第34卷 第14期 计 算 机 工 程 2008年7月
Vol.34 No.14 Computer Engineering July 2008
·安全技术· 文章编号：1000—3428(2008)14—0155—03 文献标识码：A 中图分类号：TP309.2

基于混沌系统的对称图像加密方案

颜世银，钱海峰，李志斌
(华东师范大学计算机科学技术系，上海 200062)

摘 要：采用基于外部密钥的复合混沌系统对数字图像进行对称加密，由外部密钥产生混沌系统的初始值。复合混沌系统由2个Logistic系统构成，一个用来产生置换矩阵，对图像进行像素置换操作，另一个用来产生灰度扰乱序列。在对灰度扰乱过程中，使用基于外部密钥导出的子密钥对灰度扰乱序列进行采样。在置换过程中，通过加入置换矩阵信息，使位置置换和像素替代过程以简单有效的方式复合，提高系统的耦合性。

关键词：复合混沌系统；图像加密；动态采样；Logistic映射

Symmetric Image Encryption Scheme Based on Chaotic System

YAN Shi-yin, QIAN Hai-feng, LI Zhi-bin
(Department of Computer Science, East China Normal University, Shanghai 200062)

【Abstract】This paper presents a new image encryption scheme based on composite chaotic system. An external key and two logistic systems are employed for encryption, which shuffle positions and substitute grey values of pixels of image. In the process of grey value substituting, it adopts sub key induced by external key to take samples. Through adding information of shuffle position matrix into the process of grey value substituting, processes of shuffle positions and substitute grey values are mixed in a simple and effective way, which improves coupling property of the encryption system.

【Key words】composite chaotic system; image encryption; dynamic sample; Logistic maping

1 概述

计算机网络技术的飞速发展以及多媒体技术的广泛应用使越来越多的数字图像通过各种媒介传输。很多应用(如网络个人相册、医学图像系统)需要加密敏感的图像数据。但是数字图像数据和一般的文本数据有很多不同，如数据量大、相邻像素相关性强，导致传统加密方法(如DES, RSA)并不适合直接对图像加密。近年来，基于混沌理论的数字图像加密方法受到重视并得到广泛的研究。

基于混沌的图像加密技术主要有像素替代、位置置换以及2种技术的结合。像素替代：混沌序列作为密钥和图像每个像素执行逐位的XOR或者XNOR操作，经证明，其抵御已知/选择明文攻击的性能很差。位置置换：利用混沌映射转换像素位置，不改变灰度统计特性，可以基于统计信息进行攻击。2种技术的结合能有效抵御上述攻击。但在某些方案中，位置置换过程和像素替代过程只存在时间上的先后关系，耦合性差，仍存在被攻击的可能[1]。另外，把混沌系统初始值和系统参数直接作为密钥存在缺陷[2]，因此，外部密钥(external key)被引入到加密系统中[3-4]。

针对一维混沌系统可以通过相空间重构进行识别的问题，一些研究者采用演化规律更复杂的高维混沌系统改善安全性[5]，但需要对高维连续系统进行离散化，在一定程度上影响了效率。文献[6]提出了取样一维混沌序列来改善安全性。本文提出一个基于复合混沌系统的图像加密方案。通过位置置换和像素替代，采用外部密钥和2个Logistic混沌系统对图像进行加密。数值实验和安全性分析表明，该加密方案密钥空间大、密钥敏感性高、扩散性和扰乱性能良好、容易实现且安全性能优越。

2 加密方案

2.1 相关介绍

(1)外部密钥：采用长度为80 bit的密钥[4]：

$$K = k_1 k_2 \cdots k_{20} \tag{1}$$

其中，字符 k_i $(i = 1, 2, \cdots, 20)$ 为十六进制。每2个字符(即每8 bit)构成一个子密钥 K_i $(i = 1, 2, \cdots, 10)$。于是，外部密钥表示为子密钥形式为

$$K = K_1 K_2 \cdots K_{10} \tag{2}$$

(2)混沌系统原型：本文采用的Logistic混沌系统原型为

$$x_{n+1} = \mu x_n (1 - x_n) \tag{3}$$

其中，x_n 为映射变量，$0 \le x_n \le 1$；μ 为系统参数，$0 \le \mu \le 4$。当3.569 945 6$\cdots \le \mu \le 4$时，系统进入混沌状态。

