IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 1, 2020, accepted January 21, 2020, date of publication January 24, 2020, date of current version February 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2969252

Reversible Data Hiding in Encrypted Images With Dual Data Embedding

YINGQIANG QIU[1], QICHAO YING[2], XIAODAN LIN[1], YAOWEN ZHANG[1], AND ZHENXING QIAN[3], (Member, IEEE)

[1]College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China
[2]School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China
[3]School of Computer Science, Shanghai Institute of Intelligent Electronics and Systems, Fudan University, Shanghai 200433, China

Corresponding author: Zhenxing Qian (zxqian@fudan.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61572308, Grant U1736213, Grant U1636206, and Grant 61525203, and in part by the Scientific Research Foundation of National Huaqiao University under Grant 17BS509.

ABSTRACT Typical reversible data hiding in encrypted image (RDH-EI) methods merely embed data in the encrypted domain, ignoring the requirement of the image owner for data embedding. To address this issue, this paper proposes a novel RDH-EI method with dual data embedding based on generalized integer transformation (GIT). The image owner first vacates embedding room, and performs data embedding before image encryption. After data encryption, the remote server utilizes the vacated room to further embed additional data into the encrypted image. The embedded data by the image owner and remote server can be extracted exactly, and the original image can be recovered losslessly. The experimental results prove the effectiveness of the proposed method.

INDEX TERMS Reversible data hiding, encrypted image, generalized integer transformation, dual data embedding.

I. INTRODUCTION

Reversible data hiding (RDH) has the capability of recovering the cover medium exactly after the embedded data is extracted. For this reason, RDH is widely applied in some sensitive scenarios, i.e., military or medical images, where distortions are forbidden and the accurate recovery of the original cover medium is required. Until now, most RDH techniques have been investigated for uncompressed images. In general, there are four basic fundamental strategies for RDH in image spatial domain, namely, lossless compression [1], [2], difference expansion (DE) [3]–[6], histogram shifting (HS) [7]–[10], and optimal coding [11]–[14]. Besides, many RDH techniques have been proposed for compressed images, particularly for JPEG images [15]–[17].

With the growing development of cloud computing and cloud storage, protection for data security and privacy during transmission has attracted extensive attentions. To securely share a secret image with the recipient, encryption technique is widely referred to for its effectiveness in privacy protection. In some scenarios, the remote server wishes to embed some additional data, such as the copyright data, timestamp, into the encrypted image. However, experiments have shown that though the traditional RDH methods achieve good rate-distortion performance, they are generally not efficient for encrypted images. Therefore, RDH in encrypted images (RDH-EI) has attracted considerable research interest in the past decade. Typically, the image owner first encrypts the cover image. Then the remote server embeds additional data into the encrypted image reversibly while ensuring that the authorized receiver can extract the embedded data exactly and/or recover the original image losslessly.

Generally, RDH-EI methods can be classified into two classes [18]: vacating room after encryption (VRAE) and vacating room before encryption (VRBE). Figure 1 provides the general framework for VRAE and VRBE. In VRAE methods, the remote server generates embedding room by modified some pixel values of the encrypted image directly. VRAE methods can be further grouped into three categories, i.e., data extraction in plaintext domain [19]–[23], data extraction in the cipher domain [24]–[30], and data extraction in both domains [31]–[34]. These VRAE methods achieve decent performances of RDH in encrypted image. But since the entropy of the encrypted image is usually maximized because of encryption, the net payloads are relatively low. In contrast, VRBE methods can achieve higher payloads by

The associate editor coordinating the review of this manuscript and approving it for publication was Mamoun Alazab.

