对一类超混沌图像加密算法的密码分析与改进*

朱从旭† 孙克辉

(中南大学信息科学与工程学院, 长沙 410083)

(2011 年 10 月 26 日收到; 2011 年 11 月 16 日收到修改稿)

对一种超混沌图像加密算法及其改进版进行了安全性分析, 结果表明该类算法的置乱过程都是与混淆过程相脱离的, 且混淆过程的加密公式简单; 因此都不能抵抗选择明文和选择密文攻击. 进而提出了一种改进的加强型超混沌图像加密算法; 改进算法包含两轮像素值替代加密操作, 并使得密文与明文、密钥之间的关系更复杂. 安全性分析和实验测试表明, 改进算法不仅克服了原算法不能抵御选择明文和选择密文攻击的缺陷; 而且具有时间开销更小和抗差分攻击性能更好的优势.

关键词: 超混沌, 图像加密, 选择明文攻击, 选择密文攻击

PACS: 05.45.Gg

1 引 言

混沌一直是非线性科学领域非常活跃的研究对象 [1−4]. 由于混沌信号具有对初始条件的极端敏感性、无周期性、伪随机性等复杂特性, 与保密通信及密码学之间有着天然的联系; 于是人们联想到利用混沌信号进行信息加密. 自 Matthews 提出混沌加密的思想以来 [5], 有关混沌密码和混沌保密通信方案的研究逐步成了国际电子通信领域的研究热点 [6,7]. 由于混沌密码与传统密码的密钥产生机理不同, 使混沌密码在加密实时性方面具有更大的优势. 特别是对于图像、视频等多媒体数据来说, 由于这类信息的数据量大, 且相邻的数据之间具有很强的相关性; 导致传统密码学对于诸如图像信息的加密显得效率低、不能满足实时性需要. 而混沌密码却正好在大数据量信息加密场合具有优越性. 因此, 关于混沌图像加密方法的研究备受人们关注 [8−10].

众所周知, 一种好的密码算法应该具有足够大的密钥空间以便抵抗强力攻击; 应该对明文和密钥都极端敏感, 以便很好地抵抗差分攻击; 密文分布应该随机均匀、相邻数据不相关, 以便抵抗统计分析. 虽然低维混沌系统由于形式简单而具有计算时间开销小的优点; 但由于其密钥空间小, 序列的复杂度不高, 导致密码系统安全性不高. 而高维混沌系统尤其是超混沌系统, 由于密钥空间大、具有两个以上正的 Lyaponuv 指数、更复杂和难以预测的非线性行为, 使用超混沌系统加密数据无疑会提高密码系统的安全性. 正因为超混沌系统在保密通信中的巨大应用潜力, 从而引起学者们对超混沌系统的建模研究发生浓厚兴趣 [11]. 随着现代计算机系统性能的不断提高, 研究超混沌图像加密算法更具现实性. 文献 [12] 提出了一种基于超混沌系统的图像加密算法 (本文称之为 HIE 算法), 但文献 [12] 算法的加密密钥与待加密图像的明文无关, 显然不能抵抗选择明文攻击. 最近, 王静等人 [13] 在分析文献 [12] 算法安全性缺陷的基础上提出了一种改进型超混沌图像加密算法 (本文称之为 IHIE 算法), 该算法设计了使加密密钥与明文相关的策略, 因此能提高抵抗选择明文攻击的能力, 并且具有思路清晰的优点. 但该算法总体上还是沿用了文献 [12] 的

* 国家自然科学基金 (批准号: 61073187, 61161006), 湖南省自然科学基金 (批准号: 10JJ6093) 和广东省自然科学基金 (批准号: S2011010001581) 资助的课题.

† E-mail: zhucx@csu.edu.cn

像素位置置乱和像素值替代加密的基本结构. 而像素值加密阶段只有一轮替代操作, 使得一个明文像素值的变化只能影响该像素后面的密文像素值; 且密文和密钥及明文之间的关系不够复杂. 因此, 其最终加密密钥序列同样可以破解. 为了获得高安全与高效率的图像加密方案, 本文在分析该类算法共同缺陷的基础上, 提出了改进的加强型超混沌图像加密算法, 其核心思想是: 使得加密公式的密文和密钥、明文之间的关系更复杂化; 且替代加密两轮. 这样, 攻击者凭选择的明、密文对很难解出加密的中间密钥; 同时两轮替代扩散操作增强了密文对明文的敏感性. 且改进算法省略了像素置乱操作, 能提高加密速度. 理论分析和仿真实验表明, 改进算法不仅可有效地抵抗选择明 (密) 文攻击, 同时在抗差分攻击能力和加密速度等方面都具有更好的性能.

2 一类超混沌密码算法概述

2.1 HIE 算法

文献 [12] 提出的典型超混沌密码算法包括对图像的像素位置置乱和对像素值替代变换两个部分. 其基本思想如下:

在图像置乱过程中, 先由一维的 Logistic 混沌映射迭代生成两个整数编号序列 $\boldsymbol{r}=\{r_i, i=1,2,\cdots,M\}$ 和 $\boldsymbol{c}=\{c_i, i=1,2,\cdots,N\}$, 其中 r_i 是 [1, M] 范围的整数、c_i 是 [1, N] 范围的整数; M 和 N 分别是待加密图像的像素行数、列数. 然后, 利用行、列编号序列 $\boldsymbol{r}$ 和 $\boldsymbol{c}$ 先后对图像实施行、列置乱. 若原始图像、行置乱图像和行列均置乱后的图像矩阵分别用 $\boldsymbol{P}$, $\boldsymbol{P}^{\rm r}$ 和 $\boldsymbol{P}^{\rm rc}$ 表示, 则行置乱操作是将原图像 $\boldsymbol{P}$ 的第 r_i 行移到新图像 $\boldsymbol{P}^{\rm r}$ 的第 i 行; 列置乱操作是将经行置乱后的图像 $\boldsymbol{P}^{\rm r}$ 的第 c_i 列移到新图像 $\boldsymbol{P}^{\rm rc}$ 的第 i 列. 完成所有行、列置乱操作后即得到置乱图像 $\boldsymbol{P}^{\rm rc}$.

在图像像素值替代过程中, 先用超混沌陈系统迭代生成四个混沌实数值 x_j, $j=1,2,3,4$; 然后将实数的 x_j 经下式改造成 [0, 255] 范围的整数:

$$x_j=\mathrm{mod}((|x_j|-\mathrm{floor}(|x_j|))\times 10^{14},256). \quad (1)$$

式中, $|x|$ 表示取 x 的绝对值; $\mathrm{mod}(x,y)$ 表示取 x 除以 y 后的余数. 接着再按下式计算得到一个 [0, 3] 范围的整数 $\bar{x}_1$,

$$\bar{x}_1=\mathrm{mod}(x_1,4). \quad (2)$$

然后, 根据 $\bar{x}_1$ 的值不同选择 x_i 中不同的三个数构成一组三元密钥 $\boldsymbol{BX}=\{BX_1, BX_2, BX_3\}$, 若 $\bar{x}_1=0$, 则 $\boldsymbol{BX}=\{x_1, x_2, x_3\}$; 若 $\bar{x}_1=1$, 则 $\boldsymbol{BX}=\{x_1, x_2, x_4\}$; 若 $\bar{x}_1=2$, 则 $\boldsymbol{BX}=\{x_1, x_3, x_4\}$; 若 $\bar{x}_1=3$, 则 $\boldsymbol{BX}=\{x_2, x_3, x_4\}$. 并使用该三元密钥组加密图像 $\boldsymbol{P}^{\rm rc}$ 中 3 个连续的像素. 加密公式为

$$\begin{aligned} C_{3(i-1)+1}&=P^{\rm rc}_{3(i-1)+1}\oplus BX_1,\\ C_{3(i-1)+2}&=P^{\rm rc}_{3(i-1)+2}\oplus BX_2,\\ C_{3(i-1)+3}&=P^{\rm rc}_{3(i-1)+3}\oplus BX_3, \end{aligned} \quad (3)$$

其中, $\oplus$ 表示按二进制位进行的异或运算. $i=1,2,3,\cdots$ 表示迭代轮数, 每一轮迭代得到 3 个密钥并加密 3 个像素; 反复迭代超混沌系统直到图像所有像素加密完毕, 就得到最终加密图像 $\boldsymbol{C}$.

