JPEG 比特流加密域大容量可逆数据隐藏

郑梦阳[1], 李增辉[2], 陈潭升[3], 董梦瑶[1], 和红杰[1], 陈 帆[1]*

[1]西南交通大学 信息科学与技术学院 成都 中国 611756
[2]西南交通大学 茅以升学院 成都 中国 611756
[3]西南交通大学 数学学院 成都 中国 611756

摘要 兼顾加密 JPEG 图像的隐藏容量和安全性, 提出一种 JPEG 比特流加密域可逆数据隐藏新方法。该算法设计了一种块间置乱与块内加密相结合的 JPEG 比特流加密方法, 不仅实现对图像块的伪随机置乱, 还实现了熵编码块的霍夫曼编码和扩展位的全加密, 降低信息泄露可能性的同时, 提高了算法抵抗唯密文攻击的能力。同时, 该算法生成的加密 JPEG 比特流与 JPEG 解码标准兼容, 解码得到的加密图像类似随机噪声且与原始图像大小相同, 所有图像块熵编码都可以用来隐藏附加信息, 有效解决了隐藏容量与安全性之间的矛盾。对比分析了算法的安全性、文件大小和隐藏容量等性能。实验仿真结果表明本文算法能有效抵抗唯密文攻击, 隐藏容量是现有最新同类算法的 4 倍以上。

关键词 可逆数据隐藏; 图像加密; JPEG 比特流; 隐藏容量; 唯密文攻击
中图法分类号 TP309 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2018.11.05

Larger- Capacity Reversible Data Hiding in Encrypted JPEG Bitstream

ZHENG Mengyang[1], LI Zenghui[2], CHEN Tansheng[3], DONG Mengyao[1], HE Hongjie[1], CHEN Fan[1*]

[1] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China
[2] School of Mao Yi Sheng Honors, Southwest Jiaotong University, Chengdu 611756, China
[3] School of Mathematics, Southwest Jiaotong University, Chengdu 611756, China

Abstract Considering the embedding capacity and security of encrypted JPEG images, a new method of reversible data hiding in encrypted JPEG bitstream is proposed. The proposed method designs a JPEG bitstream encryption method which combines inter-block scrambling and intra-block encryption. It not only realizes pseudo-random scrambling of entropy coding blocks, but also implements full encryption of entropy coding blocks including Huffman coding and extension bits. This reduces the possibility of information leakage and improves the ability of the algorithm to resist the only-cipher attack. Moreover, the encrypted JPEG bitstream generated by the proposed algorithm is compatible with the JPEG decoding standard, and the decoded encrypted image is similar to random noise and has the same size as the original image. This allows all entropy coding blocks in JPEG image to be used to hide additional information, effectively solving the contradiction between embedding capacity and security. Performances of the algorithm, including security, file-size of marked-encrypted JPEG image and embedding capacity are compared and analyzed. Experimental results demonstrate that the proposed algorithm is not vulnerable to the only-cipher attack and the embedding capacity is more than 4 times that of the latest similar algorithm.

Key words reversible data hiding, image encryption, JPEG bitstream, embedding capacity, only-cipher attack

1 引言

随着云计算的快速发展, 云端存储数字图像安全和隐私保护引起了广泛关注, 促生了图像加密域可逆信息隐藏(RDH-EI: Reversible Data Hiding in Encrypted Image)技术[1]。传统的明文域可逆信息隐藏技术在数字图像中隐藏附加信息以方便其管理, 同时在接收端, 授权接收者能够提取出附加信息并重建原始图像。与传统的明文域可逆信息隐藏技术相比, RDH-EI 技术不仅通过加密实现对云端数字图像隐私保护, 而且能在密文图像中隐藏附加信息。该技术为云、大数据等环境中的数据安全、隐私保护等

通讯作者: 陈帆, 博士, 副教授, Email: fchen@swjtu.cn。
本课题得到国家自然科学基金(No. 61872303, No. 61461047)和四川省科技厅科技创新人才计划(No. 2018RZ0143)的资助。
收稿日期: 2018-08-15; 修改日期: 2018-10-03; 定稿日期: 2018-10-10

提供有效技术手段, 在云存储、公共医疗、隐秘通信等领域有着很好的应用前景。

2011 年, 张新鹏[2]提出了一种联合 RDH-EI 方法。该算法利用流密码按位异或生成加密图像, 通过分块并翻转每块的最低有效位(LSB: least significant 比特)实现信息的隐藏, 利用定义的“波动函数”实现信息的提取和原始图像重建。为提高 RDH-EI 算法无损提取附件数据和重建原始图像的能力(即可逆性), 研究者利用边信息[3]、邻域相关性等, 提出了多种“波动函数优化”的 RDH-EI 算法[3-5]。不过, 上述算法的信息提取和图像解密必须同时进行(称其为联合的 RDH-EI), 不能在密文图像中直接提取附加信息, 因此, 难以满足不同的应用场合。此外, 算法的最大隐藏容量也需要进一步提高。与联合 RDH-EI 算法相对应, 可分离 RDH-EI 算法[6-15]不仅能实现在密文图像中无损地提取附加信息, 实现了加密者和隐藏者的工作相互独立, 提高了 RDH-EI 算法的实用性。同时, 可分离 RDH-EI 算法的可逆性、最大隐藏容量等性能也得到很大提高。例如, 现有可分离 RDH-EI 算法对不同图像的平均最大隐藏容量从小于 0.1 bpp [6], 扩展至 0.3 bpp [7-8]、0.5 bpp[9-11], 再进一步扩展至 1bpp[12-13]甚至更大[14-16]。上述算法的共同特点是利用对称密码系统生成加密图像, 因此加密密钥需要通过安全信道传输。为了进一步拓展应用范围, 研究者还提出了基于公钥密码系统的 RDH-EI 算法[17-19], 该类算法的安全性高, 密钥容易管理、隐藏容量较大 [17]。不过, 利用公钥密码系统进行加密时间复杂度较高, 并且密文数据存在着数据扩展的问题, 从而导致加密图像文件增大。

众所周知, 图像上传、下载和保存, 其效率都与图像的文件大小密切相关。图像压缩能有效减少自然图像的文件大小, 提高数字图像的存储与传输效率。不过, 由于加密图像类似随机噪声, 无论是无损压缩, 还是有损压缩都很难有效减小密文图像的文件大小。JPEG 是一种图像/视频有损压损标准压缩标准, JPEG 格式的图像文件尺寸较小, 被广泛应用于互联网和数码相机领域。因此, 基于 JPEG 比特流的 RDH-EI 算法 [20-25]引起了研究者的关注。2014 年文献[20]首次提出了一种基于 JPEG 比特流 RDH-EI 方法。该算法对熵编码的扩展位进行异或加密生成加密 JPEG 比特流, 加密过程既不改变霍夫曼编码(兼容 JPEG 编码标准), 也不增加码流长度。不过, 该算法不能抵抗文献[24]提出的唯密文攻击方法, 而且该算法的隐藏容量较小(512×512 图像的隐藏容量为 750 比特)。这主要是由于 JPEG 图像中存在的冗余信息较少, 而且加密后的 JPEG 比特流还必须与 JPEG 编解码兼容, 因此, JPEG 比特流 RDH-EI 算法研究极具挑战。目前研究者也提出了可分离 JPEG 比特流 RDH-EI 算法[21-22], 实现了加密者和隐藏者的工作相互独立, 且算法的安全性和隐藏容量也得到了提高。例如, 512×512 图像, 文献[21]和[22]隐藏容量的平均值分别为 1230 和 1110 比特。显然, 与空域图像的 RDH-EI 算法相比, 隐藏容量仍然较低, 该数量级的隐藏容量很难满足一些实际应用的要求。因此, 如何提高 JPEG 比特流 RDH-EI 算法的隐藏容量, 是 JPEG 比特流 RDH-EI 算法研究的难点和重点所在。