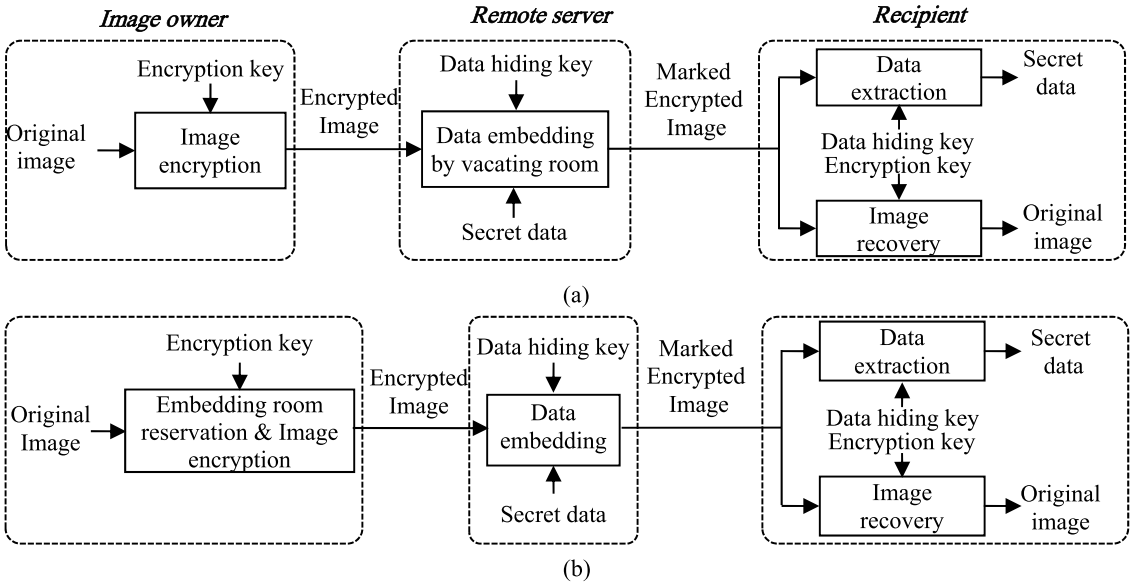


FIGURE 1. Framework of (a) VRAE and (b) VRBE method.

utilizing the spatial correlation of the original image to vacate embedding room before image encryption. Ma *et al.* [35] proposes the first VRBE algorithm by generating embedding room with traditional HS-based RDH method. In [36], Zhang *et al.* vacates embedding room before image encryption by embedding data into the estimated error of pixels. In the separable and error-free RDH-EI method proposed by Xu and Wang [37], interpolation-errors of non-sample pixels are generated by interpolation techniques, and then the interpolation-errors are encrypted with a specific encryption mode while the sample pixels are encrypted by a stream cipher. After encryption, additional bits can be embedded into interpolation-errors by histogram shifting and difference expansion technique. To better exploit the correlation between neighboring pixels, Cao *et al.* [38] achieves a large vacated room by compressing image patches with sparse representation technique. In some RDH-EI schemes for homomorphic encrypted images [39], [40], preprocessing is performed on the plaintext-image to vacate room before image encryption.

In order to improve the payload of RDH-EI, several methods utilize the correlations of higher bit-planes of the original image, esp. the most significant bit (MSB), to vacate room before image encryption. In [41], with binary-block embedding, the binary bits in lower bit-planes of the original image are embedded into its higher bit-planes to vacate room for data embedding after image encryption. Puteaux et al. proposes two RDH-EI approaches by using MSB prediction [42], i.e., high-capacity reversible data hiding approach with correction of prediction errors (CPE-HCRDH) and high-capacity reversible data hiding approach with embedded prediction errors (EPE-HCRDH), in which data embedding is realized by substituting the MSB values in the encrypted image. These works are further improved by using other higher bit-planes prediction rather than the MSB-plane only in [43] or by using two-MSBs prediction in [44]. In [45], Yi and Zhou proposes a separable RDH-EI method by using parametric binary tree labeling, which is improved in [46]. Chen and Chang [47] compresses the MSB planes of images to vacate room before image encryption, which can achieve high-capacity data embedding in image encryption domain. In [48], Yin *et al.* proposes a high capacity RDH-EI algorithm based on multi-MSB prediction and Huffman coding. Since these methods generate embedding room by modifying higher bit-planes or MSBs bit-planes, the quality of the decrypted marked image is not good.

Though promising in embedding performance, the above-mentioned RDH-EI methods usually ignore the requirements of reversible data embedding for the image owner. In [49], Qiu *et al.* proposes a new RDH-EI method to allow dual data embedding, which includes data embedding for the image owner. However, the vacated room for data embedding in encrypted image is relatively small and fixed, and the technique of generalized reversible contrast mapping introduces more distortion when vacating room for encrypted image and embedding data for image owner. This paper proposes a novel RDH-EI approach to further address this issue. With the efficient technique of generalized integer transformation (GIT), the image owner achieves data embedding and embedding room reservation adaptively while introducing less distortion. After image encryption, the remote server can easily embed additional data by LSB-planes replacement. The embedded data by the image owner and the remote server can be exactly extracted and the original image can be perfectly recovered after image decryption.

The remainder of this paper is organized as follows. The generalized integer transformation algorithm for RDH is described in Section II. The details of the proposed RDH-EI

method are proposed in Section III. Experimental results and analyses are provided in Section IV. Finally, this paper is concluded in Section V.

