Software Engineering and Applications 软件工程与应用, 2025, 14(2), 511-527
Published Online April 2025 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2025.142045

Hans汉斯

基于综合压缩策略的加密域可逆信息隐藏算法

董海卿

上海理工大学光电信息与计算机工程学院，上海

收稿日期：2025年3月17日；录用日期：2025年4月21日；发布日期：2025年4月29日

摘　要

加密图像可逆数据隐藏(Reversible Data Hiding in Encrypted Images, RDHEI)属于多媒体信息安全技术重要研究内容。该技术不仅能够通过加密的载体图像传输秘密数据，还能在不损失信息的情况下恢复载体图像，并准确提取秘密数据。加密后腾出空间(Vacating Room After Encryption, VRAE)是RDHEI的一个常用框架，但其嵌入容量较低。本文提出了一种基于子块的加密方法，即在同一块内使用相同的密钥进行逐位流加密，以保留块内像素间的相关性。为增强安全性，使用加密密钥K_e置乱图像子块的排列顺序。在数据隐藏过程中，通过分析子块内像素连续相同的高位平面数量，将该数量作为子块的块标签。引入参数T，通过块标签与参数T进行比较，为不同子块选择最优压缩策略。具体来说，对于块标签大于T的子块，采用相同位平面压缩策略，并以熵编码形式进行标记；对于块标签小于T的子块，采用保留差值的压缩策略。实验结果表明，在保证安全性和可逆性的前提下，提出方法在嵌入性能上实现了显著的优化效果。

关键词

加密后腾出空间(VRAE)，块级流加密，综合压缩策略

Reversible Data Hiding in Encrypted Images Based on Comprehensive Compression Strategy

Haiqing Dong

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Mar. 17th, 2025; accepted: Apr. 21st, 2025; published: Apr. 29th, 2025

文章引用: 董海卿. 基于综合压缩策略的加密域可逆信息隐藏算法[J]. 软件工程与应用, 2025, 14(2): 511-527.
DOI: 10.12677/sea.2025.142045

Abstract

Reversible Data Hiding in Encrypted Images (RDHEI) is an important research topic in multimedia information security technology. This technology can not only transmit secret data through encrypted carrier images, but also recover carrier images without losing information and accurately extract secret data. Vacating room after encryption (VRAE) is a common framework for RDHEI, but its embedding capacity is low. In this paper, we employ a block-based stream encryption method, i.e., bit-by-bit stream encryption using the same key within the same block to preserve the correlation between the pixels within the block. To enhance security, the ordering of image blocks is scrambled using the encryption key K_e. During the data hiding process, the number of consecutively identical high planes of pixels within a sub-block is calculated using that number as the block label of the sub-block. This paper introduces parameter T and compares it with block labels to select the optimal compression strategy for different sub blocks. Specifically, for blocks with labels greater than T, the same bit plane compression strategy is used and labeled in entropy encoding form; For blocks with labels less than T, a compression strategy that preserves the difference value is adopted. The experimental results show that the proposed method achieves significant optimization effects in embedding performance while ensuring safety and reversibility.

Keywords

Vacating Room after Encryption (VRAE), Block-Based Stream Encryption, Comprehensive Compression Strategy

1. 引言

随着云计算的快速发展，越来越多的图像数据被上传至云服务器。然而，在上传过程中，这些图像面临着保密性、身份验证和完整性等多方面的安全隐患。因此，保护多媒体数据的安全已成为一项关键任务。为了增强用户对隐私保护的信心，云平台鼓励用户对图像进行加密处理。此外，云平台可以在加密图像中嵌入额外信息，用于版权保护、秘密数据传输或实现其他安全功能。为满足这一需求，加密域的可逆信息隐藏技术(Reversible Data Hiding in Encrypted Images, RDHEI)应运而生。

目前，RDHEI 主要包括两种框架：加密前腾出嵌入空间(Reserving Room Before Encryption, RRBE)和加密后腾出嵌入空间(Vacating Room After Encryption, VRAE)。

由于 RRBE 框架对相邻像素间具有较强的相关性的明文图像进行无损压缩，所以 RRBE 算法通常能实现高嵌入率。内容所有者将压缩后的图像加密，并发送给数据隐藏者。数据隐藏者通过提取辅助信息，确定当前加密图像中可用于嵌入的数据空间。数据隐藏者将经嵌入密钥加密后的数据隐藏至加密图像中，并将携带加密数据的加密图像发送给接收者。接收者根据自身所持有的密钥获取相应的内容。Puteaux 和 Puech [1]提出在预处理过程中替换 MSB 平面并生成标签图。Yi 等学者[2]将原始图像划分为块，将划分的块分为好块或坏块，通过压缩好块来腾出容纳秘密数据的可用空间。Yin 等学者[3]在使用霍夫曼编码压缩标签之前，将原始像素的 8 位二进制与预测像素的 8 位二进制进行比较，并标记连续相同的高位平面。Puteaux 和 Puech [4]提出了一种递归方案，该方案对像素位平面进行可逆调整，以在解码步骤中检测和纠正所有预测误差较大的像素。

基于 VRAE 的算法通过无损压缩加密图像释放嵌入空间，这一过程通常由数据隐藏者执行。由于加密算法破坏像素间的相关性，使加密图像的像素值呈现接近均匀分布，因此基于 VRAE 的 RDHEI 算法的嵌入容量并不高。但 VRAE 算法具有更强的安全性。内容所有者只执行加密操作，无需额外嵌入辅助信息来标记嵌入空间的起止位置，将显著提升数据与原始图像的安全性。多位学者为了提高 VRAE 算法的嵌入容量，提出了若干基于分块策略的图像加密算法，即内容所有者将图片分成固定尺寸且互不重叠的像素块，使用加密密钥产生与块个数等量的随机数，子块内所有像素与同一随机数进行加密运算。这类加密算法能够保留子块内像素间的相关性，数据隐藏者将利用该相关性腾出嵌入空间。Liu 等人[5]通过使用块加密方法，在加密块中保留冗余。使用冗余矩阵表示方法生成容纳秘密数据的可用空间。Wang 和 He [6]通过使用随机数与块内像素进行逐位异或运算，保留了块内像素位平面间的相关性。Qiu 等人[7]使用随机数与块内像素进行相加再取模运算，像素间差值加密前后保持不变。为了进一步提升嵌入性能，Gao 等人[8]在 Wang 和 He [6]的基础上缩减辅助信息长度，但牺牲了部分嵌入空间，最终呈现出更高的嵌入率。

本文提出了一种基于综合压缩策略的加密域可逆信息隐藏算法。该算法采用基于子块的流加密技术，旨在最大限度地将原始图像子块内的相关性转移至加密后的图像对应子块中，并通过打乱子块排列顺序提升加密效果。为了优化压缩性能，本文引入了两种压缩方案对子块进行处理，并通过引入参数 T 作为压缩方案组合的关键指标，以实现最佳的压缩效果。实验结果表明，所提出的方法在嵌入容量方面显著优于 Wang 等人提出的方法[5] [6] [8]。具体地，本工作创新点如下：

(1) 采用块级流加密与块级置乱加密相结合的加密算法，实验证明，该加密算法既可保留子块内像素位平面间的相关性，并且能保证安全性；

(2) 将子块内像素连续相同的高位平面数量作为块标签，并采用熵编码对块标签进行标记，以最小化辅助信息的存储长度；

(3) 引入组合参数 T 作为综合压缩策略的关键决策因素，用于指导压缩策略的选择与实施；

(4) 提出方法在两个标准数据集上进行实验，以验证其优势。结果表明，所提出的 RDHEI 方法在有效载荷方面优于现有的基于 VRAE 的 RDHEI 方法。对于 BOSSbase [9]数据集和 BOWS-2 [10]数据集，所提出的 RDHEI 方法的平均有效载荷嵌入容量分别为 2.3268 bpp 和 2.3197 bpp。