(3)本文的混沌系统：采用2个Logistic混沌系统，μ 均取为3.999 9。位置置换过程采用的混沌系统 X 为

$$X_{n+1} = 3.999\ 9 X_n (1 - X_n) \tag{4}$$

系统初值的计算如下：

1)利用式(5)计算得到实数 X_{01}：

$$X_{01} = \frac{\mathrm{mod}(\sum_{i=1}^{10} (K_i)_{10}, 1\ 000)}{1\ 000} \tag{5}$$

其中，$(K_i)_{10}$ 表示 K_i 的十进制数值。

2)选取子密钥 K_4, K_5, K_6，利用式(6)计算得到另一个具有24位精度的实数 X_{02}：

作者简介：颜世银(1983－)，男，硕士研究生，主研方向：信息安全技术；钱海峰，副教授；李志斌，教授、博士生导师
收稿日期：2007-08-10 **E-mail：**ysyaaa@163.com

$$X_{02}=(K_{41}\times 2^0+K_{42}\times 2^1+\cdots+K_{48}\times 2^7+K_{51}\times 2^8+K_{52}\times 2^9+\cdots+K_{58}\times 2^{15}+K_{61}\times 2^{16}+K_{62}\times 2^{17}+\cdots+K_{68}\times 2^{23})/2^{24} \quad (6)$$

其中， K_{ij} 表示 K_i 的第 j bit 位上的值(0 或 1)。

3)利用式(7)计算系统初值 X_0 ：

$$X_0=(X_{01}+X_{02})-floor(X_{01}+X_{02}) \quad (7)$$

其中， $floor(\cdot)$ 表示不大于操作数“ · ”的最大整数。式(7)的目的是只取 $X_{01}+X_{02}$ 的小数部分。

通过上述几步构造的用于位置置换的混沌系统初值，使初值对外部密钥足够敏感，且精度得到保证。

灰度替代过程采用的 Logistic 系统 Y 为

$$Y_{n+1}=3.999\ 9Y_n(1-Y_n) \quad (8)$$

初值 Y_0 产生的方式和 X_0 相似，不同之处在于，计算 Y_{02} 时选取子密钥 K_7,K_8,K_9 。

2.2 算法描述

本文用 $\boldsymbol{M}$ 表示明文灰度图像，大小为 $m\times n$ ， $\boldsymbol{M}(x,y)$ $(1\le x\le m,1\le y\le n)$ 表示位置 (x,y) 处的灰度值。用 $\boldsymbol{C}$ 表示密文图像。

2.2.1 位置置换过程

根据图像 $\boldsymbol{M}$ 的大小产生 $m\times n$ 长度的混沌序列 $\{X_1,X_2,X_3,\cdots,X_{m\times n}\}$ ，用该序列按照列优先的方式构造 $m\times n$ 的矩阵 $\boldsymbol{E}$。对 $\boldsymbol{E}$ 的元素进行排序后，按照列优先的位置重新放置，启用位置矩阵 $\boldsymbol{L}$，在 $\boldsymbol{L}$ 对应的位置依次记录排序后每个矩阵元素在原始矩阵 $\boldsymbol{E}$ 中的位置。然后把图像 $\boldsymbol{M}$ 中的像素按照 $\boldsymbol{L}$ 中的位置重新排列得到 $\boldsymbol{M}'$ ，实现置换过程。

