基于MSB二维标记的加密图像可逆数据隐藏

杨尧林[1],和红杰[1],陈 帆[2],郭炀炀[1]
(1. 信号与信息处理四川省高校重点实验室,四川成都 611756; 2. 西南交通大学计算机与人工智能学院,四川成都 611756)

摘 要: 针对基于标记编码的加密图像可逆数据隐藏存在图像冗余未充分利用和信息泄露问题,提出一种基于MSB(Most Significant Bit)二维标记的加密图像可逆数据隐藏(Reversible Data Hiding in Encrypted Image,RDH-EI)算法.为提高算法的嵌入容量,在二维标记图生成阶段,根据原始与预测像素值构造出差异序列,生成MSB二维标记(l_1,l_2).第一维l_1和第二维l_2分别记录原始与预测像素值初始连续相同MSBs位数和后继连续相反MSBs(Subsequent Consecutive Opposite MSBs,SCO-MSBs)位数.SCO-MSBs的使用提高像素冗余的利用率,结合范式哈夫曼编码实现嵌入容量的提升.为提高算法的安全性,在伪标记图与加密图像构造阶段,将二维标记图生成的编码流与保存所有辅助信息的额外数据流进行有效信息合并生成原始流后加密,同时在构造加密图像过程中生成用于标识可嵌入位置的伪标记图.原始流加密能有效防止图像信息泄露,伪标记图则用于确定嵌入的预留空间位置.实验结果表明,与现有同类算法相比,本文算法能防止标记图泄露并抵抗唯密文攻击,嵌入容量提高0.208 bpp以上,且算法实现完全可逆的同时,运行时间将近现有算法的1/4.

关键词: 可逆数据隐藏;图像加密;二维标记;有效信息合并;伪标记图
基金项目: 国家自然科学基金(No.U1936113,No.61872303)
中图分类号: TN957.52 **文献标识码:** A **文章编号:** 0372-2112(2023)04-0993-13
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20210571

Reversible Data Hiding in Encryption Images Based on MSB Two-Dimensional Label

YANG Yao-lin[1], HE Hong-jie[1], CHEN Fan[2], GUO Yang-yang[1]
(1. *Sichuan Province Key Laboratory of Signal & Information Processing*, *Chengdu*, *Sichuan* 611756, *China*;
2. *School of Computing and Artificial Intelligence*, *Southwest Jiaotong University*, *Chengdu*, *Sichuan* 611756, *China*)

Abstract: For the problems of insufficient utilization of image redundancy and information leakage in reversible data hiding in encryption images based on label coding, a reversible data hiding in encrypted images (RDH-EI) based on MSB (Most Significant Bit) two-dimensional label is proposed. In order to improve the embedding capacity of the algorithm, in the two-dimensional label map generation stage, the difference sequence is constructed according to the original and predicted pixel values, and the MSB two-dimensional label (l_1,l_2) is generated. The first dimension l_1 and the second dimension l_2 respectively record the original and predicted pixel values with the same initial consecutive same MSBs and subsequent consecutive opposite MSBs (SCO-MSBs). The use of SCO-MSBs improves the utilization of pixel redundancy, combined with canonical Huffman coding to achieve higher embedding capacity. In order to improve the security of the algorithm, in the construction stage of the pseudo-label map and the encrypted image, the encoded stream generated by the two-dimensional label map and the extra data stream storing all auxiliary information are used to generate the original stream through effective information merging and then encrypted, and in the process of constructing the encrypted image, a pseudo-label map for identifying the embedding position is generated. The original stream encryption can effectively prevent the leakage of image information, and the pseudo-label map is used to determine the position of the embedded reserved room. The experimental results show that compared with the existing similar algorithms, the proposed algorithm can effectively prevent the leakage of the label map and resist the ciphertext-only attacks, the embedding capacity is increased by more than 0.208 bpp, and the algorithm achieves completely reversible while running time is nearly 1/4 of the existing algorithm.

收稿日期:2021-05-06;修回日期:2022-01-14;责任编辑:孙瑶

Key words： reversible data hiding; image encryption; two-dimensional label; effective information merging; pseudo label map

Foundation Item(s)： National Natural Science Foundation of China (No.U1936113, No.61872303)

1 引言

随着云存储的发展,更多用户将图像等多媒体文件上传至云中存储. 近年来,云端用户隐私信息和图像等数据泄露事件使得数据安全及隐私保护成为用户关注的热点. 加密图像可逆数据隐藏(Reversible Data Hiding in Encrypted Image,RDH-EI)可为云存储的场景提供有效解决方案[1]. 图像加密将原始图像转化为随机噪声,避免图像内容的泄露. 利用可逆数据隐藏技术能在不扩展图像尺寸基础上在图像中嵌入用户信息等用于隐私数据或管理等标签,且能无损地提取出来用于识别. 目前,该技术已应用于医学图像管理、法律取证、军事应用等方面[2~6].

现有RDH-EI算法中,根据嵌入方法的不同可大致分为三类:第一类为基于直方图移位的RDH-EI[7,8],通过统计像素值或误差值的分布,在峰值点中嵌入秘密数据;第二类是基于差异扩展的RDH-EI[9,10],通过修改相邻像素的差值来嵌入秘密数据;第三类为基于无损压缩的RDH-EI[11,12],将原始图像的冗余进行压缩来腾出空间用于嵌入秘密数据. 其中,基于无损压缩的RDH-EI具有较大的嵌入容量,被更多学者研究.

现有基于无损压缩RDH-EI算法,以嵌入容量的提高作为主要研究方向. Yi等人[13]提出自适应位级数据嵌入(Adaptive Bit-level Data Embedding,ABDE)方法,根据预测误差将像素分为可嵌入像素与不可嵌入像素2类,可嵌入像素又根据2种嵌入编码策略分为3类或4类,分别用2位或3位进行标记,并选取嵌入率较高的策略进行标记压缩. 但该算法在可逆恢复时仅将预测误差在[−1, 1]或[−2, 1]范围内像素作为可嵌入像素,限制了容量的提高. 在此基础上,Yi等人[14]提出参数二叉树标记(Parametric Binary Tree Labeling,PBTL)方法,在可嵌入像素个数与编码长度间做出权衡,使嵌入容量具有显著提升. 该算法与文献[13]存在相同的问题,仅使用了局部像素的相关性. 随后,Wu等人[15]提出了改进的参数二叉树标记算法,利用原始图像整体的相关性,在减少参考像素个数的同时提高预测精度,使预测误差分布更加集中,从而获得更高的嵌入容量. 上述3种算法中,都采用了将不均匀的预测误差用等长编码标记的策略,该策略会造成空间浪费. 为进一步提高冗余空间利用率,学者们提出采用变长编码标记的方法. Yi等人[16]提出二值块嵌入(Binary-Block Embedding,BBE)算法,将灰度图像的8个位面看作二进制图像,对每个位面分块,对各块分类后采用变长编码进行标记. 但该算法忽略了自然图像各位面之间的相关性. Chen等人[17]在BBE的基础上进行改进,通过迭代预测与相邻位平面异或结合的方式来增加全0块的个数,从而更适用于改进的BBE算法来减少编码长度. 此外,哈夫曼编码作为性能较优的变长编码方式,可用在RDH-EI中来实现更高的压缩性能. Fu等人[18]提出自适应编码算法中,将图像分块后,统计各块MSBs(Most Significant Bits)类别数并采用对应的哈夫曼编码实现压缩. 但该算法与文献[14]存在相同问题,未利用图像的全局相关性. 2020年,Yin等人[19]采用全局预测算法,对比原始与预测像素值的MSBs相同位并转化为标记. 在预留空间过程中,根据标记值反映原始与预测像素值MSBs相同的位数及随后位必定相反的特性,将原始与预测像素值相同的MSBs预留用于数据嵌入,剩余LSBs则作为未标记位保持不变. 该算法相较于文献[17],以BOSSBase图像库为例的嵌入容量平均值提高0.156 bpp. 通过研究发现,Yin算法中像素的未标记位依旧存在冗余,如果能利用这些冗余,将进一步提高嵌入容量.