II. GENERALIZED TRANSFORMATION FOR RDH

An integer transformation for RDH was originally proposed by Wang *et al.* [5], which was extended to a further generalized form in [6]. Without loss of generality, for $x \in \mathbb{N}$, functions $f(x, 2^m)$, $h(x, 2^m)$ and $g(x, 2^m)$ can be defined as follows:

$$\begin{cases} f\left(x, 2^m\right) = \left\lceil (2^m - 1) \cdot x / 2^m \right\rceil \\ h\left(x, 2^m\right) = x - f\left(x, 2^m\right) = \left\lfloor \dfrac{x}{2^m} \right\rfloor \\ g\left(x, 2^m\right) = x - 2^m \cdot h\left(x, 2^m\right) \\ \quad = f(x, 2^m) - (2^m - 1) \cdot h(x, 2^m), \end{cases} \tag{1}$$

where $m \in \mathbb{N}$, $\lfloor \cdot \rfloor / \lceil \cdot \rceil$ is the floor/ceiling function respectively, $f(x, 2^m)$ and $h(x, 2^m)$ form a complementary function pair, and $g(x, 2^m)$ stands for the m bits LSBs (least significant bits) of x. For an integer array $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{N}^n$, the rounded average value is defined as

$$a\left(\mathbf{x}\right) = \begin{cases} \lfloor \bar{\mathbf{x}} \rfloor, & \textit{if } \bar{\mathbf{x}} - \lfloor \bar{\mathbf{x}} \rfloor < 0.5 \\ \lceil \bar{\mathbf{x}} \rceil, & \textit{otherwise}, \end{cases} \tag{2}$$

where $\bar{\mathbf{x}} = \sum_{\mathbf{i=1}}^{\mathbf{n}} x_i / n$. The generalized integer transformation (GIT) algorithm for RDH can be defined as

$$\begin{cases} y_1 = 2^m \cdot x_1 - 2^m \cdot f\left(a\left(\mathbf{x}\right), 2^m\right) + s_1 \\ y_2 = 2^m \cdot x_2 - 2^m \cdot f\left(a\left(\mathbf{x}\right), 2^m\right) + s_2 \\ \cdots \\ y_{n-1} = 2^m \cdot x_{n-1} - 2^m \cdot f\left(a\left(\mathbf{x}\right), 2^m\right) + s_{n-1} \\ y_n = 2^m \cdot x_n - \left(2^m - 1\right) \cdot a\left(\mathbf{x}\right), \end{cases} \tag{3}$$

where $\mathbf{s} = (s_1, s_2, \ldots, s_{n-1}) \in \mathbb{N}_{2^m}^{n-1}$ stands for the embedded data array, each $s_i \in \{0, 1, \ldots, 2^m - 1\}$ $(i = 1, 2, \ldots, n-1)$. This way, $(n-1) \cdot m$ bits $\mathbf{s}$ can be embedded into integer array $\mathbf{x}$ to obtain $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \mathbb{N}^n$.

Since

$$s_i = g(y_i, 2^m), \quad i = 1, 2, \ldots, n-1, \tag{4}$$

and

$$x_i = h(y_i, 2^m) + (2^m - 1) \cdot a(h(\mathbf{y}, 2^m)) + g(y_n, 2^m), \quad i = 1, 2, \ldots, n, \tag{5}$$

$\mathbf{s}$ can be extracted exactly and $\mathbf{x}$ can be recovered losslessly, once we have $\mathbf{y}$.

III. PROPOSED GIT-BASED RDH-EI METHOD

The proposed RDH-EI framework is depicted in Figure 2. The image owner first embeds his secret data into the cover image $\mathbf{I}$ and reserves room for data embedding after image encryption, then encrypts the pre-processed image to obtain the encrypted marked image $\mathbf{I}^{EM}$, which is uploaded to a remote server. The remote server embeds secret data into $\mathbf{I}^{EM}$ to get the dual marked & encrypted image $\mathbf{I}^{MEM}$. At the recipient side, the authorized receiver can exactly extract secret data embedded by the remote server. After decryption, the secrete data embedded by image owner can be extracted exactly and the cover image can be recovered losslessly.

