DOI:10.11721/cqnuj20130420

一种改进的基于 Arnold 映射的 Hash 加密算法*

向　宇
(重庆广播电视大学 学生处，重庆 400052)

摘要:Arnold 映射 Hash 加密算法是一种二维混沌系统与 hash 函数相结合的加密方法,一般以 Arnold 映射的初值、迭代次数、Hash 值的编成方法、Hash 值的位数等作为加密密钥。由于该算法的不可逆性,使由密文到明文的逆向攻击失效,但明文和选择性明文攻击仍对该算法有一定的攻击效果。为更加有效地抵御各类明文、选择性明文的攻击,本文对该加密算法的 Arnold 映射初值、迭代次数等 2 个关键密钥进行改进,通过增加 Arnold 映射初值的个数,以及将迭代次数从常量拓展到变量的方法,构造出一个增强的 Arnold 映射 Hash 加密算法,从而进一步增加保密强度,提高 Arnold 映射 Hash 加密算法对明文、选择性明文攻击的抵抗能力。通过对实验数据的混乱与散布性质分析,改进后的 Arnold 映射 Hash 加密算法的平均变化位数和每位平均变化概率更加接近理想状况下的 64 位和 50%的变化概率,算法的保密性能更加良好。

关键词:混沌系统;Arnold 映射;Hash;加密算法
中图分类号:TP38　　**文献标志码**:A　　**文章编号**:1672-6693(2013)04-0103-06

保密通信在通信技术、计算机网络广泛运用的今天已深受人们重视,利用混沌系统设计加密算法则是一种常见的方法[1]。人们对使用 Arnold 混沌系统进行图形图像加密的技术已进行了大量深入的研究。2006 年,赵立强等人提出了一种基于广义 Arnold 混沌映射和 Hénon 混沌映射的小波水印算法[2]。2008 年,左黎明提出了一种基于 Arnold 混沌映射与 M 序列相结合的数字水印技术和水印信息置乱效果评价方法[3]。2012 年,王丽丽利用 Arnold 变换以及彩色图像的置乱度定义,求解 Lorenz 混沌系统的动力学方程,得到 3 个混沌序列;然后对 Arnold 置乱后的每个颜色分量进行置乱处理的算法,克服了 Arnold 变换和 Lorenz 混沌系统的缺点,能够抵抗多种攻击[4]。类似的研究成果还有很多,但很少从单向加密的角度考虑。本文则运用 Arnold 混沌系统与 Hash 函数相结合,通过密钥改进,构造出一个改进的基于 Arnold 映射的 Hash 加密算法,提升了基于 Arnold 映射的单向 Hash 加密算法的保密性能。

1 Arnold 映射与 Hash 函数

1.1 Arnold 映射

Arnold 映射又称为 cat 映射,是一种二维混沌系统,矩阵表述为 $\begin{pmatrix}x_{n+1}\\y_{n+1}\end{pmatrix}=\boldsymbol{C}\begin{pmatrix}x_n\\y_n\end{pmatrix} \bmod 1$,其中 $\boldsymbol{C}=\begin{pmatrix}1&1\\1&2\end{pmatrix}$,即 $\begin{pmatrix}x_{n+1}\\y_{n+1}\end{pmatrix}=\boldsymbol{C}\begin{pmatrix}x_n\\y_n\end{pmatrix} \bmod 1=\begin{pmatrix}1&1\\1&2\end{pmatrix}\begin{pmatrix}x_n\\y_n\end{pmatrix} \bmod 1$。或表述为方程组 $\begin{cases}x_{n+1}=x_n+y_n(\bmod 1)\\y_{n+1}=x_n+2y_n(\bmod 1)\end{cases}$,其中 mod 1 表示只取小数部分,即 $x \bmod 1=x-[x]$([ ]表示取整),因此(x_n,y_n)的相空间被限制在单位正方形 $[0,1]\times[0,1]$ 内[5-7]。

从 Arnold 映射的示意图(图 1)中,可以清楚地看到产生混沌运动的 2 个因素:拉伸(乘以矩阵 $\boldsymbol{C}$ 使 x,y 都变大)和折叠(取模使 x,y 又折回单位矩形内)。