2. 方法

本文算法的框架如图 1 所示，主要分为三个步骤：1) 图像加密：内容所有者使用加密密钥对原始图像进行加密，并将加密图像传输给数据隐藏者。2) 数据隐藏：数据隐藏者对加密图像进行相关性分析，以确定可用于数据隐藏的空间，并将加密后的数据隐藏至加密图像中。3) 数据提取与图像恢复：接收者根据所持有的密钥，选择性提取秘密信息或无损恢复原始图像。

2.1. 图像加密

块加密过程分为两部分：块级流加密和块置换。首先，对尺寸为 $H\times W$ 的原始图像 I 进行分块，将原始图像划分为 N 个尺寸为 $S\times S$ 且互不重叠的子块 $\{B_1, B_2, \cdots, B_N\}$，其中 $N=\left\lfloor \frac{H}{S} \right\rfloor \times \left\lfloor \frac{W}{S} \right\rfloor$。

1) 块级流加密

步骤 1：像素分解。块内的所有像素被分解为 8 位二进制数：

$$p_{i,j,k}=\left\lfloor \frac{P_{i,j}}{2^{8-k}} \right\rfloor \bmod 2,\ i=1,2,\cdots,N;\ j=1,2,\cdots,S\times S;\ k=1,2,\cdots,8 \tag{1}$$

式中，i 作为块的索引，j 作为块内像素的索引，$P_{i,j}$ 表示 B_i 中的第 j 个像素，其二进制形式的第 k 位比特值表示为 $p_{i,j,k}$，其中 $k=1$ 为最高位，运算符 mod 表示取余运算。

步骤 2：块级流加密。使用加密密钥 K_{e1} 生成长度为 N 的伪随机数组 R，并将每个元素分解为 8 位二进制数：

$$r_{i,k}=\left\lfloor \frac{R_i}{2^{8-k}} \right\rfloor \bmod 2 \tag{2}$$

式中，R_i 表示随机数组 R 中的第 i 个元素，其二进制形式的第 k 位比特值表示为 $r_{i,k}$。

具体加密过程通过公式(3)表示

$$e_{i,j,k}=r_{i,k}\oplus p_{i,j,k} \tag{3}$$

式中，$e_{i,j,k}$ 表示 B_i 中第 j 个加密像素的第 k 位比特值，符号 $\oplus$ 表示比特位异或运算。加密像素的十进制数值 $E_{i,j}$ 可表示为：

$$E_{i,j}=\sum_{k=1}^{8}2^{k-1}\times e_{i,j,k} \tag{4}$$

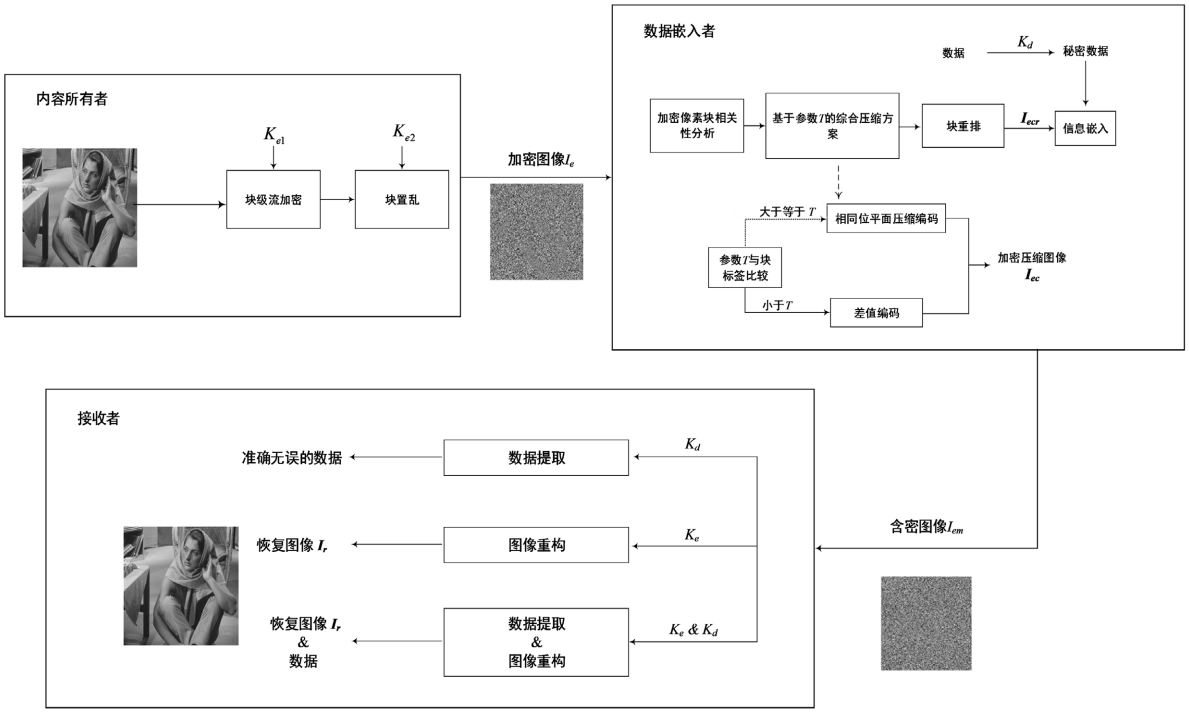
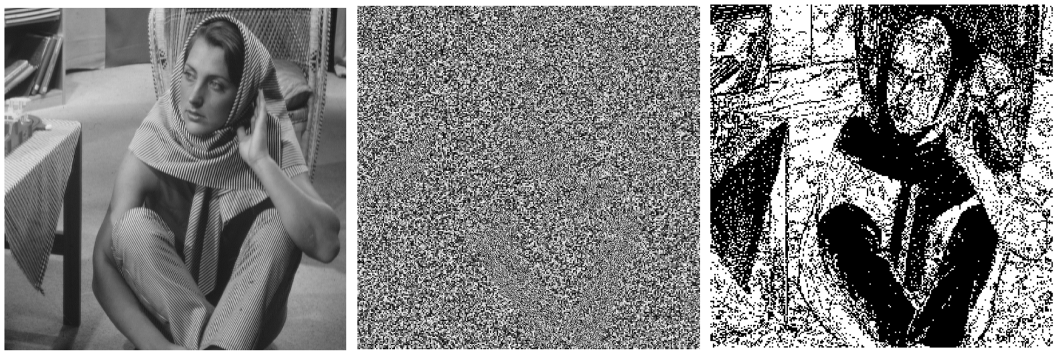


Figure 1. Framework of the proposed algorithm
图 1. 算法框架

图 2(a)、图 2(b)展示了标准测试图像 Barbara 及其仅使用块级流加密得到的加密图像。加密图像的内容完全不可辨认。由于采用了块级流加密，每个加密块与原始块在复杂度上具有较高的相似性，因此通过复杂度分析，仍能在一定程度上揭示图像的部分信息。复杂度分析的步骤如下：首先，将加密图像划分为大小相等的非重叠块，并计算每个块内像素的平均值。接下来，计算每个像素与其均值之间的差值。最后，对于所有差值小于或等于预定义阈值的平滑块，将所有像素值设为 255，而其他块的像素值则设为 0。图 2(b)展示了以阈值 5 进行复杂度分析后的结果，如图 2(c)所示，图像的轮廓已初步显现。因此，单