在兼顾嵌入容量的同时,安全性也是RDH-EI不可忽视的一个方面. 文献[15,16,19]采用异或加密生成加密图像,尽管加密图像类似随机噪声,但未改变像素空间位置. 根据位置未改变的特点,结合原始图像像素间的相关性,Khelifi[20]提出一种针对异或加密的唯密文攻击,攻击者在得到多幅加密图像的条件下,能以较高的概率估计出多个MSB平面的异或加密矩阵,从而得到加密图像的原始内容,因此采用异或加密存在安全隐患. 此外,文献[15,19]为使管理者确定位置嵌入秘密数据,标记图无法通过加密密钥进行加密,从而未持有加密密钥的未授权者也能从加密图像中无损地恢复标记图. 分析发现,标记图中包含原始图像的纹理信息,会造成原始图像内容的泄露. 因此在不影响数据嵌入的情况下,对标记图加密来防止信息泄露且提高加密算法的安全性成为研究的关键问题.

综上,在基于无损压缩RDH-EI中,文献[19]在嵌入容量上相较于现有文献具有较高优势. 算法中加密图像生成过程中分为4部分:①标记图生成,通过对比原始及预测像素值初始连续相同MSBs位数生成标记图;②哈夫曼编码,将标记图进行哈夫曼编码生成编码流;③图像加密,对原始图像加密生成中间加密图像;④标记图嵌入,将编码流及包含编码表和编码流长度的额外数据流嵌入中间加密图像中生成加密图像. 该算法中存在两个问题:一是在生成标记过程中,仅记录原始与预测像素值相同MSBs位数,但原始像素值未标

记位中与预测像素值存在多数位相反的特性,因此未标记位中仍存在冗余未用于腾出空间;二是在无加密密钥情况下,标记图能够通过加密图像直接恢复,将导致图像内容泄露,且加密阶段仅采用异或加密,无法有效抵抗唯密文攻击. 为进一步提高嵌入容量和安全性,本文算法提出一种基于MSB二维标记的加密图像可逆数据隐藏算法. 主要贡献有以下2个方面:

(1)提出二维标记的方法. 利用像素未标记位中存在的冗余,使得更多的MSBs用于腾出空间,来提高嵌入容量.

(2)提出有效信息合并和伪标记图构造策略. 在伪标记图与加密图像构造阶段,将与原始图像有关的编码流与额外数据进行有效信息合并得到的原始流进行加密提高标记图的安全性,同时伪标记图的构造在不泄露原始图像内容的情况下标明预留空间位置,且有效地降低算法的时间复杂度.

2　MSB二维标记策略

基于文献[19]提出的标记策略,研究发现其中未标记位存在冗余未利用的问题. 通过进一步使用像素冗余,提出了MSB二维标记策略. 下面通过分析文献[19]存在的问题,引申出本文标记策略过程,并根据示例说明MSB二维标记策略腾出空间的提高,具体过程描述如下.

文献[19]生成标记的过程中,先将原始像素值o与预测像素值p分别转化为二进制序列$o^k(k=1,2,\cdots,8)$和p^k,随后从MSB到LSB(Least Significant Bit)比较o^k和p^k,得到相同的位数并作为标记. 如图1所示,当原始像素值与预测像素值分别为156和144时,从MSB到LSB比较后可得有4位相同,故标记为4. 此时,该像素能够嵌入5位,因为在恢复过程中,由标记可知原始像素值与预测像素值前4位MSBs相同,第5位相反.

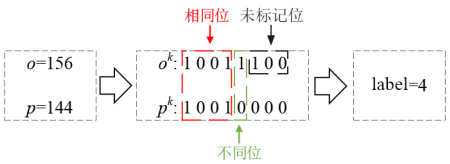


图1　文献[19]标记生成示意图

分析可知,上述算法恢复原始像素的实质:标记代表预测像素值与原始像素值从MSB到LSB各比特相同或相反的关系,结合预测像素值的可恢复性,得到原始像素值. 观察图1可以看出,除前4位MSBs相同外,第5,6位均相反,说明该像素中依旧存在冗余未被利用. 基于此,提出MSB二维标记策略,生成标记的过程如下.

Step1　差异序列的构造. 将原始像素值o与预测像素值p转化为二进制序列,分别定义为$o^k(k=1,2,\cdots,8)$和p^k,转化式如下:

$$x^k=\left\lfloor\frac{x}{2^{k-1}}\right\rfloor \bmod 2,\quad k=1,2,\cdots,8 \tag{1}$$

再由式(2)构造出o^k和p^k的差异序列d^k为

$$d^k=o^k\oplus p^k \tag{2}$$

其中,$\oplus$为异或操作. 因此当$d^k=0$时,说明o^k和p^k相同;反之,说明o^k与p^k相反.

Step2　二维标记的生成. 首先,定义二维标记(l_1,l_2),其中l_1表示首"0"串长度,l_2表示次"1"串长度. 为计算l_1与l_2,从MSB到LSB观察差异序列d^k,先统计连续"0"的个数,记为l_1,再统计随后连续"1"的个数,记为l_2,此时有$l_1,l_2\in\{0,1,2,\cdots,8\}$,$l_1+l_2\leqslant 8$. 且统计首"0"串长度$l_1$的式如(3)所示:

$$l_1=\sum_{k=1}^{8}m^k \tag{3}$$

其中,

$$m^k=\begin{cases}1-d^k, & k=8\\ (1-d^k)\times m^{k+1}, & 1\leqslant k\leqslant 7\end{cases} \tag{4}$$

统计次"1"串长度l_2则如式(5)所示:

$$l_2=\sum_{k=1}^{8-l_1}n^k \tag{5}$$

其中,

$$n^k=\begin{cases}d^k, & k=8-l_1\\ d^k\times n^{k+1}, & 1\leqslant k\leqslant 7-l_1\end{cases} \tag{6}$$

在恢复阶段,通过二维标记中首"0"串长度可确定原始像素值o与预测像素值p存在l_1位初始连续相同MSBs,次"1"串长度确定l_2位后继连续相反MSBs(Subsequent Consecutive Opposite MSBs, SCO-MSBs),且第l_1+l_2+1位相同. 因此采用二维标记时,该像素能够腾出l_1+l_2+1位空间用于嵌入. 同样以$o=156$,$p=144$为例,二维标记生成示意图如图2所示. 由原始像素值与预测像素值生成的差异序列为d^k:00001100,首"0"串长度$l_1=4$,次"1"串长度$l_2=2$,故采用二维标记时,该像素能够腾出7位用于嵌入,相较于文献[19]提高2位.

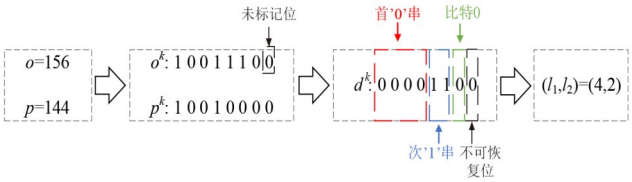


图2　二维标记生成示意图

3　基于MSB二维标记编码的RDH-EI

本文算法主要包括3部分:

(1)加密图像的生成. 图像拥有者持有加密密钥

K_e,将原始图像O进行预处理和加密生成加密图像E.

(2)数据嵌入与提取. 持有数据隐藏密钥K_m的管理者在原始图像内容未知的情况下,将秘密数据D加密后嵌入加密图像中生成含嵌图像M,且能在含嵌图像M中无损提取秘密数据D.

(3)图像恢复. 持有加密密钥K_e的授权者可通过含嵌图像M得到与原始图像O一致的恢复图像R,完全实现图像可逆恢复. 下面对每部分进行详细描述.

3.1 加密图像的生成

加密图像的生成主要由4个阶段组成:

(1)二维标记图的生成. 图像拥有者将原始图像O进行预测得到预测图像P,根据MSB二维标记策略确定二维标记图L.

(2)范式哈夫曼编码. 根据二维标记图的统计特征生成范式哈夫曼编码表后,对二维标记图编码得到编码流ξ.

(3)原始流的生成. 将恢复时需要的额外数据流φ生成后,与编码流ξ组成原始流S.

(4)伪标记图与加密图像构造. 对原始流S加密后生成加密流S',结合预留空间τ生成中间加密图像$\overline{E}$,确定为标记图后,将伪标记图信息m嵌入$\overline{E}$中生成加密图像E.