1.2 Hash 函数

单向函数可简单描述为:若映射 $h:X\to Y$ 对 $\forall x\in X$,$h(x)$都容易计算;但反过来,给定一个 $h(x)$要求 x 在计算上是困难的,满足以上条件的函数称之为单向函数。

Hash 函数是一种特殊的单向函数,它满足以下 4 个条件:1) 输入为任意长度的序列,但输出长度固定;2)

* **收稿日期**:2013-03-22　**网络出版时间**:2013-07-20 19:23
资助项目:重庆市自然科学基金(No. CSTC2012JJA40052);重庆市教委科技计划项目(No. KJ110628;No. KJ120615;No. KJ120630);重庆市优秀人才支持计划(2010)
作者简介:向宇,男,讲师,硕士,研究方向为计算机加密技术,E-mail:yaodaolangzi@qq.com
网络出版地址:http://www.cnki.net/kcms/detail/50.1165.N.20130720.1923.201304.103_017.html

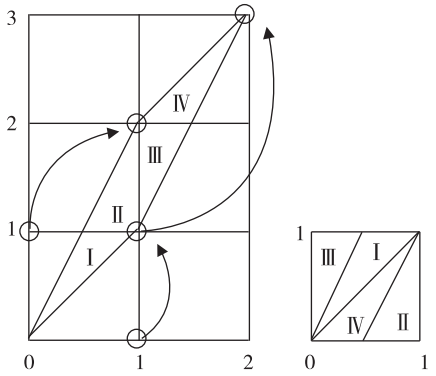

图1 Arnold映射示意图

不可逆性:即已知 $c=h(m)$,求 m 计算困难,除穷举外没有好办法;3) 防伪造性:已知 $c=h(m)$,求 n 使 $h(n)=c$ 计算困难;4) 初值敏感性:$c=h(m)$ 中 c 的每一 bit 都与 m 的每一 bit 相关,并有高度的敏感性,即每改变 m 每一 bit 都会对 c 产生明显的影响。

这些出色的性质使 Hash 函数可用来以较小的计算和存储代价保护数据的完整性和正确性,同时安全性也得到了保障,这对于单向加密技术有极其重要的意义。

2 基于 Arnold 映射的 Hash 函数构造

2.1 基于 Arnold 映射的 Hash 函数的一般构造原理

基于 Arnold 映射的 Hash 函数的构造一般设计思想为:将输入的原始明文以 bit 为单位,经过线性变换,保证明文内容的每个 bit 都得到应用,从而获得 2 个在 [0,1] 中的值。将这 2 个值作为 Arnold 映射的初始值(初始条件)进行迭代运算,最后将 x_R, x_{2R}, x_{3R}, y_R, y_{2R}, y_{3R}(R 为某一设定值)6 个迭代结果分别映射成 a_1 bit, a_2 bit, a_3 bit, a_4 bit, a_5 bit, a_6 bit 的二进制数,然后再从这些二进制数当中分别选择 b_1 bit, b_2 bit, b_3 bit, b_4 bit, b_5 bit, b_6 bit,合起来组成 128 bit 的 Hash 值。

具体的算法描述如下:

1) 设待加密明文 C 共有 N 个字节,将待明文 C 按字节转换为其对应的 ASCII 码,记为:$c_1, c_2, \cdots, c_N$;再将 $c_1, c_2, \cdots, c_N$ 线性变换为 [0,1] 范围内的数,整个消息变为一个大数组,记为 M,其中,数的个数即明文字节数为 N;

2) 设 Arnold 映射初始值为 x_1 和 y_1,令 $x_1=\frac{m_1}{A}-\left[\frac{m_1}{A}\right]$, $y_1=\frac{m_2}{B}-\left[\frac{m_2}{B}\right]$,其中 A、B 取大数,[ ] 表示取整运算。