纯依赖块级流加密无法有效保障图像的安全性。为此，本章提出了在块级流加密的基础上，进一步引入块置乱加密算法，以增强图像的安全性。

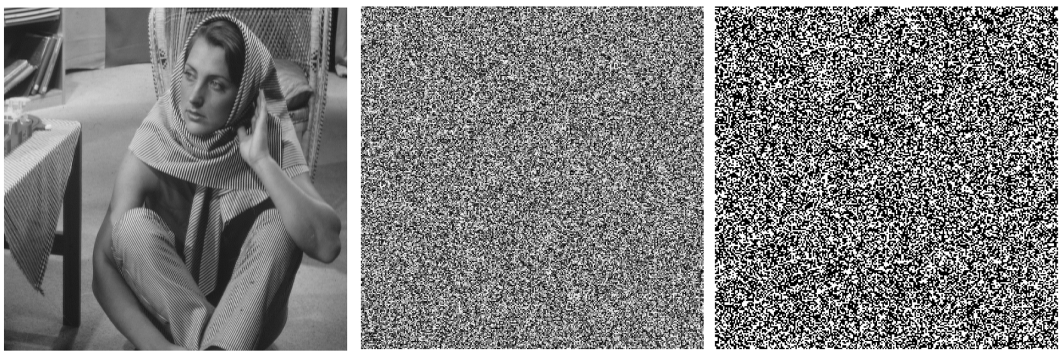


(a) (b) (c)

Figure 2. The images processed in the first encryption stage and its analysis results
图 2. 第一加密阶段处理后的图像及其分析结果

2) 块级置乱加密

通过加密密钥生成一个包含从 1 到 N 的元素的随机排列数组 R_2，并将该数组用作图像块的新排列索引。利用该索引打乱图像块之间的排列顺序，从而使图像块的内容分布更加随机，最终生成加密图像 I_e。



(a) (b) (c)

Figure 3. The images processed in the first encryption stage and its analysis results
图 3. 第二加密阶段处理后的图像及其分析结果

从图 3(a)~(c)可以看出，复杂度分析无法有效揭示图像的内容。因此，所提出的加密方法能够确保数据传输的安全性，即便加密图像被非法用户截获，也无法获取有效的图像信息。

2.2. 综合压缩方案

在上述加密算法中，通过像素块的排列顺序的置乱，相邻像素块之间的相关性不再保留。而位于同一子块内的像素使用相同的密钥进行加密运算，该加密方式可保留部分像素之间的相关性。图 4 展示了 2×2 像素块的加密运算示例，原始图像块为[179, 180, 169, 174]，经过加密后变为[209, 214, 203, 204]。在加密前，图像块内四个像素的连续相同位平面数量为 3；加密后，连续相同的位平面数量保持不变，仍为 3。该加密方法确保了加密前后，任意两个像素的二进制位中相同位和不同位的数量一致，即保留了子块内像素位平面之间的相关性。

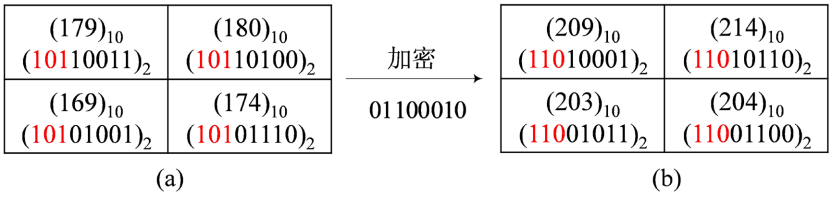


Figure 4. Example of 2 × 2 block level encryption. (a) Original block; (b) Encrypted block
图 4. 2 × 2 块级流加密示例。(a) 原始子块；(b) 加密子块

当图像中的像素块数值虽然非常接近，但相同位平面个数较少，这通常是由于十进制数值向二进制转换时产生的进位效应所导致的。为了充分利用图像中的冗余，针对此类像素块，采用压缩方案二进行处理。如图 5 所示，加密图像块[200, 183, 182, 193]展示了该压缩方案的应用。图 5(b)中载体像素的低 5 位存储其对应的差值，第三位(第一位为最高位)作为符号位，其中 0 表示正号或零值，1 表示负号。剩余的高位部分可作为嵌入空间，示例中每个载体像素可提供 2 比特的嵌入容量。然而使用相同位平面压缩方案(即图 5(a)中的红色部分)，子块内每个载体像素仅能提供 1 比特的嵌入空间。

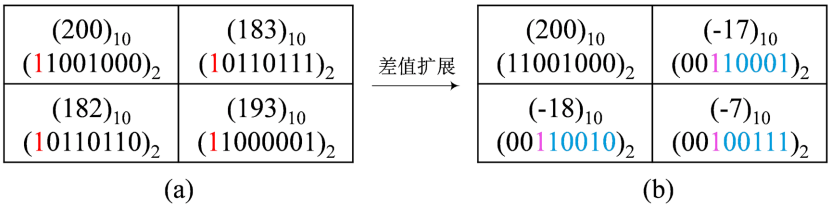


Figure 5. Example of difference expansion processing in 2 × 2 block. (a) Encrypted block; (b) The block after differential expansion processing
图 5. 2 × 2 子块差值扩展处理示例。(a) 加密子块；(b) 差值扩展处理后子块

1) 压缩方案一——相同位平面压缩编码

压缩方案一通过利用像素块内各像素相同位平面间的相关性来有效释放更多的嵌入空间。鉴于这种相关性在加密图像的所有子块中普遍存在，我们将该压缩方案作为主要的压缩方法，压缩方案二将作为对该主方案的补充。

图 4 示例相同位数量为 3，首个像素作为参考像素，在嵌入阶段保持不变，剩余三个像素的前 3 位可以作为嵌入空间。图 6 为尺寸为 2×2 的块内像素组成结构，P_r 作为参考像素以保证后续的像素恢复；$C_j\left(j=2,3,\cdots,S\times S\right)$ 用来腾出嵌入空间，运载额外信息。具体过程表示为

$$\mathrm{d_j}=\mathrm{dif}\left(\mathrm{P_r},\mathrm{C_j}\right),\mathrm{j}=2,3,\cdots,\mathrm{S}\times\mathrm{S} \tag{5}$$

式中 $\mathrm{dif}\left(\mathrm{a},\mathrm{b}\right)$ 将返回 a 与 b 不相同的低有效位的最大索引。例如，$\mathrm{dif}\left(77,86\right)=5$，因为 $77=\left(01001101\right)_2$、$86=\left(01010110\right)_2$。

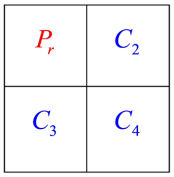


Figure 6. Composition of pixels within the 2 × 2 block
图 6. 2 × 2 的块内像素组成结构

使用块标签 *mark* 表示当前像素 C_j 可用于嵌入额外信息的空间，其计算公式为：

$$\text{mark} = 8 - \max\left(\text{dif}\left(\text{P}_\text{r}, \text{C}_\text{j}\right)\right),\ \text{j} = 2, \cdots, \text{S} \times \text{S} \tag{6}$$

式中，函数 max 为取最大值运算，为了减少辅助信息长度，块标签将使用霍夫曼编码进行标记。

2) 压缩方案二——差值编码

由于压缩方案一忽略了二进制形式下相关性较低但数值接近的块的压缩。因此，提出压缩方案二作为补充。首先计算差值，公式如下：

$$D_j = P_r - C_j, j = 2,3,\cdots,S \times S \tag{7}$$

式中，D_j 是参考像素 P 与像素 C_j 之间的差值，

$$D_{\max} = \max\left\{\left|D_2\right|, \left|D_3\right|, \cdots, \left|D_{s \times s}\right|\right\} \tag{8}$$

式中，$D_{\max}$ 表示取集合 $\left\{\left|D_j\right|, j = 2,3,\cdots,S \times S\right\}$ 的最大值。将 $D_{\max}$ 转化为二进制表示，其最短有效表示长度记为 M。令块内像素 C_j 的 $e_{i,j,M}$ 至 $e_{i,j,1}$ 存放 $\left|D_j\right|$ 的二进制位；$e_{i,j,M+1}$ 存放符号位，1 表示为负数，0 表示为正数；$e_{i,j,8}$ 至 $e_{i,j,M+2}$ 则作为嵌入空间。

