一种基于分段线性映射的分组密码算法

马 洁 张元清
(重庆教育学院美术系 重庆 400067) (重庆教育学院计现系 重庆 400067)

摘 要 提出了一种新的基于混沌理论的分组密码算法,把128比特的明文加密为128比特的密文。整个加密过程包含了8个轮变换,每一个轮变换由替换变换、移位变换和置换变换3部分组成。所有的轮密钥都由128位的比特流K和由分段线性映射产生的128比特随机二进制序列导出。理论与实验分析表明该算法克服了一些纯混沌密码系统的固有缺陷,具有较高的性能。

关键词 分组密码,混沌映射,模乘运算,置换
中图法分类号 TP309.7 **文献标识码** A

Block Cryptosystem Based on Piecewise Linear Map

MA Jie ZHANG Yuan-qing
(Fine Arts Department of Chongqing Education College, Chongqing 400067, China)
(Computer Department of Chongqing Education College, Chongqing 400067, China)

Abstract Based on the study of some existing chaotic encryption algorithms, a new block cipher was proposed. The proposed cipher encrypts 128-bit plaintext to 128-bit ciphertext blocks. It consists of eight computationally identical rounds transformation. All roundkeys are derived from K and a 128-bit pseudorandom binary sequence generated from a chaotic map. Analysis shows that the proposed block cipher does not suffer from the flaws of pure chaotic cryptosystems and possesses high security.

Keywords Block cipher, Chaotic map, Modulo multiplication, Permutation

1 引言

大多数基于混沌的软件加密技术都是使用混沌映射来产生伪随机序列。然而,Wheeler等人[9,10]指出,当混沌系统用有限精度的计算机来实现时,数字化的混沌系统表现出许多明显不同的行为,它们的数字动力学行为也远不如连续混沌系统的动力学行为,例如,非常短的周期、依赖于特定的数字精度等。目前已经提出了很多基于混沌的密码算法[3-8,11,13],但这些算法都没有能很好地克服由于混沌数字化所引起的性能退化问题。在本文中,结合混沌和代数群上的⊙运算,构造了一种新的且更具安全性的混沌分组密码。

本文第2节详细地描述了所提出的加密算法;第3节通过仿真实验验证算法的正确性;第4节从理论上分析了算法的安全性和性能;最后是对本文的总结。

2 新的加密算法

2.1 伪随机数序列的产生

具有良好性质的伪随机数序列在保密通信和密码学中有着广泛的应用。文献[8]提出了一个具有良好随机统计特性的一维分段线性混沌映射,其定义如下:

$$F(p,x)=\begin{cases}x/p, & x\in[0,p)\\ (x-p)/(1/2-p), & x\in[p,1/2]\\ F(p,1-x), & x\in[1/2,1]\end{cases} \quad (1)$$

此处,p是控制参数,且$p\in(0,1/2)$。该混沌映射式(1)在区间[0,1]上具有下面的一些比较好的统计特性。

本文提出的新的加密算法要求信息的发送者和接收者知道4个密钥参数p_1,p_2,x_0,K,并且要求$0<p_1,p_2<1/2$,$p_1\neq p_2$,K是一个128比特的二进制密钥串。然后定义如下两个具有相同初值(x_0)的离散分段线性映射来产生拟混沌轨道$\{x_1(i)\},\{x_2(i)\}$:

$$F(p_1,x_1(0)):x_1(i+1)=F(p_1,x_1(i)) \quad (2)$$

$$F(p_2,x_2(0)):x_2(i+1)=F(p_2,x_2(i)) \quad (3)$$

此处,$x_1(0)=x_2(0)=x_0$,并且$i=0,1,2,\cdots$。

首先,分别用离散混沌映射$F(p_1,x_1(0))$和$F(p_2,x_2(0))$产生两个拟混沌序列(为了性能更好,可以让映射先行迭代N_0次):$x_1(1),x_1(2),\cdots,x_1(i),\cdots,x_2(1),x_2(2),\cdots,x_2(i),\cdots$。

然后,定义伪随机二进制序列PRN,对其中的每一位有

$$\mathrm{prn}_i=\begin{cases}0, & \text{if } x_1(i)>x_2(i)\\ \text{no output}, & \text{if } x_1(i)=x_2(i)\\ 1, & \text{if } x_1(i)<x_2(i)\end{cases} \quad (4)$$