3) 进行迭代运算:① 设迭代轮数为 r,令 $r=R\times\left(\left[\frac{N}{R}\right]\right)+K$,其中 [ ] 表示取整运算,且 R 取值为正整数常量,K 可取大于 0 的整数常量;② 将 x_1 和 y_1 作为 Arnold 映射方程的迭代初始值进行 r 轮迭代,Arnold 映射方程为 $\begin{cases}x_{n+1}=x_n+y_n \pmod 1\\ y_{n+1}=x_n+2y_n \pmod 1\end{cases}$。

4) 将 x_R, x_{2R}, x_{3R}, y_R, y_{2R}, y_{3R} 等 6 个迭代结果分别通过线性变换映射成 a_1 bit, a_2 bit, a_3 bit, a_4 bit, a_5 bit, a_6 bit 的二进制数,然后分别从中取出 b_1 bit, b_2 bit, b_3 bit, b_4 bit, b_5 bit, b_6 bit,合并起来组成 128 bit 的 Hash 值。

说明:在第 4)步中,从 $a_i(i=1,2,3,\cdots,6)$ bit 分别取 $b_i(i=1,2,3,\cdots,6)$ bit 时,可从 $a_i(i=1,2,3,\cdots,6)$ bit 中任意位置取出,而这个位置将作为密钥的一部分。$a_i(i=1,2,3,\cdots,6)>b_i(i=1,2,3,\cdots,6)$,且 $a_i(i=1,2,3,\cdots,6)\in\mathbf{Z}$, $b_i(i=1,2,3,\cdots,6)\in\mathbf{Z}$。

2.2 改进的基于 Arnold 映射的 Hash 函数的原理及算法

改进的基于 Arnold 映射的 Hash 函数构造原理与上述一般构造方法类似,但在以下 2 个方面进行了改进:1) 初始值由一组(2 个)增加到两组(4 个),用 2 次 Arnold 映射分别使用两组初始值进行迭代;2) 将 2 次 Arnold 映射迭代的轮数进行分别设置,将原方案中的迭代次数密钥 K 由常量改为变量,从而增强保密效果。

具体的算法描述如下:1) 设待加密明文 C 共有 N 个字节,将待明文 C 按字节转换为其对应的 ASCII 码,记为:$c_1, c_2, \cdots, c_N$;再将 $c_1, c_2, \cdots, c_N$ 线性变换为 [0,1] 范围内的数,整个消息变为一个大数组,记为 M,其中,数的个数即明文字节数为 N;

2) 设 Arnold 映射初始值为 x_1 和 y_1、x_2 和 y_2,令 $x_1=\frac{m_1}{A}-\left[\frac{m_1}{A}\right]$, $y_1=\frac{m_2}{B}-\left[\frac{m_2}{B}\right]$, $x_2=\frac{m_3}{U}-\left[\frac{m_3}{U}\right]$, $y_2=\frac{m_4}{V}-\left[\frac{m_4}{V}\right]$,其中 A、B、U、V 取大数,[ ] 表示取整运算;

3) 进行迭代运算:① 设迭代轮数为 r_1 和 r_2,令 $r_1=R\times\left(\left[\frac{N}{R}\right]\right)+K_1$, $r_2=R\times\left(\left[\frac{N}{R}\right]\right)+K_2$,其中 [ ] 表示取整运算,且 R 取值为正整数变量,K_1 和 K_2 可取大于 0 的整数变量,使 Arnold 映射的迭代轮数不再是常量,从而增加破解算法的难度;② 将 x_1 和 y_1、x_2 和 y_2 作为 Arnold 映射方程的迭代初始值分别进行 r_1 和 r_2 轮迭代。

4) 将 x_R, x_{2R}, x_H, x_{2H}, y_R, y_{2R}, y_H, y_{2H} 等 8 个迭代结果分别通过线性变换映射成 a_1 bit, a_2 bit, a_3 bit, a_4 bit, a_5 bit, a_6 bit, a_7 bit, a_8 bit 的二进制数,然后分别从中取出 b_1 bit, b_2 bit, b_3 bit, b_4 bit, b_5 bit, b_6 bit, b_7 bit, b_8 bit 合并起来

