文章编号: 1000-5641(2019)06-0061-12

基于自适应与全局置乱的图像加密新算法

贾忠祥, 柳银萍

(华东师范大学 计算机科学与软件工程学院, 上海 200062)

摘要: 针对现有的基于比特位层面的图像加密算法中, 比特位置乱过程中存在的局限性和局部性问题, 提出了一种比特位全局置乱的加密算法, 即在置乱过程中, 位平面的重组及之后的置乱操作均随机进行, 整个置乱过程不只局限在某些位平面之内进行, 由此达到全局置乱的效果. 该加密算法运用了混沌映射系统, 可以同时实现像素的置乱和扩散操作; 另外, 为了增加对明文的敏感性和有效抵抗攻击, 加入了位平面的自适应过程, 该过程利用图像不同位平面数据之间的异或运算来进一步修改图像数据. 经实验表明: 该加密算法对明文和密钥非常敏感, 可有效抵抗选择明文攻击, 且密文图像像素分布均匀, 具有良好的图像加密效果.

关键词: 比特位层面; 图像加密; 全局置乱; 混沌映射系统; 自适应

中图分类号: TP309.7 **文献标志码**: A **DOI:** 10.3969/j.issn.1000-5641.2019.06.007

Novel image encryption algorithm based on self-adaptive diffusion and combined global scrambling

JIA Zhong-xiang, LIU Yin-ping

(*School of Computer Science and Software Engineering*, *East China Normal University*, *Shanghai* 200062, *China*)

Abstract: In light of the limitations and locality problems of the bit permutation process in existing bit-level encryption algorithms, a novel algorithm for self-adaptive diffusion and combined global scrambling was proposed. In the diffusion phase, the reorganization and subsequent scrambling process of bit planes is performed randomly, so that the scrambling process is not limited to just some bit planes, and we achieve the effect of global scrambling. The algorithm employed several chaotic mapping systems, allowing permutation and diffusion operations to be implemented simultaneously. To increase sensitivity to plain images and effectively resist attacks, the self-adaptive process of bit planes was added, thereby further modifying the original image data based on given rules. Simulation results demonstrate that the proposed scheme is sensitive to plain images and keys, and can resist Chosen-Plaintext Attack effectively. The cipher image has uniform pixel distribution and good image encryption.

收稿日期: 2018-10-31

基金项目: 国家自然科学基金(11435005, 11871328); 上海市科学技术委员会重点项目(18511103105)

第一作者: 贾忠祥, 男, 硕士研究生, 研究方向为计算机应用. E-mail: jonariguez@163.com.

通信作者: 柳银萍, 女, 教授, 博士生导师, 研究方向为计算机数学、计算机软件与理论.
E-mail: ypliu@cs.ecnu.edu.cn.

Keywords: bit-level; image encryption; global scrambling; chaotic map systems; self-adaptive

0 引 言

视觉是人们获得感知信息的主要方式, 视觉信息的主要来源是图像. 在计算机网络技术和通信技术快速发展的同时, 在网络环境中存储和传输的数字图像的数量更是急剧增多. 数字图像成为了网络中信息交流的主要信息载体. 然而随着图像数量的增加, 暴露在网络中的信息也越来越多, 其中包括一些个人的隐私信息、企业乃至国家的涉密信息, 所以网络图像信息的安全问题日益受到重视. 数字图像与一般的文本不同, 具有数据量大、相关性强以及冗余度高等特点, 这使得传统的密码学方法, 如DES(Data Encryption Standard)和AES(Advanced Encryption Standard)应用到图像加密场合的效果并不理想. 在此背景下, 基于混沌的图像加密策略引起了广泛的关注[1-7].

混沌系统具有伪随机性、遍历性和初值敏感性等特点, 其在密码学的应用中具有明显的优势. 同时, 混沌序列可以由混沌系统简单高效地生成, 故能满足加密需求. 图像加密一般采用置乱–扩散机制: 置乱过程是通过改变像素的位置来实现像素置乱; 扩散过程是对像素值的修改. 早期所研究的图像加密算法都只是基于像素层面的置乱和扩散, 并不能打破图像固有的特性, 故而加密效果一般[4-7]. 因此, 近年来越来越多的研究着眼于基于比特位层面的加密, 并获得了理想的加密效果. 混沌系统被广泛地运用到图像加密机制的各个阶段. 汪彦等提出了改进的 Lorenz 混沌系统[8], 在混沌系统里加入了 x^2 项和 e^{xy} 项, 其混沌行为表现得更为复杂, 并基于该改进的混沌系统进行图像加密. 毛骁骁等设计了分数阶统一混沌系统[9], 并运用该系统产生置换以对明文图像的像素值进行置乱, 最后通过扩散得到密文图像. Ping 等利用 Tent 和 Henon 混沌映射系统, 第一阶段对行和列分别进行置乱, 第二阶段分别对 8 个位平面各自进行置乱[10]; 但其不足之处是在第二阶段的置乱过程中没有位平面之间的比特位交换. Zhang 等发现了图像的两个内在特性: 高位相邻的位平面大概率具有相反的取值; 图像数字矩阵中 0 和 1 的个数分布在宏观上稳定, 而在微观上分布不均匀[11]. 针对这两个特性, 文中运用 CML(Coupled Map Lattice)对位平面进行置乱, 在置乱中奇数层和偶数层的位平面的数据可以各自相互置乱; 但其存在的不足是奇数层和偶数层之间不存在比特位的交换置乱, 即置乱具有局限性.