在上述研究的基础上, 2018 年, Qian 等人[23]提出了一种基于 JPEG 比特流 RDH-EI 新方法。该方法基于加密密钥将图像块随机分为两部分: (1) 加密图像块: 该类图像块的熵编码不用于隐藏附加信息, 而是将其熵编码(包括霍夫曼编码和扩展编码)异或加密后隐藏到 JPEG 图像的头文件中; (2) 携密图像块: 该类图像块用于隐藏附加信息, 将其熵编码的 DC 系数重新进行 DPCM 编码, 通过对 AC 系数的直方图平移和修改头文件中的 AC 系数熵编码表实现附加信息的隐藏。该算法通过对图像块随机划分实现图像块置乱, 提高算法抵抗唯密文攻击的能力; 同时, 对加密图像块熵编码全加密并隐藏在 JPEG 头文件中, 进一步提高了算法的安全性。不过, 该算法存在以下问题: (1) **存在安全性与隐藏容量之间的矛盾**: 加密图像块越多, 算法的安全性越高, 但隐藏容量越少。这是因为, 加密图像块的熵编码采用全加密, 用于隐藏信息的携密图像块熵编码扩展位不加密, 且霍夫曼编码不加密。这意味者, 携密图像块越多, 隐藏容量越大, 但没有加密的霍夫曼编码也越多。兼顾安全性和隐藏容量, 文献[23]的携密图像块约为图像的 1/4, 隐藏容量可达 3000 比特以上, 为文献[21]和[22]的两倍以上; (2) **存在信息泄露隐患:** 携密图像块熵编码的扩展位和霍夫曼编码没有加密, 这样可以保持与传统 JPEG 编解码的兼容, 但是, 霍夫曼以明文的形式存在使其存在着原图图像信息泄露的安全风险。

针对上述问题, 本文提出一种新的用于 JPEG 比特流的加密域可逆数据隐藏算法。为提高算法的隐藏容量, 所有图像块都做为携密图像块, 即隐藏在 JPEG 头文件的加密图像块个数为 0, 使加密图像大小与原始图像相同, 有效提高了算法隐藏容量。为了兼顾安全性、兼容性和码流长度, 本文提出一种块间置乱与块内加密相结合的 JPEG 比特流加密方法, 设计了一种块内 AC 系数置乱的加密方法, 不仅实现对

图像块的伪随机置乱, 还实现了对图像块熵编码的霍夫曼编码和扩展位的全加密, 有效解决了隐藏容量与安全性之间的矛盾。同时, 加密后的 JPEG 比特流满足 JPEG 解码标准, 能够被正确解码为 JPEG 图像, 还尽可能减少了加密 JPEG 比特流的增长。最后对算法的安全性、加密 JPEG 比特流增量和隐藏容量等性能进行了讨论分析。实验仿真结果表明, 本文算法提高了安全性, 同时具有较高的隐藏容量, 本文算法隐藏容量是文献[23]中算法的 4 倍以上。

2 图像块熵编码全加密

要使所有图像块都能携带附加信息, 同时通过对图像块的熵编码实现全加密, 保证加密 JPEG 比特流的原始信息不泄露。如此, 本文设计一种图像块熵编码全加密且兼容 JPEG 编解码的加密算法, 是本文算法的关键所在。为便于描述 JPEG 比特流的结构, 本文采用文献[23]中的字符缩写注释表, 如表 1 所示。

表 1 字符缩写注释表

Table 1 Character abbreviation comment table

缩写	解释
SOI	图像开始标记符
JH	JPEG 头文件
ECS	图像熵编码
DCC	DC 系数熵编码
DCH	DC 系数的霍夫曼编码
DCA	DC 系数的扩展位编码
ACC	AC 系数熵编码
ACH	AC 系数的霍夫曼编码
ACA	AC 系数的扩展位编码
EOI	图像结束标记符
EOB	图像块结束标记符

原始图像 X 的 JPEG 比特流 J 可用下式表示,

$$\mathrm{J}=\{\mathrm{SOI},\mathrm{JH},\mathrm{ECS}_1,\cdots,\mathrm{ECS}_n,\cdots,\mathrm{ECS}_{\mathrm{N}},\mathrm{EOI}\} \tag{1}$$

其中, ECS_n 表示第 n 个图像块的熵编码, N 为图像块的个数。每个图像块的熵编码 ECS_n 由一个 DC 系数熵编码、多个 AC 系数熵编码和结束标识组成。

$$\mathrm{ECS}_n=\{\mathrm{DCC}^{\langle n\rangle},\mathrm{ACC}^{\langle n,1\rangle},\cdots,\mathrm{ACC}^{\langle n,i\rangle},\cdots,\mathrm{ACC}^{\langle n,l\rangle},\mathrm{EOB}\} \tag{2}$$

其中, 图像块 DC 和 AC 系数熵编码均包括霍夫曼编码和扩展位编码两部分, 即

$$\mathrm{DCC}^{\langle n\rangle}=\{\mathrm{DCH}^{\langle n\rangle},\mathrm{DCA}^{\langle n\rangle}\} \tag{3}$$

$$\mathrm{ACC}^{\langle n,i\rangle}=\{\mathrm{ACH}^{\langle n,i\rangle},\mathrm{ACA}^{\langle n,i\rangle}\} \tag{4}$$

为了使加密后的 JPEG 比特流与 JPEG 解码标准兼容, 即加密 JPEG 比特流能够解码为类似随机噪声的图像, 文献[20]只对图像块的 DCA 和 ACA 进行加密, 而 DCH 和 ACH 没有加密。既然图像块中存在不被加密的部分, 因此该加密方法存在一定的安全隐患。以熵编码“00 1”为例进行说明, 其中“00”为熵编码的霍夫曼编码, “1”为熵编码扩展位编码, 使用加密密钥仅对扩展位异或加密得到的熵编码密文为“00 0”。由于霍夫曼编码并未变化, 因此在解码过程中, 对“00”解码获得对应的游程编码(R, S)为(0, 1), 根据 JPEG 标准 VLI 编码表, 如表 2 所示, S 为 1 时对应的扩展位取值范围为 1 或–1, 即攻击者有 1/2 的概率正确获得该熵编码扩展位的值, 因此存在被攻击者暴力攻破的安全隐患。

为使图像块熵编码中霍夫曼和扩展位编码实现全加密, 且能利用 JPEG 解码标准得到类随机噪声的解码图像, 本文提出一种对图像块熵编码全加密方法, 包括以下五个步骤:

***Step* 1:** 对某个图像块的熵编码 ECS_n 进行解码, 得到该图像块的 DCT 系数 C_n,

$$\mathrm{C}_n=\{\mathrm{DC}_n,\mathrm{AC}_{n,1},\cdots,\mathrm{AC}_{n,t},\cdots,\mathrm{AC}_{n,63}\} \tag{5}$$

C_n 是由一个 DC 系数 DC_n 和 63 个 AC 系数 $\mathrm{AC}_{n,t}$ 构成, 将该图像块最后一个非 0AC 系数在 C_n 中的位置记为 T^{n}。

***Step* 2:** 对该图像块中的部分 AC 系数 $\{\mathrm{AC}_{n,1},\cdots,\mathrm{AC}_{n,\mathrm{T}^n}\}$ 进行分组, 得到分组序列 B_n。

$$\mathrm{B}_n=\{\mathrm{B}_{n,1},\cdots,\mathrm{B}_{n,r},\cdots,\mathrm{B}_{n,\mathrm{R}^n}\} \tag{6}$$

其中, $\mathrm{R}^n\leqslant\mathrm{T}^n$, 并且根据下式对 AC 系数 C_n 中前 T^n 个系数进行分组, 即将一个或连续多个为 0 的 AC 系数分为一组, 非 0 的 AC 系数单独作为一组, 最终生成 R^n 个系数分组 $\mathrm{B}_{n,r}$。

$$\mathrm{B}_{n,r}=\begin{cases}\left\langle \mathrm{AC}_{n,t+1},\cdots,\mathrm{AC}_{n,t+x}\right\rangle, & \text{if连续}x\text{个AC为0}\\ \left\langle \mathrm{AC}_{n,t}\right\rangle, & \text{otherwise}\end{cases} \tag{7}$$

***Step* 3:** 基于加密密钥对分组序列 B_n 中前 R^n–1 个系数分组 $\mathrm{B}_{n,r}$ 进行置乱加密。

首先, 对前 R^n-1 个系数分组 $\mathrm{B}_{n,r}$ 中非 0 的 AC 系数分组基于加密密钥进行随机置乱, 然后, 将剩余为 0 的 AC 系数分组分别随机插入在置乱后的非 0

