ f_4$	1000	$f_3 \circ f_1 \circ f_2 \circ f_4$	1100	$f_4 \circ f_3 \circ f_2 \circ f_1$
0001	$f_1 \circ f_4 \circ f_2 \circ f_3$	0101	$f_2 \circ f_4 \circ f_1 \circ f_3$	1001	$f_3 \circ f_2 \circ f_4 \circ f_1$	1101	$f_4 \circ f_2 \circ f_1 \circ f_3$
0010	$f_1 \circ f_2 \circ f_4 \circ f_3$	0110	$f_2 \circ f_1 \circ f_3 \circ f_4$	1010	$f_3 \circ f_4 \circ f_1 \circ f_2$	1110	$f_4 \circ f_1 \circ f_3 \circ f_2$
0011	$f_1 \circ f_3 \circ f_4 \circ f_2$	0111	$f_2 \circ f_3 \circ f_4 \circ f_1$	1011	$f_3 \circ f_1 \circ f_4 \circ f_2$	1111	$f_4 \circ f_2 \circ f_3 \circ f_1$

3.3. 基于混沌动态 S-Box 的 K-HF 构造

图 2 和图 3 为本文提出的基于混沌动态 S-Box 的 K-HF 设计流程图.在 K-HF 中,首先将任意长度的原始数据 M 经填补、划分成各为 $4n$ Byte 的子信息段 $M_1, M_2, \cdots, M_i, \cdots$,之后经混沌迭代、函数变换、S-Box 替换和函数查找表生成 $4n$ Byte 的 Hash 值.现将基于混沌动态 S-Box 的 K-HF 构造算法描述如下:

1)令初始密钥 $K = (\alpha, x_0, m)$,其中 α, x_0 和 m 分别为混沌映射(1)的参数、初值和初始迭代次数.以混沌 0-1 序列(4)初始化各寄存器为

$$A = (b_0, b_1, \cdots, b_{8n-1}),$$

$$B = (b_{8n}, b_{8n+1}, \cdots, b_{2*8n-1}),$$

$$C = (b_{2*8n}, b_{2*8n+1}, \cdots, b_{3*8n-1}),$$

$$D = (b_{3*8n}, b_{3*8n+1}, \cdots, b_{4*8n-1}).$$

2)将长度为 $4n$ Byte 的子信息段 M_i 均分为 4 部分 M_{i1}, M_{i2}, M_{i3} 和 M_{i4},各为 n Byte.

3)根据函数 $f'(A,B,C,D,E) = ((A + E) \oplus \bar{B}$

$\oplus C + D) \lll 1$ 计算

$$A = f'(A, B, C, D, M_{i1}),$$
$$B = f'(B, A, C, D, M_{i2}),$$
$$C = f'(C, B, A, D, M_{i3}),$$
$$D = f'(D, B, C, A, M_{i4}).$$

4)由变换后 A, B, C 和 D 的最后 bit 位和表 1 确定 A, B, C 和 D 在混沌动态 S-Box 中的查找替换次序，即将 A, B, C 和 D 与 1,2,3 和 4 分别对应，依函数 $F(*)$ 的分函数下标顺序为混沌替换次序，如当 $F(*)$ 为 $f_3 \circ f_1 \circ f_4 \circ f_2$ 时，分函数的下标顺序为 3142，则替换次序为 $CADB$，将 $CADB$ 从左到右的各字节高 4 位记为 high_k，低 4 位记为 low_k，定义一个缓冲区 R，令 $R_k = S[\text{high}_k][\text{low}_k]$，其中 $0 \leqslant k < 4n$.

5)对 $0 \leqslant k < 4n$，将 R_k 依次赋值给 A, B, C 和 D，其中 $A = (R_0, R_1, \cdots, R_{n-1})$，$B = (R_n, R_{n+1}, \cdots, R_{2n-1})$，$C = (R_{2n}, R_{2n+1}, \cdots, R_{3n-1})$，$D = (R_{3n}, R_{3n+1}, \cdots, R_{4n-1})$.

6)将 A, B, C 和 D 的最后 bit 位取出构成 TT_1，查表 1 得相应函数 $F(*)$，并计算 $F(A, B, C, D)$.

7)判断原始数据是否完毕，是则输出 $4n$ Byte 的 hash 值 $ABCD$ 否则 $i = i + 1$，转 2).