本文基于 Tent 和 Henon 混沌系统, 提出了新的比特位平面的全局置乱算法, 对高位(在二进制表示中权重较大的位)和低位(在二进制表示中权重较小的位)采用不同的操作: ① 对明文图像进行位平面分解, 并进行自适应的异或运算来修改高 4 位的位平面. ②位平面的置乱分为两个阶段: 第一阶段, 分别对高 4 位和低 4 位组成的图像进行像素置乱; 第二阶段, 通过两两分组的方法形成 4 个平面, 进一步组成一幅新图像后再次进行像素的置乱. ③经过扩散得到最终加密图像. 这样就没有了文献 [11] 中位平面置乱时的分组限制, 加密效果良好. 经实验分析可知, 本文提出的图像加密算法能够有效地降低相邻像素之间的相关性, 有效增强抵抗选择明文攻击和已知明文攻击的能力, 最终的加密图像像素分布均匀.

1 算法原理

1.1 混沌系统

混沌是非线性动力系统的固有特性, 是普遍存在于非线性系统中的现象. 混沌系统的伪随机性、遍历性和初值敏感性等特点, 促使它在加密领域被引起广泛的关注. 本文用到了以

下混沌系统.

(1) Tent 混沌系统, 方程为

$$x_{n+1}=\begin{cases}\mu x_n, & 0<x_n<\dfrac{1}{2},\\ \mu(1-x_n), & \dfrac{1}{2}\leqslant x_n<1,\end{cases}\tag{1}$$

其中, $x_n\in(0,1)$, $\mu\in(0,2)$, x_n 和 μ 均为实数. 当 $\mu>1$ 时, 系统处于混沌状态.

(2) Henon 混沌系统, 方程为

$$\begin{cases}x_{n+1}=1-ax_n^2+y_n,\\ y_{n+1}=bx_n,\end{cases}\tag{2}$$

是一个二维的离散动力系统. 公式(2)中, x 和 y 是状态变量, a 和 b 为大于 0 的控制参数. 为了方便处理离散的数字图像, 采用其对应的完全离散映射

$$\begin{cases}x_{n+1}=1-ax_n^2+y_n \bmod N,\\ y_{n+1}=bx_n+c \bmod N,\end{cases}\tag{3}$$

其中, a、b、c都是控制参数. 当$b=1$时, Henon 映射称为保面积映射, 此时 Henon 映射是可逆的, 其逆映射为

$$\begin{cases}x_n=y_{n+1}-c \bmod N,\\ y_n=x_{n+1}-1+ax_n^2 \bmod N.\end{cases}\tag{4}$$

使用 Tent 映射系统可以产生一维的混沌序列, 或称为密钥流. Henon 映射系统可以用作坐标变换, 用于对像素置乱的过程中.

1.2　算法特性

在 Henon 映射中有 a 和 c 两个控制参数需要确定, 出于安全性考虑, 该控制参数由密钥流确定. 用来产生密钥流的 Tent 混沌系统是一个迭代系统, 根据初始值和参数值可以得到所需长度的密钥流序列.

在对加密效果的评价中, 像素个数变化率(Number of Pixels Change Rate, NPCR)是表示当明文图像的某个像素发生改变时, 经过加密得到的密文图像与由加密原文图像得到的密文图像的像素变化率. 实际上, 像素个数变化率描述的是加密算法对明文变化的敏感性, 详细内容将在第 2.5 节讨论. 讨论像素个数变化率的目的在于, 如果采用某些攻击方法, 如选择明文攻击和已知明文攻击, 来攻击加密算法, 攻击者可以选择一定数量的特定明文图像和对应密文图像, 通过发现其中的规律以破解加密算法所采用的密钥. 因此, 理想的加密算法要对明文具有足够的敏感性. 上述的敏感性至少要满足两个基本要求: ① 如果某个像素点的值发生变化, 则加密过程会不同; ② 如果没有像素点的值发生变化, 但是任意两个或多个像素点的位置发生了交换或变化而其他点保持不变, 则加密过程也会不同.

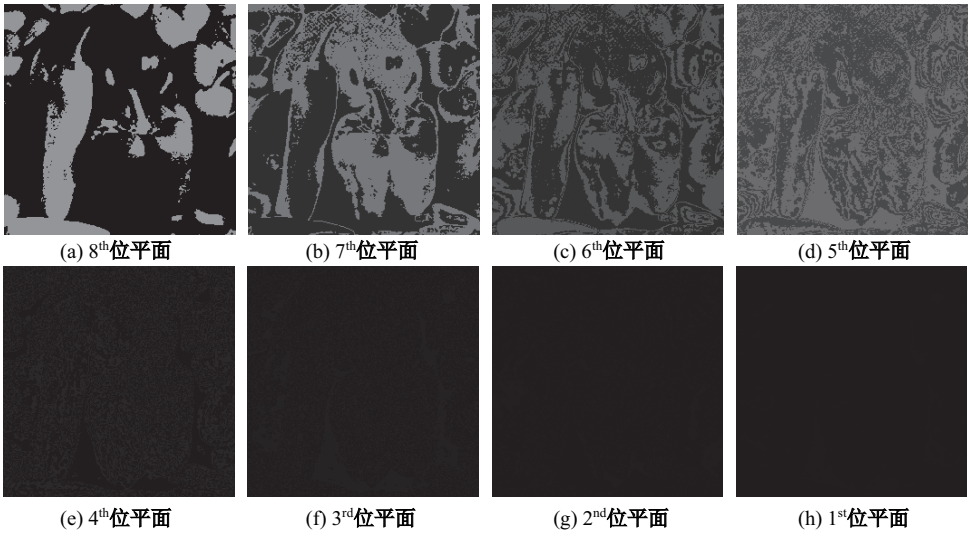
综上, 在确定 Tent 映射系统中的控制参数时, 既需考虑明文图像中每个像素点的像素值, 又要考虑像素点的位置信息.