A. GENETATION OF THE ENCRYPTED MARKED IMAGE

The image owner embeds some secret data into original image while reserving embedding room for the encrypted version. Then he encrypts the marked image before image uploading. As is shown in Figure 3, the processing procedure can be divided into three main stages: image partition, reversible data embedding, and image encryption. At the beginning, the image owner divides the cover image into non-overlapped blocks, and classifies these blocks into two parts, namely, *part* ***A*** and *part* ***B***. Then, the owner's secret data and some auxiliary data (including multi-level location map, i.e. *MLM*, and LSB-planes of *part* ***A***) for RDH are embedded into the blocks of *part* ***B***. At last, *part A* and *marked part* ***B*** are rearranged and encrypted to generate $\mathbf{I}^{EM}$, which is then to be uploaded to the remote server. The embedding room for the remote server is reserved in encrypted *part* ***A***.

1) IMAGE PARTITION

Without loss of generality, assume the cover image $\mathbf{I}$ is a gray-scale image sized by $N_1 \times N_2$ with 256 different gray levels, each pixel in $\mathbf{I}$ falls into [0, 255].

First, the image owner divides $\mathbf{I}$ into non-overlapping blocks with $n = n_1 \times n_2$ pixels, and form these blocks to n-dimensional pixel arrays $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$, $\mathbf{x}^{(j)} = (x_1^{(j)}, x_2^{(j)}, \cdots, x_n^{(j)})$, $j = 1, 2, \ldots, N$, and $N = \lfloor N_1/n_1 \rfloor \times \lfloor N_2/n_2 \rfloor$. After embedding $m \cdot (n-1)$ bits data into $\mathbf{x}^{(j)}$ to obtain $\mathbf{y}^{(j)}$, the distortion introduced by integer transformation

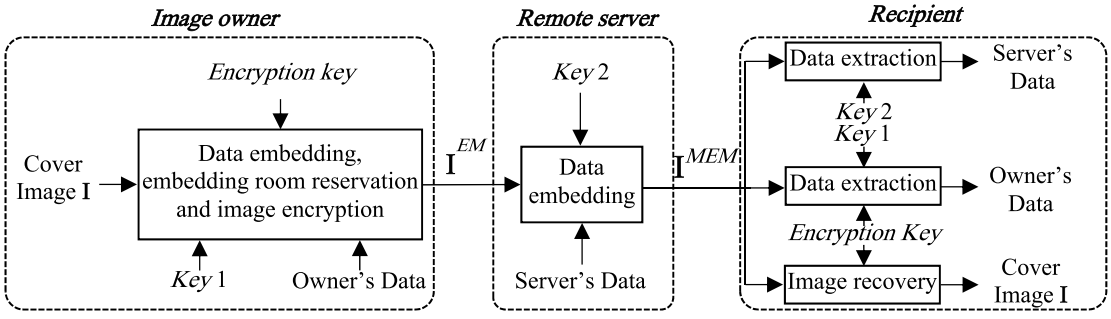


FIGURE 2. Framework of the proposed dual embedding VRBE method.

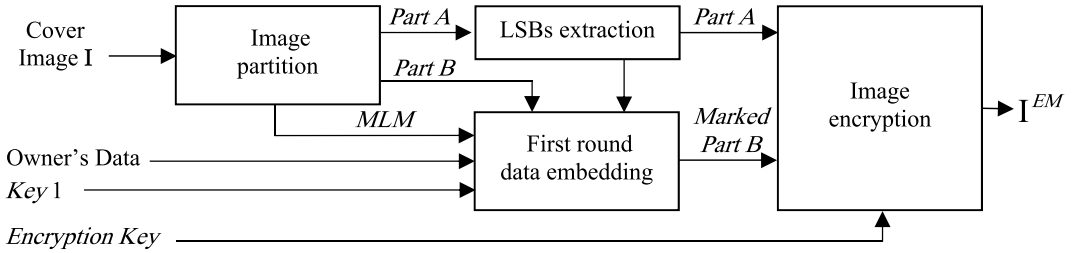


FIGURE 3. Sketch of the processing procedure at the image owner side.

would approximately be

$$\mathrm{MSE}(\mathbf{y}^{(j)}, \mathbf{x}^{(j)}) \approx \left(2^{m^{(j)}} - 1\right)^2 \cdot \mathrm{SSE}(\mathbf{x}^{(j)})/n, \tag{6}$$

where

$$\mathrm{SSE}\left(\mathbf{x}^{(j)}\right) = \sum\nolimits_{i=1}^{n} \left(x_i^{(j)} - a\left(\mathbf{x}^{(j)}\right)\right)^2. \tag{7}$$

Here, $\mathrm{SSE}\left(\mathbf{x}^{(j)}\right)$ is the sum of the squared errors of $\mathbf{x}^{(j)}$. With fixed n for special pixel arrays, larger $m^{(j)}$ would enable larger data embedding capacity, yet the introduced distortion might be larger accordingly.