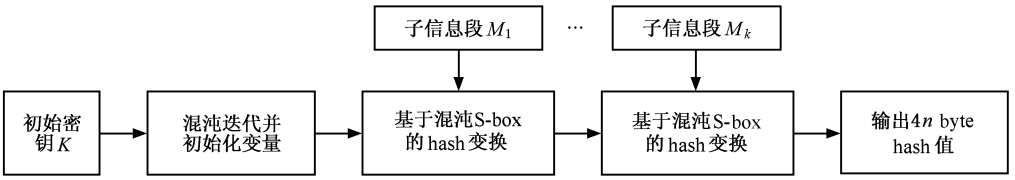

图 2　K-HF 总体设计流程图

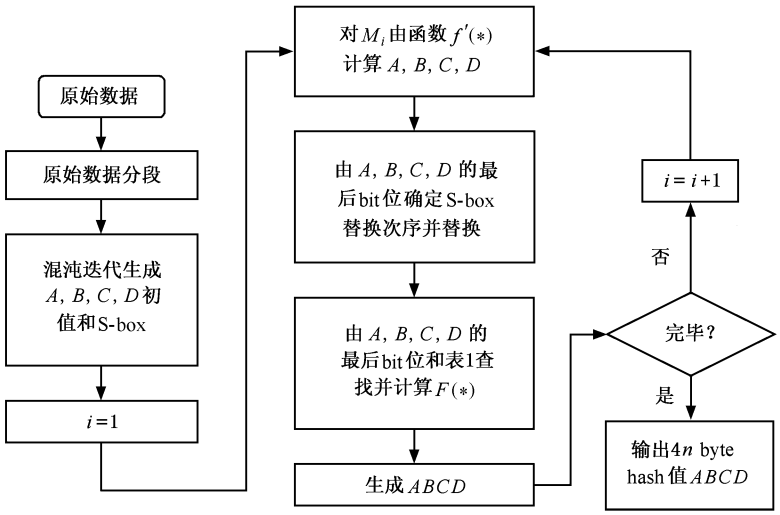

图 3　K-HF 详细设计流程图

4. 基于混沌动态 S-Box 的 K-HF 性能分析

4.1. K-HF 的密钥空间分析

一个安全的 K-HF 应该有足够大的密钥空间来抵御暴力攻击，同时暴力攻击的复杂度依赖于 Hash 摘要和密钥的长度. 图 2 中的初始密钥 $K = (\alpha, x_0, m)$ 可以在实数范围内取值，理论上的密钥空间为无穷大，能抵御暴力搜索攻击. 然而，由于计算机有效字长精度效应的影响，初始密钥只能以有限计算精度的实数来表示，致使密钥 $K = (\alpha, x_0, m)$ 只能在有限的范围内取值，初始密钥空间不再是无穷大. 因此，有必要考察有效计算精度条件下的 K-HF 抵御暴力搜索攻击的能力.

为了考察有效计算精度条件下的 K-HF 抵御暴力搜索攻击的能力，验证在双精度 64 位浮点型范围

内初始密钥 $K=(\alpha,x_0,m)$ 取值的有效性，仿真中 α 和 x_0 在取值空间内随机选取 $\alpha'=0.49245454$，$x'_0=0.649938$，并针对双精度的有效计算精度 10^{-15}，设 $\alpha'_1=\alpha'+10^{-15}$，$\alpha'_2=\alpha'-10^{-15}$，$x'_{01}=x'_0+10^{-15}$，$x'_{02}=x'_0-10^{-15}$. 以 (α',x'_0)，(α'_1,x'_0)，(α'_2,x'_0)，(α',x'_{01}) 和 (α',x'_{02}) 分别作混沌映射(1)的参数和初值，将混沌迭代得到的0-1序列(4)分别记为 B'，B'_1，B'_2，B'_3 和 B'_4. 图3是 B' 与其他0-1序列差别比特数目 n_1，n_2，n_3，n_4 的线性表示. 另外，对 $0<|\alpha-0.5|<0.01$ 和 $x_0\in(0,1)$ 范围内的其他数据做同样实验，均得到类似图4所示效果.