1.3 加密原则

基于像素层面的加密算法的效果不够理想. 因此, 目前越来越多的研究都是基于比特位层面的加密[1-3,10-11]. 灰度图像通常只有一个道, 且像素值的取值范围为[0, 255]. 所以对于数字灰度图像, 如果把所有像素值都用二进制表示, 那么这些 0 和 1 因为位置的不同, 所携带的信息量也不同, 具体第 i 位比特位所携带的信息可以用公式

$$p(i)=\frac{2^{i-1}}{\sum\limits_{j=0}^{7}2^{j}},\ i=1,\ 2,\ \cdots,8 \tag{5}$$

来计算[11]. 根据式(5)可知, 高位携带的信息更多一些. 图 1 所示是一张图像的每一位所组成的图像, 其中图 1(a)–图 1(h)分别对应由像素值第 8, 7,$\cdots$, 1 位二进制位所组成的位平面.



(a) 8th位平面 (b) 7th位平面 (c) 6th位平面 (d) 5th位平面

(e) 4th位平面 (f) 3rd位平面 (g) 2nd位平面 (h) 1st位平面

图 1 不同位平面所携带的信息

Fig. 1 Information carried by the different bit planes

有研究表明, 达到理想的加密效果需要改变的比特位的数目只占全部比特位的 3.3%[11]. 因为图像普遍存在着局部分布不均匀的问题, 所以加密的目标重点并非尽量多地改变数据, 而是尽量改变数据的分布, 使其均匀. 因此, 为了权衡加密效果和算法开销, 算法应注重改变图像数据的分布.

1.4 图像加密算法

基于以上分析, 本文提出了如下加密算法, 具体步骤为如下.

(1) 对于一幅大小为 $N\times N$ 的原始明文图像 P, S、μ 和 k 的计算公式分别为

$$\begin{cases} S=\sum\limits_{i,j}P_{ij}\otimes i'\otimes j', \\ i'=i \bmod 256, \\ j'=j \bmod 256, \end{cases} \tag{6}$$

$$\mu=\alpha\times 2^{\frac{S}{N\times N\times 256}}, \tag{7}$$

$$k=S \bmod 10^3+10^3, \tag{8}$$

其中, P_{ij} 为图像 P 中位置(i, j)处的像素值, $\otimes$代表异或运算, α 是输入密钥, μ 作为 Tent 混沌系统的控制参数, k是系统初始迭代次数. 同时, S 将作为密钥传送给解密方.

(2) 将图像中每一个像素值都用 8 位二进制表示, 由$N \times N$个像素值的第 i 位二进制位组成一个位平面 bit_i, i=1, 2,$\cdots$,8, 其中 bit_i 表示第 i 个位平面.

(3) 用 bit_1异或 bit_5, 对其他位平面也做同样对应的操作, 可形式化地表示为

$$\mathrm{bit}_j = \mathrm{bit}_j \otimes \mathrm{bit}_{j-4}, \quad j = 5, 6, 7, 8, \tag{9}$$

这是自适应操作的过程. 位平面之间的异或是指两位平面中对应点的像素值进行异或运算.

(4) 输入密钥 x_0, 作为 Tent 映射系统初始值, 对 Tent 映射系统迭代 k 次以消除初态影响.

(5) 对 Tent 系统再迭代 3 次, 得到 3 个实数 value_1, value_2, value_3, 利用式(10)–(12)计算a_i, c_i, k_i, i=1, 2, 3 时的值. 具体公式分别为

$$a_i = \mathrm{floor}(\mathrm{value}_i \times 10^3) \bmod 250 + 5, \tag{10}$$

$$c_i = \mathrm{floor}(\mathrm{value}_i \times 10^6) \bmod 250 + 5, \tag{11}$$

$$k_i = \mathrm{floor}(\mathrm{value}_i \times 10^7) \bmod 5 + 1, \tag{12}$$

其中, $a_i \in$[5, 254] 可以保证在后续步骤中置乱时减少极端情况的发生, c_i 亦同理.