当子块的 $D_{\max}$ 大于 63 时，即使采用压缩方案二，仍无法腾出嵌入空间。该子块的位置需要被标记，以便无失真地重构原始图像。具体的标记步骤如下：

步骤 1：计算子块的 $D_{\max}$。当 $D_{\max} > 63$ 时，标记位 $s_1 = 0$；当 $D_{\max} \le 63$ 时，$s_1 = 1$。

步骤 2：当 $s_1 = 0$ 时，当前子块的压缩嵌入操作中止。否则，继续执行步骤 3。

步骤 3：当 $s_1 = 1$ 时，当前子块可以被压缩。使用对应差值区间的码字标记。

差值编码表如表 1 所示，为了缩小两个压缩方案产生的辅助信息长度。在图像 Airplane、Barbara、Peppers、Boat、Lake、Baboon 的测试实验中发现，最大差值处于[0, 1]、(1, 3]区间上数目远远少于其他区间范围，最终将开始的 6 个差值区间缩减至表 1 的 4 个差值区间。将[0, 1]、(1, 3]、(3, 7]三个区间合并为[0, 7]区间。

Table 1. Difference coding table
表 1. 差值编码表

$D_{\max}$	[0, 7]	(7, 15]	(15, 31]	(31, 63]
码字	00	01	10	11

3) 参数 T 确定

通过递归遍历位平面，并以嵌入率作为评价指标，来确定最优的组合方案。参数 T 为得到最佳组合方案所对应的位平面。为了保证可逆性，每个像素最低位在嵌入过程中保持不变，最终块标签归纳为 8 种情况，即 0、1、2、3、4、5、6、7。当块标签大于或等于 T 时，子块使用压缩方案一进行压缩；当块标签小于 T 时，子块使用压缩方案二进行压缩。设块标签小于 T 的块有 X 块，那么大于等于 T 的块个数为 $N - X$ 块，则两种方法的总嵌入容量分别由公式(9)和公式(10)计算得到。

$$EC_1 = \sum_{i=1}^{N-X} mark_i \times \left(S^2 - 1\right) - C_{\text{mark}} \tag{9}$$

式中，$mark_i$ 表示第 i 个使用压缩方案一的块的块标签，C_{mark} 表示标记块标签消耗的总长度，EC_1 为使用相同位平面压缩的块产生的总嵌入容量。

$$EC_2=\left(S^2-1\right)\left(4X_4+3X_3+2X_2+X_1\right)-\left(X_4+X_3+X_2+X_1\right)\times 3-X_0\times 1 \tag{10}$$

式中，X_0、X_1、X_2、X_3、X_4分别代表处于差值区间[64, 255]、(31, 63]、(15, 31]、(7, 15]、[0, 7]的块数量且$X_0+X_1+X_2+X_3+X_4=X$，EC_2为使用差值编码的块产生的总嵌入容量。最终，T值确定公式为：

$$T=\arg\max_T\left(EC_1+EC_2\right) \tag{11}$$

在加密图像压缩前，首先利用上述公式对图像进行分析计算，以确定每幅图像的最佳参数T。

以 Barbara 图为例，块尺寸设置为2×2，当T值为 2 时，压缩性能最佳，对应的压缩方案一的块标签编码表如表 2 所示。

Table 2. Block label codewords for image Barbara
表 2. 图像 Barbara 的块标签码字

块标签	2	3	4	5	6	7
码字	10	11	01	000	0010	0011

当子块的块标签 *mark* 大于等于 2，该块使用压缩方案一。以图 7(a)为例，该子块的块标签等于 3，大于参数 T，使用压缩方案一，载体像素的前 3 位被释放作为嵌入空间，块标签对应的霍夫曼码字 11 替换前两个载体像素的最高位，剩余部分用于嵌入额外信息。若子块的块标签小于 2，使用压缩方案二。在图 7(b)、图 7(c)中，首个载体像素的最高位为标记位s_1替换(指示当前块是否可以使用差值编码进一步压缩)。图 7(b)块标签为 0，小于 T，该块使用差值编码，每个载体像素的低 5 位存放与参考像素之间的差值，第 3 位(注：最高位为第一位)存放差值的符号位。载体像素的首位被释放，用于存放生成的辅助信息，最终该块能够在不消耗额外信息位的情况下无损恢复；若使用压缩方案一处理该块，为了能够可逆恢复当前块，将消耗额外的信息位嵌入辅助信息(用于记录该块的块标签)。图 7(c)中块标签为 1，示例使用差值编码，在嵌入辅助信息后，最终腾出 3 比特的额外空间；相比之下，若使用压缩方案一，该块在嵌入辅助信息之前已腾出 3 比特空间，但嵌入辅助信息后，腾出的额外空间少于 3 比特，因此其嵌入性能逊色于压缩方案二。图 7(d)块标签为 0，且$D_{\max}=116>63$，使用差值编码仍无法腾出额外的嵌入空间，首个载体像素的最高位被s_1替换，s_1置 0。

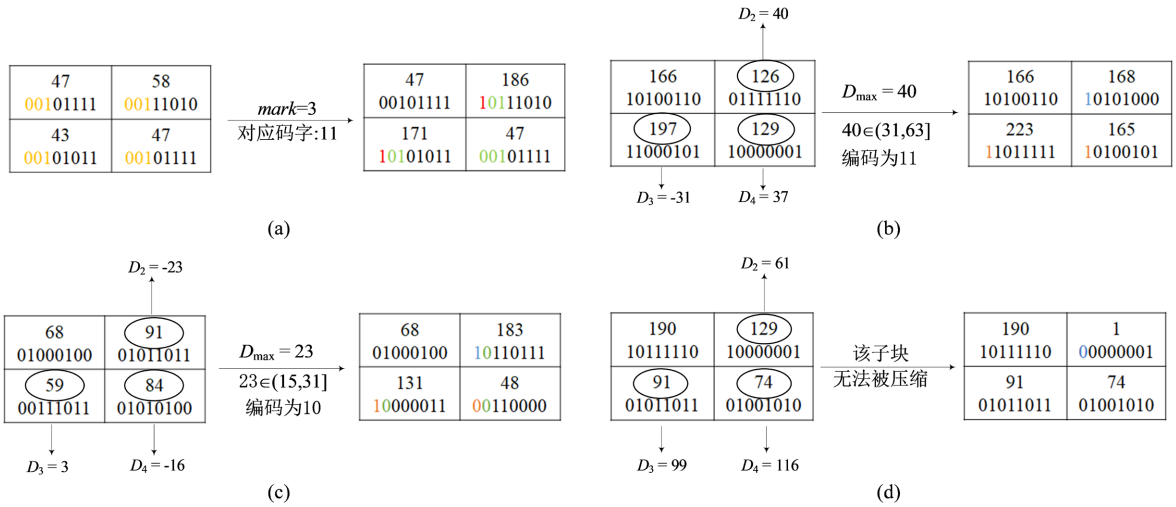


Figure 7. Example of compression process
图 7. 压缩过程示例

最终，提出方法基于组合参数 T 实现相同位平面压缩和差值编码的最佳结合，为每个数据块选择最优的压缩方案，经过压缩后得到的图像记为 I_{ec}。为实现算法的可逆性，需要大小为 $\left\lfloor \frac{H}{S} \right\rfloor \times \left\lfloor \frac{W}{S} \right\rfloor$ 的位置图指示每个数据块采用的压缩方法，其中 1 表示该数据块使用差值编码，而 0 表示采用相同位平面压缩。使用算术编码压缩位置图，以减小辅助信息开销。

2.3. 块重排

使用压缩方案一的块按照光栅扫描顺序进行收集，使用压缩方案二的块采用反向光栅扫描顺序进行收集，以实现数据块的有效排列。图 8(a)展示了 16 个数据块的初始布局，图 8(b)则呈现了相应的位置信息图。图 8(c)为所有数据块进行重新排列的结果。这一过程在给定位置图的条件下是完全可逆的。经过块重排后的图像记为 I_{ecr}。