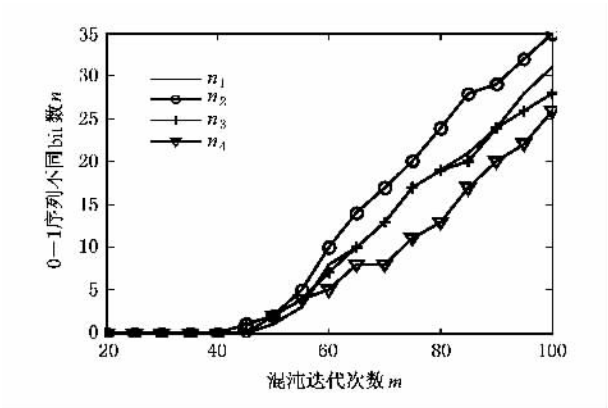

图4 0-1序列差别数比较

图4的仿真结果表明，α 或 x_0 在双精度允许范围内的任何微小扰动，经混沌迭代50次以后均可得到差别很大的0-1混沌序列，即当混沌迭代次数 $m>50$ 时能保证 α 和 x_0 在双精度范围内取值的有效性. 另外，当 $0<|\alpha-0.5|<0.01$ 时，混沌映射(1)在区间(0,1)中满足均匀分布[19]. 因此，混沌参数 α 在双精度取值范围内的取值形式为 $0.49\alpha_1\alpha_2\cdots\alpha_{13}$ 和 $0.51-0.00\alpha'_1\alpha'_2\cdots\alpha'_{13}$，去掉0.49和0.51两个点得 α 取值空间为 $2\times(10^{13}-1)$；同理初值 $x_0\in(0,1)$ 的取值空间为 $10^{15}-1$；由于64位无符号整数最大值为 $2^{64}-1$，故初始混沌迭代次数 m 的取值空间为 $2^{64}-52$. 因此，在双精度范围内密钥 K 的取值空间为 $[2\times(10^{13}-1)]\times(10^{15}-1)\times(2^{64}-52)$，约 2^{158}. 而现有暴力搜索攻击破解K-HF的密钥取值空间大约为 2^{64}，可见，本文所提方案是比较安全的.

4.2. 基于混沌动态S-Box的K-HF安全性分析

本文提出的K-HF利用初始密钥 K 和扩展Tent混沌映射生成寄存器初值和混沌动态S-Box，并在相应函数运算后经S-Box替换、函数查找表生成 $4n$ Byte Hash值. 其中，混沌动态S-Box由0-1混沌序列构成，随密钥 $K=(\alpha,x_0,m)$ 的改变而改变. 混沌系统的确定性和初值敏感性使得混沌动态S-Box对秘密密钥 K 极端敏感，且混沌映射(1)不仅具有白噪声统计特性，还在区间(0,1)满足均匀分布. 因此，对没有掌握密钥 K 的敌手，要想通过分析一组 $M\in\Sigma$ 及对应的 $h_k(M)$ 求其他信息 M' 的 $h_k(M')$ 或 $h_k(M')$ 的信息 M' 的方式来破解本体制，就必须重构函数 $F(*)$ 和S-Box. 由混沌S-Box的构成元素不具有唯一性可知：混沌动态S-Box替换是不可逆的，且 $F(*)$ 是由混沌序列与原始信息的函数结果共同确定. 因此，所构造的K-HF的安全性是与混沌系统和迭代函数 $F(*)$ 的安全性的乘积成正比的.

5. 基于混沌动态S-Box的K-HF仿真结果及分析

一个好的K-HF，不仅要有很好的单向不可逆性，还应该具备如下特性：1)密钥敏感性，即密钥的任何微小变化将产生截然不同的Hash摘要；2)初值敏感性，产生的摘要值的每一比特都应该是原始数据每一比特非常复杂、敏感的函数；3)Hash摘要值应均匀分布于摘要空间，以抵御统计分析攻击[2]. 针对上述问题及其他安全性需求，随机选取 $\beta=0.59864$，$n=4$（即最后生成128 bit的Hash结果），做了如下仿真实验.

5.1. K-HF文本仿真分析

5.1.1. K-HF的密钥敏感性分析

仿真1 随机选取 $\alpha_1=0.494534$，$x_1=0.754535$，$m_1=100$ 和文本 f“At the heart of MATLAB is a new language that you must learn before you can fully exploit its power. This isn't as hard as it might sound; you can learn the basics of MATLAB very quickly. You will be rewarded with high-productivity, high-creativity computing power that will change the way you work.”.以 $K_1=(\alpha_1,x_1,m_1)$，$K_2=(\alpha_1-10^{-15},x_1,m_1)$，$K_3=(\alpha_1+10^{-15},x_1,m_1)$，$K_4=(\alpha_1,x_1-10^{-15},m_1)$，$K_5=(\alpha_1,x_1+10^{-15},m_1)$，$K_6=(\alpha_1,x_1,m_1-1)$，$K_7=(\alpha_1,x_1,m_1+1)$ 作初始密钥，分别求得的基于混沌动态S-Box的128位Hash摘要值的16进制形式如下，将后6次Hash结果与第一次Hash结果比较，改

变的比特数目分别为 62，68，59，67，58，68，平均改变 63.667，与理想改变值 64 仅差 0.333.