(6) 将bit_5, bit_6, bit_7, bit_8分别放置到对应的方框 1, 2, 3, 4 中, 如图 2 所示. 由此形成了一个 $2N \times 2N$的图像, 图像上的每个像素点只包含 1 位二进制数 0 或 1, 以 a_1 和 c_1 为控制参数, 利用 Henon 混沌映射系统对该图像的像素点置乱 k_1 轮, 每轮置乱方式为: 对每一个位置(x_n, y_n)的像素点, 迭代 1 次 Henon 系统得到(x_{n+1}, y_{n+1}), 将(x_n, y_n)处的像素点移动到(x_{n+1}, y_{n+1}). 同理, 对低 4 位的位平面做相同的操作, 即将bit_1, bit_2, bit_3, bit_4分别对应到方框 1, 2, 3, 4 中, 以a_2, c_2 为参数, 置乱 k_2 轮.

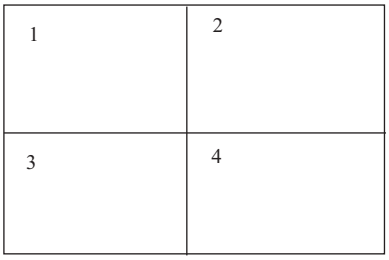


图 2　位平面的重组放置方式

Fig. 2　Reorganized placement of the bit planes

(7) 继续对 Tent 映射系统迭代 8 次, 得到$T = \{t_1, t_2, \cdots, t_8\}$, 对 T 进行间接寻址升序排序, 得到位置向量P^{T}=$\{t'_1, t'_2, \cdots, t'_8\}$. 令 $p[i]$=j 表示 t_i 在 P^{T} 中的位置为 j, 即

$$P_j^{\mathrm{T}} = P_{p[i]}^{\mathrm{T}} = T_i = t_i, \quad i = 1, 2, \cdots, 8, \tag{13}$$

则 p 形成一个置换. 令B=$\{\mathrm{bit}_1, \mathrm{bit}_2, \cdots, \mathrm{bit}_8\}$, 利用 P 对 B 置乱后得到序列 R, 即有

$$R_{p[i]} = B_i = \mathrm{bit}_i, \quad i = 1, 2, \cdots, 8, \tag{14}$$

显然 R 是$\{\mathrm{bit}_1, \mathrm{bit}_2, \cdots, \mathrm{bit}_8\}$的一个排列.

(8) 从上述步骤可知, R_i代表了某个位平面. 类似于步骤 (6), 分别把$\{R_8, R_1\}$, $\{R_7, R_2\}$, $\{R_6, R_3\}$, $\{R_5, R_4\}$放到对应的方框 1, 2, 3, 4 中形成一个新图像. 因为每个方框里是两个位平面, 所以新图像的每个像素点是一个包含两位二进制位的数. 最后以 a_3 和c_3(在步骤(5)中计算得到)为 Henon 的系统参数置乱该图像 k_3 轮. 例如, 假设 R_i=bit_i, $i = 1, 2, \cdots, 8$, 并且在方框 1 中有一个像素点的像素值为$(11)_2$=$(2^7+2^0)_{10}$=129, 如果置乱过后该点移到了方框 3 中, 则像素值变为$(11)_2$=$(2^5+2^2)_{10}$=36, 相当于 8^{th}位平面上的一个 1 和 1^{st}位平面上的一个 1 分别移动到了 6^{th}位平面和 3^{rd}位平面上. 其他情况同理. 最后按照对应的权重和位平面将该 $2N\times 2N$ 的图像复原成 $N \times N$ 的密文图像, 由此得到中间密文图像.

(9) 继续对 Tent 映射系统迭代$L = (N \times N)$次, 得到实数序列Q=$\{q_1, q_2, \cdots, q_L\}$, 对其中的每一个数取小数点后面的 1, 3, 5位组成 3 位数并对 256 取余得到整数序列I=$\{i_1, i_2, \cdots, i_L\}$, 将中间密文图像按行优先的顺序展开成一维序列$P$=$\{p_1, p_2, \cdots, p_L\}$. 对中间密文图像进行扩散操作, 采用的方式为

$$P'[i] = \begin{cases} P[i] \otimes S_0, & i = 1, \\ P[i] \otimes (P'[i-1] + I[i]) \otimes I[i], & i > 1, \end{cases} \tag{15}$$

其中, 加法为模 256 下的同余加法, S_0 为输入密钥. 最后把 P' 变换成二维数字图像矩阵, 便得到最终的密文图像. 图 3 所示为图像加密的流程.

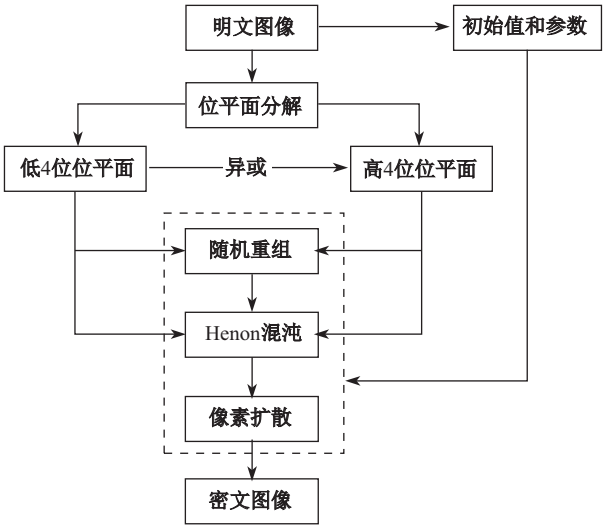


图 3 图像加密过程

Fig. 3 Diagram of the image encryption process

1.5 图像解密算法

加密算法中每一步均可逆, 故解密算法是加密算法的逆过程. 图像解密的流程如图 4 所示.