比如：

$$\boldsymbol{E}=\begin{bmatrix}0.174\ 7 & 0.092\ 7 & 0.383\ 9 & 0.651\ 4\\ 0.576\ 6 & 0.336\ 2 & 0.945\ 8 & 0.908\ 1\\ 0.976\ 3 & 0.892\ 4 & 0.204\ 9 & 0.333\ 9\end{bmatrix}\longrightarrow\boldsymbol{L}=\begin{bmatrix}4 & 12 & 2 & 11\\ 1 & 5 & 10 & 8\\ 9 & 7 & 6 & 3\end{bmatrix}$$

$$\boldsymbol{M}=\begin{bmatrix}m_{11} & m_{12} & m_{13} & m_{14}\\ m_{21} & m_{22} & m_{23} & m_{24}\\ m_{31} & m_{32} & m_{33} & m_{34}\end{bmatrix}\xrightarrow{L}\boldsymbol{M}'=\begin{bmatrix}m_{12} & m_{34} & m_{21} & m_{24}\\ m_{11} & m_{22} & m_{14} & m_{23}\\ m_{33} & m_{13} & m_{32} & m_{31}\end{bmatrix}$$

2.2.2 像素替代过程

(1)确定采样次数 T。如果每个采样因子作用于混沌序列的次数 T 为 16，则一个采样因子对原始序列采样 16 次。

(2)构造动态采样因子 S_i ：

$$S_i=((K_i)_{10}\bmod N)+1,\quad i=1,2,\cdots,9 \quad (9)$$

分别利用整数 $S_1,S_2,S_3,\cdots,S_9$ 对原始混沌序列进行 T 次采样(具体见步骤(5))。这里的 N 为整数常量，如 3, 4, 5，取值不易过大，否则影响效率。

(3)通过式(10)改变子密钥的值：

$$(K_i')_{10}=((K_i)_{10}+(K_{10})_{10}+round(mean(K_1,K_2,\cdots,K_{10})))\bmod 256,\quad i=1,2,\cdots,9 \quad (10)$$

其中， $round(mean(K_1,K_2,\cdots,K_{10}))$ 是常量，通过对初始子密钥十进制平均值取整得到。用改变后的密钥构造新的采样因子。

(4)反复执行步骤(2)、步骤(3)，直到产生数量为 $m\times n/t$ 的采样因子。

(5)对原始混沌序列采样，得到采样后的序列 $\{Y_1',Y_2',\cdots,Y'_{m\times n}\}$。本方案要把采样后序列的元素映射为[0, 255]内的整数，为得到均匀分布在[0, 255]内的整数序列，只在原始序列中选择处于[0.2, 0.8]区间上的值。由 Y_0 迭代得到 $Y_1'\in[0.2,0.8]$。用式(11)进行采样：

$$Y_{i+1}'=f^{(n)}(Y_i'),\quad i=1,2,\cdots,m\times n-1 \quad (11)$$

其中， $n\ge s$ 且 $Y_{i+1}'\in[0.2,0.8]$， s 代表当前使用的采样因子； $f^{(n)}(\cdot)$ 表示对混沌方程迭代 n 次。式(11)保证混沌方程由当前值 Y_i' 迭代大于等于 s 次后得到第 1 个属于区间[0.2, 0.8]的值作为 Y_{i+1}'。

(6)利用式(12)将 $\{Y_1',Y_2',\cdots,Y'_{m\times n}\}$ 映射到区间[0.2, 0.8]上，得到 $\{Y_1'',Y_2'',\cdots,Y''_{m\times n}\}$ ：

$$Y_i''=\mathrm{mod}(round(\frac{(Y_i'-0.2)\times 255}{0.8-0.2}),\ 256) \quad (12)$$