由上述过程可知, 该算法的加密密钥 $\{BX_1, BX_2, BX_3\}$ 与明文无关, 攻击者只要已知一组明、密文对 $\boldsymbol{P}^{\rm rc}$ 和 $\boldsymbol{C}$, 根据异或运算 $C(i)=P^{\rm rc}(i)\oplus B(i)$ 三个量的变换关系, 就很容易破解出加密密钥序列 $\boldsymbol{B}$: $B(i)=P^{\rm rc}(i)\oplus C(i)$. 此外, 由于像素位置置乱和像素值替代变换两个过程独立, 也很容易由一些特殊的明、密文对破解出行列置乱序列 $\boldsymbol{r}$ 和 $\boldsymbol{c}$. 一旦替代加密序列 $\boldsymbol{B}$ 和行列置乱序列 $\boldsymbol{r}$, $\boldsymbol{c}$ 均被破解后, 即可利用它们破解别的密文.

2.2 IHIE 算法

王静与蒋国平 [13] 最近提出的改进型超混沌图像加密算法是针对文献 [12] 算法的改进版, 可简称为 IHIE 算法. IHIE 算法也是由置乱与替代两个过程组成, 它对 HIE 算法作了两处重要改进, 其一是生成置乱过程的密钥序列不用一维混沌映射, 而改用超混沌系统; 其二是像素值替代操作阶段的加密密钥序列改成与待加密图像明文相关, 这样可以增强抗选择明 (密) 文攻击的能力. 文献 [13] 两类操作过程中都是使用方程

$$\begin{aligned} \dot{x}_1&=a(x_2-x_1)+x_4,\\ \dot{x}_2&=dx_1-x_1x_3+cx_2,\\ \dot{x}_3&=x_1x_2-bx_3,\\ \dot{x}_4&=x_2x_3+ex_4 \end{aligned} \quad (4)$$

所描述的超混沌系统 [14] 产生密钥. (4) 式中, a, b, c, d 和 e 为系统参数, 当 $a=35$, $b=3$, $c=12$, $d=7$, e 处于 (0.085, 0.798) 区间时, 系统 (4) 是超混沌的. 下面先给出文献 [13] 算法的简要描述, 然后再分析其安全性.

设大小为 $M\times N$ 的明文 (plaintext) 图像像素矩阵记为 $\boldsymbol{P}$, 明文经行 (row) 置乱后的像素矩阵记为 $\boldsymbol{P}^{\rm r}$, 明文经列 (column) 置乱后的像素矩阵记为 $\boldsymbol{P}^{\rm c}$, 明文经行、列两重置乱后的像素矩阵记为 $\boldsymbol{P}^{\rm rc}$, 最终加密图像矩阵记为 $\boldsymbol{C}$. 文献 [13] 利用 Runge-Kutta 算法将超混沌系统预迭代 N_0 次 (文中取 $N_0=200$), 并丢弃前 N_0 次迭代产生的数据用于防止过渡效应带来的有害影响. 系统 (4) 每迭代一次时, 由第一个状态变量 x_1 的值计算如下 r 值:

$$r=\mathrm{mod}((|x_1|-\mathrm{floor}(|x_1|))\times 10^{14},M), \tag{5}$$

式中, floor(x) 表示取小于或等于 x 的最大整数. 显然, $r\in[0,M-1]$. 反复迭代超混沌系统直到产生 M 个完全不同的 r 值, 记为序列 $\boldsymbol{r}=\{r_i,i=0,1,\cdots,M-1\}$. 根据序列 $\boldsymbol{r}$ 对矩阵 $\boldsymbol{P}$ 进行行置换, 行置换后所得的结果像素矩阵记为 $\boldsymbol{P}^{\rm r}$, 则 $\boldsymbol{P}^{\rm r}$ 矩阵中的元素与 $\boldsymbol{P}$ 矩阵中的元素有如下位置对应关系:

$$P^{\rm r}(i,j)=P(r_i,j). \tag{6}$$

同理, 系统 (4) 每次迭代时也由 x_2 计算如下 c 值:

$$c=\mathrm{mod}((|x_2|-\mathrm{floor}(|x_2|))\times 10^{14},N), \tag{7}$$

其中, $c\in[0,N-1]$. 反复迭代系统 (4) 直到产生 N 个完全不同的 c 值, 记为序列 $\boldsymbol{c}=\{c_j,j=0,1,\cdots,N-1\}$. 根据 $\{c_j,j=0,1,\cdots,N-1\}$ 对行置换矩阵 $\boldsymbol{P}^{\rm r}$ 进行列置换, 置换后的矩阵记为 $\boldsymbol{P}^{\rm rc}$, 则矩阵 $\boldsymbol{P}^{\rm rc}$ 与矩阵 $\boldsymbol{P}^{\rm r}$ 中的元素对应关系为

$$P^{\rm rc}(i,j)=P^{\rm r}(i,c_j). \tag{8}$$

结合 (6) 和 (8) 式, 不难推出 $\boldsymbol{P}^{\rm rc}$ 矩阵中的元素与原始图像矩阵中元素的关系是

$$P^{\rm rc}(i,j)=P(r_i,c_j). \tag{9}$$

接下来, 使用超混沌系统产生的混沌序列对置乱后的图像 $\boldsymbol{P}^{\rm rc}$ 进行像素值替代加密. 首先按照

$$x_i=\mathrm{mod}((|x_i|-\mathrm{floor}(|x_i|))\times 10^{14},256) \tag{10}$$

将每一轮迭代所得的 4 个混沌实数状态值改造为 [0,255] 范围的整数密钥. (10) 式中, $i=1,2,3,4$. 并按下式计算 $\bar{x}_1$:

$$\bar{x}_1=\mathrm{mod}((x_1+x_2+x_3+x_4),4), \tag{11}$$

即 $\bar{x}_1$ 是 [0, 3] 范围的整数. 每次 (第 i 次) 迭代根据 $\bar{x}_1$ 选择一组不同混沌密钥组合构成一个三元密钥组 $\boldsymbol{B}=(BX_1,BX_2,BX_3)$: 若 $\bar{x}_1=0$, 选择 $\boldsymbol{B}=\{x_1,x_2,x_3\}$; 若 $\bar{x}_1=1$, 选择 $\boldsymbol{B}=\{x_1,x_2,x_4\}$; 若 $\bar{x}_1=2$, 选择 $\boldsymbol{B}=\{x_1,x_3,x_4\}$; 若 $\bar{x}_1=3$, 选择 $\boldsymbol{B}=\{x_2,x_3,x_4\}$. 每次用一组密钥对图像中 3 个连续的像素进行加密, 加密公式为

$$\begin{aligned} C_{3(i-1)+1}&=P^{\rm rc}_{3(i-1)+1}\oplus K_1,\\ C_{3(i-1)+2}&=P^{\rm rc}_{3(i-1)+2}\oplus K_2,\\ C_{3(i-1)+3}&=P^{\rm rc}_{3(i-1)+3}\oplus K_3, \end{aligned} \tag{12}$$

其中, (K_1,K_2,K_3) 为每次迭代对应的最终加密密钥, 它们与每次迭代所得的三元混沌密钥组 (BX_1, BX_2, BX_3) 有关; 并且还与前一次所加密的 3 个像素的输出密文值有关, (K_1, K_2, K_3) 的计算公为

$$\begin{aligned} K_1&=\mathrm{mod}(BX_1\oplus C_{3\times(i-1)},256),\\ K_2&=\mathrm{mod}(BX_2\oplus C_{3\times(i-1)+1},256),\\ K_3&=\mathrm{mod}(BX_3\oplus C_{3\times(i-1)+2},256), \end{aligned} \tag{13}$$

引入 (13) 式产生最终加密密钥正是文献 [13] 对文献 [12] 算法改进的主要地方. 在 (12) 和 (13) 式中, $i=1,2,\cdots,\mathrm{ceil}(M\times N)/3-1$ 表示第 i 次超混沌迭代. $P^{\rm rc}_1$, $P^{\rm rc}_2$, $\cdots$ 分别表示置乱图像的像素序列, 而 C_1, C_2, $\cdots$ 则分别表示最终加密图像的像素序列. 特别地, 在加密第 1 个像素时用到的 C_0 是一个由加密者预先指定的常数 (可以作为密钥).