加密过程中的密钥有 α, S_0, x_0, S, 其中 α, S_0 和 x_0 为输入密钥, S为根据明文图像利用式(6)计算得到. 解密步骤如下.

(1) 设密文大小为$N \times N$, 根据式(7)和式(8)计算得出 μ 和 k, 以 μ 为式(1)Tent 映射系统的控制参数, x_0 为初始值, 迭代 k 次以消除初态影响.

(2) 通过对系统进行迭代得到实数 value_1, value_2, value_3, 实数序列T=$\{t_1, t_2, \cdots, t_8\}$和$Q$=$\{q_1, q_2, \cdots, q_L\}$, 其中$L = N \times N$, 利用式(10)–(14)分别算得 a_i, c_i, k_i, i=1, 2, 3 时的值和关于 8 个位平面的排列 R, 并由序列 Q 按加密过程所述方法得到整数序列I=$\{i_1, i_2, \cdots, i_L\}$.

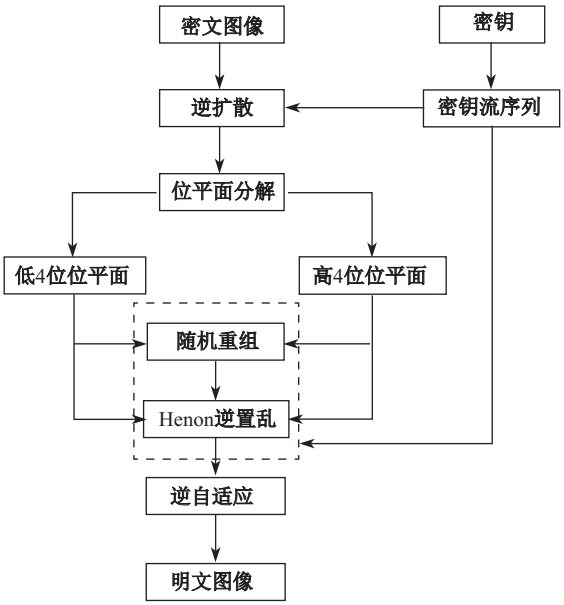


图 4　图像解密过程

Fig. 4　Diagram of the image decryption process

(3) 将密文图像按行优先展开为一维序列$P' = \{p'_1, p'_2, \cdots, p'_L\}$, 利用式(15)的逆运算

$$P[i] = \begin{cases} P'[i] \otimes I[i] \otimes (P'[i-1] + I[i]), & i > 1, \\ P'[i] \otimes S_0, & i = 1, \end{cases} \tag{16}$$

得到中间密文P, 该过程为逆扩散过程.

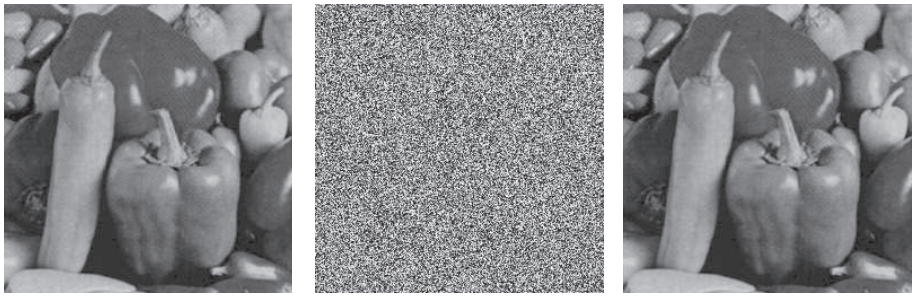
(4) 将中间密文 P 位平面分解得到 8 个位平面 $\text{bit}_i, i = 1, 2, \cdots, 8$, 根据 R 将 8 个位平面分成 4 组$\{R_8, R_1\}$, $\{R_7, R_2\}$, $\{R_6, R_3\}$, $\{R_5, R_4\}$, 并分别放到图 2 对应的方框 1, 2, 3, 4 中形成一个新平面 Y, 以a_3, c_3为控制参数, 利用式(4)所示的 Henon 逆映射对 Y 中的每个点进行置乱, 该过程为 Henon 逆置乱. 同理, 分别对高 4 位和低 4 位形成的平面进行 Henon 逆置乱过程.

(5) 自适应过程的逆过程, 即对于位平面进行一次式(9)的计算, 可得到原明文图像的位平面, 然后将 8 个位平面按正常顺序重组, 将每个像素点的 8 个 01 位表示转换成十进制数即得到最终的解密图像.

2　实验结果和性能分析

仿真实验的明文图像为灰度图像 Peppers, 大小为 256 × 256 像素, 其中初始值x_0 = 0.567 812 345 6, α = 0.987 654 321 01, S_0 = 155, 根据式(6)算得S = 7 714 235. 图 5(a)所示为明文图像 Peppers, 根据图 5(c)可知, 解密算法能够成功解密得到原图. 仿真实验以 Visual

Studio 2015+Opencv3.3 作为实验环境.