To prevent the overflow/underflow during transforming $\mathbf{x}^{(j)}$ to $\mathbf{y}^{(j)}$, each marked transformed pixel value in $\mathbf{y}^{(j)}$ should be restrained within [0, 255]. Let

$$C = \{\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{N}_{256}^n : 0 \le x_i \le 255,\ i = 1, 2, \ldots, n\} \tag{8}$$

be the set of all possible pixel value arrays. Define subsets C_m satisfying

$$\begin{aligned} C_m = \{\mathbf{x} \in C : 0 \le x_i - f(a(\mathbf{x}), 2^m) \le 2^{(8-m)} - 1 \\ (i = 1, 2, \ldots, n-1), \\ 0 \le 2^m \cdot x_n - (2^m - 1) \cdot a(\mathbf{x}) \le 255\}. \end{aligned} \tag{9}$$

If $\mathbf{x}^{(j)}$ belongs to C_m, there would be no underflow/overflow during embedding $m \cdot (n-1)$ bits into $\mathbf{x}^{(j)}$. To alleviate the distortion of GIT, we keep $\mathrm{MSE}(\mathbf{y}^{(j)}, \mathbf{x}^{(j)}) \le T/n$, T is a predefined integer threshold and $T > 0$. Those blocks selected for data embedding should satisfy $\mathrm{SSE}(\mathbf{x}^{(j)}) \le T$ and $\mathbf{x}^{(j)} \in C_1$. According to the principle, if $\mathrm{SSE}(\mathbf{x}^{(j)}) \le T$ and $\mathbf{x}^{(j)} \in C_1$, $\mathbf{x}^{(j)}$ is classified into *first-round embedded category FE* ($\mathbf{x}^{(j)} \to FE$) as *part* ***B***; otherwise, $\mathbf{x}^{(j)}$ is classified into *second-round embedded category SE* ($\mathbf{x}^{(j)} \to SE$) as *part* ***A***. Figure 4 illustrates the diagram of classification. Multi-level location map *MLM* is used to identify blocks classification, where the first-level location map *FLM* is used to distinguish whether $\mathbf{x}^{(j)} \in FE$ (corresponding $m^{(j)} \ge 1$ adaptively) or $\mathbf{x}^{(j)} \in SE$ (corresponding $m^{(j)} = 0$). In *FLM*, "1" is assigned for $\mathbf{x}^{(j)} \in FE$ and "0" for $\mathbf{x}^{(j)} \in SE$, other levels of *MLM* will be introduced in the next section. In Figure 5, after classification, all blocks belonged to *part* ***A*** are rearranged successively to the front of the image, followed in turn by the blocks of *part* ***B***. With *FLM*, the original image can be recovered from the rearranged image exactly. The one or more LSB-planes of *part* ***A*** are reserved for data embedding by the remote server after image encryption and uploading. Thus, the image owner should embed the auxiliary data with his secret data into *part* ***B***.

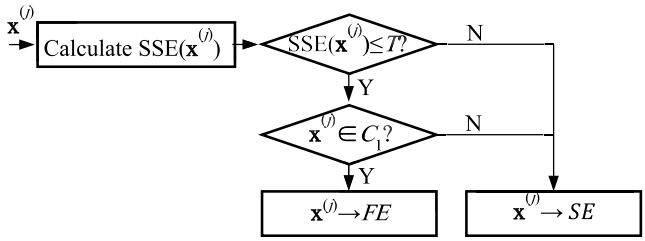


FIGURE 4. The classification of block $\mathbf{x}^{(j)}$.

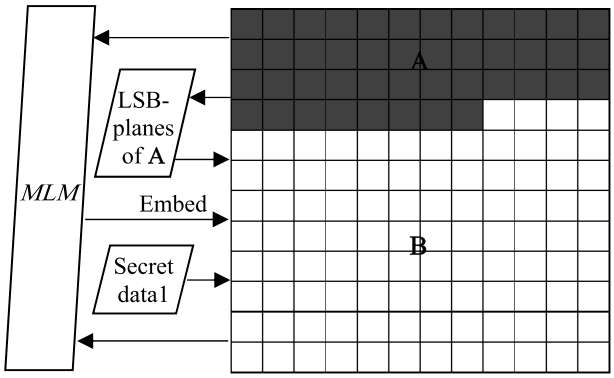


FIGURE 5. Illustration of blocks rearrangement.