事实上, (12) 和 (13) 式可以合并写成

$$C_j=P^{\rm rc}_j\oplus\{\mathrm{mod}[(B_j\oplus C_{j-1}),256]\}, \tag{14a}$$

(14a) 式中, $\{B_j|j=1,2,\cdots,M\times N\}=\boldsymbol{B}$, 此序列 $\boldsymbol{B}$ 就是由前述所有三元组混沌密钥 (BX_1,BX_2,BX_3) 连接成的总混沌密钥序列; 而 mod($B_j\oplus C_{i-1}$, 256) 即为第 j 个最终加密密钥 K_j, 序列 $\boldsymbol{K}=\{K_j|j=1,2,\cdots,M\times N\}$ 即为总的最终加密密钥序列. 对 $j=1,2,\cdots,M\times N$ 所对应的每个像素值 $P^{\rm rc}_j$ 都采用 (14a) 式进行加密变换后, 便得到最终加密图像 $\boldsymbol{C}$. 因为, $B_j\in[0,255]$, $C_{j-1}\in[0,255]$; 所以, $B_j\oplus C_{j-1}\in[0,255]$, 于是, mod($B_j\oplus C_{j-1}$, 256) $=B_j\oplus C_{j-1}$. 故 (14a) 式又可

以简化为

$$C_j = P_j^{\mathrm{rc}} \oplus B_j \oplus C_{j-1}, \tag{14b}$$

其中, $j = 1, 2, \cdots, M \times N$. 综上所述, 文献 [13] 算法核心思想主要体现在两个公式上: 即像素位置置乱 (9) 式和像素值替代加密 (14a) 式. (9) 式中的行列置乱序列 $\boldsymbol{r}$, $\boldsymbol{c}$ 只与混沌系统参数及初值有关; 而 (14a) 式的最终加密密钥 $K_j = B_j \oplus C_{i-1}$ 虽然与待加密明文 (前一点的密文) 有关, 但其中的混沌密钥序列 $\boldsymbol{B}$ 与加密图像明文无关. 因此, 若能破解 $\boldsymbol{r}$, $\boldsymbol{c}$ 和 $\boldsymbol{B}$ 序列, 就能解密所要破解的密文.

3 IHIE 算法的破译

根据 Kerckhoff 提出的现代密码学原理, 加密系统的算法可以公开, 其安全性应完全决定于密钥. 即密码分析者可以知道加密算法, 但不知道密钥. 对密码系统的攻击按难易程度有 4 个级别, 按照从难到易的顺序依次排列为: 唯密文攻击、已知明文攻击、选择明文攻击和选择密文攻击. 只有加密算法能够抵御所有的攻击, 才能认为该算法是安全的. 下面将利用选择明文攻击对文献 [13] 算法进行破译 (也可用选择密文攻击).

选择明文攻击的定义是: 攻击者有机会使用密码机, 因此可选择一些明文, 并产生相应的密文. 对于文献 [13] 算法进行选择明文攻击, 只要选择两组明、密文对即可 (在确保其中一组明文图像矩阵不出现重复像素值前提下). 假定选择的 2 组大小为 $M \times N$(与要破解的密文图像等大) 明文矩阵为 $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, 它们的数据矩阵形式分别为

$$\boldsymbol{P}_1 = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}, \tag{15a}$$

$$\boldsymbol{P}_2 = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ \cdots & \cdots & \cdots & \cdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{pmatrix}. \tag{15b}$$

并得到了 $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ 所对应的密文矩阵 $\boldsymbol{C}_1$, $\boldsymbol{C}_2$. 假定各明文矩阵和各密文矩阵的元素按行优先次序排成一维像素序列后分别用 $P_1(j)$, $P_2(j)$ 和 $C_1(j)$, $C_2(j)$ 的形式表示, 则根据文献 [13] 加密 (14b) 式, 两幅密文图像的像素与其明文图像像素、密钥的关系为

$$C_1(j) = P_1^{\mathrm{rc}}(j) \oplus B(j) \oplus C_1(j-1), \tag{16a}$$

$$C_2(j) = P_2^{\mathrm{rc}}(j) \oplus B(j) \oplus C_2(j-1), \tag{16b}$$

其中, $P_i^{\mathrm{rc}}(j)$ 表示第 i 个 ($i = 1, 2$) 选择明文矩阵 $\boldsymbol{P}_i$ 经行列置乱后所得矩阵序列化后的第 j 个元素. 若要破解密文矩阵 $\boldsymbol{C}$ 得到其相应的明文矩阵 $\boldsymbol{P}$, 可以通过下面的 4 个步骤实现. 即先后破解出像素替代加密的密钥序列 $\boldsymbol{B}$、行置乱序列 $\boldsymbol{r}$ 和列置乱序列 $\boldsymbol{c}$; 因为这三个序列与明文无关, 因此利用 $\boldsymbol{B}$, $\boldsymbol{r}$ 和 $\boldsymbol{c}$ 序列即可解密出密文矩阵 $\boldsymbol{C}$, 得到 $\boldsymbol{C}$ 对应的明文矩阵 $\boldsymbol{P}$.

第1步 破解像素替代密钥序列 $\boldsymbol{B}$. 因为第一幅选择明文图像由全零像素组成, 因此其行列置乱图像矩阵仍然为全零矩阵, 即对所有 j 都满足 $P_1^{\mathrm{rc}}(j) = 0$; 于是, (16a) 式简化为 $C_1(j) = B(j) \oplus C_1(j-1)$, 故可解出 $B(j)$ 为

$$B(j) = C_1(j) \oplus C_1(j-1), \tag{17}$$

其中, $j = 1, 2, \cdots, M \times N$; 而 $C_1(0)$ 为未知常数 C_0, 所以 $B(1)$ 无法求出, 但有 $B(1) \oplus C_0 = C_1(1)$.

第2步 破解出行、列置乱图像矩阵 $\boldsymbol{P}_2^{\mathrm{rc}}$. 首先由密文序列 $\boldsymbol{C}_2$, 借助已经破解的 $\boldsymbol{B}$ 序列, 解密出对应的置乱图像矩阵的所有像素 $P_2^{\mathrm{rc}}(j)$. 注意 $C_2(0)$ 也为同一未知常数 C_0, 且有 $B(1) \oplus C_0 = C_1(1)$, 所以根据 (16b) 式得出 $P_2^{\mathrm{rc}}(j)$ 为

$$P_2^{\mathrm{rc}}(j) = C_2(j) \oplus B(j) \oplus C_2(j-1), \tag{18a}$$

$$P_2^{\mathrm{rc}}(1) = C_2(1) \oplus C_1(1), \tag{18b}$$

其中, $j = 2, \cdots, M \times N$. 所有 $P_2^{\mathrm{rc}}(j)$ 求出后, 转化成二维矩阵即得到行列置乱图像 $\boldsymbol{P}_2^{\mathrm{rc}}(j)$.

第3步 破解行、列置乱序列 $\boldsymbol{r}$, $\boldsymbol{c}$. 在解出 $\boldsymbol{P}_2^{\mathrm{rc}}$ 之后, 将 P_2^{rc} 与 $\boldsymbol{P}_2$ 对比, 由于 $\boldsymbol{P}_2$ 中的像素无重复值, 于是按下列步骤 1) 到 4) 的操作算法即可求出行、列置乱序列 $\boldsymbol{r}$, $\boldsymbol{c}$:

1) 初始化, $i \leftarrow 1, j \leftarrow 1; n \leftarrow 0$.

2) 在 P_2^{rc} 中取位置坐标为 (i, j) 的像素 $P_2^{\mathrm{rc}}(i, j)$; 将此像素的值与原始图像矩阵中的元素逐个比较, 一定能在 $\boldsymbol{P}_2$ 的某个唯一的位置 (x_i, y_j) 处找到一个元素 $P_2(x_i, y_j)$, 满足关系 $P_2(x_i, y_j) = P_2^{\mathrm{rc}}(i, j)$; 则可知原始图像的第 x_i 行是被置换到 P_2^{rc} 中第 i 行, 原始图像的第 y_j 列是被置换到 P_2^{rc} 中第 j 列; 则执行操作: $r(i) \leftarrow x_i$, $c(j) \leftarrow y_j$; $n \leftarrow n + 1$.

3) 如果 $i < M$, 则 $i \leftarrow i+1$; 如果 $j < N$, 则 $j \leftarrow j+1$.

4) 如果 $n \leqslant \max(M,N)$, 则重复执行步骤 2) 与 3); 如果 $n > \max(M,N)$, 意味着所有 $r(i)$ 和 $c(j)$ 值已被求出, 则算法结束. 这里, $\max(M,N)$ 表示求 (M,N) 中的较大者.