2.2 密钥编排

本文加密算法所有的轮密钥$K^{(r)}$($r=1,2,\cdots,8$)都是由密钥K(128比特)和上面描述的方法产生的伪随机二进制序列PRN(128比特)通过下面的密钥扩展算法keyschedule(K,PRN)生成的。

到稿日期:2008-10-08 返修日期:2008-12-14

马 洁(1979-),女,硕士,讲师,主要研究方向为计算机应用、信息安全、电子商务等;张元清(1968-),女,副教授,主要研究方向为计算智能。

for(r=1;r≤8;r++)

$K^{(r)}$=PRN⊕(K<<<16·(r−1))

此处,⊕表示按位异或,<<<表示 K 循环左移 16·(r−1)位。

2.3 替换变换

替换变换是一个非线性的字(16 比特)变换,独立地作用在每一个 16 比特的子块上。变换算法 MessageSub($K^{(r)}$,$C^{(r-1)}$)过程如下。

第 1 步　把 $K^{(r)}$ 分成 8 个 16 比特的子块 $K_i^{(r)}$(1≤i≤8);把 $C^{(r-1)}$ 也分成 8 个 16 比特的子块 $C_i^{(r-1)}$(1≤i≤8)。此处,$K^{(r)}$ 是由 2.2 节的方法产生的第 r 轮的轮密钥,$C^{(r-1)}$ 表示第(r−1)轮的输出。

第 2 步　对应于 1≤i≤8,将 16 比特的 $K_i^{(r)}$,$C_i^{(r-1)}$ 转换成十进制数,然后对这两个十进制数进行群($Z_{2^{16}+1}^*$,⊙)上的模乘运算,即 $I_i^{(r)}$ ← $Int2Bin$ ($Bin2Int$ ($K_i^{(r)}$) ⊙$Bin2Int$ ($C_i^{(r-1)}$))。

第 3 步　将第 2 步得到的 8 个 16 比特 $I_i^{(r)}$(1≤i≤8)拼接,得到替换变换后的 128 比特的中间结果:$I^{(r)}$ ← ($I_1^{(r)}$ $I_2^{(r)}$ $I_3^{(r)}$ $I_4^{(r)}$ $I_5^{(r)}$ $I_6^{(r)}$ $I_7^{(r)}$ $I_8^{(r)}$)。

注意,算法中的 $Bin2Int$(·)表示把 16 比特的二进制数变换成对应的十进制数,$Int2Bin$(·)表示把十进制数变换成对应的 16 比特的二进制数。群 $Z_{2^{16}+1}^*=\{a|a\in 1,2,\cdots,2^{16}\}$,⊙表示群($Z_{2^{16}+1}^*$,⊙)上的模乘运算,即两个元素的乘积再模上 $2^{16}+1$。对 $\forall a,b\in Z_{2^{16}+1}^*$,有 $a\odot b=(a\cdot b)_{2^{16}+1}\in Z_{2^{16}+1}^*$。

由于 $0\notin Z_{2^{16}+1}^*$,而 $2^{16}=65536\in Z_{2^{16}+1}^*$,所以用 2^{16} 代替 0。因此,如果一个操作数是 0 时,用 2^{16} 代替。同样,如果结果等于 2^{16},则用 0 代替。另外,在计算群($Z_{2^{16}+1}^*$,⊙)的逆元时,将采用扩展的 Euclidean 算法。

显然,MessageSub 变换是可逆的,并且其逆变换也是非线性的字变换。其可逆性由群($Z_{2^{16}+1}^*$,⊙)的可逆性决定,即 $C_i^{(r-1)}\leftarrow(K_i^{(r)})^{-1}\odot I_i^{(r)}$,$(K_i^{(r)})^{-1}$ 是 $K_i^{(r)}$ 在群 $Z_{2^{16}+1}^*$ 上的逆元。记 MessageSub 的逆变换为 Inv_ MessageSub。

2.4 移位变换

在移位变换过程中,不同的子块根据轮密钥的不同,循环左移不同的位数。其具体算法 MessageShift ($K^{(r)}$,$I^{(r)}$)过程如下。

第 1 步　把 $K^{(r)}$ 分成 8 个 16 比特的子块 $K_i^{(r)}$(1≤i≤8),把 $I^{(r)}$ 分成 8 个 16 比特的子块 $I_i^{(r)}$(1≤i≤8)。此处,$K^{(r)}$ 是由 2.2 节的方法产生的第 r 轮的轮密钥,$I^{(r)}$ 表示第 r 轮替换变换(见 2.3 节)的输出。