的 AC 系数分组之前, 最后, 得到加密分组序列 $\tilde{\mathrm{B}}_n$。

$$\tilde{\mathrm{B}}_n=\left\{\tilde{\mathrm{B}}_{n,1},\cdots,\tilde{\mathrm{B}}_{n,r},\cdots,\mathrm{B}_{n,\mathrm{R}^n}\right\} \tag{8}$$

为了更好理解置乱加密方法, 以下举例进行说明。假如该图像块的 DCT 系数 $\mathrm{C}_n=\{-3,0,4,5,8,1,13,0,\cdots,0\}$, 则 DC 系数为 -3, 最后一个非零 AC 系数为 13, $\mathrm{R}^n=\mathrm{T}^n=6$。该图像块中的部分 AC 系数为 $\{0,4,5,8,1,13\}$, 并对该部分的 AC 系数进行分组, 由于该部分系数中不存在连续为 0 的 AC 系数, 故分组序列为 $\mathrm{B}_n=\{\langle 0\rangle,\langle 4\rangle,\langle 5\rangle,\langle 8\rangle,\langle 1\rangle,\langle 13\rangle\}$, 然后, 基于加密密钥对分组序列 B_n 中前 5 个分组进行随机置乱, 得到加密分组序列为 $\tilde{\mathrm{B}}_n=\{\langle 8\rangle,\langle 5\rangle,\langle 0\rangle,\langle 4\rangle,\langle 1\rangle,\langle 13\rangle\}$。

若该图像块的 DCT 系数 $\mathrm{C}_n=\{-3,0,0,5,0,8,0,0,0,6,13,0,\cdots,0\}$, 则 DC 系数为 -3, 最后一个非零 AC 系数为 13, $\mathrm{R}^n=7,\mathrm{T}^n=9$。该图像块中的部分 AC 系数为 $\{0,0,5,0,8,0,0,0,6,13\}$, 并对该部分的 AC 系数进行分组, 连续为 0 的 AC 系数分为一组, 其余单独作为一个分组, 故分组序列为 $\mathrm{B}_n=\{\langle 0,0\rangle,\langle 5\rangle,\langle 0\rangle,\langle 8\rangle,\langle 0,0,0\rangle,\langle 6\rangle,\langle 13\rangle\}$, 然后基于加密密钥对分组序列中前 6 个分组中的非 0 的 AC 系数分组($\{\langle 5\rangle,\langle 8\rangle,\langle 6\rangle\}$)进行随机置乱, 得到置乱后的非 0AC 系数分组序列为 $\{\langle 6\rangle,\langle 8\rangle,\langle 5\rangle,\langle 13\rangle\}$, 最后将剩余为 0 的系数分组($\{\langle 0,0\rangle,\langle 0\rangle,\langle 0,0,0\rangle\}$), 随机插入到置乱后的非 0AC 系数分组之前, 得到加密分组序列为 $\tilde{\mathrm{B}}_n=\{\langle 0\rangle,\langle 6\rangle,\langle 0,0\rangle,\langle 8\rangle,\langle 0,0,0\rangle,\langle 5\rangle,\langle 13\rangle\}$;

***Step* 4:** 将上述该图像块熵编码的加密分组序列 $\tilde{\mathrm{B}}_n$ 中所包含的所有 AC 系数与 $63-\mathrm{T}^n$ 个为 0 的 AC 系数依次拼接, 生成随机置乱系数 $\tilde{\mathrm{C}}_n$, $\tilde{\mathrm{C}}_n=\{\mathrm{AC}'_1,\mathrm{AC}'_2,\cdots,\mathrm{AC}'_{Tn-1},\mathrm{AC}_{Tn}\cdots\mathrm{AC}_{63}\}$, 并通过下式得到置乱 AC 系数 C'_n,

$$\mathrm{C}'_n=\begin{Bmatrix}\mathrm{C}_n,\mathrm{T}^n=0.1\\ \tilde{\mathrm{C}}_n,1<\mathrm{T}^n\leqslant 63\end{Bmatrix} \tag{9}$$

然后, 对置乱 AC 系数 C'_n 进行熵编码, 并与图像块的熵编码 ECS_n 中的 DC 系数熵编码 $\mathrm{DCC}^{\langle n\rangle}$ 和图像块结束标记符组成系数置乱熵编码 $\mathrm{ECS}_n^{\mathrm{T}}$。

$$\mathrm{ECS}_n^{\mathrm{T}}=\left\{\mathrm{DCC}^{\langle n\rangle},\mathrm{ACC}^{\mathrm{T}\langle n,1\rangle},\cdots,\mathrm{ACC}^{\mathrm{T}\langle n,i\rangle},\cdots\mathrm{ACC}^{\mathrm{T}\langle n,l\rangle},\mathrm{EOB}\right\} \tag{10}$$

***Step* 5:** 基于加密密钥对置乱熵编码 $\mathrm{ECS}_n^{\mathrm{T}}$ 中 DC 系数的扩展位 $\mathrm{DCA}^{\langle n\rangle}$ 和 AC 系数的扩展位 $\mathrm{ACA}^{\mathrm{T}\langle n,l\rangle}$ 按位异或加密, 生成该图像块的全加密熵编码 ECS'_n。

$$\mathrm{ECS}'_n=\left\{\mathrm{DCC}'^{\langle n\rangle},\mathrm{ACC}'^{\langle n,1\rangle},\cdots,\mathrm{ACC}'^{\langle n,l\rangle},\cdots,\mathrm{ACC}'^{\langle n,l\rangle},\mathrm{EOB}\right\} \tag{11}$$

本节中图像块熵编码全加密的方法不仅对图像块熵编码中的扩展为进行加密, 而且, 由于对图像块中的 AC 系数进行了置乱, 改变了其 AC 系数的霍夫曼编码内容, 从而实现了对 AC 系数霍夫曼编码内容的加密。

3 算法

本章提出一种大容量的 JPEG 比特流 RDH-EI 算法, 该算法主要分为加密, 隐藏, 提取, 解密四个过程, 算法的具体流程如图 1 所示。

3.1 基于图像块熵编码全加密的 JPEG 比特流加密

对 JPEG 比特流加密, 主要是通过对图像块熵编码全加密和图像块熵编码置乱两个部分来实现。

***Step* 1:** 内容拥有者根据加密密钥, 按照 2 中的算法对原始 JPEG 比特流 J 中所有的图像块熵编码 $\mathrm{ECS}_n,n=1,2,\cdots,\mathrm{N}$ 进行图像块熵编码全加密。

然后, 将图像开始标记符 SOI, JPEG 头文件 JH, 所有图像块的全加密熵编码 ECS'_n 以及图像结束标记符 EOI 依次拼接, 生成熵编码全加密比特流 J'。

$$\mathrm{J}'=\left\{\mathrm{SOI},\mathrm{JH},\mathrm{ECS}'_1,\cdots,\mathrm{ECS}'_n,\cdots,\mathrm{ECS}'_{\mathrm{N}},\mathrm{EOI}\right\} \tag{12}$$

***Step* 2:** 基于加密密钥对所有图像块的全加密熵编码 ECS'_n 进行随机置乱, 由于置乱后的图像块熵编码的 DC 系数随机分布, 不符合 JPEG 编码中 DPCM(差分脉冲调制编码)编码, 故需要提取置乱后所有图像块的 DC 系数编码进行解码, 得到所有的 DC 系数 $d=\{d_1,d_2,d_3,\cdots,d_N\}$, 并根据下式(13)重新计算,

$$d_n^s=\begin{cases}d_1\\ d_n-d_{n-1}\end{cases},n=1,2,3,\cdots,\mathrm{N} \tag{13}$$