其中， $i=1,2,\cdots,m\times n$。

(7)由式(13)、式(14)进行像素灰度替代：

$$C(1)=(L(1)\bmod 256)\oplus(\sum_{i=1}^{10}(K_i)_{10}\bmod 256)\oplus((M'(1)+Y_1'')\bmod 256) \quad (13)$$

$$C(i)=(L(i)\bmod 256)\oplus((M'(i)+Y_i'')\bmod 256)\oplus\ C(i-1) \quad (14)$$

其中， $i=2,3,\cdots,m\times n$ ； $L(i),M'(i)$ 表示按照列优先方式从矩阵 $\boldsymbol{L},\boldsymbol{M}'$ 取得的第 i 个元素。根据式(12)建立 $C(i)$ 与 $C(i-1)$ 的关系，以这种方式扩散明文到密文。在 $C(1)$ 中加入密钥信息，以增加密钥敏感性。这里的 $\oplus$ 指按位异或操作。最后用所有的 $C(i)$ 构造最终的加密矩阵 $\boldsymbol{C}$，也就是密文。

综上所述，整个加密过程如下：(1)基于外部密钥，构造 2 个混沌系统 X, Y 的初值。(2)根据混沌系统 X 迭代出的元素的大小，构造位置置换矩阵，对图像像素进行位置置换操作。(3)由外部密钥导出的子密钥动态采样混沌系统 Y 产生的原始序列，并将采样后的序列映射到[0, 255]的整数域上，结合外部密钥以及位置置换矩阵的信息，对置换后的图像执行灰度替代操作。

解密过程是加密过程的逆操作。对应加密公式(式(13)和式(14))，解密公式为

$$M'(1)=((L(1)\bmod 256)\oplus(\sum_{i=1}^{10}(K_i)_{10}\bmod 256)\oplus C(1)+256-Y_1'')\bmod 256 \quad (15)$$

$$M'(i)=((L(i)\bmod 256)\oplus C(i-1)\oplus C(i)+256-Y_i'')\bmod 256 \quad (16)$$

其中， $i=2,3,\cdots,m\times n$。

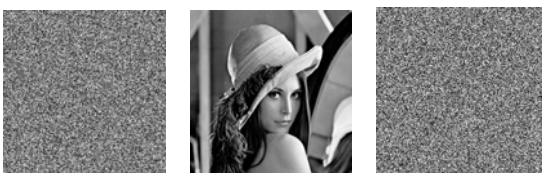
3 实验结果及分析

本文采用 Matlab 7.3 平台，选取 256×256 的 Lena 灰度图像(图 1)进行实验。式(9)中 N=3，采样次数 T=16。

图 1 明文图像

3.1 密钥敏感性分析

选择’A0C2EF52BBE5CA894405’作为外部密钥，图 2(a)、图 2(b)为加解密效果。用密钥’A0C2EF52BBE5CA894406’解密图 2(a)的效果如图 2(c)所示。



(a)密文图像 (b)解密图像 (c)错误解密图像

图 2 实验结果

表 1 给出了密文图像之间的相关系数，其中用于加密的密钥如下：

key1：‘A0C2EF52BBE5CA894405’

key2：‘A0C2EF52BBE5CA894406’

key3：‘A0C2EF53BBE5CA894405’

key4：‘B0C2EF52BBE5CA894405’

表 1　密文图像相关系数

密钥		相关系数
key1	key2	-0.002 979
key1	key3	-0.000 699
key1	key4	0.004 413
key2	key3	0.002 645
key2	key4	0.000 062
key3	key4	0.004 907

上述 4 个密钥之间都只有 1 位或 2 位不同，由相关系数考察密钥敏感性。相关系数定义如下：

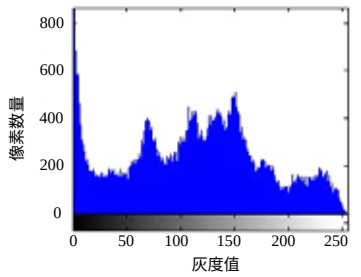
$$r_{xy}=\mathrm{cov}(x,y)\Big/\sqrt{D(x)D(y)} \tag{17}$$

其中，$\mathrm{cov}(x,y)=E((X-E(X))(Y-E(Y)))$ 为协方差；方差 $D(x)=E(x^2)-E^2(x)$，均值为 $E(x)$；x,y 分别代表 2 个密文图像同一位置上像素灰度值。从中可以看出，对应位置像素的相关性可视为 0，表明密文对密钥具有高度的敏感性。