3.1.1 二维标记图的生成

原始图像生成预测图像后,采用MSB二维标记策略生成所有像素的二维标记,构成二维标记图. 由MSB二维标记策略可知,当预测方法越准确,腾出的嵌入空间越大,嵌入容量越高. 兼顾嵌入容量和时间复杂度,常用的预测方法主要有中值边缘检测(Median Edge Detector, MED)预测、梯度调整预测(Gradient Adjusted Predictor, GAP)及菱形预测[21]. 其中,GAP预测精度高于MED预测,但GAP的复杂度也高于MED预测[22]. 菱形预测精度高于MED与GAP,但保留的参考像素较多[21]. 为便于对比文献[19]的标记性能,本文采用与文献[19]相同的MED预测方法.

$$\overline{p}(i,j)=\begin{cases}\max\left(o(i-1,j),o(i,j-1)\right),\\ \min\left(o(i-1,j),o(i,j-1)\right),\\ o(i-1,j)+o(i,j-1)-o(i-1,j-1),\end{cases}$$

Step2 二维标记图的生成. 根据MSB二维标记策略,由原始像素值$o(i,j)$与预测像素值$p(i,j)$初始连续相同MSBs位数和SCO-MSBs位数生成对应的首"0"串长度$l_1(i,j)$与次"1"串长度$l_2(i,j)$,组成二维标记$(l_1(i,j),l_2(i,j))$. 最终构造出二维标记图L.

3.1.2 范式霍夫曼编码

在得到二维标记图后,需要转化为二进制序列作为加密图像的一部分,转化过程则通过范式哈夫曼编码实现. 在编码过程中统计出二维标记图中各标记的概率后,构成范式哈夫曼编码表. 最后,通过编码表将二维标记图转化为编码流.

Step1 概率统计. 在二维标记中有$0\leqslant l_1,l_2\leqslant 8$,$l_1+l_2\leqslant 8$,当且仅当$l_1=8$时,$l_2=0$,其余情况下次"1"串长度至少为1,即$l_2\geqslant 1$,所以二维标记的总类别数为$\sum_{i=0}^{7}\sum_{j=1}^{8-i}1+1=37$. 假设所有的二维标记定义为$\overline{L_k}$,$k=1,2,\cdots,37$,在二维标记图中出现的概率为$f_k$,则各二维标记的概率计算式为

Step1 图像预测. 设原始图像O的大小为$W\times H$,$o(i,j),1\leqslant i\leqslant W,1\leqslant j\leqslant H$为原始图像中第$i$行第$j$列的像素值. 在预测算法中,除$o(1,1)$作为参考像素$o_{\text{ref}}$外,首行首列的其余像素使用线性预测方法,通过左侧或上侧的像素得到预测值. 除首行首列的其余像素则采用MED预测算法. 因此预测图像P中预测像素值$p(i,j)$的计算式为

$$p(i,j)=\begin{cases}o(i,j), & i=1 \text{ and } j=1\\ o(i,j-1), & i=1 \text{ and } 1<j\leqslant H\\ o(i,j-1), & 1<i\leqslant W \text{ and } j=1\\ \overline{p}(i,j), & 2\leqslant i\leqslant W \text{ and } 2\leqslant j\leqslant H\end{cases}\tag{7}$$

其中,$\overline{p}(i,j)$表示通过MED预测算法得到的预测像素值,如式(8)所示. 通过式(7)与式(8)计算后可得预测图像P.

$$\begin{aligned}&o(i-1,j-1)\leqslant\min\left(o(i-1,j),o(i,j-1)\right)\\ &o(i-1,j-1)\geqslant\max\left(o(i-1,j),o(i,j-1)\right)\\ &\text{其他}\end{aligned}\tag{8}$$

$$f_k=\frac{\sum_{i=1}^{W}\sum_{j=1}^{H}\text{isequal}\left((l_1(i,j),l_2(i,j)),\overline{L}_k\right)}{W\times H}\tag{9}$$

其中,isequal(a,b)表示当a与b完全相同时为1,反之为0.

Step2 范式哈夫曼编码表的构造. 根据范式哈夫曼编码算法生成哈夫曼编码表T,其中包括两部分,二维标记$\overline{L}_k,k=1,2,\cdots,37$以及对应的编码$C_k$,在编码表中以编码长度为最高关键字,二维标记为次关键字进行排序.

Step3 编码流的生成. 按照从上到下,从左到右的顺序扫描二维标记图,如式(10)所示,按照范式哈夫曼编码表依次将各二维标记$(l_1(i,j),l_2(i,j))$转化为编码ξ_x.

$$\xi_x=C_k,\ \text{if isequal}\left((l_1(i,j),l_2(i,j)),\overline{L}_k\right)\tag{10}$$

其中,$x=1,2,\cdots,W\times H$且$x=(j-1)\times W+i$.

将编码$\xi_1,\xi_2,\cdots,\xi_{W\times H}$连接,组成编码流$\xi$,其长度$l_\xi$为

$$l_\xi = \sum_{k=1}^{37} f_k \times (W \times H) \times l_k \tag{11}$$

其中,l_k表示编码C_k的长度.

3.1.3 原始流的生成

为可逆地恢复原始图像,除生成的编码流ξ外,还需要将原始图像中的参考像素o_{ref},无法通过二维标记恢复的未标记位B_{o}、范式哈夫曼编码表T以及编码流的长度l_ξ转化为比特流进行存储. 将上述生成的比特流定义为额外数据流φ,结合编码流ξ组成可恢复原始图像的原始流S.

Step1　额外数据流的构造. 将参考像素o_{ref},未标记位长度l_B,未标记位B_{o},范式哈夫曼编码表T及编码流长度l_ξ转化的比特流合并构造为额外数据流φ. 转化过程如下:参考像素o_{ref}仅有一个像素,用8位存储. 未标记位B_{o}为所有原始像素值中无法通过二维标记恢复的LSBs,其长度l_B为

$$l_B = \sum_{i=1}^{W}\sum_{j=1}^{H} l_b(i,j) \tag{12}$$

其中,$l_b(i,j)$表示像素值$o(i,j)$中无法恢复的原始LSBs长度,即

$$l_b(i,j) = \begin{cases} 0, & l_1(i,j)+l_2(i,j)+1 \geqslant 8 \\ 7-l_1(i,j)-l_2(i,j), & \text{其他} \end{cases} \tag{13}$$

在转化范式哈夫曼编码表T时,由于编码长度从小到大有序,因此只需记录二维标记以及每种长度编码的个数即可. 其中,二维标记共有37类,用$\lceil \log_2 37 \rceil = 6$位表示;每种长度编码的个数先统计出编码长度最大值$l_{\max}$,用6位记录后,再统计出长度为1~$l_{\max}$包含的编码个数并转化为$(\log_2 37) = 6$位的二进制序列. 因此存储编码表需要的长度$l_T$为

$$l_T = 37 \times 6 + 6 + l_{\max} \times 6 \tag{14}$$

此外,未标记位的长度l_B和编码流的长度l_ξ分别用$\lceil \log_2 (W \times H \times 8) \rceil$位记录. 最终可得构成的额外数据流的长度$l_\varphi$为

$$l_\varphi = 8 + l_B + l_T + 2 \times \lceil \log_2 (W \times H \times 8) \rceil \tag{15}$$

Step 2　有效信息合并. 额外数据流φ与编码流ξ作为恢复原始图像的有效信息,按照额外数据流在前,编码流在后的顺序,将两者采用位连接的方法合并为原始流S.

3.1.4 伪标记图与加密图像构造

在构造加密图像过程中,主要实现3个目的:①为避免标记图泄露,对原始流进行加密;②为保证加密图像与原始图像尺寸一致,预留空间τ需满足一定长度;③为标识嵌入位置,构造伪标记图并嵌入加密图像中. 实现步骤如下.

Step1　原始流加密. 根据加密密钥K_{e}生成与原始流相同长度的伪随机比特流R_1和伪随机置乱序列R_2后,通过R_1对原始流异或加密后再通过R_2进行置乱加密,生成加密流S'.

Step2　预留空间长度的确定. 根据已知的额外数据流长度l_φ和编码流长度l_ξ,确定预留空间τ的长度l_τ为

$$l_\tau = W \times H \times 8 - l_\varphi - l_\xi - \lceil \log_2 (W \times H \times 8) \rceil \tag{16}$$

其中,$\lceil \log_2 (W \times H \times 8) \rceil$位用于记录伪标记图信息$m$. 预留空间$\tau$则通过随机序列填充.

Step3　伪标记图的生成. 将加密流S',预留空间τ及$\lceil \log_2 (W \times H \times 8) \rceil$位连接生成加密图像流$S_{\mathrm{e}}$. 再将$S_{\mathrm{e}}$分为大小为$W \times H$的8个位平面$\overline{E}_i, i = 1, 2, \cdots, 8$,按照式(17)将位平面结合为中间加密图像$\overline{E}$,即

$$\bar{E} = \sum_{i=1}^{8} \bar{E}_i \times 2^{i-1} \tag{17}$$

统计预留空间在加密图像中的起始位置$(x,y), 1 \leqslant x \leqslant W, 1 \leqslant y \leqslant H$以及占有的完整位面个数$n$. 随后以$(x,y)$为分界点,生成前面全为$n$,后面全为$n+1$,大小为$W \times H$的伪标记图$L'$.