Hash K_1：815*BA*4*E*2*BE*95*E*26*B*717*A*2420*A*6*AB*175*C*，

Hash K_2：50598954*AE*4*F*70*CFAE*4*CAE*40739*DFE*2*B*，

Hash K_3：595*FE*3*BD*18*A*8*A*82*CED*9911*D*5*EF*6*CAAB*4，

Hash K_4：*C*103*B*2*C*481*ADED*3*DAF*11*E*3312*CDF*8*C*5*F*，

Hash K_5：3*C*74*CA*91*B*51*B*32*BB*0387*B*1*BB*7*BBB*7*F*64，

Hash K_6：*F*75508*D*0029*CD*27*EF*90*F*007*B*5385101*B*，

Hash K_7：*C*78*A*507*B*935*EDC*27*B*9211*CC*3*E*5*A*4*D*4*B*3.

仿真 2　将 K-HF 初始密钥 $K=(\alpha,x,t)$ 的各个分量在有效取值范围内分别做不同幅度的扰动，并使用扰动前后密钥对随机文本计算 hash 值，得扰动前后 hash 摘要改变 bit 数目如图 5 所示.

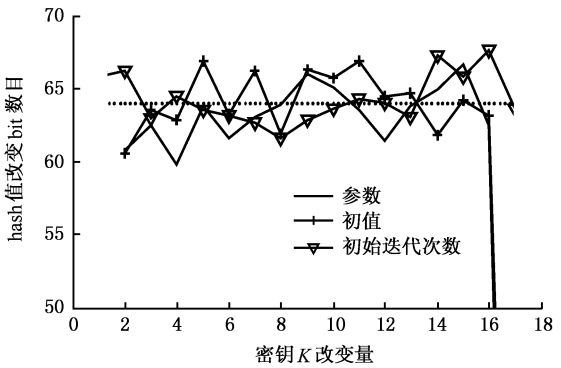

图 5　密钥敏感性仿真图(横坐标分别表示参数和初值改变量 V 的 $-\log_{10}(V)$，或初始迭代次数改变量)

上述仿真结果表明，初始密钥 K 中分量 α 或 x_0 在 10^{-16} 精度范围内、分量 m 在正整数范围内的微小扰动，所得到的 Hash 值每 bit 均以约 50% 的概率发生了变化，说明基于混沌动态 S-Box 的 K-HF 具有极高的密钥敏感性.

5.1.2. K-HF 的数据敏感性分析

文本 2 将文本 1 的第一个大写字母改为小写，文本 3 将文本 1 的 its 改为 ita，文本 4 将文本 1 的 high-creativity 改为 high creativity，文本 5 将文本 1 的 work 改为 works，用 K_1 作初始密钥，求得各文本的 128 位 Hash 函数值的 16 进制形式为

文本 1：815*BA*4*E*2*BE*95*E*26*B*717*A*2420*A*6*AB*175*C*，

文本 2：1156*D*910*D*651*EB*6*EA*5*A*515*E*6*C*2336*AE*5，

文本 3：5*AD*6101652*ABF*9*A*7111710*F*362*FCE*3*B*9，

文本 4：53*BCDCF*1*D*6*F*35*ABB*5*B*445575058*AF*704，

文本 5：16*E*88*C*7*DC*3*F*3*E*8*D*571*BD*0435*A*6*F*810*E*1.

将文本 1 的 Hash 结果与其他结果比较，改变的比特数目分别为 61，70，59，62，平均为 62.5，与理想状态的 64 仅差 1.5. 仿真结果说明，在初始密钥不变的情况下，原始数据的微小变化都将引起 Hash 值的很大改变，有很高的数据敏感性，即 Hash 值是原始数据每一 bit 极端敏感的函数.

5.2. “雪崩效应”统计分析

为了隐藏明文信息的冗余度，Shannon 提出了混乱与散布的概念，也叫“雪崩效应”. 加密体制中要求充分利用密文空间，K-HF 中同样如此，不仅要求相应明文与对应的 Hash 值不相关，同时还要做到密钥的敏感性. 理想的 K-HF 应该是初值和密钥极端敏感而复杂的函数值，即密钥或初值的每 bit 变化都将引起结果每 bit 以 50% 的概率发生变化，遵循“雪崩效应”. 为了测试提出的 K-HF 在统计意义上的“雪崩效应”定义[5,9].