组成 128 bit 的 Hash 值。

说明：第 4)步中，从 $a_i(i=1,2,3,\cdots,8)$bit 分别取 $b_i(i=1,2,3,\cdots,8)$bit 时，可从 $a_i(i=1,2,3,\cdots,8)$bit 中任意位置取出，而这个位置将作为密钥的一部分。$a_i(i=1,2,3,\cdots,8)>b_i(i=1,2,3,\cdots,8)$，且 $a_i(i=1,2,3,\cdots,8)\in\mathbf{Z}$，$b_i(i=1,2,3,\cdots,8)\in\mathbf{Z}$。

3 计算机仿真

运用 2.2 中的改进算法进行计算机仿真实验，实验所用计算机设备主要配置如表 1 所示。

在如表 1 所示的硬件设备条件下，在 Windows XP 中运用 Matlab 6.1 系统进行仿真实验，仿真测试前 17 个步骤简单描述为：

表 1　基于 Arnold 映射的单向 hash 函数算法计算机仿真实验设备配置表

CPU	英特尔 Core i5-2430M @ 2.40GHz 双核
内存	2 GB(尔必达 DDR3 1 333 MHz)
硬盘	日立 HTS545050B9A300 (500 GB / 5 400 转/分)
显示器	LG LGD02E9(14 英寸)

1) 取待加密明文 1 为原始明文进行加密，获得密文 1，并将其作为加密效果对比标准。

2) 将明文 1 中第 1 个字符由“9”更改为“8”，形成明文 2，进行第 2 次加密，并将获得的密文 2 与密文 1 进行对比。

3) 将明文 1 中最后一个字符由“Z”更改为“z” 形成明文 3，进行第 3 次加密，获得密文 3 与密文 1 进行对比。

4) 将明文 1 末尾添加“.”形成明文 4，进行第 4 次加密，获得密文 4 与密文 1 进行对比。

5) 将明文 1 中“M”和“N”交换位置，形成明文 5，进行第 5 次加密，获得密文 5 与密文 1 对比。

6) 将明文 1 中“S”删除，缩短明文长度，形成明文 6，进行第 6 次加密，获得密文 6 与密文 1 对比。

7) 将明文 1 中插入字符串“Xiang” 形成明文 7，进行第 7 次加密，获得密文 7 与密文 1 对比。

8) 将明文 1 中的“0”更换为特殊符号“@” 形成明文 8，进行加密，获得密文 8 与密文 1 对比。

9) 将明文 1 中的“9876543210”的子串与“abcdefg”的子串进行换位，形成明文 9，再进行第 9 次加密，获得密文 9 与密文 1 对比。

10) 将明文 1 中第 1 个字符由“9”更改为“n”，形成明文 10，进行第 10 次加密，并将获得的密文 10 与密文 1 进行对比。

11) 将明文 1 中最后 1 个字符由“Z”更改为“X” 形成明文 11，进行第 11 次加密，获得密文 11 与密文 1 进行对比。

12) 将明文 1 末尾添加“0”形成明文 12，进行第 12 次加密，获得密文 12 与密文 1 进行对比。

13) 将明文 1 中“K”和“L”交换位置，形成明文 13，进行第 13 次加密，获得密文 13 与密文 1 对比。

14) 将明文 1 中“R”删除，缩短明文长度，形成明文 14，进行第 14 次加密，获得密文 14 与密文 1 对比。

15) 将明文 1 中插入字符串“Yu” 形成明文 15，进行第 15 次加密，获得密文 15 与密文 1 对比。

16) 将明文 1 中的“a”更换为特殊符号“#” 形成明文 16，进行加密，获得密文 16 与密文 1 对比。

17) 将明文 1 中的“9876543210”的子串与“HIJKLMN”的子串进行换位，形成明文 17，再进行第 17 次加密，获得密文 17 与密文 1 对比。

基于 Arnold 映射的单向 Hash 函数算法计算机仿真实验实验数据如表 2～表 4 所示，其中表 2 为上述 9 个明文序列，表 3 为算法改进前后 9 个明文序列对应的密文序列，表 4 为算法改进前后 9 个密文对应的 128 bit Hash 值。