第4步 解密待破解的密文矩阵 $\boldsymbol{C}$, 得出对应的明文经行列置乱后的数据矩阵 $\boldsymbol{P}^{\mathrm{rc}}$. 根据文献 [13] 加密算法, $\boldsymbol{B}$, $\boldsymbol{r}$ 和 $\boldsymbol{c}$ 序列只与混沌系统初值及参数有关, 而不随明文图像而变, 因此, 由加密 (14) 式和已经得到的 $\boldsymbol{B}$, 可以求出 $\boldsymbol{P}^{\mathrm{rc}}$ 的各个元素 $P^{\mathrm{rc}}(j)$ 为

$$P^{\mathrm{rc}}(j) = C(j) \oplus B(j) \oplus C(j-1),\ j > 1 \quad (19a)$$

$$P^{\mathrm{rc}}(1) = C(1) \oplus C_1(1),\ j = 1. \quad (19b)$$

对所有 j 值运用 (19a) 或 (19b) 式, 即可求出 $\boldsymbol{P}^{\mathrm{rc}}$ 的所有元素, 也就得到了矩阵 $\boldsymbol{P}^{\mathrm{rc}}$.

第5步 反行、列置乱, 破解出原始明文图像 $\boldsymbol{P}$. 先根据列置乱序列 $\boldsymbol{c}$, 由 $\mathbf{P}^{\mathrm{rc}}$ 进行反列置乱, 即可得到 $\boldsymbol{P}^{\mathrm{r}}$; 再根据行置乱序列 $\boldsymbol{r}$, 由 $\boldsymbol{P}^{\mathrm{r}}$ 进行反行置乱, 即可得到 $\boldsymbol{P}$.

为了直观地证明上述分析的合理性, 下面通过一组简单具体数据的仿真实验进行验证, 假设攻击者有机会使用文献 [13] 的加密机, 选择两组 3×3 明密文矩阵对, 明文矩阵分别为

$$\boldsymbol{PX} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\boldsymbol{PY} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix},$$

获得了对应的密文矩阵分别为

$$\boldsymbol{CX} = \begin{pmatrix} 214 & 152 & 198 \\ 146 & 134 & 209 \\ 234 & 57 & 33 \end{pmatrix},$$

$$\boldsymbol{C}Y = \begin{pmatrix} 222 & 151 & 192 \\ 150 & 131 & 215 \\ 233 & 62 & 32 \end{pmatrix},$$

现在要破译密文矩阵 $\boldsymbol{C}$, $\boldsymbol{C}$ 的元素按行优先顺序写成一维形式是: $\boldsymbol{C}$ = [211, 98, 5, 86, 3, 6, 110, 120, 114], 破解任务是求解出 $\boldsymbol{C}$ 所对应的明文矩阵 $\boldsymbol{P}$.

第1步 由第 1 组明、密文对 $\boldsymbol{PX}$ 和 $\boldsymbol{CX}$, 解出替代加密密钥序列 $\boldsymbol{B}$. 由 (17) 式, 按一维序列方式逐个求出 $\boldsymbol{B}$ 中的每个元素: $\boldsymbol{B}$ = [214, 152, 198, 146, 134, 209, 234, 57, 33] $\oplus$ $[C_0, 214, 152, 198, 146, 134, 209, 234, 57]$, 将两个序列中对应位置的数分别异或, 即可得到下列结果:

$[B(2),\ B(3),\ B(4),\ B(5),\ B(6),\ B(7),\ B(8),\ B(9)] = [78, 94, 84, 20, 87, 59, 211, 24]$; 而 $B(1) = 214 \oplus C_0$, 反过来有 $B(1) \oplus C_0 = 214$.

第2步 由第 2 组明密文对 $\boldsymbol{PY}$ 和 $\boldsymbol{CY}$, 破解出第 2 幅已知明文图像的行列置乱图像 $\boldsymbol{PY}^{\mathrm{rc}}$. 先由 (19a) 式求出 $\boldsymbol{PY}^{\mathrm{rc}}$ 中的后面 8 个元素, 得

$$\begin{aligned} PY^{\mathrm{rc}}(2:9) =& CY(2:9) \oplus B(2:9) \oplus CY(1:8) \\ =& [151, 192, 150, 131, 215, 233, 62, 32] \\ & \oplus [78, 94, 84, 20, 87, 59, 211, 24] \\ & \oplus [222, 151, 192, 150, 131, \\ & 215, 233, 62] \\ =& [7, 9, 2, 1, 3, 5, 4, 6]; \end{aligned}$$

再由 (19b) 式求出 $\boldsymbol{PY}^{\mathrm{rc}}$ 中的第 1 个元素 (注意: $CY(0) = CX(0) = C_0$, $B(1) \oplus C_0 = 214$)

$$\begin{aligned} PY^{\mathrm{rc}}(1) &= CY(1) \oplus B(1) \oplus CY(0) \\ &= 222 \oplus B(1) \oplus C_0 = 222 \oplus 214 = 8. \end{aligned}$$

将求出的像素值序列排列成二维矩阵形式, 即得

$$\boldsymbol{PY}^{\mathrm{rc}} = \begin{pmatrix} 8 & 7 & 9 \\ 2 & 1 & 3 \\ 5 & 4 & 6 \end{pmatrix}.$$

第3步 破解行、列置乱序列 $\boldsymbol{r}, \boldsymbol{c}$. 按照前述算法, 利用 $\boldsymbol{PY}^{\mathrm{rc}}$ 与原始明文矩阵 $\boldsymbol{PY}$ 的数据进行对比即可求出 $r(i)$ 和 $c(j)$; $i, j = 1, 2, 3$. 具体操作方法叙述如下: 将 $\boldsymbol{PY}^{\mathrm{rc}}$ 中位置 (1,1) 处的像素值 8 与 $\boldsymbol{PY}$ 的元素逐个比较, 可以在 $\boldsymbol{PY}$ 中唯一位置 (3,2) 处找到; 所以, $r(1) = 3$, $c(1) = 2$. 将 $\boldsymbol{PY}^{\mathrm{rc}}$ 中位置 (2,2) 处的像素值 1 与 $\boldsymbol{PY}$ 的元素逐个比较, 可以在 $\boldsymbol{PY}$ 中唯一位置 (1,1) 处找到; 所以, $r(2) = 1$, $c(2) = 1$. 将 $\boldsymbol{PY}^{\mathrm{rc}}$ 中位置 (3,3) 处的像素值 6 与 $\boldsymbol{PY}$ 的元素逐个比较, 可以在 $\boldsymbol{PY}$ 中唯一位置 (2,3) 处找到; 所以, $r(3) = 2$, $c(3) = 3$. 故行置乱序列为 $\boldsymbol{r} = \{3, 1, 2\}$; 而列置乱序列是 $\boldsymbol{c} = \{2, 1, 3\}$.

注意到这里的 $c(3)$ 是一个弱置乱密钥; 因为 $c(3)$ =3, 这意味着原始明文中的第 $c(3)$ 列置换到新图像中的列是同样的第 3 列, 出现不动点.

第4步 由待破解的密文矩阵 $\boldsymbol{C}$ 求出其对应明文的置乱矩阵 $\boldsymbol{P}^{\rm rc}$. 先由式 (19a) 求出 $\boldsymbol{P}^{\rm rc}$ 中的后面 8 个元素, 得

$$\begin{aligned} P^{\rm rc}(2:9) =& C(2:9)\oplus B(2:9)\oplus C(1:8)\\ =&[142,162,148,83,1,57,188,162]\\ &\oplus[78,94,84,20,87,59,211,24]\\ &\oplus[174,142,162,148,83,1,57,188]\\ =&[110,114,98,211,5,3,86,6]. \end{aligned}$$

再由 (19b) 式求出 $\boldsymbol{P}^{\rm rc}$ 中的第 1 个元素 (注意: $C(0)=CX(0)=C_0$, $B(1)\oplus C_0=214$)

$$\begin{aligned} P^{\rm rc}(1) =& C(1)\oplus B(1)\oplus C(0)\\ =&174\oplus B(1)\oplus C_0=174\oplus 214=120. \end{aligned}$$

将求出的像素值序列排列成二维矩阵形式, 即得

$$\boldsymbol{P}^{\rm rc}=\begin{pmatrix} 120 & 110 & 114\\ 98 & 211 & 5\\ 3 & 86 & 6 \end{pmatrix}.$$

第5步 反置乱得到原始明文图像矩阵. 先根据列置乱序列 $\boldsymbol{c}=\{2,1,3\}$, 由 $\boldsymbol{P}^{\rm rc}$ 进行反列置乱 (第 1 列移到第 2 列; 第 2 列移到第 1 列; 第 3 列移到第 3 列, 即不动), 即可得到

$$\boldsymbol{P}^{\rm r}=\begin{pmatrix} 110 & 120 & 114\\ 211 & 98 & 5\\ 86 & 3 & 6 \end{pmatrix}.$$

再根据行置乱序列 $\boldsymbol{r}=\{3;1;2\}$, 由 $\boldsymbol{P}^{\rm r}$ 进行反行置乱 (第 1 行移到第 3 行, 第 2 行移到第 1 行, 第 3 行移到第 2 行), 即可得到

$$\boldsymbol{P}=\begin{pmatrix} 211 & 98 & 5\\ 86 & 3 & 6\\ 110 & 120 & 114 \end{pmatrix}.$$

可见, 文献 [13] 的算法对文献 [12] 的算法改进不彻底, 依然不能抵御选择明文攻击.