Step4　加密图像的生成. 将伪标记图的重要信息:分界点(x,y)与n分别转化为$\lceil \log_2 W \rceil + \lceil \log_2 H \rceil$位和$\lceil \log_2 8 \rceil = 3$位作为伪标记图信息存储至中间加密图像$\bar{E}$的LSB最后$\lceil \log_2 (W \times H \times 8) \rceil$位中,生成加密图像$E$.

为更好地介绍伪标记图与加密图像的构造过程,以图3为例进行说明. 原始流加密生成加密流后,与预留空间和$\lceil \log_2 (W \times H \times 8) \rceil$位连接后生成中间加密图像. 该图像共分为8个位平面,最外侧为MSB,最内侧为LSB. 8个位平面中的内容依次说明如下:前4个MSBs中均为加密流;第5个位平面中包含加密流与预留空间2部分,且以坐标(x,y)作为分界点;最后3个位平面中除最后$\lceil \log_2 (W \times H \times 8) \rceil$位外均为预留空间. 此时,该图像在$(x,y)$前可在3LSBs嵌入,在$(x,y)$后可在4LSBs嵌入. 为在数据嵌入阶段嵌入秘密数据,需告知管理者可嵌入位置,因此构造以(x,y)为分界点,前面全为3,后边全为4的伪标记图. 随后将分界点(x,y)与$n=3$两部分伪标记图信息嵌入至中间加密图像最后$\lceil \log_2 (W \times H \times 8) \rceil$位生成加密图像. 在数据嵌入阶段中,管理者根据LSB中存储的伪标记图信息重构伪标记图,确定预留空间位置.

3.2 数据嵌入与提取

持有信息隐藏密钥K_{m}的管理者,可在加密图像E中嵌入秘密数据D,以实现对密文图像的管理. 数据嵌入过程中,将秘密数据D加密后嵌入到预留空间τ中生

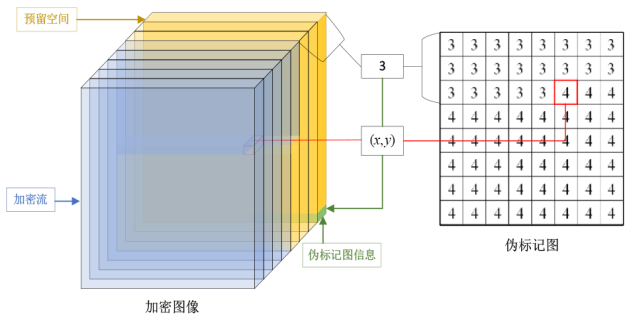


图3 标记图与加密图像的构造过程示例

成含嵌图像M. 数据提取则是数据嵌入的逆过程. 为保证秘密数据的安全性,秘密数据在嵌入前先要加密再嵌入. 为适用于不同应用,秘密数据D可以是比特流或图像形式,需提供不同的加密方式. 当秘密数据D为比特流时,可选择复杂度较低的异或加密,也可选择安全性较高的DES[23]和AES[24]等加密方法;当秘密数据D为图像时,可选择现有的安全性高的图像加密方法. 例如,混合混沌加密[25]、双加密结合方案[26,27]等.

Step1 数据嵌入. 通过加密图像E中LSB的最后$\lceil \log_2(W\times H\times 8)\rceil$位得到伪标记图信息,进而恢复以$(x,y)$为分界点,前面为$n$,后面为$n+1$的伪标记图$L'$. 随后,根据$L'$采用位替换的将加密的秘密数据$D'$依次嵌入至$E$的$n$LSBs或$(n+1)$LSBs中生成含嵌图像$M$.

Step2 数据提取. 与数据嵌入步骤类似,通过含嵌图像M中LSB的最后$\lceil \log_2(W\times H\times 8)\rceil$位恢复伪标记图$L'$,随后从$M$中$n$LSBs或$(n+1)$LSBs提取加密的秘密数据$D'$,用信息隐藏密钥$K_m$对$D'$解密恢复原始的秘密数据$D$.

3.3 图像恢复

在持有加密密钥K_e的接收者得到含嵌图像M后,可恢复出与原始图像O完全相同的恢复图像R. 在恢复过程中,先从含嵌图像M中得到加密流S',解密后得到的原始流S. 通过S包含的范式哈夫曼编码表T和编码流ξ恢复出二维标记图L后,结合参考像素o_{ref}和未标记位B_o生成恢复图像R.

Step1 原始流的恢复. 通过含嵌图像M恢复伪标记图L'后,根据L'中标记确定预留空间长度l_τ. 将含嵌图像展开为八个位平面并排列为比特流的形式,此时前$W\times H\times 8-l_\tau-\lceil \log_2(W\times H\times 8)\rceil$位即为加密流$S'$,根据加密密钥$K_e$依次进行置乱解密和异或解密后恢复原始流$S$.

Step2 二维标记图的恢复. 根据原始流S恢复额外数据流φ与编码流ξ,额外数据流φ中包括参考像素o_{ref},未标记位B_o,范式哈夫曼编码表T及编码流长度l_ξ. 结合范式哈夫曼编码表与编码流恢复二维标记图L.

Step3 恢复图像的生成. 定义$r(i,j),1\leqslant i\leqslant W,1\leqslant j\leqslant H$为恢复图像$R$中第$i$行第$j$列的像素值. 首先将参考像素$o_{ref}$作为$r(1,1)$,对其余像素,采用与式(1)相同的预测算法,得到预测像素值$p(i,j)$后,对应位置恢复像素值$r(i,j)$可根据二维标记$(l_1(i,j),l_2(i,j))$与未标记位B_o得出. 预测像素值$p(i,j)$与恢复像素值$r(i,j)$从MSBs到LSBs有$l_1(i,j)$位初始连续相同MSBs及$l_2(i,j)$位SCO-MSBs,且第$l_1(i,j)+l_2(i,j)+1$位MSB相同. 恢复像素值$r(i,j)$剩余$7-l_1(i,j)-l_2(i,j)$位LSBs通过未标记位B_o进行恢复. 将所有的像素值恢复后,得到恢复图像R. 图像恢复为加密图像的生成的逆过程,恢复图像R与原始图像O完全相同,算法实现完全可逆.

4 实验结果与分析

在本节中,设计相关实验来验证算法的性能,实验主要包括5部分:①二维标记性能分析;②嵌入容量;③安全性分析;④可逆性分析;⑤时间复杂度分析. 实验选取如图4所示的八幅测试图像,分别为Lena,Jetplane,Barbara,Peppers,Boat,Lake,Crowd和Baboon. 同时使用包含了1 338幅图像的UCID[28]与包含了10 000幅图像的BOSSbase[29]和BOWS2[30]的3个图像库进行实验. 其中,UCID图像库全部变为512×512大小的图像.

4.1 二维标记性能分析

本算法在腾出空间阶段采用二维标记策略,结合范式哈夫曼编码来进一步提高文献[19]的嵌入容量. 在本文和文献[19]中,与嵌入容量EC相关的参数主要包括总容量TC,编码流长度l_ξ,额外数据流长度l_φ. 下面从这3个方面对比分析.

图像的总容量为各类标记总容量之和,在文献[19]中,标记值表示原始像素值与预测像素值相同的MSBs位数,与本文中二维标记的l_1是一致的,因此各标记的总容量为

$$\overline{\mathrm{TC}}(l_1)=\begin{cases}8, & l_1\geqslant 8\\ l_1+1, & \text{其他}\end{cases} \tag{18}$$

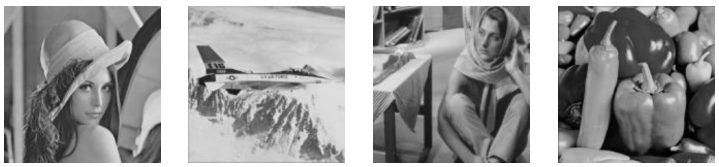
而本文算中各标记的总容量为

$$\overline{\overline{\mathrm{TC}}}(l_1)=\begin{cases}8, & l_1+l_2\geqslant 8\\ l_1+l_2+1, & \text{其他}\end{cases} \tag{19}$$

则本文算法中除首行首列像素外,各标记的总容量增长量为

$$\Delta\mathrm{TC}(l_1,l_2)=\overline{\overline{\mathrm{TC}}}(l_1,l_2)-\overline{\mathrm{TC}}(l_1) \tag{20}$$

文献[19]中,首行首列的像素作为参考像素,无法嵌入信息. 本文算法将第一个像素作为参考像素且保存在额外数据流中,首行首列像素同样采用二维标记腾出空间,从而进一步提高总容量.