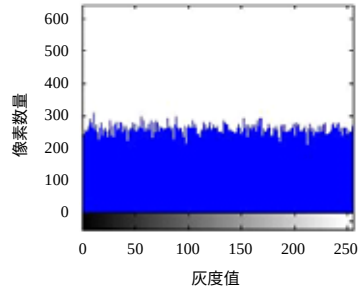
3.2　统计特性分析

以密钥“ A0C2EF52BBE5CA894405 ”加密的结果为例分析统计特性。

(1)灰度直方图。图 3(a)是原始图像(图 1)的像素灰度直方图，图 3(b)则对应密文图像(图 2(a))。可以看出，明文的像素值统计信息在密文图像中没有体现，明密文之间统计上的相关性非常小，表明方案能够有效抵御基于像素值统计的攻击。



(a)明文图像



(b)密文图像

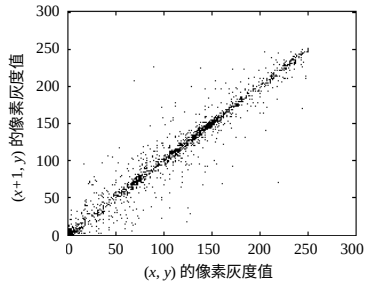
图 3　图像灰度直方图

(2)相邻像素相关性。比较明密文图像水平和垂直方向相邻像素相关性。利用式(17)计算相关系数，其中的 x, y 分别代表图像内部相邻位置的像素灰度值。表 2 为实验结果。

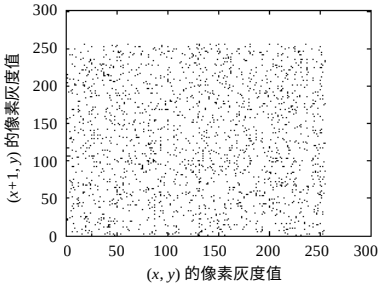
表 2　图像内部相邻像素相关系数

	原始图像	加密图像
水平	0.969 3	-0.002 7
垂直	0.906 2	0.002 5

图 4、图 5 分别描述了在图像中随机选取 2 000 个点水平和垂直方向的相关性。

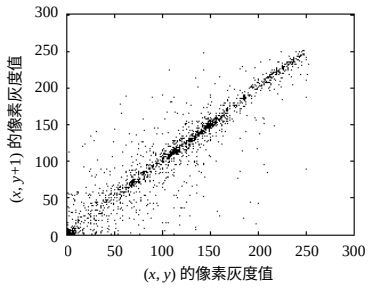


(a)明文图像

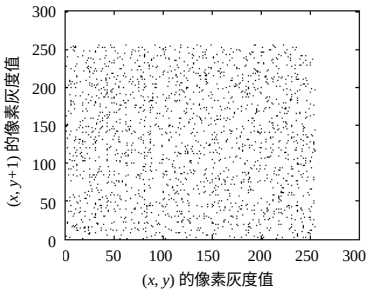


(b)密文图像

图 4　水平方向相邻像素的相关性



(a)明文图像



(b)密文图像

图 5　垂直方向相邻像素的相关性

结果表明，原始图像中相邻像素之间存在很高的相关性，而加密图像相邻像素之间的相关性可以忽略为 0。

分析像素灰度直方图和相邻像素相关性得出结论：明文图像统计结构已经扩散到密文中，方案有很好的扩散性能。

3.3　密钥空间分析

本文的方案采用的密钥为 80 bit，密钥空间可达 2^{80} ($\approx 1.208\ 9\times10^{24}$)，加密系统的熵为 80，密钥空间大小足够抵御穷举攻击。只需要修改算法某些细节，就可以使用更长的密钥，如 128 bit，容易实现且不会影响算法效率。

4　结束语

本方案使用外部密钥构造混沌系统初值，导出子密钥，

(下转第 160 页)