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

(a)

1	0	1	0
0	0	1	0
0	0	0	0
1	0	0	1

(b)

2	4	5	6
8	9	10	11
12	14	15	1
3	7	13	16

(c)

Figure 8. Example of block rearrangement
图 8. 块重排示例

2.4. 数据隐藏

在图像 I_{ecr} 中按照以下步骤完成信息的嵌入。

步骤 1：数据加密。使用 K_d 加密秘密数据。

步骤 2：为确保可逆性，霍夫曼表长和霍夫曼表依次嵌入至前几个数据块中。

步骤 3：将图 9 的信息按顺序嵌入至子块腾出的嵌入空间中。首先是被霍夫曼表替代的数据块原始信息位，随后嵌入压缩后的位置图的长度及具体内容，以及实行压缩方案二时被替换的原始信息位的长度及具体内容。以上信息将用于在接收者拥有加密密钥的情况下，辅助其重构原始载体图像。最后嵌入经过 K_d 加密的秘密数据得到含秘图像 I_{em}。通过这种顺序化的嵌入过程，确保了在拥有相应访问权限的情况下，接收者能够无损恢复原始图像并准确提取秘密数据。

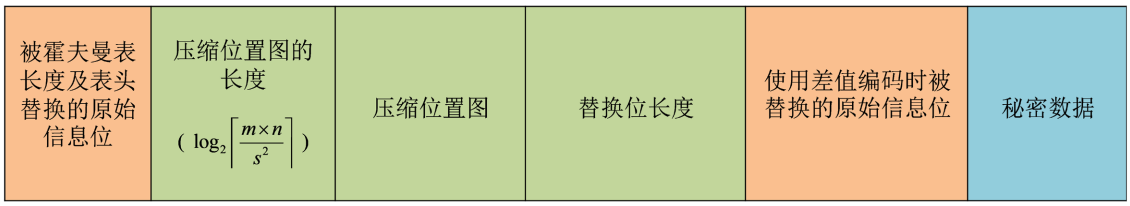


Figure 9. The embedding order of auxiliary information and secret data
图 9. 辅助信息及秘密数据的嵌入顺序

2.5. 数据提取及图像恢复

图 10 展示了数据提取与图像恢复的总体流程框图，以及四种接收场景。

场景 1：在没有任何访问权限的情况下，图像被划分为若干个 $S\times S$ 大小的不重叠数据块。首先，从前几个数据块中按顺序提取霍夫曼编码表长、霍夫曼表。根据霍夫曼表，识别每个数据块的块标签。识别子块标签，依次提取出图 9 所示的嵌入信息。根据以上信息，获得加密图像 I_e 和加密数据。由于接收者缺少解密所需的密钥最终接收者无法获得任何有效信息。

场景 2：接收者拥有 K_d 时，他/她能够准确提取秘密数据并解密数据，但无法恢复出原始载体图像。

场景 3：接收者拥有 K_e 时，他/她可以恢复出原始的载体图像但无法解密秘密数据。

场景 4：接收者拥有 K_d 和 K_e 时，他/她可以准确地提取数据并恢复出原始的载体图像。

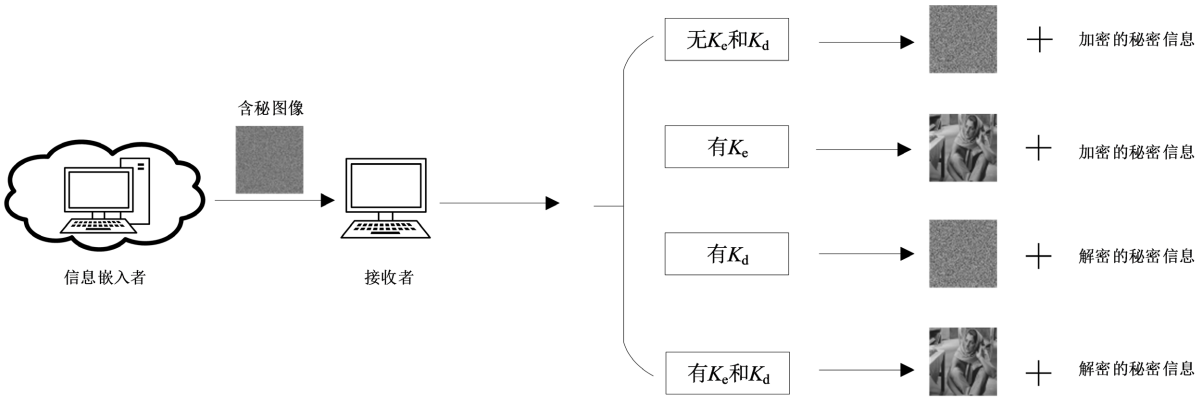


Figure 10. Flow chart of data extraction and image restoration
图 10. 数据提取与图像恢复流程图

3. 实验结果

本节通过一系列实验验证算法的性能。在 2.1 节中，首先分析了所提方法的安全性。2.2 节中分析了算法的可逆性。2.3 节中讨论了压缩方案的嵌入性能、组合参数 T 对嵌入率的影响及其在数据库中的分布情况，并展示了本算法与所提方法同为 VRAE 框架的五种主流算法的对比结果。实验主要采用了六幅测试图像(见图 11)，分别为 Airplane、Barbara、Peppers、Boat、Lake 和 Baboon。此外，为进一步验证方法的通用性，在 BOSSbase [9]和 BOWS-2 [10]两个数据集上进行了扩展实验。

Figure 11. Six test images
图 11. 六幅测试图像

3.1. 安全性分析

所提方法能够有效保障秘密信息与载体图像的安全性。在数据隐藏过程中，秘密信息首先通过加密算法(K_d)进行加密处理。即使非法访问者能够从载体图像中提取出部分加密数据或仅受到(K_e)加密的图像内容，由于其缺乏合法的访问密钥(即 K_e 和 K_d)，因此无法对提取的数据进行解密，亦无法恢复出原始图像。

1) 直方图统计分析

图 12 分别给出了原始 Barbara 图、加密 Barbara 图以及含秘加密 Barbara 图的像素值分布。可以看出原始 Barbara 图像的像素值分布具有显著特征，而加密 Barbara 图与含秘加密 Barbara 图的像素值分布均匀，且难以区分。从直方图分布无法推断出原始图像的具体内容。因此，加密算法能够有效防止攻击者仅凭直方图推测明文内容。

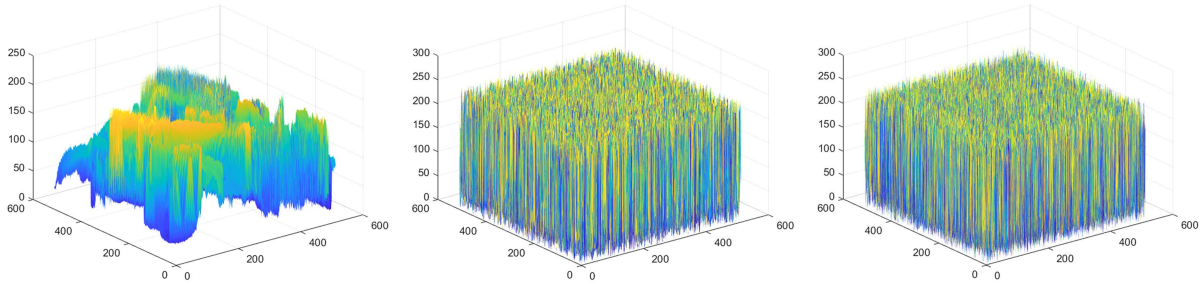


Figure 12. Distribution of pixel values in different images
图 12. 不同图像像素值分布

2) 信息熵

熵是表征信息源混沌程度的重要指标，计算公式如下：

$$H(x)=-\sum_{i=1}^{h}p(x_i)\log_2 p(x_i) \tag{12}$$

熵值越大，信息源的随机性和不确定性越强。对于 8 位灰度图像而言，当图像的熵值接近 8 时，表示图像的内容越混乱，从而使攻击者难以从中提取有效信息。表 3 展示了原始图像、加密图像以及含秘加密图像的熵值。实验结果表明，加密图像及含秘加密图像的熵值均高于原始图像，并且接近最大值 8，进一步验证了所提方法在图像安全性方面的有效性。此外，实验结果还表明，当图像被划分为 2 × 2 大小的子块时，其熵值达到最高；随着子块尺寸的增大，加密图像和含秘图像的熵值逐渐降低，但仍显著高于原始图像，并接近最大值 8。结果表明，子块尺寸对图像熵值的影响虽存在，但加密算法在不同尺寸下均能有效维持图像的高熵特性。