(a) Lena　(b) Jetplane　(c) Baboon　(d) Peppers

(e) Boat　(f) Lake　(g) Crowd　(h) Baboon

图4　测试图像

与文献[19]中的标记方法相比,二维标记策略中标记的总类别数有所提高,最多为37类,因此在编码流长度与额外数据流长度方面发生增长. 由于编码长度与各标记出现的频率有关,此处将除首行首列外,各类标记出现的次数定义为$n(l_1,l_2)$. 本文中通过范式哈夫曼编码对二维标记图编码生成编码流,因此均匀分布时,该编码方法效率最差. 假设当l_1相同时的类别数量全部相同,即平均数量$n'(l_1)$为

$$n'(l_1)=\left\lfloor \frac{\sum_{l_2=1}^{8-l_1} n(l_1,l_2)}{8-l_1} \right\rfloor \tag{21}$$

则此时l_1对应的各标记对应的编码长度平均值$\overline{\overline{l_\xi}}'(l_1)$近似为

$$\overline{\overline{l_\xi}}'(l_1)\approx \left\lceil \log_2 \frac{(W-1)\times(H-1)}{n'(l_1)} \right\rceil \tag{22}$$

故除首行首列外,各标记的编码长度增长量近似为

$$\Delta l_\xi' \approx \overline{\overline{l_\xi}}'(l_1)-\overline{l}_\xi(l_1) \tag{23}$$

其中,$\overline{l}_\xi(l_1)$表示文献[19]中标记l_1对应的编码长度.

在额外数据流方面,为与文献[19]进行对比,需除去未标记位B_o. 此时假设l_{max}达到最大值,即$l_{max}=36$时,额外数据流长度为494位,相较于文献[19]中的52位,提高了442位,对图像嵌入容量的影响较小,因此在针对具体图像的分析过程中,主要从总容量和编码流长度上说明二维标记策略引起的嵌入容量增长量. 此时各标记的嵌入容量增长量约为

$$\Delta EC'(l_1,l_2)\approx \Delta TC(l_1,l_2)-\Delta l_\xi'(l_1) \tag{24}$$

当标记总容量的增长量大于编码长度的增长量,则嵌入容量呈现正增长;反之为负增长.

下面以Lena图像为例进行实验,测试结果如表1所示. 可以看出当$l_1=0$时,文献[19]的$\overline{TC}(0)=1$,$\overline{l}_\xi(0)=5$. 同时该标记数量为9 823,则二维标记中,当$l_2=1,2,\cdots,8$时,平均数量$n'(0)=1\,227$,编码长度平均值$\overline{\overline{l_\xi}}'(0)\approx 8$,则有$\Delta l_\xi'(0)\approx 3$. 此时当$\Delta TC(0,l_2)\geqslant 3$时,嵌入容量实现正增长. 观察表中数据,二维标记(0,1),(0,2),(0,3),(0,4),(0,5),(0,6),(0,7),(0,8),各标记的总容量分别为2,3,4,5,6,7,8,8位,相较于文献[19],前7个标记中,总容量增长量ΔTC为l_2位,但在标记(0,8)中,由于$l_1+l_2\geqslant 8$,因此ΔTC仅为$7-l_1$位. 且嵌入容量增长量ΔEC在$\Delta TC\geqslant 3$时为正,与估计的当$\Delta TC(0,l_2)\geqslant 3$时,嵌入容量实现正增长一致. 本文算法在$l_1=0$时的数量分布上,呈现以标记(0,5)为峰值,向上下两侧减少的趋势,由于上侧标记的总容量增长量较少,相应的嵌入容量产生负增长. 但在下侧标记中,随着总容量增长量的提高,嵌入容量也实现正增长. 最终本文相较于文献[19],在$l_1=0$时实现嵌入容量的提高. 其余标记同理,虽然在$l_1=4$时虽然嵌入容量增长量为负,但Lena图像整体的嵌入容量依旧提高78 663位,即0.300 bpp.

4.2　嵌入容量

在管理者对图像进行管理时,嵌入容量越大说明嵌入的秘密信息越多,即能嵌入更多与图像有关的标签,从而实现便捷分类管理. 为说明本文算法在嵌入容量上的性能,实验过程分为2个阶段:先以8幅测试图像为例,对比分析本文算法相较于文献[19]的提高. 再分别以测试图像和2个图像库为例,与现有的4篇相关文献进行对比,分析本文算法在嵌入容量上的优势.

4.2.1　与文献[19]的对比

在本文算法和文献[19]嵌入容量对比中,从总容量TC,编码流长度l_ξ以及额外数据流长度l_φ以及图像嵌入率rate进行测试. 同样地,为与文献[19]形成对比,本文的额外数据流需减去未标记位B_o.

以8幅测试图像为例,对比结果如表2所示. 以Lena为例可以看出,在总容量上,根据式(2),本文算法

表1 文献[19]与本文算法的标记策略对比

文献[19]					本文算法					增长量		
标记	数量	$\overline{\mathrm{TC}}$/位	$\overline{l}_\xi$/位	EC/位	二维标记	数量	$\overline{\overline{\mathrm{TC}}}$/位	$\overline{\overline{l}}_\xi$/位	EC/位	ΔTC/位	Δl_ξ/位	ΔEC/位
–1	1 023	—	—	—	—	1 023	7 646	4 395	3 251	7 646	4 395	3 251
0	9 823	1	5	4	(0,1)	15	2	11	–9	1	6	–5
					(0,2)	306	3	10	–7	2	5	–3
					(0,3)	901	4	8	–4	3	3	0
					(0,4)	1 501	5	8	–3	4	3	1
					(0,5)	2 439	6	7	–1	5	2	3
					(0,6)	2 280	7	7	0	6	2	4
					(0,7)	1 156	8	8	0	7	3	4
					(0,8)	1 225	8	8	0	7	3	4
1	9 778	2	5	3	(1,1)	223	3	11	–8	1	6	–5
					(1,2)	1 241	4	8	–4	2	3	–1
					(1,3)	2 510	5	7	–2	3	2	1
					(1,4)	2 482	6	7	–1	4	2	2
					(1,5)	1 585	7	8	–1	5	3	2
					(1,6)	908	8	8	0	6	3	3
					(1,7)	829	8	9	–1	6	4	2
2	15 343	3	4	1	(2,1)	1 812	4	7	–3	1	3	–2
					(2,2)	2 443	5	7	–2	2	3	–1
					(2,3)	3 651	6	6	0	3	2	1
					(2,4)	3 598	7	6	1	4	2	2
					(2,5)	1920	8	7	1	5	3	2
					(2,6)	1919	8	7	1	5	3	2
3	33 368	4	3	1	(3,1)	6 464	5	5	0	1	2	–1
					(3,2)	9 742	6	5	1	2	2	0
					(3,3)	8 240	7	5	2	3	2	1
					(3,4)	4 515	8	6	2	4	3	1
					(3,5)	4 407	8	6	2	4	3	1
4	44 478	5	2	3	(4,1)	16 507	6	4	2	1	2	–1
					(4,2)	13 601	7	4	3	2	2	0
					(4,3)	7 186	8	5	3	3	3	0
					(4,4)	7 184	8	5	3	3	3	0
5	53 281	6	2	4	(5,1)	25 723	7	3	4	1	1	0
					(5,2)	13 664	8	4	4	2	2	0
					(5,3)	13 894	8	4	4	2	2	0
6	41 487	7	3	4	(6,1)	20 832	8	4	4	1	1	0
					(6,2)	20 655	8	4	4	1	1	0
7	23 939	8	4	4	(7,1)	23 939	8	4	4	0	0	0
8	29 624	8	4	4	(8,0)	29 624	8	3	5	0	–1	1
Total	—	1 469 869	793 712	676 157	—	—	1 914 640	1 159 820	754 820	444 771	366 108	78 663

的每个标记与文献[19]相比多腾出ΔTC位,结合首行首列像素容量的提高量,整幅图像提高44万位. 但标记类别更多导致编码长度及额外数据流长度也有提高,编码长度提高量约为36万位,额外数据流仅提高292位. 由于总容量的提高量高于编码长度与额外数据流,最终的嵌入率依旧提高0.299 bpp. 统计8幅图像嵌入率提高量的平均值,本文算法相较于文献[19]能提高0.267 bpp.