第 2 步　对应于 1≤i≤8,执行如下操作,$T_i^{(r)}\leftarrow I_i^{(r)}$ <<< ($Bin2Int(K_i^{(r)})$ mod 2^4)。

第 3 步　将第 2 步得到的 8 个 16 比特 $T_i^{(r)}$(1≤i≤8)拼接,得到移位变换后的 128 比特的中间结果:$T^{(r)}\leftarrow$ ($T_1^{(r)}$ $T_2^{(r)}$ $T_3^{(r)}$ $T_4^{(r)}$ $T_5^{(r)}$ $T_6^{(r)}$ $T_7^{(r)}$ $T_8^{(r)}$)。

注意,<<< 表示 $I_i^{(r)}$ 循环左移 ($Bin2Int(K_i^{(r)})$ mod 2^4)位,$Bin2Int$(·)的含义同 2.3 节。MessageShift 移位变换的可逆变换是循环右移,即 $I_i^{(r)}\leftarrow T_i^{(r)}$ >>> ($Bin2Int$ ($K_i^{(r)}$) mod 2^4)。记这种可逆变换为 Inv_ MessageShift。

2.5 置换变换

在置换变换过程中,实质是对 128 比特的 8 个 16 比特子块进行重排列。其具体算法 MessagePermu ($K^{(r)}$,$T^{(r)}$)过程如下。

第 1 步　把 $K^{(r)}$ 分成 16 个 8 比特的子块 $K_i^{(r)}$(1≤i≤16),把 $T^{(r)}$ 分成 8 个 16 比特的子块 $T_j^{(r)}$(1≤j≤8)。此处,$K^{(r)}$ 是由 2.2 节的方法产生的第 r 轮的轮密钥,$T^{(r)}$ 表示第 r 轮移位变换(见 2.4 节)的输出。

第 2 步　计算 $w_r=g(\bigoplus_{i=1}^{16}K_i^{(r)})$,得到 1～8 的一个排列 w_r。然后按排列 w_r 的顺序重新排列 $T_j^{(r)}$(1≤j≤8),得到 $T_{i_1}^{(r)}$ $T_{i_2}^{(r)}$ $T_{i_3}^{(r)}$ $T_{i_4}^{(r)}$ $T_{i_5}^{(r)}$ $T_{i_6}^{(r)}$ $T_{i_7}^{(r)}$ $T_{i_8}^{(r)}$。

注意,映射 g 的映射构造算法如下:

Ⅰ)由混沌映射 $F(p_1,x_1(i))$产生一个新的混沌状态 $x_1(i+1)$。

Ⅱ)通过模 8 加 1 操作,抽取 $x_1(i+1)$的前 8 个不同的数字位得到 1～8 的一个排列。如果这次操作失败,即状态 $x_1(i+1)$中的数字位通过模 8 加1 操作不能得到 1～8 的一个排列或者得到的排列前面已经出现过,则转 Ⅰ)继续,直到得到 256 个不同的 1～8 的排列为止,如表 1 所列。

表 1　i 与 1,2,3,4,5,6,7,8 的排列 w_i 之间的关系

i		w_i
Decimal	Binary	
0	00000000	w_0
1	00000001	w_1
…	……	…
254	11111110	w_{254}
255	11111111	w_{255}

显然,MessagePermu 变换也是可逆的,其逆变换记作 Inv_ MessagePermu。

2.6 加密与解密过程

加密算法步骤如下(Encryption (p_1,p_2,x_0,K,P),其加密过程示意如图 1 所示。

第 1 步　按照 2.1 节的方法产生 128 比特的二进制随机序列 PRN。

第 2 步　按照 2.2 节的算法得到 8 个轮密钥 $K^{(r)}$(1≤r≤8)。

第 3 步　把 P 赋给 $C^{(0)}$,即 $C^{(0)}\leftarrow P$。

第 4 步　对应于 r 从 1 到 8,执行下面的操作

$I^{(r)}\leftarrow$MessageSub($K^{(r)}$,$C^{(r-1)}$)

$T^{(r)}\leftarrow$MessageShift($K^{(r)}$,$I^{(r)}$)

$C^{(r)}\leftarrow$MessagePermu($K^{(r)}$,$T^{(r)}$)