然后, 对 d_n^s 进行霍夫曼编码, 得到新的 DC 系数熵编码 $\mathrm{DCC}^{s\langle n\rangle}$,

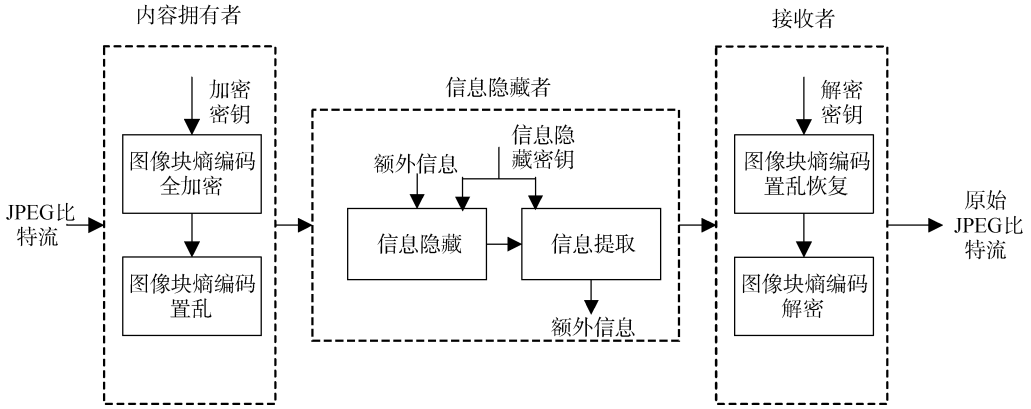


图 1　算法流程图

Figure 1　Algorithm flow chart

$$\mathrm{DCC}^{s\langle n\rangle}=\left\{\mathrm{DCH}^{s\langle n\rangle},\mathrm{DCA}^{s\langle n\rangle}\right\} \tag{14}$$

并替换原有的 DC 系数熵编码, 生成图像块熵编码密文 ECS_n^*。

$$\mathrm{ECS}_n^*=\left\{\mathrm{DCC}_1^{*\langle n\rangle},\mathrm{ACC}^{*\langle n,1\rangle},\mathrm{ACC}^{*\langle n,2\rangle},\cdots,\mathrm{ACC}^{*\langle n,l\rangle},\mathrm{EOB}\right\} \tag{15}$$

最后将图像开始标记符 SOI, JPEG 头文件 JH, 所有的图像块熵编码密文 ECS_n^* 以及图像结束标记符 EOI 依次拼接, 得到密文比特流 J^*。

$$\mathrm{J}^*=\left\{\mathrm{SOI},\mathrm{JH},\mathrm{ECS}_1^*,\cdots,\mathrm{ECS}_n^*,\cdots,\mathrm{ECS}_{\mathrm{N}}^*,\mathrm{EOI}\right\} \tag{16}$$

3.2　信息隐藏

文献[23]提出一种 JPEG 比特流的直方图平移的隐藏方法, 该方法主要根据 AC 系数的 RLC 编码(游程/算术编码)平移 AC 系数的值, 也即是改变 AC 系数熵编码扩展位的值, 实现附加信息的隐藏。

对每个密文图像块熵编码中的 AC 系数熵编码进行解码得到:

$$\mathrm{ACH}^{*\langle n,i\rangle}\rightarrow\mathrm{R}^{\langle n,i\rangle}/\mathrm{S}^{\langle n,i\rangle} \tag{17}$$

$$\mathrm{ACA}^{*\langle n,i\rangle}\rightarrow\mathrm{V}^{\langle n,i\rangle} \tag{18}$$

$\mathrm{R}^{\langle n,i\rangle}$ 为两个相邻非零 AC 系数之间 0 的个数, $\mathrm{S}^{\langle n,i\rangle}$ 为 AC 系数的霍夫曼编码 $\mathrm{ACH}^{*\langle n,i\rangle}$ 在 JPEG 霍夫曼编码表中对应的码长, $\mathrm{V}^{\langle n,i\rangle}$ 为 AC 系数值的编码。

云端使用信息隐藏密钥对长度为 L_u 的附加信息 U_0 按位异或加密, 得到附加信息 $\mathrm{U},\mathrm{U}=\left\{u_1,u_2,\cdots,u_k,\cdots,u_{L_u}\right\}$。采用文献[23]中方法通过修改 $\mathrm{V}^{\langle n,i\rangle}$, 将附加信息 U 隐藏到每个图像块熵编码中,

$$\mathrm{V}_m^{\langle n,i\rangle}=\begin{cases}\mathrm{V}^{\langle n,i\rangle}-1,\mathrm{V}^{\langle n,i\rangle}<-1\cap R^{\langle n,i\rangle}=0\\ \mathrm{V}^{\langle n,i\rangle}-u_k,\mathrm{V}^{\langle n,i\rangle}=1\cap R^{\langle n,i\rangle}=0\\ \mathrm{V}^{\langle n,i\rangle}+u_k,\mathrm{V}^{\langle n,i\rangle}=1\cap R^{\langle n,i\rangle}=0\\ \mathrm{V}^{\langle n,i\rangle}+1,\mathrm{V}^{\langle n,i\rangle}>1\cap R^{\langle n,i\rangle}=0\\ \mathrm{V}^{\langle n,i\rangle},\quad\text{otherweise}\end{cases} \tag{19}$$

记该 JPEG 比特流最大隐藏容量为 β, 则有,

$$\beta=\mathrm{num}(\mathrm{R}^{\langle n,i\rangle}=0\&\mathrm{V}^{\langle n,i\rangle}=\pm1),n\leqslant\mathrm{N},i\leqslant l \tag{20}$$

也即是最大隐藏容量 β 的值等于在 JPEG 比特流中所有 AC 系数 RLC 编码为 (0,±1) 的个数。

最后, 将携密 AC 系数重新熵编码, 得到携密图像块熵编码 ECS_n^m,

$$\mathrm{ECS}_n^m=\left\{\mathrm{DCC}_1^{\langle n\rangle},\mathrm{ACC}_m^{\langle n,1\rangle},\cdots,\mathrm{ACC}_m^{\langle n,1\rangle},\mathrm{EOB}\right\} \tag{21}$$

将图像开始标记符 SOI, JPEG 头文件 JH, 所有的携密图像块熵编码 ECS_n^m 以及图像结束标记符 EOI 依次拼接得到携密比特流 J_m。

$$\mathrm{J}_m=\left\{\mathrm{SOI},\mathrm{JH},\mathrm{ECS}_1^m,\mathrm{ECS}_2^m,\cdots,\mathrm{ECS}_{\mathrm{N}}^m,\mathrm{EOI}\right\} \tag{22}$$

3.3　信息提取

云端对携密比特流 J_m 中的携密图像块熵编码 ECS_n^m 中的 AC 系数熵编码 $\mathrm{ACC}^{m\langle n,i\rangle}$ 进行解码,

$$\mathrm{ACC}^{m\langle n,i\rangle}=\left\{\mathrm{ACH}^{m\langle n,i\rangle},\mathrm{ACA}^{m\langle n,i\rangle}\right\},i\leqslant l \tag{23}$$

$$\mathrm{ACH}^{m\langle n,i\rangle}\rightarrow\mathrm{R}_m^{\langle n,i\rangle}/\mathrm{S}_m^{\langle n,i\rangle} \tag{24}$$

$$\mathrm{ACA}^{m\langle n,i\rangle}\rightarrow\mathrm{V}_m^{\langle n,i\rangle} \tag{25}$$

其中, $\mathrm{R}_m^{\langle n,i\rangle}$ 为相邻非零 AC 系数之间 0 的个数, $\mathrm{S}_m^{\langle n,i\rangle}$ 为

AC 系数的霍夫曼编码 $\mathrm{ACH}^{m\langle n,i\rangle}$ 在霍夫曼编码表中对应的码长, $\mathrm{V}_m^{\langle n,i\rangle}$ 为 AC 系数值的大小。

并通过下式提取附加信息 U, $\mathrm{U}=\{u_1,u_2,\cdots,u_{L_u}\}$,

$$u_k=\begin{bmatrix}0, \left|\mathrm{V}_m^{\langle n,i\rangle}\right|=1\cap \mathrm{R}_m^{\langle n,i\rangle}=0\\ 1, \left|\mathrm{V}_m^{\langle n,i\rangle}\right|=2\cap \mathrm{R}_m^{\langle n,i\rangle}=0\end{bmatrix} \tag{26}$$