Table 3. Information entropy results between test images, encrypted images, and encrypted images containing secret data in different sizes
表 3. 不同尺寸下测试图像与加密图像、含秘加密图像间信息熵值结果

图像	原始图像	2×2		4×4		8×8	
		加密图像	含秘加密图像	加密图像	含秘加密图像	加密图像	含秘加密图像
Airplane	6.7025	**7.9989**	**7.9963**	7.9986	7.9920	7.9965	7.9851
Barbara	7.4464	**7.9993**	**7.9971**	7.9988	7.9942	7.9976	7.9929

续表

Peppers	7.5937	**7.9990**	**7.9955**	7.9987	7.9934	7.9975	7.9891
Boat	7.1914	**7.9993**	**7.9929**	7.9989	7.9956	7.9980	7.9899
Lake	7.4842	**7.9992**	**7.9941**	7.9989	7.9883	7.9974	7.9752
Baboon	7.3583	**7.9991**	**7.9955**	7.9990	7.9895	7.9989	7.9918

3) 相关性分析

表 4 与表 5 分别展示了原始图像与在不同子块尺寸下处理得到的加密图像、含秘加密图像的 *PSNR* (峰值信噪比)与 *SSIM* (结构相似性)数据。*PSNR* 值越小，表明两幅图像之间的差异越大，相似性越低。在六幅测试图像的 *PSNR* 实验中，不同尺寸下测试图像与加密图像、含秘加密图像间信息熵均小于 10 dB，表明原始图像与加密图像或含秘加密图像间的差异性极为显著，几乎无法从加密或含秘加密图像中辨识出原始图像的内容。此外，表 5 中的 *SSIM* 数据进一步支持了这一结论：不同尺寸下测试图像与加密图像、含秘加密图像之间的结构相似性均趋近于 0，表明它们在结构上几乎不存在相似性，充分证明了所提出的加密算法在保障图像内容、秘密数据的安全性方面的有效性。

Table 4. PSNR between test images, encrypted images, and encrypted images containing secret data in different sizes
表 4. 不同尺寸下测试图像与加密图像、含秘加密图像间 PSNR 值结果

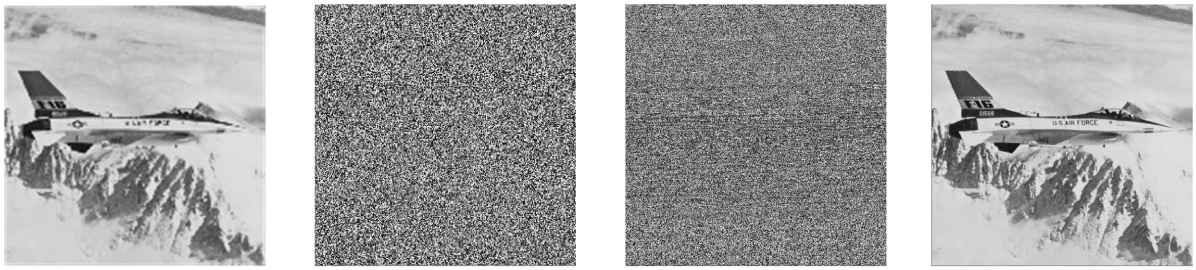
测试图像	尺寸	加密图像	含秘加密图像	测试图像	尺寸	加密图像	含秘加密图像
Airplane	2×2	7.9946	7.9883	Boat	2×2	9.3010	9.2996
	4×4	8.0267	7.9188		4×4	9.3116	9.2841
	8×8	8.0313	7.9379		8×8	9.3124	9.2677
Barbara	2×2	9.1531	9.1016	Lake	2×2	8.2247	8.1917
	4×4	9.1543	9.1236		4×4	8.2149	8.2043
	8×8	9.1651	9.1476		8×8	8.2199	8.1965
Peppers	2×2	8.8599	8.8907	Baboon	2×2	9.5415	9.1686
	4×4	8.8611	8.8911		4×4	9.5581	9.1420
	8×8	8.8632	8.9011		8×8	9.5742	9.1563

Table 5. SSIM between test images, encrypted images, and encrypted images containing secret data in different sizes
表 5. 不同尺寸下测试图像与加密图像、含秘加密图像间 SSIM 值结果

测试图像	尺寸	加密图像	含秘加密图像	测试图像	尺寸	加密图像	含秘加密图像
Airplane	2×2	0.0054	−0.0019	Boat	2×2	0.0064	0.0182
	4×4	0.0147	0.0101		4×4	0.0038	0.0094
	8×8	0.0005	0.0025		8×8	0.0160	0.0179
Barbara	2×2	0.0044	0.0026	Lake	2×2	0.0053	0.0049
	4×4	0.0116	0.0094		4×4	0.0107	0.0156
	8×8	−0.0094	0.0101		8×8	−0.0192	−0.0235
Peppers	2×2	0.0024	0.0049	Baboon	2×2	0.0077	0.0073
	4×4	0.0101	0.0126		4×4	0.0105	0.0184
	8×8	−0.0009	0.0014		8×8	0.0249	0.3151

3.2. 可逆性分析

以 Airplane 图像为例，图 13 展示了所提方法生成的视觉效果。加密图像和含秘加密图像的视觉效果均表现为类似噪声的图像，原始图像的主观特征在其中无法被辨识。通过视觉评估，可以确认恢复后的图像与原始图像完全一致，这一结果也得到了 *PSNR* 指标的证实。



(a) (b) (c) (d)

Figure 13. Four image forms of Airplane
图 13. Airplane 的 4 种图像形态

表 6 中展示恢复图像和原始图像的 *PNSR* 和 *SSIM* 值。六幅测试图像与得到的恢复图像之间的 *PSNR* 均趋于+∞，*SSIM* 均等于 1，恢复图像与原始图像完全相等，证明所提算法具有可逆性。

Table 6. PSNR and SSIM between test images and restored images in different sizes of blocks
表 6. 不同子块尺寸下测试图像与恢复图像间的 PSNR 与 SSIM

图像	2×2		4×4		8×8	
	PNSR/dB	*SSIM*	*PNSR*/dB	*SSIM*	*PNSR*/dB	*SSIM*
Airplane	+∞	1	+∞	1	+∞	1
Barbara	+∞	1	+∞	1	+∞	1
Peppers	+∞	1	+∞	1	+∞	1
Boat	+∞	1	+∞	1	+∞	1
Lake	+∞	1	+∞	1	+∞	1
Baboon	+∞	1	+∞	1	+∞	1

3.3. 嵌入性能分析

提出方法使用参数 T 作为两种压缩方案的组合指标。由于图像内容的差异，不同图像对应的组合参数 T 略有不同。图 14(a)为六幅测试图像且子块尺寸设置为 2×2 中参数 T 与嵌入率之间的曲线关系。在图像“Airplane”中，当 $T=3$ 时，嵌入率达到了最大值 2.0936 bpp；在图像“Lake”中，$T=1$ 时，嵌入性能最佳，达到最大嵌入率 1.2982 bpp。在图像“Barbara”、“Peppers”、“Lake”和“Baboon”中，$T=2$ 和 $T=3$ 时的嵌入率相近，其中 $T=2$ 时的嵌入率性能略优于 $T=3$。