表 2　基于 Arnold 映射的 Hash 函数算法仿真实验明文列表

明文编号	明文序列
明文 1	9876543210abcdefgHIJKLMNopqRSTuvwXYZ
明文 2	8876543210abcdefgHIJKLMNopqRSTuvwXYZ
明文 3	9876543210abcdefgHIJKLMNopqRSTuvwXYz
明文 4	9876543210abcdefgHIJKLMNopqRSTuvwXYZ.
明文 5	9876543210abcdefgHIJKLNMopqRSTuvwXYZ
明文 6	9876543210abcdefgHIJKLMNopqRTuvwXYZ
明文 7	9876543210abcdefgHIJKLMNopqRSTuvwXiangXYZ
明文 8	987654321@abcdefgHIJKLMNopqRSTuvwXYZ
明文 9	abcdefg9876543210HIJKLMNopqRSTuvwXYZ
明文 10	n876543210abcdefgHIJKLMNopqRSTuvwXYZ
明文 11	9876543210abcdefgHIJKLMNopqRSTuvwXYX
明文 12	9876543210abcdefgHIJKLMNopqRSTuvwXYZ0
明文 13	9876543210abcdefgHIJLKMNopqRSTuvwXYZ
明文 14	9876543210abcdefgHIJKLMNopqSTuvwXYZ
明文 15	9876543210abcdefgHIJKLMNopqYuRSTuvwXYZ
明文 16	9876543210#bcdefgHIJKLMNopqRSTuvwXYZ
明文 17	HIJKLMNabcdefg9876543210opqRSTuvwXYZ

表3 基于Arnold映射的Hash函数算法仿真实验获得密文列表

密文编号	改进前的密文序列	改进后的密文序列
密文1	489FCDBBAFDDB1EECC1B422C8498211E	6744BEF772130591D6645D6371890385
密文2	3415500B4469BB8B33BDF8362650EC2C	028A3EF233B29E39257B351D1269ACDF
密文3	BC448B5F055E72BFC64E4508AA746183	6744BEF72B390BBAD6645D63406EDF75
密文4	2A847390C6F4DCA51D8C166EB7F50D89	D48AB63BA4B10C35216E2FBC246BFF12
密文5	ED07D755E551C1FE4FABA5EFA66BA893	6744BEF7E2B12041D6645D63E47C3511
密文6	5C3BA0AD36DDB98C964C22BA63A2EF3C	8D9BE296431033764238A869072AF27B
密文7	605C72EA080B69D00AD137EF7B6CF5AB	C60ABE241E0A02398EAE1D76AE9DB6F3
密文8	3097DDBEDB2A7B6270C9FD726D57CCD5	F6D5437972130591B9FC723171890385
密文9	AF247E4A7FEF28EF3939D72C3A543AC1	69D0D29D678DD2C17EAD3B347DD5B3D5
密文10	72F1D0CC13E3E337263C4B8058A06224	410241885FA7E408BF713DAFA555426F
密文11	A87BAE53C6E15EAA59E8595C1836FAB3	6744BEF75CD91B05D6645D631F1C3ACE
密文12	30D7D48955910CDCC8030D9ABD7399DE	D48AB63BFECDD368216E2FBC70A786A5
密文13	345FA2ECF407B1325BF473A4FBC93E53	6744BEF7F3AC453FD6645D638871DFE4
密文14	992E80D1B6C07C9370D67F1E7405D2FC	8D9BE2965B64489A4238A869E9F2F158
密文15	9B2A3F0AC8C3729906F0FE92C48F87EC	552CDBC7B713BAB17344FAFB8E41C7F3
密文16	0A0C0332C70F5CC63EDC965995174DB3	BAB590B172130591563376E4571890385
密文17	54DC6A9D9EE0DC5424FCA93AE0F5205A	CA417DA49520759CB9514866F7D40ACA