第 5 步　输出密文 C,即 $C\leftarrow C^{(8)}$。

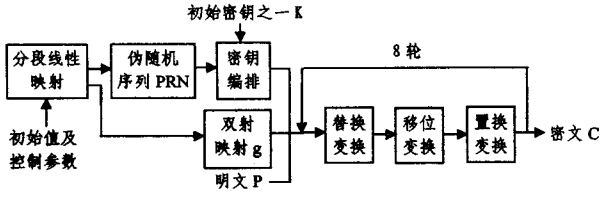

图 1　加密算法框图

在解密过程中,每一轮的变换顺序是与加密过程相反的。其算法 Decryption (p_1,p_2,x_0,K,C)具体步骤如下。

第 1 步　按照 2.1 节的方法产生 128 比特的二进制随机序列 PRN。

第 2 步　按照 2.2 节的算法得到 8 个轮密钥 $K^{(r)}$(1≤r≤8)。

第 3 步　把密文 C 赋给 $C^{(8)}$，即 $C^{(8)} \leftarrow C$。

第 4 步　对应于 r 从 1 到 8，执行下面的操作

$T^{(r)} \leftarrow \text{Inv\_MessagePermu}(K^{(r)}, C^{(r)})$

$I^{(r)} \leftarrow \text{Inv\_MessageShift}(K^{(r)}, T^{(r)})$

$C^{(r-1)} \leftarrow \text{Inv\_MessageSub}(K^{(r)}, I^{(r)})$

第 5 步　输出明文 P，即 $P \leftarrow C^{(0)}$。

3　仿真实验结果

在一个密码系统中，安全性是首要的问题。下面将从理论和仿真实验方面来阐述本文提出的加密算法的安全性。

在实验中，为了评估算法的性能，采用了一个 4000 字节的文本文件和一个 256×256 像素的灰度图像文件。设 $x_0 = 0.436567349535648$，$p_1 = 0.485734534345379$，$p_2 = 0.234579834895896$，$K =$ "*abcdefghijklmnop*"。

图 2 表明，该算法能够正确地加/解密文件。注意，当用本文的算法来加密文本文件时，在密文中可能存在一些不可打印的字符。

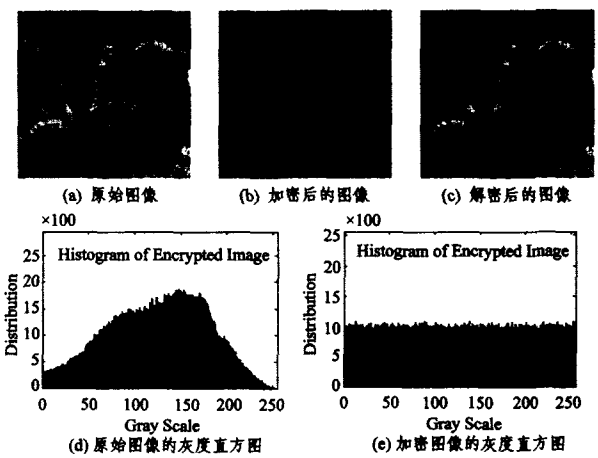

图 2　Tiger 图像加密实验

4　安全性与性能分析

4.1　密钥空间

在下面的分析中，采用了 IEEE 754[12] 浮点数标准。根据本文对密钥序列的产生方法可以知道，式(2)、式(3)中参数 p_1，p_2 对最后得到的伪随机密钥序列 PRN 的性能影响非常大。因此需要仔细地选择参数 p_1，p_2。因为 $0 < p_1, p_2 < 1/2$，$p_1 \neq p_2$，所以 p_1，p_2 的第 1 位 $d_1^{(1)}, d_1^{(2)} \in \{0,1,2,3,4\}$，又 K 的长度为 128 位，则本文提出的算法的密钥空间约为 $(5 \cdot 10^{14}) \times (5 \cdot 10^{14}) \times (10^{15}) \times 2^{128} = 2.5 \times 10^{44} \times 2^{64} \approx 2^{271.5}$。

4.2　排列分析

排列几乎是所有的传统密码系统的基本操作。在许多密码系统中，排列只是根据设计者预先定义的方式重新排列输入元素，与密钥无关。在实际的密码分析过程中，由于这种排列很容易被差分分析攻破，因此它对算法安全性几乎没有什么意义。然而，在本文提出的加密算法中，排列是与密钥相关的，不同的消息块有不同的排列方式，从而增加了密码分析的难度。

4.3　统计测试

根据 Shannon 理论，一个密码系统在抗统计攻击方面应该具有很好的性质。下面的实验表明本文的密码系统保留了这个好的特性。实验结果如图 3(第 3 节的文本加密前后的统计直方图)所示。我们发现密文的直方图分布已经相当均匀了，并且完全不同于明文的直方图分布。