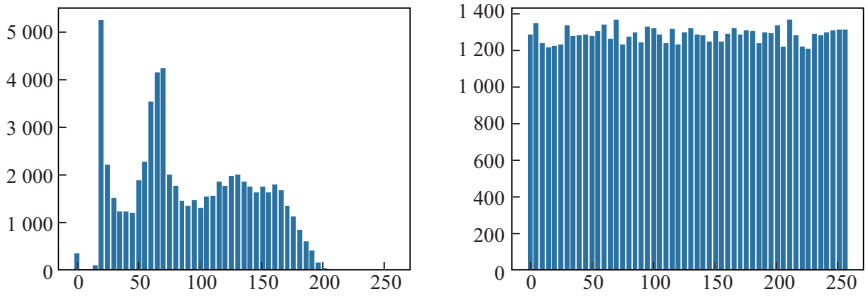
(a) **明文图像** (b) **密文图像** (c) **解密图像**

图 5 明文图像、密文图像和解密图像

Fig. 5 Plain image, cipher image, and decrypted image

2.1 直方图分析

像素分布直方图能直观地揭示图像中的像素值分布. 明文图像的像素值一般会分布不均, 攻击者可以利用统计特性攻击来获得图像中的有用信息; 而密文图像使得攻击者难以从中得到有价值的信息. 密文图像中的像素分布越均匀, 则加密算法越理想. 图像 Peppers 的明文和密文的像素分布直方图如图 6 所示.



(a) **明文图像像素直方图** (b) **密文图像像素直方图**

图 6 明文图像和密文图像的像素分布直方图

Fig. 6 Histogram of plain image and cipher image pixels

2.2 相关性分析

理想的加密算法需要降低明文图像中相邻像素间的强相关性, 该相关性可由相关系数定量描述. 计算图像相关系数的方法为, 在图像中随机选取 10 000 个像素点存入 X 序列, 再把这些点的相邻点存入 Y 序列, 两者的相关系数定义为

$$\rho_{x,y}=\frac{E\{[x-E(x)][y-E(y)]\}}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{17}$$

其中,

$$E(x)=\frac{1}{n}\sum_{i=1}^{n}x_i, \tag{18}$$

$$D(x)=\frac{1}{n}\sum_{i=1}^{n}[x_i-E(x)]^2. \tag{19}$$

在二维数字图像中, 图像的相邻关系包括: 水平、垂直、对角这 3 种. 表 1 所示为不同加密算法的明文图像和密文图像的相关性分析结果.

表 1　不同加密方法的相关性比较

Tab. 1　Correlation comparison between different encryption methods

图像	算法	水平	垂直	对角
Peppers	本文	0.005 06	-0.004 98	-0.004 11
	文献[10]	0.096 88	0.002 31	0.011 63
	文献[9]	0.003 00	0.005 40	0.006 50
	文献[7]	0.000 71	0.002 16	0.014 89
Lena	本文	–0.002 71	–0.004 07	0.008 80
	文献 [10]	0.003 63	–0.004 52	0.002 08
	文献 [9]	–0.001 9	–0.003 5	0.007 90
Couple	本文	0.000 35	–0.000 84	0.00164
	文献 [8]	0.002 40	–0.001 50	–0.008 10
Baboon	本文	0.005 15	0.000 30	0.003 13
Sailboat	本文	0.003 63	–0.000 26	–0.007 59

由表 1 可知, 密文图像的相关系数的绝对值非常接近于 0. 这也定量地说明了密文像素的弱相关性. 通过比较, 本文算法相关性方面的表现要优于其他文献中的加密算法.

除了利用相关系数表示之外, 以相邻两点的像素值分别作为横、纵坐标描绘出散点图, 该散点图也可以将其相关性直观地展示出来, 如图 7 所示. 图 7(a)、图 7(c)和图 7(e)分别是明文图像中具有水平、垂直和对角关系的相邻像素形成的散点图, 而与其对应的密文图像的散点图分别为图 7(b)、图 7(d)和图 7(f). 从图 7 中可知, 明文图像的相邻像素之间有明显的相关性, 而对于密文图像, 散点图中的点分布均匀, 毫无规律, 表明本文加密算法对破坏相关性有良好的表现.

2.3　信息熵分析

借助信息熵的概念能定量地描述图像像素的混乱程度和分布情况. 设 X 为随机变量, $X=\{x_1, x_2, \cdots, x_n\}$, $p(x_i)$为x_i的出现概率, 则关于 X 的信息熵 $H(X)$ 为

$$H(X)=-\sum_{i=1}^{n} p(x_i)\log_2 p(x_i), \tag{20}$$

其中, $p(x_i)\in(0,1)$, $p(x_i)$之和等于 1, 约定$0\cdot\log_2 0=0$.

由信息熵的表达式$H(X)$可知, X分布均匀时, 熵值达到最大值. 在灰度图像中像素值共有 256 个不同取值, X分布均匀等价于每个像素值在图像中出现的频率为 1/256, 此时熵的理想值为 8. 运用本文提出的加密方案进行加密得到的密文的熵值为 7.997 2, 这说明其加密效果良好. 表 2 展示的是对不同图像加密之后的信息熵值及与其他不同加密算法效果的对比.