表2　文献[19]与本文算法在嵌入容量上的对比

测试图像	算法	TC/位	l_ξ/位	l_φ/位	rate/bpp
Lena	文献[19]	1 469 869	793 712	52	2.579
	本文	1 914 640	1 159 820	344	2.878
	提高	444 771	366 108	292	0.299
Jetplane	文献[19]	1 587 492	792 365	52	3.033
	本文	1 964 178	1 094 646	344	3.316
	提高	376 686	302 281	292	0.283
Barbara	文献[19]	1 313 541	839 954	52	1.806
	本文	1 790 104	1 246 849	332	2.071
	提高	476 563	406 895	280	0.265
Peppers	文献[19]	1 386 368	783 602	52	2.299
	本文	1 865 253	1 188 976	344	2.578
	提高	478 885	405 374	292	0.279
Boat	文献[19]	1 470 649	794 800	52	2.578
	本文	1 908 999	1 162 174	344	2.848
	提高	438 350	367 374	292	0.270
Lake	文献[19]	1 302 121	785 928	52	1.969
	本文	1 791 291	1 210 354	344	2.215
	提高	489 170	424 426	292	0.246
Crowd	文献[19]	1 560 448	789 080	52	2.942
	本文	1 927 452	1 093 600	344	3.180
	提高	367 004	304 520	292	0.238
Baboon	文献[19]	1 074 525	794 539	52	1.068
	本文	1 616 592	1 269 867	332	1.321
	提高	542 067	475 328	280	0.253

4.2.2　与现有文献对比

在与现有文献的对比实验中,为公平性,设置参数如下:文献[14]设定块大小为3,$\alpha=5$,$\beta=2$;文献[15]则设定$\alpha=5$,$\beta=3$. 分别以8幅测试图像和3个数据库为例进行测试,测试结果如下.

以8幅测试图像为例进行测试的结果如表3所示,可以看出,文献[14]的容量相对较小,因为该算法在冗余较小加密图像中腾出空间. 文献[15,17,19]及本文算法则在原始图像中预留空间,因此容量相对较高. 本文算法在8幅测试图像中优于现有文献,对于适合MSBs编码的Jetplane图像,文献[19]的容量达到3.033 bpp,本文算法比其提高0.283 bpp;但对于适合预测误差编码的Peppers图像,现有文献中[15]的容量最高,为2.543 bpp. 本文算法与之比较依旧提高了0.035 bpp.

为避免偶然性,分别以3个图像库为例进行测试,结果如表4所示. 可以看出,在3个图像库中本文算法的平均嵌入容量均高于现有文献. 以UCID图像库为例,本文相较于文献[14,15,17,19],分别提高1.640 bpp,0.842 bpp,0.628 bpp,0.248 bpp. 在BOSSBase和BOWS2中,本文算法相较于平均嵌入率最高的文献,提高了0.297 bpp和0.208 bpp.

表3　最大嵌入率对比

测试图像	文献[14]	文献[15]	文献[17]	文献[19]	本文算法
Lena	2.014	2.643	2.590	2.579	2.878
Jetplane	2.191	2.685	2.922	3.033	3.316
Barbara	1.261	1.918	2.008	1.806	2.071
Peppers	2.042	2.543	2.507	2.299	2.578
Boat	1.726	2.550	2.550	2.578	2.848
Lake	1.524	2.090	2.188	1.969	2.215
Crowd	1.731	2.518	2.547	2.942	3.180
Baboon	0.723	0.968	1.219	1.068	1.321

表4　图像库最大嵌入率对比

测试图像	文献[14]	文献[15]	文献[17]	文献[19]	本文算法
UCID	1.680	2.478	2.692	3.072	3.320
BOSSBase	1.954	2.567	3.205	3.361	3.658
BOWS2	1.878	2.528	3.309	3.246	3.517

综上所述,本文算法在嵌入容量上优于现有文献,通过对文献[19]未标记位中冗余的利用,提高了嵌入容量,且在平均嵌入率方面,相较于现有算法能够提高0.208 bpp以上.

4.3　安全性分析

文献[19]中生成加密图像的步骤:(1)计算预测像素值,与原始像素进行比较生成标记;(2)对原始图像进行异或加密;(3)对标记图进行加密生成编码流;(4)将所有的辅助数据嵌入参考像素中,若参考像素无法完全嵌入,则根据标记图与被替代的原始参考像素一起嵌入到非参考像素中. 在该算法中存在两个隐患:一是为使管理者确定嵌入位置,标记图无法通过加密密钥加密,即标记图可直接从加密图像中提取恢复;二是加密图像由原始图像异或加密生成,虽然嵌入标记图等辅助数据时对其中一部分像素进行修改,但部分像素依旧保持不变,无法抵抗唯密文攻击. 下面从这两方面隐患对文献[19]及本文算法进行分析.

4.3.1　标记图信息泄露

在文献[19]和本文算法中,标记图实质是预测像素值与原始像素值MSBs的差异,对同一幅图像,本文算法中的l_1与文献[19]中的label是相同的. 实验过程中分别以Lena,Jetplane,Baboon为例,生成的标记图如图5所示. 可以看出,标记与图像的纹理程度有关,图像在纹理区域,预测像素与原始像素相同的MSBs位数较少;反之,在平滑区域,预测像素与原始像素相同的MSBs位数较多. 所以从标记图中反映出原始图像的纹理信息.

(a) Lena (b) Jetplane (c) Baboon

图5 标记图

分析文献[19]与本文算法通过加密图像恢复标记图的可能性. 在文献[19]生成的加密图像中,辅助数据未进行加密,标记图能通过辅助数据中的编码规则和编码流进行无损恢复,导致图像信息泄露. 本文算法则采用有效信息合并的方法,将额外数据流与编码流合并为原始流后进行异或和置乱加密生成加密流. 若要恢复原始的额外数据流与编码流,则需要从 $2^{l_\xi+l_\varphi}\times(l_\xi+l_\varphi)!$ 种情况中找到唯一正确的密钥. 此时,若无正确的加密密钥,很难通过加密流恢复为原始流,即无法通过加密流恢复标记图. 为说明加密密钥错误时本文算法无法正确恢复标记图,由加密流恢复的比特流与原始流的差异,定义恢复比特流相较于原始流的正确率公式如下:

$$R_a=\frac{\sum_{i=1}^{\text{len}} s(i)==s'(i)}{\text{len}}\times 100\% \tag{25}$$

其中,$s(i)$与$s'(i)$分别表示原始流与错误密钥下恢复比特流的第i位;len为原始流长度.

以Lena图像为例,测试100种错误密钥下由加密流恢复比特流的正确率,结果如图6所示. 可以看出,在加密密钥错误的情况下,正确率R_a的均值为50.02%,即恢复正确率仅有一半,无法从恢复的比特流中得出标记图信息. 因此,仅在密钥正确的情况下才能可逆恢复原始流信息,进而恢复正确的标记图.

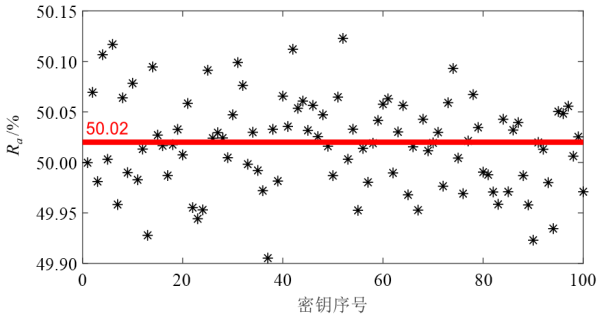


图6 错误密钥下正确率

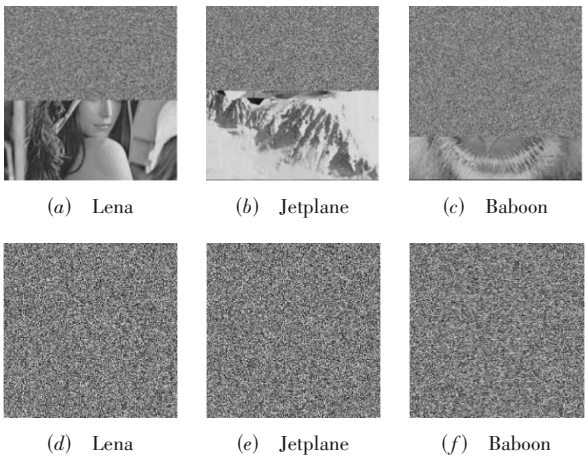
综上,文献[19]中标记图未进行加密,导致原始图像中的纹理信息被泄露. 本文则通过对原始流进行加密保证了标记图的安全性,且加密流仅在加密密钥正确的情况下才能正确恢复至原始流,有效地避免了标记图信息泄露.