4 改进的超混沌密码算法

分析 HIE 算法和 IHIE 算法的关键步骤, 发现它们共同的缺陷如下: 其一, 像素值替代加密的公式过于简单, 即 HIE 算法的 (3) 式和 IHIE 算法的 (14a) 或 (14b) 式中, 密文与明文、密钥之间的关系较为简单, 导致攻击者利用特殊的明、密文对容易破解替代密钥序列; 而该替代密钥序列又与明文无关, 破解后的替代密钥序列即可用来解密其他密文. 其二, 行、列置乱序列与明文无关, 仅决定于混沌序列; 而且又与替代过程独立, 一旦反替代得到置乱后的像素序列, 即可推断出行、列置乱序列; 故置乱过程事实上成为虚设. 其三, 该类算法对全部像素的替代操作只有一轮, 可以断定最终密文对明文的敏感性不强. 本文提出一种增强型超混沌图像加密算法, 将算法设计为 2 轮像素值的替代加密过程; 去掉像素置乱步骤; 并改造替代加密的公式, 使之复杂化. 这样, 既可以抵御攻击者对替代密钥序列分析的攻击; 同时也将提高密文对明文的敏感性; 省略置乱步骤对打破明文相邻像素的相关性不会有影响, 但可以大大缩短整个加密过程的时间.

本文改进的增强型超混沌图像加密算法的原理如图 1 所示. 图中用 $\boldsymbol{K}=\{K_1,K_2,\cdots,K_{L+1}\}$ 表示由超混沌系统生成的混沌密钥序列; 用 $\boldsymbol{P}=\{P_1,P_2,\cdots,P_L\}$ 表示明文图像的像素值序列; 用 $\boldsymbol{C}=\{C_1,C_2,\cdots,C_L\}$ 表示第一轮替代后得到的中间密文像素值序列; 而 $\boldsymbol{D}=\{D_1,D_2,\cdots,D_L\}$ 即为最后得到的密文像素值序列; $f(\cdot,\cdot)$ 代表某种非线性运算函数. 其中, $L=M\times N$ 为图像的像素总数, M, N 分别为图像的像素行数和列数. 由于第二轮的第 1 个像素加密时利用了第一轮最后像素的输出密文值, 因此, 经两轮操作即可将任何位置像素值的影响扩散到所有像素.

4.1 超混沌系统与混沌密钥序列生成

本文采用的超混沌系统模型为 [15]

$$\begin{aligned} \dot{x}_1 &= ax_1-x_2x_3,\\ \dot{x}_2 &= x_1x_3-bx_2,\\ \dot{x}_3 &= cx_1x_2-dx_3+gx_1x_4,\\ \dot{x}_4 &= kx_4-hx_2, \end{aligned} \tag{20}$$

其中 a, b, c, d, g, h 和 k 为系统的控制参数. 在 $a=8$, $b=40$, $c=2$, $d=14$, $g=5$, $h=0.2$ 和 $k=0.05$ 的条件下, 系统表现为超混沌运动. 本文在 Matlab7.1 的环境下对系统 (20) 进行仿真, 得到超混沌系统在上述参数条件下的仿真结果如图 2

所示. 系统 (20) 的超混沌吸引子有四个涡卷, 表现出比一般超混沌系统具有更加复杂的动力学特性. 从安全性角度考虑, 由于其相空间轨迹更加复杂, 因此用它产生密钥序列能够获得更高的安全性.

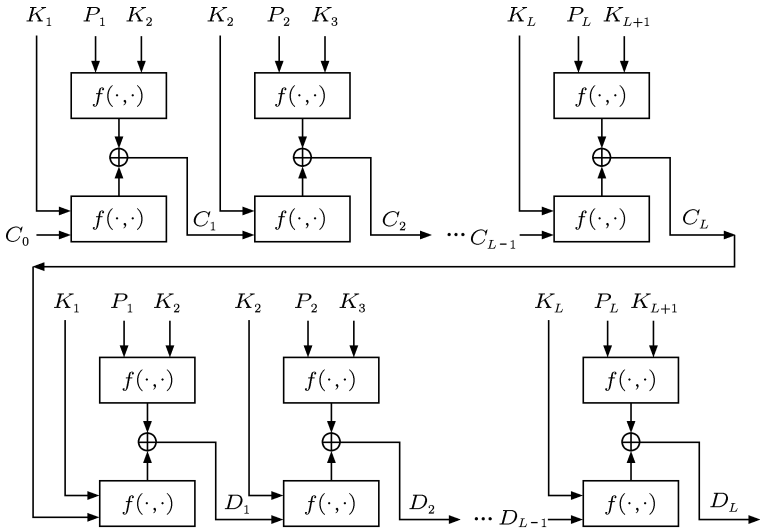

图 1 改进的增强型超混沌图像加密算法框图

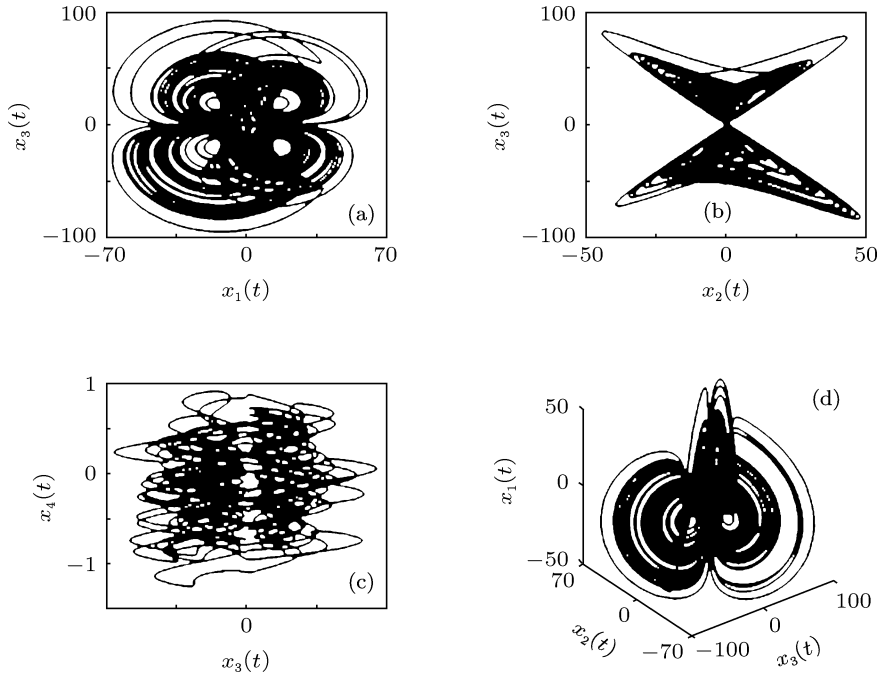

图 2 系统 (20) 的超混沌吸引子相图 (a) (x_1, x_3) 平面; (b) (x_2, x_3) 平面; (c) (x_3, x_4) 平面; (d) (x_1, x_2, x_3) 空间

先由系统 (20) 迭代生成超混沌实数序列, 然后对系统 (20) 生成的原始序列进行改造, 得到适合于图像加密的超混沌密钥序列. 生成超混沌密钥序列的方法如下:

步骤 1 超混沌系统预迭代 N_0 次, 以消除暂态过程带来的有害效应, 并增强算法对初始密钥的敏感性; 初始化一空序列 $\boldsymbol{B}$.

步骤 2 超混沌系统每迭代 1 次, 得到一组

新的状态值 $\{x_1, x_2, x_3, x_4\}$, 并计算如下 m 值: $m = \mathrm{mod}(x_1 + x_2 + x_3 + x_4, 3)$. 若 $m = 0$, 将 $\{x_1, x_3, x_2, x_4\}$ 加入序列 $\boldsymbol{B}$, 即序列 $\boldsymbol{B} = \{\boldsymbol{B}, x_1, x_3, x_2, x_4\}$; 若 $m = 1$, 则将 $\{x_2, x_4, x_3, x_1\}$ 加入序列 $\boldsymbol{B}$, 即序列 $\boldsymbol{B} = \{\boldsymbol{B}, x_2, x_4, x_3, x_1\}$. 若 $m = 2$, 则将 $\{x_4, x_1, x_3, x_2\}$ 加入序列 $\boldsymbol{B}$, 即序列 $\boldsymbol{B} = \{\boldsymbol{B}, x_4, x_1, x_3, x_2\}$.