4 混乱与散布统计性质分析

根据Shannon提出的混乱与散布的概念,理想Hash函数的散布效果应该是初值的细微变化将导致结果的每bit都以50%的概率变化,考察算法在明文发生1 bit变化的情况下,引起Hash密文结果的变化bit数,就可以知道Hash函数的混乱和散布统计性质。对混合Arnold映射Hash算法的混乱和散布统计性质统计为[8-9]:1)平均变化bit数 $\overline{B}=\frac{1}{N}\sum_{i=1}^{N}B_i$;2)平均变化概率 $P=\frac{\overline{B}}{128}\times 100\%$;3)$B$的均方差 $\Delta B=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i-\overline{B})^2}$;4)$P$的均方差

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

其中N为统计总次数,B_i为第i次测试时结果的变化bit数。

测试方法为:在明文空间中随机选取一段明文进行Hash测试,

表4 基于Arnold映射的Hash函数算法仿真实验获得密文Hash值

密文编号	改进前密文的128 bit Hash值	改进后密文的128 bit Hash值
密文1	01001000100111111100110110111011 10101111110111011011000111101110 11001100000110110100001000101100 10000100100110000010000100011110	01100111010001001011111011110111 01110010000100110000010110010001 11010110011001000101110101100011 01110001100010010000001110000101
密文2	00110100000101010101000000001011 01000100011010011011101110001011 00110011101111011111100000110110 00100110010100001110110000101100	00000010100010100011111011110010 00110011101100101001111000111001 00100101011110110011010100011101 00010010011010011010110011011111
密文3	10111100010001001000101101011111 00000101010111100111001010111111 11000110010011100100010100001000 10101010011101000110000110000011	01100111010001001011111011110111 00101011001110010000101110111010 11010110011001000101110101100011 01000000011011101101111101110101
密文4	00101010100001000111001110010000 11000110111101001101110010100101 00011101100011000001011001101110 10110111111101010000110110001001	11010100100010101011011000111011 10100100101100010000110000110101 00100000101101110001011111010111100 00100100011010111111111100010010
密文5	11101101000001111101011101010101 11100101010100011100000111111110 01001111101010111010010111101111 10100110011010111010100010010011	01100111010001001011111011110111 11100010101100010010000001000001 11010110011001000101110101100011 11100100011111000011010100010001
密文6	01011100001110111010000010101101 00110110110111011011100110001100 10010110010011000010001010111010 01100011101000101110111100111100	10001101100110111110001010010110 01000011000100000011001101110110 01000010001110001010100001101001 00000111001010101111001001111011

然后改变明文 1 bit 的值得到另一 Hash 结果，比较 2 个结果得到变化 bit 数 B_i。经 128 次测试，得到明文 1 bit 变化下的统计变化 bit 数为 $\overline{B}=63.513$，$P=49.69\%$，$\Delta B=5.065$，$\Delta P=4.148\%$。这个算法的平均变化 bit 数和每 bit 平均变化概率都已非常接近理想状况下的 64 bit 和 50%的变化概率，相当充分和均匀地利用了密文空间；而 Δ 标志着 Hash 混乱与散布性质的稳定性，越接近 0 就越稳定，这个算法的 Δ 都已很小，其混乱与散布性质是稳定的[10]。

5 算法的快速性和碰撞分析

从改进的 Hash 算法描述来看，算法的速度与原始报文长度基本成正比的关系。算法的两轮迭代的轮数分别为：$r_1=R\times\left(\left[\frac{N}{R}\right]\right)+K_1$，$r_2=R\times\left(\left[\frac{N}{R}\right]\right)+K_2$。若混沌映射对初始值具有高度的敏感性，则 R、K_1 和 K_2 的值可选择较小；否则，就应选择较大的数值。本算法中的所选择的 Arnold 映射对初值有很高的敏感性，因此 R、K_1 和 K_2 的数值可以选择较小，从而使得 r_1 和 r_2 较小，从而保证算法具有较快的速度。因此原始文本很短时只需很小的迭代步数。

所谓碰撞，是指不同的初值 Hash 映射结果相同，即发生了多对一映射。取初始文本为一字节，即包含 8 bit，ASCII 码的对应值为 0～255；同样 Hash 值也取为 8 bit，也对应为一个 0～255 的数。这样，初值空间与终值空间相同。设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 个原像的点的个数记为 $n(k)$。如果 $n(1)$ 越大，$n(0)$ 和其他各项越小，表明碰撞越少，Hash 函数的散乱能力越强。