4.3.2 唯密文攻击

Khelifi提出的有关异或加密的唯密文攻击中,利用自然图像中存在的冗余来估计异或加密矩阵. 在算法中,先将异或加密矩阵各位面的第一个比特设定为"1". 随后估计第一行的比特,再由第一行的比特估计出其余比特. 当所有比特估计后进一步改善,设定第一个像素为0~255,在256种情况中选取最佳的结果. 下面以该攻击算法为依据对比分析文献[19]及本文算法抵抗唯密文攻击的性能.

文献[19]中仅采用异或加密方法,虽然加密图像中前面的像素在嵌入辅助数据时发生修改,但后面的像素未发生变化,即相较于原始像素值仅进行了异或加密,因此依旧可以利用Khelifi提出的根据自然图像中的冗余进行攻击,存在安全隐患. 在设计攻击实验时,考虑到文献[19]在图像的后部分异或加密后未发生变化,对唯密文攻击算法进行一些调整:各位面从最后一个比特开始,从下向上估计异或加密矩阵. 最后采用该矩阵解密其余加密图像得到破解图像. 在实验过程中,分别通过文献[19]及本文算法在相同的加密密钥下对Lena,Jetplane,Baboon及100张BOSSBase中图像进行加密. 以BOSSBase中100张加密图像估计异或加密矩阵分别对Lena,Jetplane,Baboon解密得出破解图像,结果如图7所示. 图7(a)~(c)为文献[19]的破解图像,图7(d)~(f)为本文的破解图像. 从图7(a)~(c)这3幅破解图像可以看出,文献[19]破解图像的下半部分图像内容被泄露,且被泄露部分的大小与辅助数据有关. 当辅助数据较长时,加密图像中更多的像素被修改,从而在破解图像中仅能恢复较少的像素,如Baboon图像. 反之,当辅助数据较短时,破解图像中将泄露大量的原始图像信息. 观察本文得到的图7(d)~(f)破解图像,3幅图像类似随机噪声,即本文破解图像中均未泄露出原始图像的相关信息. 分析可知,本文算法中加密前图像相当于将加密图像中的加密流替换为原始流,即加密前图像包括原始流、预留空间与伪标记图信息,由于三者相邻比特间并未存在相关性,因此本文生成的加密图像无法通过文献[20]中的唯密文攻击破解.

为进一步说明本文算法能够有效抵抗上述唯密文攻击算法,从冗余度方面进行分析. 水平冗余度P_h与垂直冗余度P_v的计算式如文献[20]中式(5)与式(7). 以Lena,Jetplane,Baboon为例,按照各位面进行加密前图像冗余度测试,结果如表5所示. 可以看出,三幅图像的水平与垂直冗余度约为0.5,即本文算法中生成的加密图像无法通过唯密文攻击进行破解. 因为在本文算法中额外数据流与编码流经过异或与置乱加密,生



(*a*) Lena　(*b*) Jetplane　(*c*) Baboon

(*d*) Lena　(*e*) Jetplane　(*f*) Baboon

图7　破解图像

成的加密流近似伪随机分布,同时预留空间由伪随机序列填充,所以最终构造的加密图像相邻像素间的相关性较低.

表5　冗余度测试

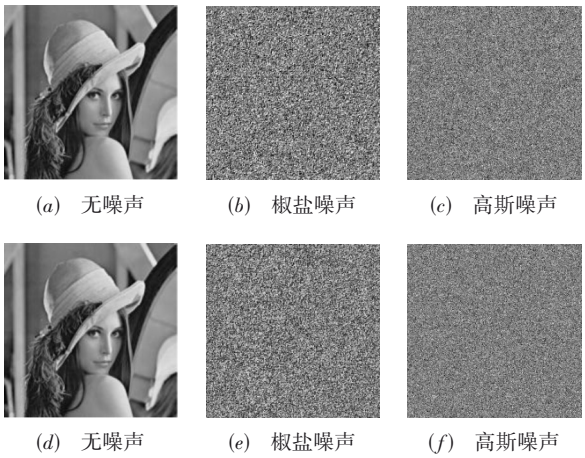
位面	Lena		Jetplane		Baboon	
	P_h	P_v	P_h	P_v	P_h	P_v
1	0.499	0.500	0.497	0.500	0.502	0.504
2	0.502	0.494	0.502	0.504	0.502	0.506
3	0.503	0.503	0.503	0.511	0.500	0.520
4	0.502	0.489	0.503	0.501	0.502	0.497
5	0.500	0.482	0.497	0.471	0.502	0.486
6	0.500	0.498	0.501	0.501	0.500	0.525
7	0.501	0.500	0.498	0.501	0.501	0.507
8	0.498	0.502	0.501	0.499	0.502	0.501

综上,本文算法采用有效信息合并的方法,将额外数据流及编码流合并为原始流,并进行异或与置乱加密,因此在无正确加密密钥时无法通过加密图像恢复标记图,同时加密前图像的冗余近似为0.5,能有效抵抗唯密文攻击.

4.4　可逆性分析

为分析文献[19]与本文算法的可逆性,分别从含嵌图像在有无噪声情况下进行图像恢复. 在含嵌图像加入噪声下,为获得恢复图像,当编码流长度不足时,将编码流进行复制,提高编码流长度.

以Lena图像为例,分别测试文献[19]与本文算法中含嵌图像在无噪声、噪声密度为0.05的椒盐噪声以及均值为0方差为0.01的高斯噪声的影响下恢复图像的效果,结果如图8所示. 图中第一行为文献[19]的恢复图像,第二行为本文算法的恢复结果. 可以看出,在无噪声的情况下,文献[19]与本文算法的恢复图像与原始图像完全一致,满足可逆性. 但在加入噪声后,文献[19]与本文算法获得的恢复图像均为类似噪声的图像. 分析原因可知,当含嵌图像中加入噪声后,会对编码表及编码流产生影响,从而无法恢复正确的标记图. 同时原始像素中的未标记位也会发生错误,结合MED预测方法的错误传递性,当一个像素值恢复发生错误时,将会对该像素右下方像素的恢复产生影响,最终导致恢复图像产生错误. 综上可知,在含嵌图像中未包含噪声时,算法可实现完全可逆,恢复图像与原始图像完全一致. 但在加入噪声后,由于哈夫曼编码的特殊性,若编码流中出现错误将会导致解码错误的特性,在加入噪声后恢复图像与原始图像截然不同. 设计鲁棒的编码方法是需要进一步研究的方向.



(*a*) 无噪声　(*b*) 椒盐噪声　(*c*) 高斯噪声

(*d*) 无噪声　(*e*) 椒盐噪声　(*f*) 高斯噪声

图8　恢复图像

4.5　时间复杂度分析

文献[19]与本文算法可以分为4个阶段:加密图像的生成、数据嵌入、数据提取、图像恢复. 由于数据提取与图像恢复分别是加密图像的生成和数据嵌入的逆过程,所以主要对比分析文献[19]与本文算法中加密图像的生成和数据嵌入2个阶段的时间复杂度,以及4个阶段的运行时间来说明本文算法的性能.

在加密图像的生成阶段,本文算法相较于文献[19]多了范式哈夫曼编码表构造的步骤,但少了标记图嵌入的步骤. 范式哈夫曼编码表构造的时间复杂度为$O(37\times\log_2 37)$. 标记图嵌入依次在像素中嵌入辅助数据,因此时间复杂度为$O(W\times H)$. 其余构造标记图与标记图编码步骤等步骤均与图像大小相关,因此时间复杂度也为$O(W\times H)$. 在数据嵌入阶段,文献[19]需要先恢复标记图,再根据标记图依次在加密图像的像素中进行信息嵌入,因此时间复杂度为$O(W\times H)$. 本文算法则直接采用位替换进行嵌入,时间复杂度为$O(1)$.