图 14(b)展示了在 BOSSbase [9]和 BOWS-2 [10]数据库中随机选择 400 张图像时，T 值的数量分布情况。在 BOSSbase 数据库中，组合参数 T 为 2 的图像数量最多，其次是组合参数 T 为 3。而在 BOWS-2 数据库中，$T=2$ 的图像数量最多，紧随其后的是组合参数 T 为 1。当 T 值为 6 时，两数据库中图像的数量均为零。

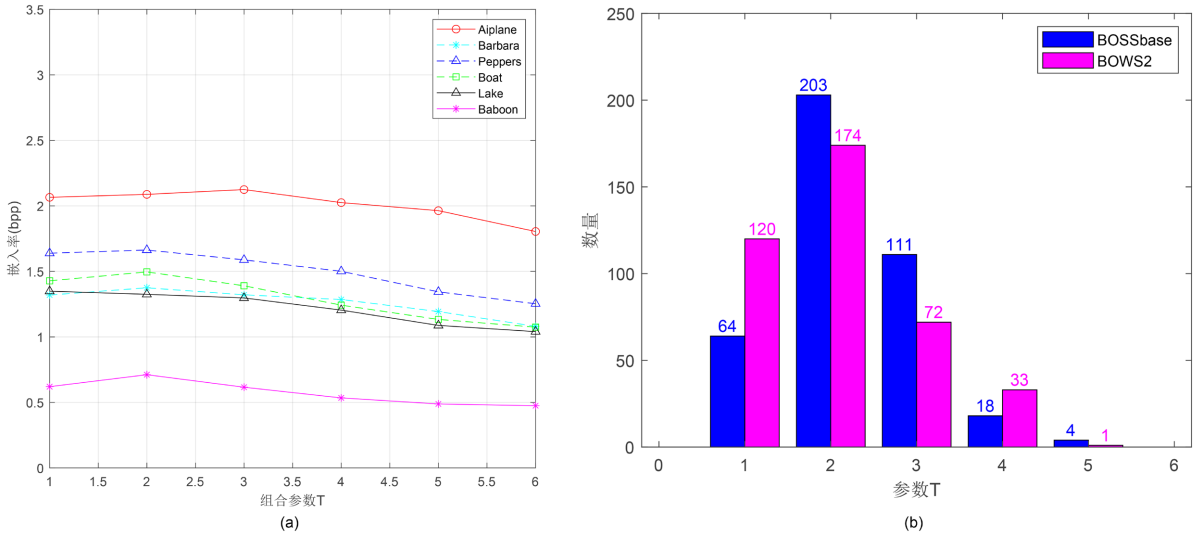


Figure 14. The embedding rate of test images under different parameters T and the distribution of the number of combined parameters T in the database
图 14. 不同参数 T 下测试图像的嵌入率及数据库中组合参数 T 的数量分布情况

图 15 展示了在六幅图像中，当 $mark < T$ 时，采用压缩方案二获得更高嵌入容量的子块数量在对应块标签下的分布情况。使用压缩方案二对应子块数量远超于使用压缩方案一的子块数量。所以在 $mark < T$ 时，使用压缩方案二会释放更多的嵌入空间，证明了该压缩方案的有效性。

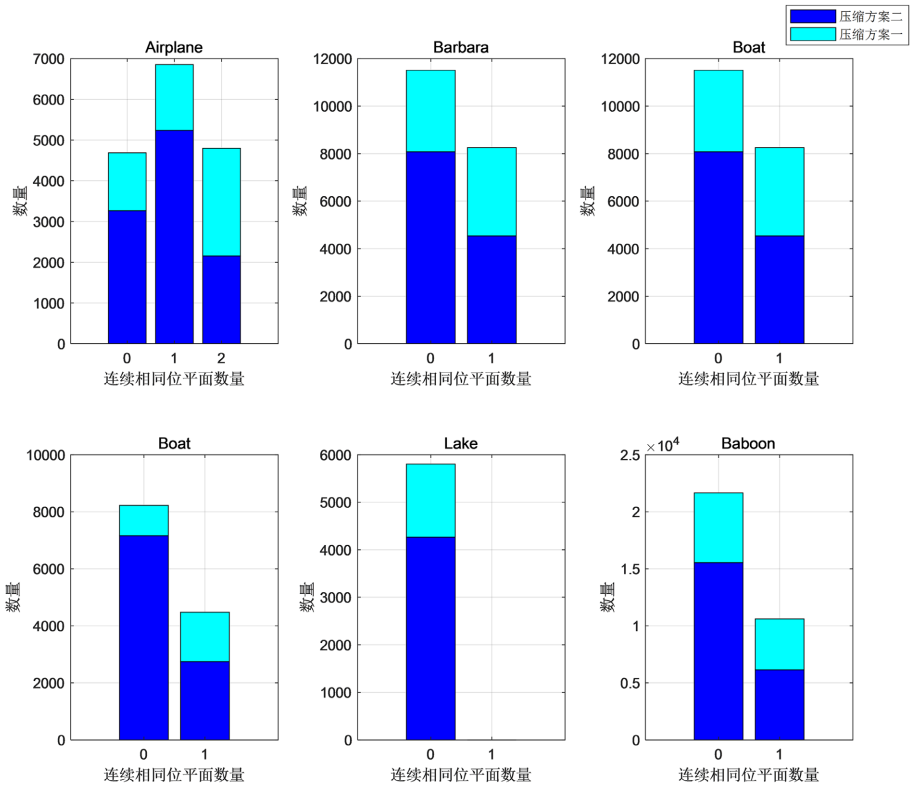


Figure 15. Distribution of block numbers after further compression using compression scheme 2
图 15. 采用压缩方案二进一步压缩的块数量分布

所提算法以图像子块作为图像压缩处理的基本单元，探讨了子块尺寸对最终嵌入率的影响。表 7 列出了六幅测试图像在不同子块尺寸下获得的最终嵌入率。在图像 Airplane 和 Peppers 中，子块尺寸设置为 4×4 时，嵌入率最高，分别为 2.1532、1.6289。而对于其余五幅测试图像，当子块尺寸为 2×2 时，嵌入率表现最佳。

Table 7. Maximum embedding rate of images in different sizes of blocks (Unit: bpp)
表 7. 不同子块尺寸下图像的最大嵌入率(单位：bpp)

测试图像	2×2	4×4	8×8
Airplane	2.0936	**2.1532**	1.6281
Barbara	**1.3153**	1.3010	0.8712
Peppers	1.6236	**1.6289**	1.0839
Boat	**1.4239**	1.4108	0.9134
Lake	**1.2982**	1.2233	0.7282
Baboon	**0.6075**	0.4267	0.1570

表 8 展示所提方法与五种基于 VRAE 框架的算法满嵌情况下的嵌入率的比较。所有算法的配置参数均设置为其最佳性能参数。在图像 Airplane、Barbara 和 Peppers 中，所提方法的嵌入性能最佳。在图像 Airplane、Barabara、Peppers、Boat 图像，比五种方法中嵌入率最佳的算法[8]分别高出 0.11 bit/像素、0.11 bit/像素、0.07 bit/像素、0.02 bit/像素，对于 512 × 512 大小的图像，嵌入率高于 0.1 bit/像素即本方法比其他算法至少多释放 26214 bits 空间。对于图像 Boat，本算法的嵌入率仅比算法[8]的最佳结果高出 0.01 bit/像素，约多释放 2621 bits 空间。在图像 Lake 和 Baboon 中，算法[8]表现出最佳的嵌入性能，其次是本方法。在图像 Lake 中，算法[8]的嵌入率比本方法高约 0.03 bit/像素，即多释放约 7823 bits；而在图像 Baboon 中，算法[8]的嵌入率比本方法高约 0.2 bit/像素，约多腾出 52428 bits。