然后使用信息隐藏密钥对提取到的附加信息 U 按位异或解密, 得到长度为 L_u 的附加信息 U_0;

3.4 JPEG 比特流解密

如图 1 所示, 对 JPEG 比特流解密主要分为图像块熵编码置乱恢复和图像块熵编码解密两个部分。

由于在信息隐藏过程中, 使用直方图平移的方法对 JPEG 比特流中所有的 AC 系数进行了修改, 所以在进行上述操作之前, 接收者要根据下式, 恢复密文图像块熵编码 ECS_n^* 中 AC 系数的扩展位 $\mathrm{ACA}^{*\langle n,i\rangle}$ 的值 $\mathrm{V}^{\langle n,i\rangle}$,

$$\mathrm{V}^{\langle n,i\rangle}=\begin{cases}\mathrm{V}_m^{\langle n,i\rangle}+1, \mathrm{V}_m^{\langle n,i\rangle}<-2\cap \mathrm{R}_m^{\langle n,i\rangle}=0\\ -1,-2\leqslant \mathrm{V}_m^{\langle n,i\rangle}\leqslant -1\cap \mathrm{R}_m^{\langle n,i\rangle}=0\\ +1,1\leqslant \mathrm{V}_m^{\langle n,i\rangle}\leqslant 2\cap \mathrm{R}_m^{\langle n,i\rangle}=0\\ \mathrm{V}_m^{\langle n,i\rangle}-1, \mathrm{V}_m^{\langle n,i\rangle}>2\cap \mathrm{R}_m^{\langle n,i\rangle}=0\\ \mathrm{V}_m^{\langle n,i\rangle}, \quad \text{其他}\end{cases} \tag{27}$$

将 AC 系数值 $\mathrm{V}^{\langle n,i\rangle}$ 重新进行熵编码, 即恢复得到密文图像块熵编码中 AC 系数的熵编码 $\mathrm{ACC}^{*\langle n,i\rangle}$, 进而可恢复出密文图像块熵编码 ECS_n^*。

最后将图像开始标记符 SOI, JPEG 头文件 JH, 所有的图像块熵编码密文 ECS_n^* 以及图像结束标记符 EOI 依次拼接, 得到密文比特流 J^*。

***Step* 1:** 在密文比特流 J^* 中, 解码所有的图像块 DC 系数的熵编码, 得到 JPEG 密文比特流中所有的 DC 系数 $d^s=\{d_1^s,d_2^s,d_3^s,\cdots,d_N^s\}$ 并通过下式进行 DPCM 解码, 恢复原始的 DC 系数值。

$$d_n\begin{cases}d_1^s\\ d_n^s+d_{n-1}^s\end{cases},n=1,2,3,\cdots,N \tag{28}$$

将差分解码 DC 系数 d_n 进行熵编码并替换密文图像块熵编码 ECS_n^* 中的 DC 系数熵编码, 然后, 根据加密密钥对所有的图像块熵编码密文进行置乱恢复。

***Step* 2:** 基于加密密钥对密文 JPEG 比特流 J^* 中所有图像块熵编码进行第 2 节的逆过程, 对图像块的全加密熵编码 ECS_n' 解密, 得到原始比特流 J, 接收者解码即可得到原始图像 X。

4 实验结果

为验证本文算法的性能, 以量化因子 Q=80、大小为 512×512 的 Lena, Peppers, Baboon, Man, Lake 和 Airplane 六幅灰度图像(如图 2 所示)为测试图像, 从加密算法的安全性、密文 JPEG 图像文件大小和最大隐藏容量三个方面进行分析与讨论。

4.1 安全性

文献[24]中提出了一种针对每个图像块熵编码, 使用 AC 系数的霍夫曼编码来分析原始图像轮廓的攻击方法, 该分析方法在密码学中属于唯密文攻击。由于文献[20]的 JPEG 比特流加密算法只是对图像块熵编码的扩展位进行加密, 并没有对图像块熵编码进行置乱, 因此保留了原始图像块的位置信息, 所以使用文献[24]中的攻击方法对 JPEG 比特流中的每个图像块熵编码进行攻击能够获取该图像的轮廓信息, 因此, 采用对只对图像块熵编码扩展位加密不能够抵挡该攻击。

为使生成的加密 JPEG 比特流与 JPEG 编码标准兼容, 文献[20]的 JPEG 比特流加密算法只对图像块熵编码的扩展位进行加密。熵编码中的霍夫曼编码没有加密且图像块位置保持不变。针对文献[20]的加密方法, 文献[24]提出了一种利用 AC 系数的霍夫曼编码, 分析并推断原始图像轮廓的攻击方法, 该分析方法属于唯密文攻击。

为提高算法抵抗唯密文攻击的能力, 本文算法从设计的加密算法包括以下两个部分: 图像块熵编码全加密和图像块熵编码块置乱。

(1) 图像块熵编码全加密

由第二部分可知, 本文设计的图像块熵编码全加密算法不仅采用与现有算法相同方法(按位异或)加密了编码的扩展位, 而且设计了一种低码流增长的 AC 系数置乱方法, 不仅实现对霍夫曼编码的加密, 而且与 JPEG 编码标准兼容。

下面首先分析仅对熵编码块扩展位加密存在的安全隐患。由 JPEG 图像编码标准可知, 图像块的 DCT 系数矩阵经过 zig-zag 扫描后, 采用游程编码(RLC)对 AC 系数进生成的编码序列(R,W), 其中 R 为两个相邻非零 AC 系数之间 0 的个数, W 为该 AC

系数的值。然后根据 JPEG 标准的 VLI 编码表, 将生成的 (R,W) 转换为中间格式 ((R,S),V) ,其中 V 为 AC 系数值, S 为对应的编码长度。

为便于理解, 表 2 给出了 JPEG 标准的 VLI 编码表的一部分, 详见参考文献[26]。下面以 RLC 编码 (0,2)为例介绍 AC 系数编码过程。由表 2 可知, V 为 2 的系数对应码长 S 为 2, 因此 RLC 编码(0,2)转换为中间格式 ((0.2),2) 。查 JPEG 头文件中 AC 系的霍夫曼编码表可知(0,2)的霍夫曼编码为 “01”, 扩展位 2 查 VLI 编码表可知为“10”。因此 RLC 编码(0,2)的编码为熵编码为“0110”, 其中前面的 01 为霍夫曼编码, 后面的 10 为扩展位编码。假设该 AC 系数的霍夫曼编码 01 未发生改变, 则扩展位异或加密后仅有四种可能取值 00, 01, 10 和 11, 查找 VLI 编码表可知 S 对应的扩展位值有–3, –2, 2 和 3, 即是攻击者有 1/4 的概率破解该 AC 系数的扩展位。更重要的是, 如果霍夫曼编码不变, 非 0AC 系数在 DCT 矩阵中的位置是固定的, 因此, 霍夫曼编码的不加密而仅对扩展位加密, 存在严重的安全隐患。

相反, 本文将每个图像块熵编码中的 AC 系数基于加密密钥进行了随机置乱, 使得在所有的图像块熵编码中, AC 系数的 RLC 编码内容 (R,W) 发生了改变, 进而致使每个 RLC 编码所对应的 (R,S),V 发生变化。所以 (R,S) 对应的霍夫曼编码发生了变化, 也即实现了对 AC 系数熵编码中霍夫曼码的加密, 同时又满足了 JPEG 标准中的编解码规范, 能够正确将 AC 系数置乱后的比特流解码为图像。

表 2 JPEG 标准 VLI 编码表

Table 2 JPEG standard VLI code table

V	S	VLI 编码
0	0	–
–1,1	1	0,1
–3, –2,2,3	2	00,01,10,11
–7,-6,⋯,-4,4,⋯,6,7	3	000,⋯,111
–15,⋯, –8,8,⋯,15	4	0000,⋯,1111
⋯	⋯	⋯
–32767,⋯, –16384,16384,⋯,32767	15	0000000000000000,⋯, 111111111111111