2) DATA EMBEDDING BY ADAPTIVE GIT

In *part* ***B***, adaptive GIT (AGIT) algorithm is used to accommodate the auxiliary data and the owner's secret data. To embed more bits and alleviate distortion, the image owner should adaptively choose $m^{(j)}$ for each $\mathbf{x}^{(j)}$ to keep $\mathrm{MSE}(\mathbf{y}^{(j)}, \mathbf{x}^{(j)}) \le T/n$. During data embedding, there would be individual cases. For example, three cases appear if $m^{(j)} \le 3$. Accordingly, all $\mathbf{x}^{(j)} \in FE$ in *part* ***B*** are classified to three different categories, and the flowchart of adaptive embedding is showed in Figure 6.

Case 1: $\mathrm{SSE}(\mathbf{x}^{(j)}) \le T/49$. If $\mathbf{x}^{(j)} \in C_3$, $m^{(j)} = 3$ can be used for embedding $3(n-1)$ bits into $\mathbf{x}^{(j)}$ without any underflow/ overflow, keeping $\mathrm{MSE}(\mathbf{y}^{(j)}, \mathbf{x}^{(j)}) \le T/n$. Thus, $\mathbf{x}^{(j)}$ is classified into the *embedded subcategory* FE_3 ($\mathbf{x}^{(j)} \to FE_3$).

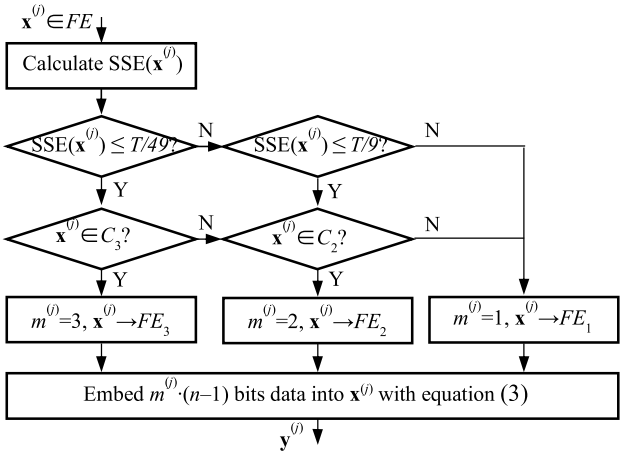


FIGURE 6. The flowchart of adaptive data embedding.

If $\mathbf{x}^{(j)} \notin C_3$ and $\mathbf{x}^{(j)} \in C_2$, $m^{(j)} = 2$ can be used for embedding $2(n-1)$ bits into $\mathbf{x}^{(j)}$ without any underflow/overflow, which also keeps MSE($\mathbf{y}^{(j)}$, $\mathbf{x}^{(j)}$) $\leq T/n$. Accordingly, $\mathbf{x}^{(j)}$ is classified into the *embedded category* FE_2 ($\mathbf{x}^{(j)} \to FE_2$). Otherwise, $m^{(j)} = 1$ should be used for embedding $(n-1)$ bits into $\mathbf{x}^{(j)}$ without any underflow/overflow, which still keeps MSE($\mathbf{y}^{(j)}$, $\mathbf{x}^{(j)}$) $\leq T/n$. Therefore, $\mathbf{x}^{(j)}$ is classified into the *embedded category* FE_1 ($\mathbf{x}^{(j)} \to FE_1$).

Case 2: $T/49 <$ SSE($\mathbf{x}^{(j)}$) $\leq T/9$. If $\mathbf{x}^{(j)} \in C_2$, $m^{(j)} = 2$ can be used for data embedding and $\mathbf{x}^{(j)} \to FE_2$. Otherwise, $m^{(j)} = 1$ should be used for data embedding and $\mathbf{x}^{(j)} \to FE_1$.

Case 3: $T/9 <$ SSE($\mathbf{x}^{(j)}$) $\leq T$, $m^{(j)} = 1$ can be used for data embedding and $\mathbf{x}^{(j)} \to FE_1$.