平均变化 bit 数

$$\bar{B}=\frac{1}{N}\sum_{i=1}^{N}B_i\ ; \tag{10}$$

平均变化概率

$$P=(\bar{B}/128)\times 100\%\ ; \tag{11}$$

B 的均方差

$$\Delta B=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i-\bar{B})^2}\ ; \tag{12}$$

P 的均方差

$$\Delta P=\sqrt{\frac{1}{N-1}\sum_{i+1}^{N}(B_i/128-P)^2}\times 100\%\ . \tag{13}$$

这里 N 为统计次数，B_i 为第 i 次测试结果时变化的 bit 数.

初值敏感性测试：使用同一初始密钥 K 计算仅差 1bit 的两段随机明文的 K-HF 值，并比较其差别比特数目，N 次比较结果如表 2 所示.

表 2　初值敏感性测试表

统计值 \ N	256	512	1024	2048	总平均
B	63.922	63.518	63.66	63.399	63.6247
ΔB	6.4372	7.1396	7.5857	7.127	7.0724
P/%	49.939	49.623	49.734	49.53	49.7065
ΔP/%	5.029	5.5778	5.9263	6.3492	5.7206

密钥敏感性测试：计算随机选取的一段明文和密钥 K 的 K-HF 值，然后保持明文不变，对 K 进行一个微小扰动，再次计算 K-HF 值并比较，N 次比较结果见表 3.

表 3 密钥敏感性测试表

统计值 \ N	256	512	1024	2048	总平均
$\bar{B}$	63.691	64.33	63.843	63.873	63.9343
ΔB	5.8318	5.5191	5.6746	5.5631	5.6471
$P/\%$	49.759	50.258	49.877	49.901	49.9488
$\Delta P/\%$	4.5561	4.3118	4.4333	4.3462	4.4188

由大量仿真数据的统计分析结果可见,无论是初始数据的微小改变还是密钥的微小扰动,生成的 Hash 摘要平均变化比特和每比特的平均变化率分别趋近于 64 bit 和 50%.仿真结果表明,基于混沌动态 S-Box 的 K-HF 在摘要空间服从均匀分布,有很好的混乱与散布性,即'雪崩效应'显著.初值和密钥的极端敏感性,以及平均稳定的散布性使攻击者无法得到任何有用的统计信息,为抵御现有的已知密文攻击和差分线性攻击提供了很好的保证.

5.3. 基于混沌动态 S-Box 的 K-HF 执行效率比较

下文对基于混沌动态 S-Box 的 K-HF 执行性能进行实验研究.实验中将提出的 K-HF 与现有混沌 Hash 方案进行仿真对比,并将文献[5]和文献[7]提出的混沌 Hash 方案分别简称为'广义 Hash'和'时空 Hash",另外,为了数据的准确性,每组数据取 100 次计算结果的平均值.在 Matlab 7.0 中,分别用基于 S-Box 的 K-HF、广义 Hash 和时空 Hash 对不同长度的随机数据计算摘要,其中 K-HF 的混沌初始迭代次数取 100,各个算法计算时间(s)见表 4.

表 4 混沌 Hash 算法时间复杂度比较表

Hash 算法 \ 文本长度/bit	50	100	150	200	1×10^3	10×10^3
K-HF/s	0.026312	0.038375	0.037937	0.048719	0.15625	1.3125
广义 Hash/s	0.018891	0.030344	0.03925	0.051172	0.21875	2.0688
时空 Hash/s	3.0719	12.419	27.877	47.641	1252.9	125100

从表 4 可以看出,时空 Hash 的计算复杂度较高,对同一数据计算摘要时所需时间最多.广义 Hash 的时间复杂度大致与数据长度成正比,在对较短数据段计算摘要时所需时间最少,但当数据长度大于 150bit 时计算摘要所用时间多于 K-HF,且数据段越长,所用时间与 K-HF 的差距越大.由以上分析可知,基于混沌动态 S-Box 的 K-HF 不仅执行性能较高,且其计算时间随数据块的增大而增大的幅度最小.仿真结果说明,基于混沌动态 S-Box 的 K-HF 没有将原始数据直接参与混沌迭代,而采用混沌 S-Box 替换,提高了 K-HF 的运行效率,兼顾了安全性与复杂度.