为了更好说明 AC 系数对图像块熵编码的改变, 以 Lena 图像中一个图像块熵编码为例进行说明。该图像块的 DCT 系数为 $\{-2,4,2,-1,-1,1,1,0,1,0,0,0,0,0,1,-1,1,\cdots,0\}$ 序列, 该序列的中间格式为 $\{\langle 2\rangle,-2,\langle 0.3\rangle,4,\langle 0.2\rangle,2,\langle 0.1\rangle,-1\langle 0.1\rangle,-1,\langle 0.1\rangle,1\langle 0,1\rangle,1,\langle 1,1\rangle,1,\langle 4,1\rangle,1,\langle 0,1\rangle,-1,\langle 0,1\rangle,1,\langle 0,0\rangle\}$, 其中 $\langle 2\rangle$,–2 为 DC 系数 –2 的中间格式, 其余为 AC 系数的中间格式, $\langle 0,0\rangle$ 为图像块编码结束标志 EOB。则根据查亮度 DC 系数表, 亮度 AC 系数表, VLI 编码表可知该图像块熵编码为 “0110110010001100000000010011100111101110000011010”。而采用本文所提出的 AC 系数置乱方法得到的置乱系列为, $\{-2,0,2,1,-1,1,0,0,0,0,4,1,-1,1,-1,1,\cdots,0\}$ 同样对该置乱序列进行熵编码得到的图像块熵编码内容为“0110111011100010000011111111111111001011010000100000100000011010”。从而可以看出采用本文所提出的置乱方法对图像块熵编码内的 AC 系数进行置乱, 能够实现对熵编码中霍夫曼编码的加密。

同时, 通过该置乱方法得到的熵编码能够满足 JPEG 的解码标准, 也即能够将加密后的比特流正确解码为图像。

(2) 图像块熵编码置乱

此外, 在 JPEG 比特流加密后, 对所有的图像块熵编码进行随机置乱, 对图像块熵编码的置乱方法保护了该图像块的位置信息, 从而提高了 JPEG 比特流加密算法的安全性。

以 Lena, Baboon 和 Man 灰度图像为例, 其加密图像如图 3 所示, 从图 3 中可以看出, 加密图像大小与原始图像大小相等(512×512), 并且加密效果类似随机噪声, 加密图像内容是不被认知的, 从而说明使用本文加密算法进行加密, 加密效果较好。

从以上两个方面进行验证, 根据密钥对图像块熵编码内的 AC 系数进行随机置乱, 能有效的改变每个图像块熵编码中的霍夫曼编码, 同时, 通过该置乱方法得到的熵编码能够满足 JPEG 的解码标准, 也即能够将加密后的比特流正确解码为图像。此外, 对 JPEG 比特流中所有的图像块熵编码进行随机置乱, 对图像块熵编码的置乱方法保护了该图像块的位置信息, 从而提高了 JPEG 比特流加密算法的安全性。

4.2 加密图像文件大小

为了验证使用本文算法在 JPEG 比特流加密, 和信息隐藏过程中对图像文件大小变化的影响, 本文从加密图像和携密图像文件大小两个方面通过实验进行验证。

为了比较文献[23]和本文算法加密图像文件大小的变化, 本节采用图 2 中量化因子 Q=80 的测试图像进行实验仿真, 加密图像文件大小对比结果如表 3 所示。

图 2　测试图像. (a)Lena,(b)Peppers, (c)Baboon, (d)Man, (e)Lake, (f)Airplane

Figure 2　test image. (a)Lena,(b)Peppers, (c)Baboon, (d)Man, (e)Lake, (f)Airplane

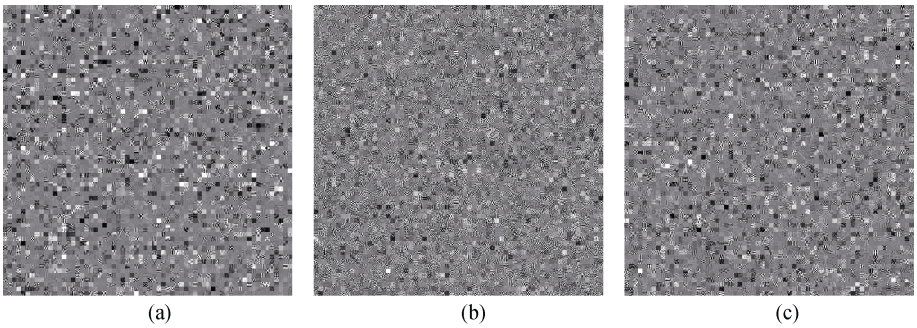


图 3　加密图像 (a)Lena 加密图像, (b)Baboon 加密图像, (a)Man 加密图像

Figure 3　encrypted images. (a)Lena encrypted image,(b)Baboon encrypted image,(c)Man encrypted image

表 3　加密图像文件大小对比

Table 3　Encrypt image file size comparison

测试图像	原始图像大小(KB)	文献[23](KB)	本文算法(KB)
Lena	37.0	37.1	39.5
Peppers	38.4	38.5	41.0
Baboon	76.8	76.7	82.3
Man	48.2	48.3	51.6
Lake	50.9	51.0	54.5
Airplane	37.8	37.8	40.3

表 3 中对加密图像文件大小对比显示, 本文算法加密图像的文件大小要稍高于文献[23]中的加密图像, 平均增长 3.5KB。由于在加密过程中进行了 AC 系数的置乱, 故导致加密图像文件大小的增长, 但是由于 AC 系数置乱实现了对 JPEG 比特流在加密过程中安全性的提高, 故传输该加密图像所占用信道资源小范围的有所增加是可以接受的。

此外, 为了验证本文所提出的 AC 系数置乱方法是一种低码流增长的 AC 系数置乱方法, 以 DCT 序列 $\{-3,0,1,0,0,2,3,4,0,\cdots,0\}$ 为例进行说明, 如果根据加密密钥对该序列的部分 AC 系数 $\{0,1,0,0,2,3,4\}$ 中除最后一个非零 AC 系数外的所有的 AC 系数进行置乱, 得到置乱后的 DCT 序列为 $\{-3,2,0,3,0,1,0,4\}$, 则该序列的熵编码为 "01100011011011111100111110011001010", 该编码长度为 35 比特, 而采用本文的置乱方法置乱后的 DCT 序列为 $\{-3,3,0,2,0,0,1,4\}$, 其熵编码为 "01100011111011101110011001001010" 为 32 比特。所以与全部置乱的方法相比本文的置乱方法比特增长长度要低, 特别是当 AC 系数中存在多个 0 的情况下, 对比效果更为明显。

由于在加密过程中, 文献[23]和本文算法都对 JPEG 比特流中所有的图像块熵编码进行了随机置乱, 而置乱后的 JPEG 比特流需要对每个图像块熵编码的 DC 系数需要重新进行一次 DPCM 编码。

但文献[23]为了提高算法的安全性将比特流中 3/4 的图像块熵编码隐藏至头部, 因此只有 1/4 的图像块熵编码, 也即是携密熵编码的 DC 系数需要进行 DPCM 编码, 而本文算法中不需要对部分图像块熵编码进行头部隐藏, 而是将所有的图像块熵编码作为携密熵编码。为了分析 DPCM 编码对文件大小变化的影响, 采用图 2 中量化因子 Q=80 的测试图像进行实验仿真, 实验结果如表 4 所示,

表 4　DPCM 编码对文件大小影响

Table 4　DPCM encoding affects file size

测试图像	原始图像大小(KB)	文献[23](KB)	本文算法(KB)
Lena	37.0	37.1	37.7
Peppers	38.4	38.5	39.2
Baboon	76.8	76.7	77.3
Man	48.2	48.3	48.8
Lake	50.9	51.0	51.7
Airplane	37.8	37.8	38.5

表 4 为本文算法将 JPEG 比特流中的图像块熵编码进行随机置乱后, 对每个图像块的 DC 系数重新做 DPCM 编码对图像文件大小的影响与文献[23]对比。由于异或加密并不会影响文件长度变化, 所以在文献[23]中, 加密过程造成的文件长度变化是由 DPCM 编码造成的。而在文献[23]中, 是根据加密密钥随机选取 1/4 的图像块熵编码作为携密熵编码, 而本文将所有的图像块熵编码作为携密熵编码, 所以需要对所有的图像块熵编码的 DC 系数进行 DPCM 编码, 故文件增长长度要大于文献[23]。