2.4　密钥敏感性

密钥敏感性表示在密钥发生改变时密文所对应的变化程度. 计算密钥敏感性常采用的方法是让密钥的数值发生很微小的变化, 并计算两次加密的不同密文的差异, 这种差异称作灰度均方差. 灰度均方差的计算公式为

$$E_{C,C'}=\frac{1}{m\times n}\sum_{i=1}^{m}\sum_{j=1}^{n}(C_{ij}-C'_{ij})^2, \tag{21}$$

其中, C 为用原始密钥x_0, S_0, α进行加密得到的密文图像, C'为改变密钥之后再次加密得到的密文图像, m 和 n 分别表示图像的高度和宽度. 本文分别用改变x_0, S_0和α后的密钥来加密以得到不同的密文图像, 并和原始密文图像进行比较, 结果如表 3 所示.

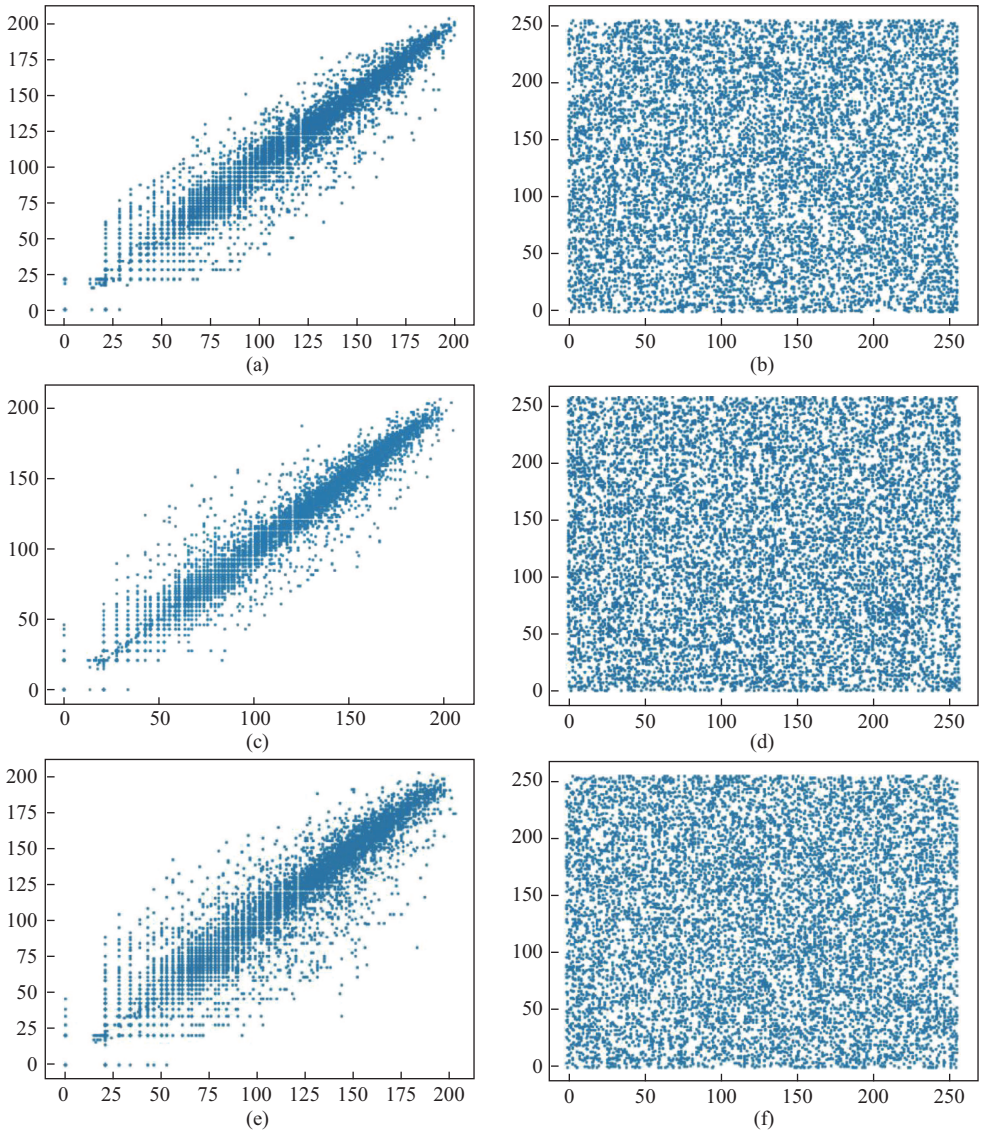


图 7 相邻像素散点图

Fig. 7 Adjacent pixel scatter plots

表 2 明文图像和密文图像的熵值

Tab. 2 Information entropy of plain image and cipher image

图像	算法	密文图像信息熵
Peppers	本文	7.997 2
	文献[10]	7.996 2
Lena	本文	7.999 3
	文献[7]	7.999 4
Couple	本文	7.992 0
	文献[8]	7.990 1
Baboon	本文	7.991 8
Sailboat	本文	7.991 3

表 3　灰度均方差

Tab. 3　Gray mean square error

密钥	灰度均方差
$(\widehat{x}_0, S_0, \alpha)$	10 876.15
$(x_0, \widehat{S}_0, \alpha)$	10 900.52
$(x_0, S_0, \widehat{\alpha})$	10 926.79

在表 3 中, $\hat{x}_0 = 0.567\,812\,345\,61$, $\hat{S}_0 = 156$, $\hat{\alpha} = 0.987\,654\,321\,02$, 即分别是由对$x_0$和$\alpha$增加 10^{-11}、对S_0增加 1 得到的.