步骤 3 重复 $(L/4+1)$ 次执行步骤 2, 将生成一个长度为 $L+4$ 的原始超混沌序列 $\boldsymbol{B}$.

步骤 4 再按照变换式

$$\begin{aligned} &\mathrm{temp} = \mathrm{floor}((|B(i)| - \mathrm{floor}(|B(i)|)) \times 10^{14}),\\ &K(i) = \mathrm{mod}(t\mathrm{emp}, 256),\\ &\qquad i = 1, 2, \cdots, L+4 \end{aligned} \tag{21}$$

对序列 $\boldsymbol{B}$ 进行改造, 得到混沌密钥序列 $\boldsymbol{K}$. 显然, $K(i) \in [0, 255]$.

4.2 两轮扩散替代加密操作

利用 4.1 生成的混沌密钥序列对图像进行两轮替代加密. 在加密过程中, 引入密文扩散机理, 并使密文与明文、密钥之间具有更复杂的非线性关系. 设明文图像的像素序列为 $\{P(i)|i = 1, 2, \cdots, L\}$; 中间密文图像的像素序列为 $\{C(i)|i = 1, 2, \cdots, L\}$; 最终密文图像的像素序列为 $\{D(i)|i = 1, 2, \cdots, L\}$.

第 1 轮扩散替代加密公式分别由

$$\begin{aligned} &\mathrm{temp} = \mathrm{mod}(P(1) + K(2), 256),\\ &C(1) = \mathrm{temp} \oplus \mathrm{mod}(K(1) + C_0, 256), \end{aligned} \tag{22a}$$

和

$$\begin{aligned} &\mathrm{temp1} = \mathrm{mod}(P(i) + K(i+1), 256),\\ &\mathrm{temp2} = \mathrm{mod}(K(i) + C(i-1), 256),\\ &C(i) = \mathrm{temp1} \oplus \mathrm{temp2},\\ &\qquad i = 2, 3, \cdots, L \end{aligned} \tag{22b}$$

表示. 其中, C_0 为加密第 1 个明文像素时引入的参数 (可以作为密钥), $C_0 \in [0, 255]$.

第 2 轮加密公式分别由

$$\begin{aligned} &\mathrm{temp1} = \mathrm{mod}(C(1) + K(2), 256),\\ &\mathrm{temp2} = \mathrm{mod}(K(1) + C(L), 256),\\ &D(1) = \mathrm{temp1} \oplus \mathrm{temp2}, \end{aligned} \tag{23a}$$

和

$$\begin{aligned} &\mathrm{temp1} = \mathrm{mod}(C(i) + K(i+1), 256),\\ &\mathrm{temp2} = \mathrm{mod}(K(i) + D(i-1), 256),\\ &D(i) = \mathrm{temp1} \oplus \mathrm{temp2}, \quad i = 2, 3, \cdots, L \end{aligned} \tag{23b}$$

表示.

从上述加密公式来看, 攻击者要想由特殊的明、密文对 $< P(i), D(i) >$ 破解出密钥序列 $\boldsymbol{K}$ 是比较困难的, 因为在第 2 轮加密操作中 $D(i)$ 与 $P(i)$ 无直接联系, 故依靠 (23a) 和 (23b) 式无法反推出 $K(i)$; 而且每一轮加密操作中, 密文与明文 (或中间密文) 以及密钥之间的关系不是简单的异或运算关系, 还包含了非线性的取模运算. 因此, 算法可以抵御选择明文攻击.

设解密图像用矩阵 $\boldsymbol{P}'$ 表示, 其像素值按逐行扫描顺序排列形式为 $\{P'(i), i = 1, 2, \cdots, L\}$, 解密过程与加密过程互为逆操作; 但解密时针对的像素操作顺序为逆序 (即从最后一个像素开始, 依次循环到第 1 个像素点).

第 1 轮解密操作公式为

$$\begin{aligned} &\mathrm{temp} = D(i) \oplus \mathrm{mod}(K(i) + D(i-1), 256),\\ &P'(i) = \mathrm{mod}(\mathrm{temp} + 256 - K(i+1), 256),\\ &\qquad i = L, L-1, \cdots, 2, \end{aligned} \tag{24a}$$

$$\begin{aligned} &\mathrm{temp} = D(1) \oplus \mathrm{mod}(K(1) + P'(L), 256),\\ &P'(1) = \mathrm{mod}(\mathrm{temp} + 256 - K(2), 256),\\ &\qquad i = 1. \end{aligned} \tag{24b}$$

第 2 轮解密操作公式为

$$\begin{aligned} &\mathrm{temp} = P'(i) \oplus \mathrm{mod}(K(i) + P'(i-1), 256),\\ &P'(i) = \mathrm{mod}(\mathrm{temp} + 256 - K(i+1), 256),\\ &\qquad i = L, L-1, \cdots, 2, \end{aligned} \tag{25a}$$

$$\begin{aligned} &\mathrm{temp} = P'(i) \oplus \mathrm{mod}(K(i) + C_0, 256),\\ &P'(i) = \mathrm{mod}(\mathrm{temp} + 256 - K(i+1), 256),\\ &\qquad i = 1. \end{aligned} \tag{25b}$$

完成上述两轮操作后, 就得到了最终的解密图像 $\boldsymbol{P}'$. 如果解密时所用的初始密钥及其他参数与加密时参数完全一致的话, 则解密图像将会与原始图像完全一致, 即 $\boldsymbol{P}' = \boldsymbol{P}$.

5 实验仿真与性能分析

实验中使用 256 × 256 的 8 位经典测试图像, 在 Matlab7.1 下仿真. (20) 式的系统参数取 $a = 8$, $b = 40$, $c = 2$, $d = 14$, $e = 5$, $g = 0.2$, $h = 0.05$. 这样, 系统 (20) 是超混沌的. 取系统状态初值为 (0.12, 0.23, 0.34, 0.45); 微分方程组迭代求解的时间步长取 0.001; 取 $N_0 = 1000$, $C_0 = 74$.

5.1 抗统计攻击的性能分析

首先对来自 University of Granada 标准测试图库 CVG 中的 Cameraman 图像进行 2 轮加密, 加密效果如图 3 所示.

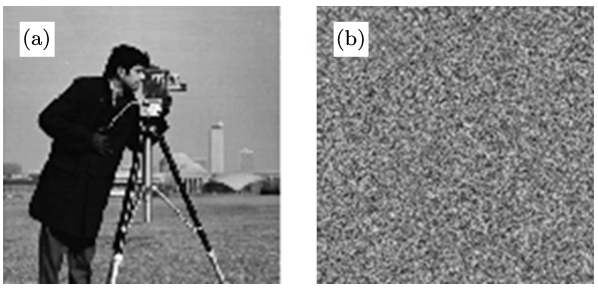

图 3 图像加密效果 (a) 原始图像; (b) 加密图像

图 4 则分别给出了原始明文图像和密文图像对应的直方图, 由图 4(a) 可见, 原始 Cameraman 图像的像素值分布是非常不均匀的; 但图 4(b) 表明, 加密后图像的像素值已经呈平坦而均匀的分布, 表明密文图像的像素值取各种可能值的概率趋于均等. 因此, 本文算法将能够有效地抵抗基于统计分析的攻击.

5.2 相关性分析

从图像中选取若干组相邻的像素对 (包括水平、垂直和对角方向的三类像素对), 用

$$\gamma = \frac{\sum_{i=1}^{M_0}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Big(\sum_{i=1}^{M_0}(x_i - \bar{x})^2\Big)\Big(\sum_{i=1}^{M_0}(y_i - \bar{y})^2\Big)}} \tag{26}$$

分别计算三种方向相邻像素之间的相关系数 [12,13]. (26) 式中, x_i 和 y_i 分别表示图像中第 i 对邻居像素的两个像素值; $\bar{x}$, $\bar{y}$ 分别为所有 x_i、所有 y_i 的平均值; M_0 为选取的相邻像素对组数; γ 即为相邻像素的相关系数. 取前述有关参数对 Cameraman 图像进行加密, 计算加密前后图像三种方向的 γ 系数 (M_0 取值为全部像素对的组数), 所得结果分别列于表 1 的第 2, 3 列中. 从表 1 结果可知, 明文图像的相邻像素是高度相关的 ($\gamma \to 1$); 但对应密文图像的相邻像素则已经几乎不相关 ($\gamma \to 0$). 表 1 第 4 列同时也给出了用文献 [13] 的算法和超混沌系统加密此图像所得密文像素的相应结果. 可见本文算法所得的密文图像具有更小的 γ 系数, 表明本文算法在省略置乱操作时, 只要增加 1 轮替代加密操作, 也能更好地达到破坏相邻像素相关性的目的, 使密文具有更好的随机分布特性.