Table 8. Comparison of embedding rates of different algorithms (Unit: bpp)
表 8. 不同算法的嵌入率比较(单位：bpp)

测试图像	Fu 等[11]	Chen [12]	Liu 等[5]	Wang 等[6]	Gao 等[8]	所提方法
Airplane	1.28	1.68	1.58	2.02	2.04	**2.1532**
Barbara	0.81	0.85	0.88	1.16	1.20	**1.3153**
Peppers	1.00	1.35	1.27	1.51	1.56	**1.6289**
Boat	0.92	1.06	1.20	1.28	1.41	**1.4239**
Lake	0.90	1.05	1.07	1.23	**1.32**	1.2982
Baboon	0.37	0.29	0.46	0.30	**0.81**	0.6075

由于算法[8]在图像 Lake 和 Baboon 中的嵌入性能优于所提方法，因此需要通过进一步实验来证明本算法的普遍优越性。为了实现这一目标，在 BOSSbase [9]和 BOWS-2 [10]数据库中随机选取了 400 张图像。表 9 展示了算法[8]与本方法在这两个数据库中的嵌入性能对比。所提方法平均嵌入率和最大嵌入率均优于算法[8]，其中平均嵌入率分别高于 Gao 约 0.0712 bit/像素、0.0654 bit/像素。

图 16 绘制了 BOSSbase [9]和 BOWS-2 [10]数据集上对 400 张测试图像的嵌入率分布提出算法在嵌入率上始终优于对比算法[8]。表明本文方法在大多数测试样本中展现出更优异的嵌入性能。

Table 9. Comparison of embedding rates between BOSSbase [9] and BOWS [10] datasets (Unit: bpp)
表 9. 基于数据集 BOSSbase [9]和 BOWS-2 [10]的嵌入率比较(单位：bpp)

方法	数据集	平均	最小	最大
所提方法	BOSSbase [9]	2.3268	0.8064	4.1552
	BOWS-2 [10]	2.3197	0.4073	4.3537
算法[8]	BOSSbase [9]	2.2516	0.8404	3.8313
	BOWS-2 [10]	2.2543	0.8268	3.4942

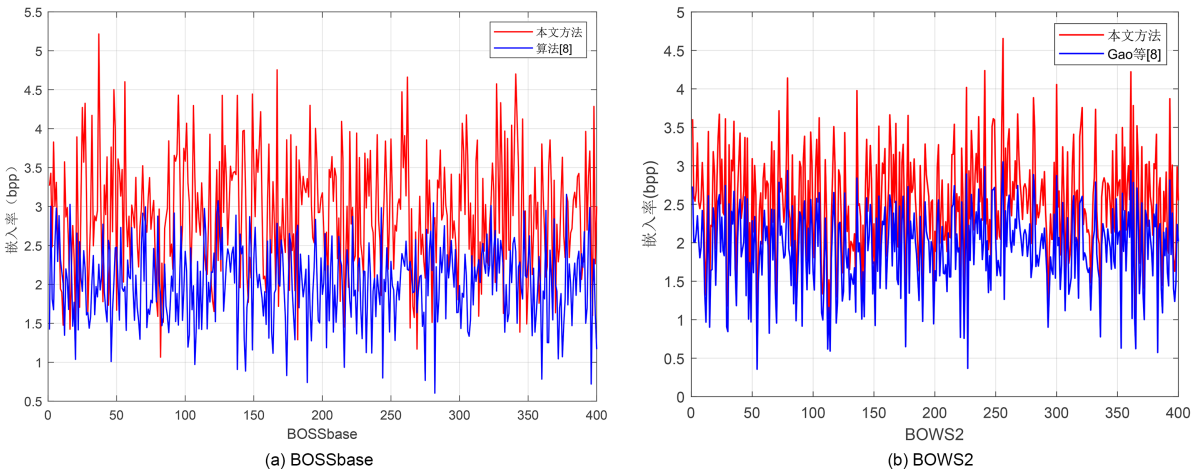


Figure 16. Embedding rate line chart based on the BOSSbase [9] and BOWS2 [10]
图 16. 基于数据集 BOSSbase [9]、BOWS2 [10]的嵌入率折线图

4. 结论

本文提出了一种新的基于 VRAE 的 RDHEI 方法。它采用分块加密方式，尽可能地将原始像素之间的相关性转移至加密图像。为了充分利用加密后子块内各像素间的相关性，本文采用基于组合参数 T 的综合压缩策略压缩加密图像，显著提升了嵌入容量。并且，本算法的信息提取和图像恢复可实现分离进行，即在只拥有图像加密密钥时，恢复出原始图像；在只拥有嵌入密钥时，无误提取秘密信息；在同时拥有上述两个密钥时，无损地恢复原始图像并且无误地提取出秘密信息。

参考文献

[1] Puteaux, P. and Puech, W. (2018) An Efficient MSB Prediction-Based Method for High-Capacity Reversible Data Hiding in Encrypted Images. *IEEE Transactions on Information Forensics and Security*, **13**, 1670-1681. https://doi.org/10.1109/tifs.2018.2799381

[2] Yi, S. and Zhou, Y. (2019) Separable and Reversible Data Hiding in Encrypted Images Using Parametric Binary Tree Labeling. *IEEE Transactions on Multimedia*, **21**, 51-64. https://doi.org/10.1109/tmm.2018.2844679

[3] Yin, Z., Xiang, Y. and Zhang, X. (2020) Reversible Data Hiding in Encrypted Images Based on Multi-MSB Prediction and Huffman Coding. *IEEE Transactions on Multimedia*, **22**, 874-884. https://doi.org/10.1109/tmm.2019.2936314

[4] Puteaux, P. and Puech, W. (2021) A Recursive Reversible Data Hiding in Encrypted Images Method with a Very High Payload. *IEEE Transactions on Multimedia*, **23**, 636-650. https://doi.org/10.1109/tmm.2020.2985537

[5] Liu, Z. and Pun, C. (2022) Reversible Data Hiding in Encrypted Images Using Chunk Encryption and Redundancy Matrix Representation. *IEEE Trans on Dependable and Secure Computing*, **19**, 1392-1394.

[6] Wang, Y. and He, W. (2022) High Capacity Reversible Data Hiding in Encrypted Image Based on Adaptive MSB Prediction. *IEEE Transactions on Multimedia*, **24**, 1288-1298. https://doi.org/10.1109/tmm.2021.3062699

[7] Qiu, Y., Ying, Q., Yang, Y., Zeng, H., Li, S. and Qian, Z. (2022) High-Capacity Framework for Reversible Data Hiding in Encrypted Image Using Pixel Prediction and Entropy Encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, **32**, 5874-5887. https://doi.org/10.1109/tcsvt.2022.3163905

[8] Gao, K., Horng, J. and Chang, C. (2022) High-capacity Reversible Data Hiding in Encrypted Images Based on Adaptive Block Encoding. *Journal of Visual Communication and Image Representation*, **84**, Article 103481. https://doi.org/10.1016/j.jvcir.2022.103481

[9] Bas, P., Filler, T. and Pevný, T. (2011) Break Our Steganographic System: The Ins and Outs of Organizing Boss. In: *Lecture Notes in Computer Science*, Springer, 59-70. https://doi.org/10.1007/978-3-642-24178-9_5

[10] Bas, P. and Furon, T. (2018) Image Database of Bows-2.

[11] Fu, Y., Kong, P., Yao, H., Tang, Z. and Qin, C. (2019) Effective Reversible Data Hiding in Encrypted Image with Adaptive Encoding Strategy. *Information Sciences*, **494**, 21-36. https://doi.org/10.1016/j.ins.2019.04.043

[12] Chen, K. (2020) High Capacity Reversible Data Hiding Based on the Compression of Pixel Differences. *Mathematics*, **8**, Article 1435. https://doi.org/10.3390/math8091435