用值域空间与定义域空间的

续表 4

密文编号	改进前密文的 128 bit Hash 值	改进后密文的 128 bit Hash 值
密文 7	01100000010111000111001011101010 00001000000010110110100111010000 00001010110100010011011111101111 01111011011011001111010110101011	11000110000010101011111000100100 00011110000010100000001000111001 10001110101011100001110101110110 10101110100111011011011011110011
密文 8	00110000100101111101110110111110 11011011001010100111101101100010 01110000110010011111110101110010 01101101010101111100110011010101	11110110110101010100001101111001 01110010000100110000010110010001 10111001111111000111001000110001 01110001100010010000001110000101
密文 9	10101111001001000111111001001010 01111111111011110010100011101111 00111001001110011101011100101100 00111010010101000011101011000001	01101001110100001101001010011101 01100111100011011101001011000001 01111110101011010011101100110100 01111101110101011011001111010101
密文 10	01110010111100011101000011001100 00010011111000111110001100110111 00100110001111000100101110000000 01011000101000000110001000100100	01000001000000100100000110001000 01011111101001111110010000001000 10111111101110001001111011010111 10100101010101010100001001101111
密文 11	10101000011110111010111001010011 11000110111000010101111010101010 01011001111010000101100101011100 00011000001101101111101010110011	01100111010001001011111011110111 01011100110110010001101100000101 11010110011001000101110101100011 00011111000111000011101011001110
密文 12	00110000110101111101010010001001 01010101100100010000110011011100 11001000000000110000110110011010 10111101011100111001100111011110	11010100100010101011011000111011 11111110110011011101001101101000 00100001011011100010111110111100 01110000101001111000011010100101
密文 13	00110100010111111010001011101100 11110100000001111011000100110010 01011011111101000111001110100100 11111011110010010011111001010011	01100111010001001011111011110111 11110011101011000100010100111111 11010110011001000101110101100011 10001000011100011101111111100100
密文 14	10011001001011101000000011010001 10110110110000000111110010010011 01110000110101100111111100011110 01110100000001011101001011111100	10001101100110111110001010010110 01011011011001000100100010011010 01000010001110001010100001101001 11101001111100101111000101011000
密文 15	10011011001010100011111100001010 11001000110000110111001010011001 00000110111100001111111010010010 11000100100011111000011111101100	01010101001011001101101111000111 10110111000100111011101010110001 01110011010001001111101011111011 10001110010000011100011111110011
密文 16	00001010000011000000001100110010 11000111000011110101110011000110 00111110110111001001011001011001 10010101000101110100110110110011	10111010101101011001000010110001 01110010000100110000010110010001 01010110001101110110111001000101 01110001100010010000001110000101
密文 17	01010100110111000110101010011101 10011110111000001101110001010100 00100100111111001010100100111010 11100000111101010010000001011010	11001010010000010111110110100100 10010101001000000111010110011100 10111001010100010100100001100110 11110101111010100000010101100101 0

测度之比来定量衡量碰撞发生程度，令 $C_e=\frac{256-n(0)}{256}$，C_e 的值越接近 1，表示碰撞程度越低；当 C_e 等于 1 时，表示完全没有碰撞发生。参数 C_e 的计算与 Hash 值长度密切相关，对于不同的 Hash 值长度，算法的 Hash 性能变化很大，C_e 值变化也较大，故难以进行精确的衡量与预测。

6 结论

改进后的基于 Arnold 映射的 Hash 加密算法，对初值有高度的敏感性，及很好的单向 Hash 函数性能，混乱与散布性质上更加均匀，通过增强初值设置环节、增加迭代次数变化等 2 个环节强加了算法抵抗攻击的能力；同时，其加密时间短、加密效率高，是一种更加快速高效的加密算法。

参考文献：

[1] Bruce Schneier. 应用密码学-协议、算法与 C 源程序[M]. 北京：机械工业出版社，2001：311-315.
Bruce S. Applied cryptography protocols, algorithms, and source code in C[M]. Beijing: China Machine Press, 2001: 311-315.