为进一步体现本文算法在时间复杂度上的优越性,对比测试了文献[19]与本文算法在4个阶段的运行时间,实验过程中测试10次512×512的Lena图像求取

平均值. 实验环境为Windows10操作系统,MATLAB R2018b,AMDR7 4800U的CPU,16 GB内存(15.4 GB可用),64位操作系统. 实验结果如表6所示. 可以看出,本文算法在加密图像的生成与数据嵌入阶段的运行时间均低于文献[19],且整体算法的运行时间相较于文献[19]减少3.652 s.

表6 运行时间 单位:s

阶段	文献[19]	本文算法	降低
加密图像的生成	1.278	0.790	0.488
数据嵌入	1.375	0.051	1.324
数据提取	1.319	0.032	1.287
图像恢复	0.995	0.442	0.553
总时间	4.967	1.315	3.652

5 结论

本文提出一种基于MSB二维标记的加密图像可逆数据隐藏算法. 定义的二维标记中包含的首"0"串长度与次"1"串长度保存原始与预测像素值初始连续相同MSBs位数与SCO-MSBs位数,能够有效地增加图像冗余的利用,结合范式哈夫曼编码实现嵌入容量的提升及图像的可逆恢复,以BOWS2图像库为例,平均嵌入率相较于现有算法提高0.208 bpp以上. 有效信息合并使恢复原始图像的有效信息与预留空间分离,结合构造的伪标记图,在不影响数据嵌入的情况下,一方面,原始流加密能够防止二维标记图被未授权者获得,且有效抵抗唯密文攻击,避免图像内容泄露;另一方面,额外数据流与编码流直接合并,使得预留空间为连续的空间,可通过伪标记图快速地确定嵌入位置,减少了额外数据流及秘密数据的嵌入时间. 在加密图像的生成与数据嵌入阶段的运行时间上仅有0.790 s和0.051 s,且总体运行时间为文献[19]的26.5%. 后续工作将围绕更高效、鲁棒的编码方法设计以及更安全的加密算法研究等方面展开.

参考文献

[1] SHI Y Q, LI X L, ZHANG X P, et al. Reversible data hiding: Advances in the past two decades[J]. IEEE Access, 2016, 4: 3210-3237.

[2] QIAN Z X, ZHOU H, ZHANG X P, et al. Separable reversible data hiding in encrypted JPEG bitstreams[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 15(6): 1055-1067.

[3] LI M, XIAO D, KULSOOM A, et al. Improved reversible data hiding for encrypted images using full embedding strategy[J]. Electronics Letters, 2015, 51(9): 690-691.

[4] BOUSLIMI D, COATRIEUX G, COZIC M, et al. A joint encryption/watermarking system for verifying the reliability of medical images[J]. IEEE Transactions on Information Technology in Biomedicine, 2012, 16(5): 891-899.

[5] 王继军, 孙泽锐, 李国祥. 图像抛物线插值空间大容量可逆信息隐藏算法[J]. 电子学报, 2019, 47(1): 137-144.
WANG J J, SUN Z R, LI G X. High capacity reversible data hiding algorithm based on parabolic interpolation space [J]. Acta Electronica Sinica, 2019, 47(1): 137-144. (in Chinese)

[6] 王继军, 李国祥, 夏国恩, 等. 图像插值空间完全可逆可分离密文域信息隐藏算法[J]. 电子学报, 2020, 48(1): 92-100.
WANG J J, LI G X, XIA G E, et al. A separable and reversible data hiding algorithm in encrypted domain based on image interpolation space[J]. Acta Electronica Sinica, 2020, 48(1): 92-100. (in Chinese)

[7] MA K D, ZHANG W M, ZHAO X F, et al. Reversible data hiding in encrypted images by reserving room before encryption[J]. IEEE Transactions on Information Forensics and Security, 2013, 8(3): 553-562.

[8] JIA Y J, YIN Z X, ZHANG X P, et al. Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting[J]. Signal Processing, 2019, 163: 238-246.

[9] WANG W Q, YE J Y, WANG T Q, et al. Reversible data hiding scheme based on significant-bit-difference expansion[J]. IET Image Processing, 2017, 11(11): 1002-1014.

[10] KE Y, ZHANG M Q, LIU J, et al. Fully homomorphic encryption encapsulated difference expansion for reversible data hiding in encrypted domain[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(8): 2353-2365.

[11] KHELIFI F, BRAHIMI T, HAN J G, et al. Secure and privacy-preserving data sharing in the cloud based on lossless image coding[J]. Signal Processing, 2018, 148: 91-101.

[12] LI F Y, ZHANG L M, WEI W M. Reversible data hiding in encrypted binary image with shared pixel prediction and halving compression[J]. EURASIP Journal on Image and Video Processing, 2020, 2020(1): 1-21.

[13] YI S, ZHOU Y C. Parametric reversible data hiding in encrypted images using adaptive bit-level data embedding and checkerboard based prediction[J]. Signal Processing, 2018, 150: 171-182.

[14] YI S, ZHOU Y C. Separable and reversible data hiding in encrypted images using parametric binary tree labeling [J]. IEEE Transactions on Multimedia, 2019, 21(1):

51-64.
[15] WU Y Q, XIANG Y Z, GUO Y T, et al. An improved reversible data hiding in encrypted images using parametric binary tree labeling[J]. IEEE Transactions on Multimedia, 2020, 22(8): 1929-1938.
[16] YI S, ZHOU Y C. Binary-block embedding for reversible data hiding in encrypted images[J]. Signal Processing, 2017, 133: 40-51.
[17] CHEN F, YUAN Y, HE H J, et al. Multi-MSB compression based reversible data hiding scheme in encrypted images[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 31(3): 905-916.
[18] FU Y J, KONG P, YAO H, et al. Effective reversible data hiding in encrypted image with adaptive encoding strategy [J]. Information Sciences, 2019, 494: 21-36.
[19] YIN Z X, XIANG Y Z, ZHANG X P. Reversible data hiding in encrypted images based on multi-MSB prediction and huffman coding[J]. IEEE Transactions on Multimedia, 2020, 22(4): 874-884.
[20] KHELIFI F. On the security of a stream cipher in reversible data hiding schemes operating in the encrypted domain[J]. Signal Processing, 2018, 143: 336-345.
[21] GUAN B, XU D W. An efficient high-capacity reversible data hiding scheme for encrypted images[J]. Journal of Visual Communication and Image Representation, 2020, 66: 102744.
[22] COLTUC D. Improved embedding for prediction-based reversible watermarking[J]. IEEE Transactions on Information Forensics and Security, 2011, 6(3): 873-882.
[23] 卡茨安. 标准数据加密算法[M]. 陈太一, 屠世桢, 译. 北京: 人民邮电出版社, 1983.
[24] 郎荣玲, 夏煜, 戴冠中. 高级加密标准(AES)算法的研究[J]. 小型微型计算机系统, 2003, 24(5): 905-908.
LANG R L, XIA Y, DAI G Z. Research on the algorithm of advanced encryption standard (AES)[J]. Mini-Micro Systems, 2003, 24(5): 905-908. (in Chinese)
[25] ALAWIDA M, SAMSUDIN A, TEH J S, et al. A new hybrid digital chaotic system with applications in image encryption[J]. Signal Processing, 2019, 160: 45-58.
[26] ZHANG L N, WEI D Y. Image watermarking based on matrix decomposition and gyrator transform in invariant integer wavelet domain[J]. Signal Processing, 2020, 169: 107421.
[27] LI Y M, WEI D Y, ZHANG L N. Double-encrypted watermarking algorithm based on cosine transform and fractional Fourier transform in invariant wavelet domain[J]. Information Sciences, 2021, 551: 205-227.
[28] SCHAEFER G, STICH M. UCID: An uncompressed color image database[C]//Storage and Retrieval Methods and Applications for Multimedia. San Jose: SPIE, 2004, 5307: 472-480.
[29] BAS P, FILLER T, PEVNÝ T. "Break Our Steganographic System": The ins and outs of organizing BOSS[C]//International Workshop on Information Hiding. Berlin: Springer, 2011: 59-70.
[30] BAS P, FURON T. Image database of BOWS-2[EB/OL]. (2017)[2021]. http://bows2.ec-lille.fr.

作者简介

杨尧林　男,1996年出生,河南许昌人. 现为西南交通大学信息科学与技术学院博士生. 主要研究方向为图像处理和加密域可逆信息隐藏.
E-mail: ylyangwr@foxmail.com

和红杰(通讯作者)　女,1971年出生,四川成都人. 现为西南交通大学信息科学与技术学院教授. 主要研究方向为信息隐藏、图像处理和深度学习.
E-mail: hjhe@swjtu.edu.cn