To distinguish $\mathbf{x}^{(j)} \in FE_1$, $\mathbf{x}^{(j)} \in FE_2$ or $\mathbf{x}^{(j)} \in FE_3$, in the second-level of *MLM*, "0" is assigned for $\mathbf{x}^{(j)} \in FE_1$ and "1" for $\mathbf{x}^{(j)} \in FE_2 \cup FE_3$. In the same way, "0" is assigned for $\mathbf{x}^{(j)} \in FE_2$ and "1" for $\mathbf{x}^{(j)} \in FE_3$ in the third-level of *MLM*. Then, the *MLM* is compressed losslessly by the JBIG algorithm to get *MLM*' with length l_3 bits, which will be embedded into $\mathbf{x}^{(j)} \in FE$ as the LSBs of $\mathbf{y}^{(j)}$. Set the number of $\mathbf{x}^{(j)} \in FE$ is $N_\mathbf{B}$, $N_\mathbf{B}$ should satisfy

$$(n-1) \cdot N_\mathbf{B} \geq l_3. \tag{10}$$

Set the owner's secret data is of l_1 bits and the reserved room of *part* $\boldsymbol{A}$ is of l_2 bits, given threshold T should satisfy

$$\sum_{j=1}^{N} m^{(j)} \cdot (n-1) \geq l_1 + l_2 + l_3. \tag{11}$$

The reserved room for data embedding after image encryption can be one or more LSB-planes of *part* $\boldsymbol{A}$, even part of single LSB-plane in *part* $\boldsymbol{A}$, depending on the requirement of the remote server. Also, in some applications, the requirement of embedding capacity is fixed. In such situations, the image owner should determine a proper threshold T to meet the embedding capacity requirement and preserve the quality of marked image. The determination of a proper threshold T can be accomplished by the binary search algorithm as shown in Algorithm I.

Algorithm 1 Determination of a Proper Threshold T for Special Embedding Capacity

```
Processing (I, EC)
/* I is the original image; */
/* EC is the embedding capacity required; */
/* Set T_l is the low threshold which cannot vacate embedding capacity larger than required;*/
/* Set T_h is the high threshold which can vacate embedding capacity larger than required;*/
begin
  while (T_h! = T_l and T_h! = T_l + 1)
    Set T_temp = ⌊(T_l + T_h)/2⌋;
    With T_temp, divide all pixel arrays into two categories
    FE and SE, then determine the transformation
    parameter m^(j) adaptively for each x^(j) ∈ FE, construct
    the multi-level location map and compress, calculate
    the net embedding capacity
    EC_temp;
    if EC_temp >= EC then
      Set T_h = T_temp;
    else
      Set T_l = T_temp;
   end if
  end while
  Set T_temp =⌊(T_l + T_h)/2⌋;
  With T_temp, calculate corresponding net embedding
  capacity EC_temp;
  If EC_temp >= EC then
    Set T = T_l;
  else
    Set T = T_h;
  end if
  return T;
end
```

For a special threshold T, the total embedding capacity is determined. Since embedding capacity for encrypted image is vacated by the image owner, the image owner takes the responsibility for the distribution of embedding capacity. Set the number of pixel arrays $\mathbf{x}^{(j)} \in SE$ as $N_\mathbf{A}$, the maximum embedding capacity for encrypted image will be less than $8 \cdot n \cdot N_\mathbf{A}$, where all pixel bit-planes in $\mathbf{x}^{(j)} \in SE$ are vacated for data embedding after image encryption. In practice, as the modifications of pixel's MSBs will introduce more distortion than that of pixel's LSBs, the vacated LSB-planes of $\mathbf{x}^{(j)} \in SE$ can be limited no larger than three, and the limited embedding capacity for encrypted image will be no larger than $3 \cdot n \cdot N_\mathbf{A}$ correspondingly. Within the maximum embedding capacity for encrypted image, the image owner can distribute some of the total embedding capacity in any proportion, for example, by vacating embedding capacity for encrypted image as more as possible or distributing the total embedding capacity equally between them. This way, the image owner achieves data embedding and embedding room reservation adaptively.

The procedure of data embedding in *part* ***B*** is described in Table 1, where we take $m^{(j)} \leq 3$ for illustration.

TABLE 1. Steps of data embedding in *part* *B* by AGIT.

Step-1. If given special embedding capacity, find the proper threshold T with Algorithm I. With a special T, classify the blocks in *part B* into three embedded subcategories FE_1, FE_2 and FE_3. Construct the Multi-level location map *MLM*. With the JBIG algorithm, *MLM* is compressed to *MLM'* with the length l_3 bits. Extract l_2 bits LSB-planes of A, denoted as LSB_A. With *Key* 1, encrypt l_1 bits of the owner's secret data to get *ES*1. Combine *MLM'*, LSB_A and *ES*1 sequentially as the data S to be embedded.

Step-2. In descending order of j, for each $\mathbf{x}^{(j)} \in FE$, with $m^{(j)}$, $\mathbf{x}^{(j)}$ is transform to $\mathbf{y}^{(j)}$ by using equation (3), where $(s_1, s_2, \ldots, s_{n-1})$ is the binary bits from S sequentially.