6. 结 论

针对现有混沌单向 Hash 算法的不足,本文提出了一个基于混沌动态 S-Box 的 K-HF 设计方案.该方案首先利用混沌迭代系统生成寄存器初值和 S-Box,然后经函数变换、混沌动态 S-Box 替换和函数查找表,最后生成 $4n$ Byte 的 Hash 值.n 的可变性,增加了系统应用的灵活性.仿真研究结果表明:1)原始数据每比特的变化均将引起 Hash 值中近 50% 比特发生改变;2)当初始迭代次数大于 50 时,K-HF 的密钥 K 在 10^{-15} 精度允许范围内的微小扰动,Hash 值的每比特都以近似 50% 的概率发生变化;3)所设计的 K-HF 密钥空间约为 2^{158},原始信息、密钥 K 和 Hash 值之间复杂而敏感的非线性关系为抵御现有的暴力攻击和统计分析攻击提供了保证.另外,该方案并没有将原始数据直接进行混沌迭代,而仅用较少的混沌序列初始化寄存器和设计 S-Box,使混沌迭代步数与初始数据的长度无关,提高了算法的运行效率,兼顾了安全性与复杂度.

[1] Feng D G, Pei D Y 1999 *Introduction to Cryptography* (Beijing: Science Press) p192(in Chinese) [冯登国、裴定一 1999 密码学导引(北京:科学出版社)第 192 页]

[2] Bakhtiari S, Safavi-Naini R, Pieprzyk J 1996 *Lecture Notes in Computer Science* **1029** 201

[3] Knudsen L, Preneel B 2002 *IEEE Trans. Inform. Theor.* **48** 2524

[4] Heileman GL, Abdallah C, Hush D R *et al* 1993 *Proceedings of International Symposium on Nonlinear Theory and Its Applications* **1** 1183

[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] Li H D, Feng D G 2003 *Chinese Journal of Computers* **26** 460 (in Chinese) [李红达、冯登国 2003 计算机学报 **26** 460]

[7] Zhang H, Wang X F, Li C 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]

[8] Sheng L Y, Cao L L, Sun K H, Wen J 2005 *Acta Phys. Sin.* **54** 4031 (in Chinese) [盛利元、曹莉凌、孙克辉、闻　姜 2005 物理学报 **54** 4031]

[9] 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]

[10] 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]

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

[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, Xiao X C 2000 *Chin. Phys.* **9** 408

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

[16] Berson T A, Gong L. Secure, keyed, and collisionful Hash functions. Technical Report, SRI International Laboratory, Menlo Park, California, 1993

[17] Yang T, Shao H H 2002 *Acta Phys. Sin.* **51** 742 (in Chinese) [杨　涛、邵惠鹤 2002 物理学报 **51** 742]

[18] Yi X, Tan C H, Siew C K 2002 *IEEE Trans. Circuits Syst.* Ⅰ **49** 1826

[19] Shujun Li, Guanrong Chen, Xuanqin Mou. 2004 *IEEE Transactions on Circuits and Systems-II: Express Briefs* **51** 12 665

Keyed one-way Hash function construction based on the chaotic dynamic S-Box*

Guo Xian-Feng[1)2)]　Zhang Jia-Shu[1)]

1) (*Key Laboratory of Signal and Information Processing of Sichuan Province, Southwest Jiaotong University, Chengdu* 610031, *China*)

2) (*College of Computer Science and Technology, Southwest University for Nationalities, Chengdu* 610041, *China*)

(Received 4 November 2005; revised manuscript received 26 December 2006)

Abstract

This paper presents a novel keyed one-way Hash function based on a chaotic dynamic S-Box together with traditional one-way Hash function construction. The proposed approach can give a chaotic Hash value by means of the look up table of functions and chaotic dynamic S-Box. Compared with the existing chaotic Hash functions, this method improves computational performance of Hash system by using the chaotic dynamical S-Box substitution in place of iterating the original message directly in chaos system. Theoretical and experimental results show that the proposed method has strong one way property, large key space, sensitivity to initial conditions and chaotic system's parameters.

Keywords: Hash function, chaos, S-Box, look-up table of functions

PACC: 0545

* Project supported by the National Natural Science Foundation of China (Grant No. 60572027), by the Program for New Century Excellent Talents in University (Grant No. NCET-05-0794), by the Foundation for Young Scientists of Sichuan Province, China (Grant No. 03ZQ026-033) the Application Basic Foundation of Sichuan Province, China (Grant No. 2006J13-110) and by the Foundation of key laboratory of National Defense Science and Technology (Grant No. 51435080104QT2201).