此外, 隐藏附加信息也会引起图像文件大小的增加, 将携密图像文件大小变化率定义为α, 表示平均隐藏 1 比特的附加信息携密文件大小变化, 单位为 KB, 计算则有,

$$\alpha = \frac{M - E}{\beta} \times 100\%(\text{KB/bit}) \tag{29}$$

其中M为携密文件大小, E为加密文件大小, β为最大隐藏容量。因此将文献[23]和本文算法中隐藏容量和文件变化大小作为测试对象进行比较, 如表 5 所示。

表 5　携密图像文件大小变化率对比

Table 5　Comparison of size change rate of image files with encryption

测试图像	携密文件大小变化率α(KB/bit)%	
	文献[23]	本文算法
Lena	0.01	0.01
Peppers	0.01	0.01
Baboon	0.01	0.01
Man	0.01	0.01
Lake	0.01	0.01

由于本文算法和文献[23]的信息隐藏算法相同, 都采用直方图平移方法进行可逆隐藏。从表 5 中的对比结果可以看出本文算法与文献[23]中的隐藏算法平均每隐藏 1 比特信息, 文件平均长度变化率相同, 大小为 0.01%KB, 也即平均每隐藏 1 比特的附加信息, 文件大小增长了 0.0001KB。

4.3　隐藏容量分析

最后, 为了验证本文算法的最大隐藏容量, 选取 Lena, Peppers, Baboon, Man, Lake 压缩因子 Q=80、大小为 512×512 的灰度图像作为测试图像进行实验, 得到的实验结果与文献[20-23]的最大隐藏容量进行对比, 结果如图 4 所示。

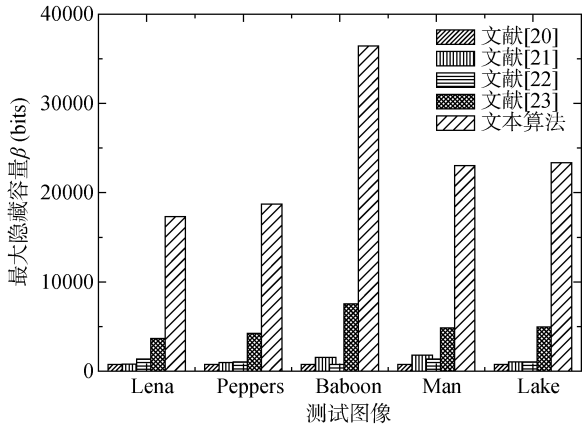


图 4　最大隐藏容量对比图

Figure 4　Maximum embedded capacity comparison diagram

文献[23]为了提高加密算法的安全性, 需要将加密熵编码隐藏到 JPEG 头文件中, 因此在 JPEG 比特流所有的图像块熵编码中, 需要根据加密密钥随机选取 3/4 的图像块熵编码, 作为加密熵编码, 其余 1/4 部分作为携密熵编码用于隐藏信息, 所以得到的加密图像大小为 256×256 。该算法的最大隐藏容量高于文献[20-23], 以 Lena 图像为例, 文献[20]的最大隐藏容量为 750bits, 文献[21]的最大隐藏容量为

798bits, 文献[22]的最大隐藏容量为 1364bits, 而文献[23]的最大隐藏容量可以达到 3667bits。然后, 采用以上 5 幅图像为测试对象, 计算其最大隐藏容量的平均值, 则文献[23]平均最大隐藏容量(5057.4bits)分别是文献[20]的 6.7 倍(750bits), 文献[21]的 4.1 倍(1230.2bits), 文献[22]的 4.6 倍(1109.8bits)。

以 Lena 图像为例, 图 4 结果显示本文算法的最大隐藏容量为 17343bits, 要高于文献[23]的最大隐藏容量。这是由于本文算法将 JPEG 比特流中所有的图像块熵编码作为信息隐藏的载体, 并通过对图像块熵编码中的 AC 系数置乱能够解决文献[22,23]中隐藏容量与安全性之间的矛盾。本文的平均最大隐藏容量为 23783.2bits, 约为文献[23]的 4.7 倍, 这是由于文献[23]中是随机选取 1/4 图像块作为携密熵编码用于隐藏信息, 而本文是选取所有的图像块熵编码携密熵编码, 对于不同图像块熵编码的隐藏容量是由该图像块本身的结构所决定, 因此选取不同的图像块熵编码作为携密熵编码, 文献[23]的最大隐藏容量可能会发生改变, 故本文的最大隐藏容量要稍高于文献[23]的 4 倍。从图 4 中可以看出对纹理复杂的 Baboon 图像的最大隐藏容量可以到达 36452bits, 与空域图像 RDH-EI 算法相反, 纹理复杂的图像的最大隐藏容量要高于纹理相对平滑的图像(Lena 等图像)的最大隐藏容量。这是由于在空域图像的 RDH-EI 算法中, 是利用像素之间的相关性进行隐藏附加信息, 所以平滑的图像的最大隐藏容量要高于纹理复杂的图像。与之相反, 对于 JPEG 图像, 与平滑的图像相比, 纹理复杂的图像在 JPEG 量化过后的高频分量部分结构相对复杂, 而本文是采用对个图像块熵编码的 AC 系数做直方图平移以隐藏附加信息, 故对于纹理复杂的图像, 其用于隐藏附加信息的 AC 系数 RLC 编码 (R,S) 为 (0,±1) 的个数较多, 因此, 对于纹理复杂的图像, 其隐藏容量要高于平滑的图像。

此外, 对于同一幅 JPEG 图像, 其量化因子 Q 取值不同, 隐藏容量也会受到影响。为了验证最大隐藏容量与量化因子的关系, 以不同量化因子 Q=10,20,…,90 的相同图像作为测试对象进行测试, 结果如图 5 所示,

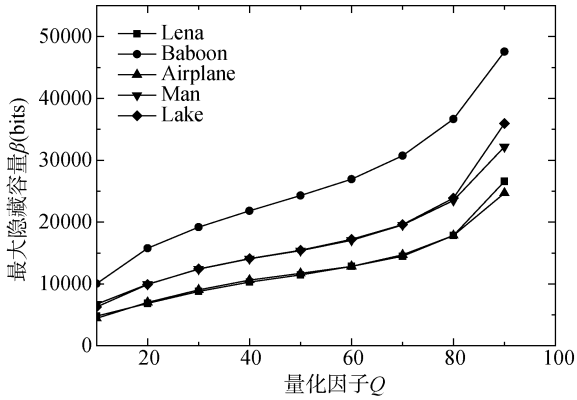


图 5 量化因子与最大隐藏容量关系

Figure 5 The relationship between quantization factor and maximum embedded capacity

从图 5 中结果可以看出随着量化因子增加, 图像的最大隐藏容量呈逐渐递增的趋势, 这是由于在量化过程中, 量化因子 Q 越大, 压缩率就越大, 从而导致每个 8×8 大小的图像块对应的 DCT 矩阵中, 高频部分系数(AC 系数)值为 0,±1 的系数存在个数越多, 由于隐藏附加信息是通过修改 RLC 编码 (R,S) 为 (0,±1) 的 AC 系数值来实现的, 因此, 量化因子 Q 取值越大, 其最大隐藏容量也就越大。

以量化因子 Q 取值为 10 和 90 不同值的条件下对 Lena 等图像进行测试, 对比加密图像文件大小变化, 携密文件大小变化率, 以及最大隐写容量各方面性能, 结果如表 6 所示。从表 6 中可以看出量化因子 Q=90 时, 对应的最大隐写容量明显高于 Q=10, 因此说明, 量化因子 Q 也是影响图像最大隐写容量的因素。

表 6 量化因子 Q=10, 90 性能对比

Table 6 Quantitative factors Q= 10,90 performance comparisons

测试图像	Q=10				Q=90			
	原始图像大小(KB)	加密图像文件大小(KB)	携密图像文件大小变化率(KB/bit)%	最大隐写容量(bits)	原始图像大小(KB)	加密图像文件大小(KB)	携密图像文件大小变化率(KB/bit)%	最大隐写容量(bits)
Lena	7.82	8.56	0.01	4636	57.8	62.0	0.01	26221
Airplane	8.59	9.22	0.01	4175	56.7	60.5	0.01	24130
Baboon	14.6	15.3	0.01	9595	111.0	118.0	0.01	47426
Man	9.34	10.1	0.01	6474	72.1	77.4	0.01	31885
Lake	10.4	11.0	0.01	5935	81.8	88.5	0.01	35763

5 结论

JPEG 比特流加密域大容量可逆数据隐藏, 根据加密密钥对每个图像块熵编码中的 AC 系数进行了置乱。在满足 JPEG 解码标准的条件下对图像块熵编码进行了全加密, 提高了加密算法的安全性。同时本文置乱方法避免了多个连续 0 系数在随机置乱的过程中分散排布, 从而造成比特流长度的增加。并且本文算法将 JPEG 比特流中所有的图像块熵编码作为信息隐藏的载体, 在提高安全性的同时, 具有很高的隐藏容量。

参考文献

[1] Y. Q. Shi., X.L. Li., and X.P. Zhang, et al ,“Reversible data hiding: Advances in the past two decades,”. *in Proc. IEEE Access,* 4,pp.3210-3237,2016.