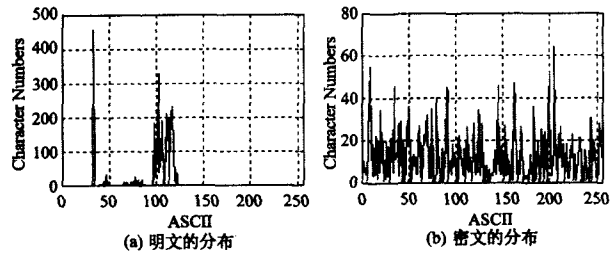

图 3　明文和密文分布

文本文件通常是由可打印字符组成的，其 ASCII 码一般在 033～126 之间。然而，经过本文的算法加密后，其密文的 ASCII 码分布在整个 0～255 之间，如图 3 所示。

4.4　密钥敏感性测试

由于本章的加密算法的密钥是由 x_0，K 两部分组成的，因此将从两个方面来进行密钥敏感性测试。

Ⅰ)保持 p_1，p_2，x_0 不变，改变密钥 K 的最后一位。修改后的密钥 $K' =$ "*abcdefghijklmnoq*"。然后用密钥 K' 和 p_1，p_2，x_0 解密图 2 (b)，实验结果如图 4(a)所示。

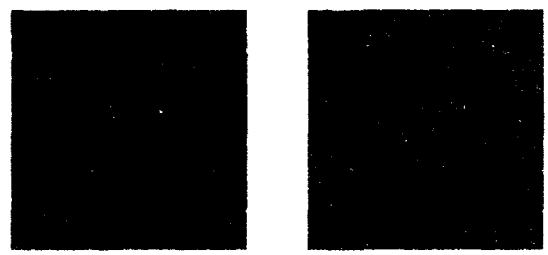

图 4　密钥敏感性测试

Ⅱ)保持 p_1，p_2，K 不变，改变 x_0 的最后一位。修改后的密钥 $x_0' = 0.436567349535649$。然后用密钥 p_1，p_2，K 和 x_0' 解密图 2 (b)，实验结果如图 4(b)所示。

实验结果表明，尽管密钥只有微小的差异，也导致了解密的失败。因此，这个新的加密算法仍然保持了密钥敏感性。同时，在实验中也发现，两个只有 2^{-15} 微小差异的初值 x_0 和 x_0'，构造的双射映射也几乎完全不同。

结束语　本文提出了一种新的基于混沌映射和代数群上运算的密码系统。在这个新的密码系统中，每个 128 比特的明文块产生一个同样长度的密文块，同时密文也依赖于明文、密钥、混沌映射和群上的运算。本算法弥补了纯混沌密码系统的一些缺陷。另外，大的密钥空间、比特位的替换与移位和基于密钥的子块排列变换都大大增强了算法的各种抗攻击能力。

参 考 文 献

[1] Yang H, et al. A new block cipher based on chaotic map and group theory [J]. Chaos, Solitons & Fractals, 2005(10): 10-16

[2] Tang G, Liao X F. A method for designing dynamical S-boxes based on discretized chaotic map [J]. Chaos, Solitons & Fractals, 2005, 23: 1901-1909

(下转第 172 页)

Int. Conf. on Genetic Algorithms. Lawrence Erlbaum, 1985: 93-100

[3] Hajela P, Lin C Y. Genetic search strategies in multicriterion optimal design[J]. Structural Optimization, 1992, 4: 99-107

[4] Ishibuchi H, Murata T. Multi-objective Genetic Local Search Algorithm and its Application to Flowshop Scheduling[J]. IEEE Trans. Syst. Man Cybern. C, Aug. 1998, 28: 392-403

[5] Jaszkiewicz A. On the performance of Multiple - objec Genetic Local Search on the 0/1 Kna psack Problem- A Comparative Experiment[J]. IEEE Transaction on Evolutionary Computation, 2002, 6: 402-412

[6] Fonseca C M, Fleming P J. Genetic algorithms for Multiobjective optimization: Formulation, discussion and generalization[C]// Proceedings of the 5th International Conference on Genetic Algorithms. San Mateo, California, 1993

[7] Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms[R]. Dept. Mechanical Engineering, Kanput, India, 1993

[8] Deb K, Agrawal S, Pratap A, et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA II[M]. Parallel Problem Solving from Nature (PPSN VI), Berlin, 2000

[9] Horn J, Nafpliotis N, Goldberg D E. A niched Pareto genetic algorithm for multiobjective optimization[C]// IEEE World Congress on Computation. Piscataway, NJ, 1994