[2] 赵立强，李艳坡，孔令富，等. 基于广义 Arnold 混沌映射和 Hénon 混沌映射的小波水印算法[J]. 河北科技师范学院学报，2006，20(4)：34-40.
Zhao L Q, Li Y P, Kong L F, et al. An image digital watermarking algorithm based on Arnold, Hénon chaotic maps and wavelet transform[J]. Journal of Hebei Normal University of Science & Technology, 2006, 20(4): 34-40.

[3] 左黎明. 一种基于 Arnold 混沌映射的数字水印技术研究[J]. 华东交通大学学报，2008，25(5)：60-64.
Zuo L M. A research of digital watermark technique based on Arnold chaotic mapping[J]. Journal of East China Jiaotong University, 2008, 25(5): 60-64.

[4] 王丽丽. 基于 Arnold 变换和 Lorenz 混沌系统的彩色水印图像加密算法[J]. 计算机系统应用，2012，21(6)：123-127.
Wang L L. Encryption algorithm of colored watermark image based on Arnold transform and lorenz chaos system[J]. Computer Systems & Applications, 2012, 21(6): 123-127.

[5] Kocarev L. Chaos-based cryptography: a brief overview[J]. IEEE Circuits and Systems Magazine, 2001, 1(3): 6-21.

[6] Wong K W, Ho S W, Yung C K. A chaotic cryptography scheme for generating short ciphertext[J]. Physics Letters A, 2003, 310(1): 67-73.

[7] Alvarez G G, Montoya F, Romera M, et al. Cryptanalysis of dynamic look-up table based chaotic cryptosystems[J]. Phys Lett A, 2004, 326(3/4): 211-218.

[8] 胡滨，范九伦. 一种带有猫映射动态置换盒的分组密码[J]. 西安邮电学院学报，2012，17(1)：19-23.
Hu B, Fan J L. A block cipher with dynamic S-boxes based on arnold map[J]. Journal of Xi'an University of Posts and Telecommunications, 2012, 17(1): 19-23.

[9] 陈仁杰，孙友林，何丹. 基于广义猫映射和 H. 264 的视频加密[J]. 物联网技术，2012，2(1)：52-55.
Chen R J, Sun Y L, He D. Video encryption based on generalized cat mapping and H. 264[J]. Internet of Things Technologies, 2012, 2(1): 52-55.

[10] 王圆妹，李涛. 基于 Arnold 变换和混沌理论的图像加密算法研究[J]. 山西大学学报：自然科学版，2012，35(1)：49-53.
Wang Y M, Li T. Image encryption algorithm based on Arnold transform and chaos theory[J]. Journal of Shanxi University: Natural Science Edition, 2012, 35(1): 49-53.

An Improved Hash Encryption Algorithm Based on Arnold Mapping

XIANG Yu

(Students' Affairs Office, Chongqing Radio & TV University, Chongqing 400052, China)

Abstract: The Hash encryption algorithm based on Arnorld mapping is realized by integration of two-dimensional chaotic system and Hash function, whose encryption keys are the initial value of the Arnorld mapping, iterations, the forming method of hash value, and the digit of hash value and so on. The irreversibility of this algorithm can deactivate the reverse attack from cryptograph to plaintext but fail to work against all the attacks such as plaintext and selective plaintext. According to this article, to improve the initial value of the Arnorld mapping and the iterations of the encryption algorithms by means of increasing the number of the initial value of the Arnorld mapping and changing the constant value of the iterations into variable value and building a more secure hash encryption algorithm can further strengthen encryption, so as to improve the resistant ability of the hash encryption algorithm against such attacks as plaintext, selective plaintext etc. And the experimental data indicates the practicability and effectiveness of the improved hash encryption algorithm based on Arnorld mapping in that an analysis of its confusion and dispersion properties shows that its changing bits and average change probability of every bit of is closer to the ideal 50% 64-bit change probability.

Key words: chaotic system; Arnorld mapping; hash; encryption algorithms

(责任编辑 黄 颖)