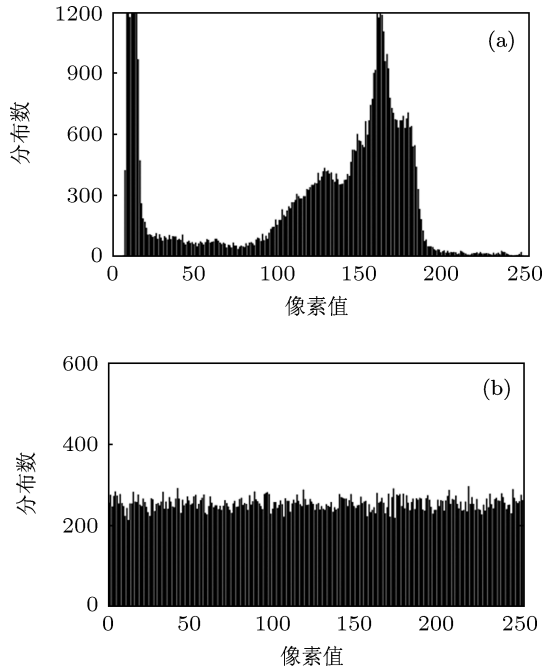

图 4 直方图分布 (a) 原始图像的直方图; (b) 加密图像的直方图

表 1 明文和密文的相邻像素相关系数

方向	明文	密文	密文 [13]
水平	0.933475	−0.000907	0.001406
垂直	0.959223	−0.000475	0.009399
对角	0.908663	0.000057	0.003087

5.3 抗差分攻击的性能分析

根据密码学原理, 一种好的密码算法应该对明文充分敏感, 这种敏感性越强, 抵抗差分攻击的能力就越强. 加密算法对明文的敏感性可以用像素数改变率 (number of pixels change rate, NPCR) 表征;

因此, NPCR 也是算法抗差分攻击性能的重要度量指标. NPCR 定义为两幅仅有一个不同像素的明文图像对应密文图像所具有的不同像素比例; 设两幅密文图像中第 (i, j) 点的像素值分别为 $C_1(i,j)$ 和 $C_2(i,j)$, 若 $C_1(i,j) = C_2(i,j)$, 定义 $D(i,j) = 0$; 若 $C_1(i,j) \neq C_2(i,j)$, 定义 $D(i,j) = 1$. 则 NPCR 的计算公式为 [16]

$$\mathrm{NPCR} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\%, \quad (27)$$

其中, M 和 N 分别是图像的像素行数与列数. 理想情况下, NPCR 的期望值为 $\mathrm{NPCR_E} = (1 - 2^{-n}) \times 100\%$, 其中的 n 为图像颜色深度. 对于 256 级颜色的灰度图像 ($n = 8$), NPCR 的理想期望值为 $\mathrm{NPCR_E} = 99.6094\%$. 本文实验共取了 100 组 Lena 图像进行加密, 每组的两个明文图像中, 仅有一个位置的像素点灰度值相差 1, 这个不同的像素点是随机地从图像不同位置选择的 (并包括起始点、中间位置点和最末尾位置点三种极端情况在内). 然后计算 100 组密文图像之间的 NPCR 值, 得到的平均值为 $\overline{\mathrm{NPCR}} = 99.6397\%$; 其余数据的详细结果则如图 5(a) 所示. 图 5(a) 表明, 本文算法 100 组实验的 NPCR 值都非常接近理想值 (图中水平线), 仅在理想值附近上下浮动. 即无论任何位置的明文像素值改变 1, 所引起的密文像素数改变率都能达到 99%以上; 这表明本文算法对明文非常敏感. 我们也将文献 [13] 的 IHIE 算法进行了实现, 然后用同样的 100 组图像进行实验, 得到 100 组密文图像之间的 NPCR 值曲线则如图 5(b) 所示, 其平均值仅为 $\overline{\mathrm{NPCR}}' = 50.4228\%$. 可见, 本文算法对明文的敏感性远远强于文 [13] 算法对明文的敏感性. 因为文 [13] 算法只在一轮像素替代操作过程中产生密文扩散效应, 因此一个位置点的明文变化只影响该点后面的密文变化. 而本文算法经两轮替代操作, 在第 2 轮替代加密过程中, 首先由前一轮最后一点的密文值影响第 2 轮第一点的密文值; 因此, 原始图像任何位置点的明文像素值发生变化, 都将扩散到所有位置的密文像素.

5.4 密钥敏感性实验

一个好的密码算法对密钥必须非常敏感 [17,18], 即两个具有微小差别的加密密钥, 应该产生完全不同的密文图像; 同样地, 两个具有微小差别的解密密钥, 对同一密文的解密结果也应该截然不同. 在本文实验中, 对来自 University of Granada 标准测试图库 CVG 中的 Einstein 图像先采用由初始参数 (x_{10}, x_{20}, x_{30}, x_{40}) = (0.12, 0.23, 0.34, 0.45) 生成的密钥进行加密, 然后采用稍微不同的初始参数 $(x_{10}, x_{20}, x_{30}, x_{40}) = (0.12, 0.23 + 10^{-10}, 0.34, 0.45)$ 生成的密钥去解密, 图 6(a) 与 (b) 分别给出了原始图像和该错误密钥的解密图像. 计算图 6(a) 与 (b) 之间的均方误差 [13], 结果为 7244.2319. 可见, 解密密钥 x_{10} 稍有误差, 解密图像与原始图像截然不同; 即算法对解密密钥非常敏感. 对初始密钥 x_{20}, x_{30}, x_{40} 进行测试, 都得到类似结果.

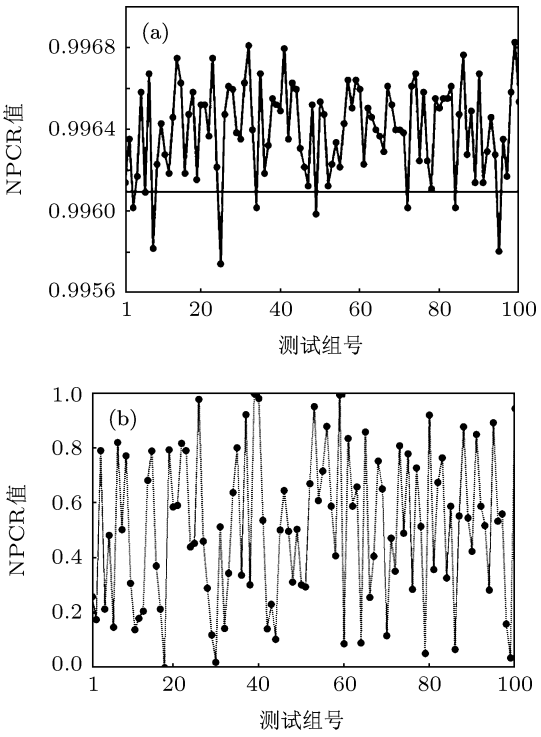

图 5 像素数改变率对比实验 (a) 本文算法密文像素数改变率; (b) IHIE 算法密文像素数改变率

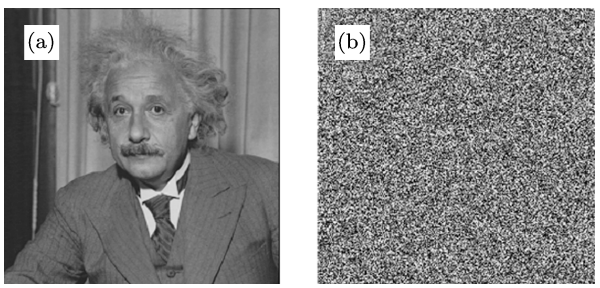

图 6 算法对解密密钥敏感性的测试结果 (a) 原始图像; (b) x_{10} 误差 10^{-10} 时得到的解密图像

然后, 测试 3 组加密密钥对 Einstein 图像加密所得密文的差别, 这三组密钥对应的初始参数 $(x_{10}, x_{20}, x_{30}, x_{40})$ 分别是 (0.12, 0.23, 0.34, 0.45), $(0.12+10^{-10}, 0.23, 0.34, 0.45)$ 和 $(0.12, 0.23+10^{-10}, 0.34, 0.45)$, 三组密文图像分别用 $\boldsymbol{C}_1$, $\boldsymbol{C}_2$ 和 $\boldsymbol{C}_3$ 表示. 图 7(a) 与 (b) 分别绘制了 $\boldsymbol{C}_1$ 与 $\boldsymbol{C}_2$ 之间、$\boldsymbol{C}_1$ 与 $\boldsymbol{C}_3$ 之间前 200 对点的差值曲线. 同时计算了 $\boldsymbol{C}_1$ 与 $\boldsymbol{C}_2$ 之间、$\boldsymbol{C}_1$ 与 $\boldsymbol{C}_3$ 之间的 NPCR 值分别为 99.6368%, 99.5941%. 由此可见, 初始密钥 x_{10} 或 x_{20} 误差 10^{-10} 时, 所得密文图像截然不同; 表明算法对加密密钥非常敏感. 进一步对 x_{30}, x_{40} 初始密钥作细微改变, 均得到类似结果.