[2] X.P. Zhang,“ Reversible data hiding in encrypted image,” *in Proc. IEEE on Signal Processing Letters,* 18(4), pp.255-258, 2011.

[3] M. Li, D. Xiao, A. Kulsoom and Y.S. Zhang. “ Improved reversible data hiding for encrypted images using full embedding strategy”. *in Proc. Electronic Letters,* 51(9), pp. 690-691, 2015.

[4] X. Liao and C.W. Shu, “Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels,” *in Proc. Journal of Visual Communication and Image Representation,* 28, pp. 21-27, 2015.

[5] Z.X. Qian, S. Dai, F. Jiang and X.P. Zhang. “Improved joint reversible data hiding in encrypted images,” *in Proc. Journal of Visual Communication and Image Representation,* 40, pp. 732-738, 2016.

[6] F.J. Huang, J.W. Huang and Y. Q. Shi, “New framework for reversible data hiding in encrypted domain,”. *IEEE Trans. Information Forensics and Security,* 11 (12), pp. 2777-2789, 2016.

[7] Z.X. Qian and X.P. Zhang, “Reversible data hiding in encrypted images with distributed source encoding,” *IEEE Trans.Circuits and Systems for Video Technology.* 26(4), pp. 636-646,2016.

[8] D.W. Xu and R.D. Wang, “Separable and error-free reversible data hiding in encrypted images,” *in Proc. Signal Processing.* 123, pp.9-21 ,2016.

[9] W.M. Zhang , K.D. Ma and N.H. Yu,“ Reversibility improved data hiding in encrypted images ,” *in Proc. Signal Processing,* 94,pp.118-127, 2014.

[10] T. Nguyen, C.C. Chang and W.C. Chang, “High capacity reversible data hiding scheme for encrypted images,” *in Proc.Signal Processing: Image Communication.* 44,pp. 84-91, 2016.

[11] B.X. Yin, F. Chen and H.J. He et al, “Separable Reversible Data Hiding in Encrypted Image with Classification Permutation,”*in Proc. IEEE Third International Conference on Multimedia Big Data.,pp.*201-204, 2017.

[12] X.C. Cao, L. Du and X.X. Wei, et al, “ High capacity reversible data hiding in encrypted images by patch-level sparse representation,” *IEEE Trans.Cybernetics,* 46(5),pp.1132-1143, 2016

[13] P. Pauline and P. William,“ An Efficient MSB Prediction-Based Method for High-Capacity Reversible Data Hiding in Encrypted Images,” *IEEE Trans. Information Foren-sics and Security,* pp. 13-7 ,2018.

[14] S. Yi and Y.C. Zhou,“Binary-block embedding for reversible data hiding in encrypted images,” *in Proc. Signal Processing,* 133, pp.40-51, 2017

[15] Z. Liu and C. Pun,“ Reversible data-hiding in encrypted images by redundant space transfer.” *in Proc. Information Sciences,*433–434, pp.188–203,2018.

[16] Y.Y. Chen, S. Yan, H.J. He, B.X. Yin, and M. Deng. The reversible information hiding method of secure encryption domain based on high turnover prediction, application number: 201810174576.8. 2018-03-02.
(陈玉钰, 鄢舒, 和红杰, 尹帮旭, 邓敏. 基于高位翻转预测的安全加密域可逆信息隐藏方法, 申请号: 201810174576.8. 2018-03-02.)

[17] X.T. Wu and B. Chen, J. Weng,“Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer,” *in Proc.J. Vis. Commun. Image* R. 41 pp. 58–64, 2016.

[18] M. Li and Y. Li.,“Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding,” *in Proc.Signal Processing* 130 pp.190–196,2017.

[19] S.J. Xiang and X.R. Luo, “Reversible Data Hiding in Homomorphic Encrypted Domain By Mirroring Ciphertext Group,”.*IEEE Trans. Circuits & Systems for Video Technology,* pp.1-1, 2017.

[20] Z. Qian, X. Zhang, and S. Wang, “Reversible data hiding in encrypted JPEG bitstream,” *IEEE Trans. Multimedia,* vol. 16, no. 5, pp. 1486-1491, 2014.

[21] J. C. Chang, Y. Z. Lu, and H. L. Wu, “A separable reversible data hiding scheme for encrypted JPEG bitstreams,” *in Proc.Signal Processing,* vol. 133, pp. 135-143, 2017.

[22] Z.X. Qian, H. Zhou, X. Zhang and W. Zhang, “Separable re-versible data hiding in encrypted JPEG bitstreams,*” IEEE Trans.Dependable and Secure Computing,* doi: 10.1109 /TDSC.2016.2634161.

[23] Z.X. Qian, H. Xu, X.R. Luo and X.P. Zhang, “New Framework of Reversible Data Hiding in Encrypted JPEG Bitstreams,” *IEEE Trans.Circuits and Systems for Video Technology,* 10.1109/TCSVT.2018.2797897.

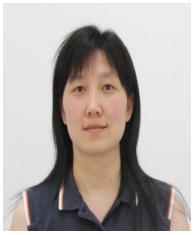
[24] S. Y. Ong, K. S. Wong, X. Qi, et al “Beyond format-compliant encryption for JPEG image,” *Signal Processing: Image Communication,* vol. 31, pp. 47-60, 2015.

[25] S. Y. Ong, K. S. Wong and K. Tanaka, “Scrambling-embedding for JPEG compressed image,” *in Proc.Signal Processing,* vol. 109, pp. 56-68, 2015.

[26] Int. Tele. Union, CCITT Recommendation T.81, “Information technology-digital compression and coding of continuous-tone Still Images - Requirements and Guidelines,” 1992.

郑梦阳 2016 年在贵州大学信息安全专业获得学士学位, 现西南交通大学信息科学与技术学院攻读硕士学位。研究领域为加密域可逆信息隐藏。Email: 405602023@qq.com

陈帆 2012 年在西南交通大学获得博士学位, 现任西南交通大学副教授, 研究领域多媒体数据安全。研究方向包括: 信息隐藏、计算机应用技术等。Email: fchen@swjtu.edu.cn



和红杰 2009 年在西南交通大学获得博士学位, 现任西南交通大学教授, 研究领域信号与信息处理。研究兴趣包括: 信息隐藏、图像处理, 深度学习等。Email: hjhe@swjtu.edu.cn

董梦瑶 2016 年, 在西南交通大学信息安全专业获得学士学位, 现西南交通大学信息科学与技术学院攻读硕士学位, 研究领域为加密域可逆信息隐藏。研究方向包括变长自恢复水印等。Email: 576343920@qq.com

李增辉 现西南交通大学茅以升学院电子信息类专业攻读本科学位。研究领域为信息安全、图像处理。研究兴趣包括: 深度学习, 可逆信息隐藏技术。Email: 1374223299@qq.com

陈潭升 现西南交通大学数学学院统计类专业攻读本科学位。研究领域为信息安全、图像修复。研究兴趣包括: 深度学习, 图像修复。Email: 897833416@qq.com