2.5　明文敏感性

采用明文敏感性分析能够定量地评估明文图像发生微小变化对密文图像的影响. 明文敏感性分析方法主要包括像素个数变化率(NPCR)和归一化平均变化强度(Unified Average Changing Intensity, UACI).

(1) NPCR 的计算公式为

$$\mathrm{NPCR} = \frac{\sum\limits_{i,j} D(i,j)}{m \times n} \times 100\%, \tag{22}$$

其中 D 的定义为

$$D(i,j) = \begin{cases} 1, & C_1(i,j) \neq C_2(i,j), \\ 0, & C_1(i,j) = C_2(i,j), \end{cases} \tag{23}$$

其中, C_1 为将原始明文图像加密得到的密文图像, C_2 为将原始明文图像中的某个像素修改 1 位二进制位得到的新图像并进行加密得到的密文图像.

(2) UACI 的计算公式为

$$\mathrm{UACI} = \frac{1}{m \times n} \sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\%. \tag{24}$$

对于不同的图像, 通过随机修改某一像素点得到新的明文图像, 并对其加密得到密文图像, 最后计算其 NPCR 和 UACI, 结果如表 4 所示. 表 4 的实验结果可表明, 本文加密算法对明文信息变化敏感, 并能有效抵抗差分攻击.

2.6　密钥空间分析

密钥空间是衡量加密算法是否能够抵抗穷举攻击的指标, 密钥空间足够大才能抵御穷举攻击. 本文加密算法中的密钥包括用于产生混沌序列的初值 x_0、系统参数 α 以及扩散阶段的参数 S_0, 其中 x_0 和 α 是双精度实数, 64 bit 的计算机中双精度实数的有效精度可达到 10^{-14}, 故密钥空间大小为 $10^{14}\times10^{14}\times256>2^{100}$, 以目前计算机的计算能力, 该加密算法能有效抵御穷举攻击.

表 4　NPCR 和 UACI

Tab. 4　NPCR and UACI performance

图像	NPCR/%	UACI/%
Peppers	99.588	33.455
Lena	99.616	33.470
Couple	99.615	33.559
Baboon	99.631	33.373
Sailboat	99.600	33.658

3 总 结

本文所提出的图像加密算法是基于比特位层面的加密, 并且置乱是全局的, 因此能有效破坏图像数据分布的局部不均匀性. 同时, 本文算法中加入了自适应过程, 给一些攻击行为, 如选择明文攻击等, 带来了更大的挑战, 并且增强了加密算法的明文敏感性. 实验数据表明, 本文算法打破了其他基于像素层面的加密算法和基于比特位层面的局部固定置乱的加密算法的局限, 相比于其他算法获得了更好的效果. 本文分别从 NPCR、UACI、信息熵、明文敏感性等角度对所提算法进行了测试实验, 结果显示, 该加密算法都有更好的表现. 因此, 本文提出的加密算法有更高的安全性和较好的应用前景.

[参 考 文 献]

[1] ZHU Z L, ZHANG W, WONG K W, et al. A chaos-based symmetric image encryption scheme using a bit-level permutation [J]. Information Sciences, 2011, 181(6): 1171-1186.

[2] WONG K W, KWOK B S H, LAW W S. A fast image encryption scheme based on chaotic standard map [J]. Physics Letters A, 2008, 372(15): 2645-2652.

[3] ALVAREZ G, LI S. Some basic cryptographic requirements for chaos-based cryptosystems [J]. International Journal of Bifurcation and Chaos, 2006, 16(8): 2129-2151.

[4] ABDULLAH A H, ENAYATIFAR R, LEE M. A hybrid genetic algorithm and chaotic function model for image encryption [J]. AEU-International Journal of Electronics and Communications, 2012, 66(10): 806-816.

[5] WANG Y, WONG K W, LIAO X F, et al. A chaos-based image encryption algorithm with variable control parameters [J]. Chaos, Solitons & Fractals, 2009, 41(4): 1773-1783.

[6] YANG H Q, LIAO X F, WONG K W, et al. A new block cipher based on chaotic map and group theory [J]. Chaos, Solitons & Fractals, 2009, 40(1): 50-59.

[7] WANG Y, WONG K W, LIAO X F, et al. A new chaos-based fast image encryption algorithm [J]. Applied Soft Computing, 2011, 11(1): 514-522.

[8] 汪彦, 涂立. 基于改进Lorenz混沌系统的图像加密新算法 [J]. 中南大学学报(自然科学版), 2017, 48(10): 2678-2685.

[9] 毛骁骁, 孙克辉, 刘文浩. 基于分数阶统一混沌系统的图像加密算法 [J]. 传感器与微系统, 2017, 36(6): 138-141.

[10] PING P, LI J H, MAO Y C, et al. Image encryption algorithm based on chaotic maps and bit reconstruction [J]. Journal of Image and Graphics, 2017, 22(10): 1348-1355.

[11] ZHANG W, WONG K W, YU H, et al. A symmetric color image encryption algorithm using the intrinsic features of bit distributions [J]. Communications in Nonlinear Science and Numerical Simulation, 2013, 18(3): 584-600.

(责任编辑: 李 艺)