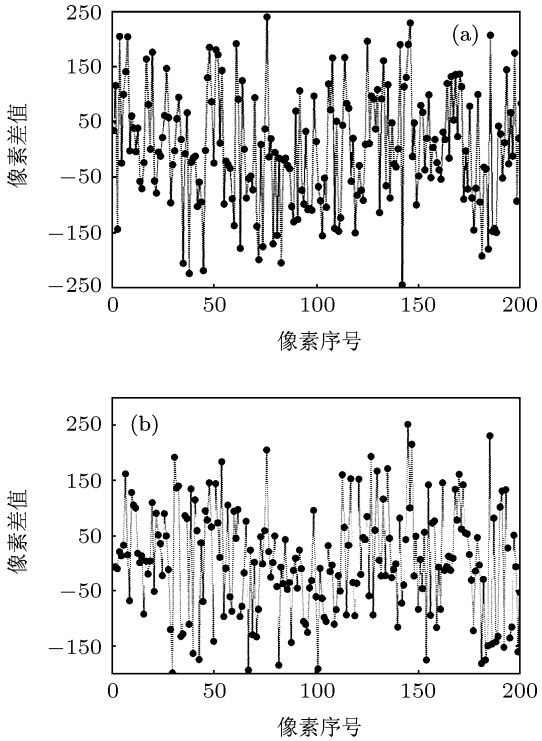

图 7 算法对加密密钥敏感性的测试结果. (a) 两组加密密钥中 x_{10} 误差 10^{-10}; (b) 两组加密密钥中 x_{20} 误差 10^{-10}

5.5 密钥空间和执行效率

本文算法采用超混沌系统的 4 个状态变量的初始值作为原始密钥, 每个数据用 16 位十进制数字的实数表示 (整数部分 1 位和小数部分 15 位); 因此, 密钥空间是 $10^{16} \times 10^{16} \times 10^{16} \times 10^{16} = 10^{64} \approx 2^{213}$, 相当于二进制的 213 比特密钥长度. 若将参数 C_0 也作为原始密钥, 则密钥空间更大. 故本文算法具有抗穷举攻击的能力. 实验在硬件环境为 Intel Celeron 2.13 GHz CPU, 2 GB 内存和 120 GB 硬盘的 PC 机及软件环境为 Windows XP, Matlab7.1 编译器的平台上进行; 最终密钥和像素采用无符号整数表示. 本文算法用两轮像素替代操作加密一幅 256 × 256 的 8 位灰度图像的平均时间约为 0.047 s. 将文献 [13] 的算法在同样的计算机系统环境下实现, 对相同的原始图像采用相同数据类型加密, 经行、列置乱和一轮像素替代加密操作的平均时间则为 0.065 s. 可见, 本文加密算法的加密速度比文献 [13] 算法要快. 文献 [13] 算法置乱阶段需要另外生成置乱密钥序列 $\boldsymbol{r}$ 和 $\boldsymbol{c}$, 所以导致算法的时间复杂度增加.

6 结 论

本文对一类超混沌图像加密算法进行了安全性分析, 发现该类算法存在下列共同缺陷: 其一, 由于像素位置置换与像素值替代加密两种处理环节独立, 因此可以分别破解密码系统的置乱密钥序列和替代密钥序列. 其二, 在像素替代加密环节, 加密公式是明文和密钥的简单异或运算; 而且替代加密只有一轮, 这样很容易由已知的明、密文对破解出替代加密密钥序列. 基于此, 我们提出了一种改进的超混沌图像加密算法, 改进算法只需要两轮像素值替代加密操作, 即可使密文对任何明文像素的变化具有高度敏感性; 并破坏相邻像素的相关性. 改进的加密公式不仅使最终加密密钥与明文相关, 而且使密文与明文、密钥之间的关系复杂化; 使算法能真正抵抗选择明文和选择密文攻击. 实验结果和安全性分析表明, 该算法的密钥空间大、密文相邻像素的相关性低、密文分布均匀. 与原超混沌图像密码算法相比, 不仅真正具备了抵抗选择明文和选择密文攻击的能力; 而且具有时间开销更小, 抗差分攻击能力更强等优势. 因此, 本文算法在图像保密通信等应用领域将具有更好的应用潜力.

[1] Pisarchik A N, Zanin M 2008 *Physica* D **237** 2638
[2] Yang D G, Liao X F, Wang Y 2009 *Chaos Soliton. Fract.* **41** 505
[3] Rontani D, Sciamanna M, Locquet A 2009 *Phys. Rev.* E **80** 066209
[4] Liu S B, Sun J, Xu Z Q 2009 *J. Computers* **4** 1091
[5] Matthews R A J 1989 *Cryptologia* XIII **29**
[6] Lin S L, Tung P C 2009 *Chaos Soliton. Fract.* **42** 3234
[7] Moskalenko O I, Koronovskii A A, Hramov A E 2010 *Phys. Lett.* A **374** 2925
[8] Wang Y, Wong K W, Liao X F, Chen G R 2011 *Appl. Soft. Comput.* **11** 514
[9] Zhu Z L, Zhang W, Wong K W, Yu H 2011 *Information Sciences* **181** 1171
[10] Tong X J, Cui M G, Wang Z 2009 *Opt. Commun.* **282** 2722
[11] Wang X Y, Wang M J 2007 *Acta Phys. Sin.* **56** 5136 (in Chinese) [王兴元, 王明军 2007 物理学报 **56** 5136]
[12] Gao T G, Chen Z Q 2008 *Phys. Lett.* A **372** 394
[13] Wang J, Jiang G P 2011 *Acta Phys. Sin.* **60** 060503 (in Chinese) [王静, 蒋国平 2011 物理学报 **60** 060503]
[14] Park J H 2005 *Chaos Soliton. Fract.* **26** 959
[15] Dadras S, Momeni H R 2010 *Phys. Lett.* A **372** 1368
[16] Sun F Y, Lü Z W 2011 *Chin. Phys.* B **20** 040506
[17] Wang X Y, Xie Y X 2011 *Chin. Phys.* B **20** 080504
[18] Wang F L 2011 *Acta Phys. Sin.* **60** 110517 (in Chinese) [王福来 2011 物理学报 **60** 110517]

Cryptanalysis and improvement of a class of hyperchaos based image encryption algorithms*

Zhu Cong-Xu† Sun Ke-Hui

(*School of Information Science and Engineering, Central South University, Changsha* 410083*, China*)

(Received 26 October 2011; revised manuscript received 16 November 2011)

Abstract

The security of a class of hyperchaotic image encryption algorithms is analysed. The results show that the shuffling process can be separated from the confusion process, and the formulas of encryption are simple, which makes the ciphertext cannot resist the attacks from chosen plaintext and ciphertext. Then we propose an improved and enhanced algorithm based on hyperchaos. The improved algorithm includes two rounds of encryption operation. The theoretical analyses and the experimental results indicate that the improved algorithm can overcome these flaws and has better cryptographic performances in resisting differential attacks and speed of encryption.

Keywords: hyperchaos, image encryption, chosen plaintext attack, chosen ciphertext attack

PACS: 05.45.Gg

* Project supported by the National Natural Science Foundation of China (Grant Nos. 61073187, 61161006), the Natural Science Foundation of Hunan Province, China (Grant No. 10JJ6093), and the Natural Science Foundation of Guangdong Province, China (Grant No. S2011010001581).

† E-mail: zhucx@csu.edu.cn