[10] Horn, Jeffrey, Nafpliotis N. Multiobjective Optimization using the Niched Pareto Genetic Algorithm [R]. IlliGAL Report 93005. University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993

[11] Zitzler E, Laumanns M, Thiele L. Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization[C]// EUROGEN 2001, Evolutionary Methods for Design, Optimization and Contorl with Applications to Industrial Problems. Septermber 2001

[12] Knowles J D, Corne D W. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimization[C]// Congress on Evolutionary Computation (CEC99). Piscataway, NJ, 1999

[13] Laumanns M, Zitzler E, Thiele L. On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization[C]// Evolutionary Multi-Criterion Optimization (EMO 2001). 2001

[14] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and strength Pareto approach[J]. IEEE Transactions on Evolutionary Computation, 1999, 3: 257-271

[15] Laumanns M, Zitzler E, Thiele L. On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization[C]// Evolutionary Multi-Criterion Optimization (EMO 2001). 2001

[16] Fonseca C M, Fleming P J. An overview of evolutionary algorithms in multi-objective optimization[J]. Evolutionary Computation, 1995, 3: 1-16

[17] 崔逊学, 李森, 方廷健. 多目标协调进化算法研究[J]. 计算机学报, 2001, 24(9): 979-984

[18] Cui Xun-Xue, Lin Chuang. A Preference-Based Multi-Objective Concordance Genetic Algorithm[J]. Journal of Software, 2005, 16(05): 761-770

[19] Veldhuizen D A V, Lamont G B. Evolutionary Computation and Convergence to a Pareto Front[C]// Late Breaking Papers at the Genetic Programming 1998 Conference. Stanford University, California, 1998

[20] Back T. Evolutionary Algorithms in Theory and Practice[M]. NewYork: Oxford University Press, 1996

(上接第 105 页)

[3] Xun Yi, Chik How Tan, Chee Kheong Siew. A New Block Cipher Based on Chaotic Tent Maps [J]. IEEE Trans. Circuits and Systems, 2002, 49(12): 1826-1829

[4] Jakimoski G, Kocarev L. Chaos and cryptography: Block encryption ciphers based on chaotic maps [J]. IEEE Trans. Circuits Syst, 2001, 48(2): 163-169

[5] Stojanovski T, Kocarev L. Chaos-based random number generators—Part I: Analysis [J]. IEEE Trans. Circuits Syst, 2001, 48 (3): 281-288

[6] Stojanovski T, Kocarev L. Chaos-based random number generators—Part II: Practical realization [J]. IEEE Trans. Circuits Syst, 2001, 48(3): 382-385

[7] Wong W K, Lee L P, Wong K W. A modified chaotic cryptographic method [J]. Comput Phys Commun, 2000, 138: 234-236

[8] Li Shujun, Mou Xuanqin, Cai Yuanlong. Pseudo - random Bit Generator Based on Couple Chaotic Systems and Its Applications in Stream-Cipher Cryptography[C]// Progress in Cryptology-IndoCrypt 2001. Lecture Notes in Computer Science. 2247, Dec. 2001: 316-329

[9] Wheeler D D. Problems with chaotic cryptosystems [J]. Cryptologia, 1989, 13(3): 243-250

[10] Wheeler D D, Mathews R A J. Supercomputer investigations of a chaotic encryption algorithm [J]. Cryptologia, 1991, 15(2): 140-152

[11] Wei Jun, Liao Xiaofeng, Wong Kwok-wo, et al. A new chaotic cryptosystem[J]. Chaos, Solitons and Fractals, 2006, 30: 1143-1152

[12] Goldberg D, Priest D. What every computer scientist should know about floating-point arithmetic [J]. ACM Comp. Surv, 1991, 23(1): 5-48

[13] Fridrich J. Symmetric Ciphers Based on Two-dimensional Chaotic Maps [J]. Int. J. Bifurcat Chaos, 1998, 8(6): 1259-84

[14] Knuth D E. Seminumerical algorithms[M]. The Art of Computer Programming 3rd edition. Reading, MA: Addison Wesley, 1998, 2

[15] Kohda T, Tsuneda A. Statistics of Chaotic Binary Sequences [J]. IEEE Transactions on information theory, 1997, 43(1): 104-112

[16] 陈勇源. 一种抗掩密分析的图像 LSB 掩密算法[J]. 重庆邮电大学学报: 自然科学版, 2008, 20(6): 758-762