Step-3. In descending order of j, for each $\mathbf{x}^{(j)} \in FE_2 \cup FE_3$, $\mathbf{y}_i^{(j)} = \mathbf{y}_i^{(j)} + 2s_i$ $(i = 1, 2, \ldots, n-1)$.

Step-4. In descending order of j, for each $\mathbf{x}^{(j)} \in FE_3$, $\mathbf{y}_i^{(j)} = \mathbf{y}_i^{(j)} + 4s_i$ $(i = 1, 2, \ldots, n-1)$. Finally, the marked *part* ***B*** is generated.

3) IMAGE ENCRYPTION

After image partition and data embedding by AGIT, the marked image $\mathbf{I}^{\boldsymbol{M}}$ is generated by rearranging *part* ***A*** and *part* ***B***. With the stream cipher, the content owner encrypts $\mathbf{I}^{\boldsymbol{M}}$ to obtain the encrypted marked image $\mathbf{I}^{\boldsymbol{EM}}$ easily. A gray value $V_{r,s}$ in $\mathbf{I}^{\boldsymbol{M}}$ can be represented eight bits as $v_{r,s}^{(0)}$, $v_{r,s}^{(1)}, \ldots, v_{r,s}^{(7)}$, where (r, s) indicates the position of the pixel, $r = 1, 2, \ldots, N_1$, and $s = 1, 2, \ldots, N_2$. Thus,

$$v_{r,s}^{(k)} = \left\lfloor V_{r,s}/2^k \right\rfloor \bmod 2, \quad k = 0, 1, \cdots, 7, \tag{12}$$

and then,

$$V_{r,s} = \sum_{k=0}^{7} v_{r,s}^{(k)} \cdot 2^k. \tag{13}$$

In the encryption stage, a random sequence $\boldsymbol{E}$ of size $8 \times N_1 \times N_2$ is generated using *Encryption key* as

$$E = \left\{ e_{r,s}^{(k)} \middle| e_{r,s}^{(k)} \in \{0, 1\} \right\}. \tag{14}$$

Then, the bit-wise exclusive-or results of the pixel bits and pseudo-random bits are calculated as

$$v'^{(k)}_{r,s} = v_{r,s}^{(k)} \oplus e_{r,s}^{(k)}. \tag{15}$$

therefore, the encrypted pixel value is

$$V'_{r,s} = \sum_{k=0}^{7} v'^{(k)}_{r,s} \cdot 2^k \tag{16}$$

Finally, the image owner should embed 32 bits information into LSBs of first 32 pixels in encrypted version of *part* ***A*** to tell remote server the number of blocks and embedding capacity he can embed his additional data into, and get the encrypted marked image $\mathbf{I}^{\boldsymbol{EM}}$. Note that after image encryption and uploading, the remote server or a third party cannot access the original image and the owner's secret data without *Encryption key*, thus the privacy of the image owner can be protected.

B. SECOND-ROUND DATA EMBEDDING IN THE ENCRYPTED IMAGE

The proposed method allows the server's secret data to be further embedded into the encrypted marked image $\mathbf{I}^{\boldsymbol{EM}}$, even if the remote server is not able to access the original image. The embedding process starts with locating the encrypted version of *part A*, denoted by A_E. Since A_E has been rearranged to the front of $\mathbf{I}^{\boldsymbol{EM}}$, the remote server has direct access to read the 32-bits information in LSBs of the first 32 encrypted pixels to obtain the number of blocks to be embedded and the allowed embedding capacity. After that, the remote server just adopts LSB replacement to substitute the LSB-planes in A_E with his additional data, which are encrypted by *Key* 2. This way, the marked encrypted marked image $\mathbf{I}^{\boldsymbol{MEM}}$ is generated. Without the data hiding *Key* 2, no one could extract the additional data the remote server embeds.

C. DATA EXTRACTING AND IMAGE RECOVERY

On the recipient side, receivers may have different authorizations. As depicted in Figure 2, only with *Key* 2, the receiver can obtain the hidden server's data; with *Encryption key* and *Key*1, the receiver can get the hidden owner's data; if with *Encryption key*, the original image can be recovered perfectly. In general, with all the three keys, the two parts of additional data can be respectively extracted from $\mathbf{I}^{\boldsymbol{EM}}$ using the steps described in Table 2, and the original image can be recovered losslessly.

TABLE 2. Extracting and image recovering.

Step-1. Divide the marked encrypted marked image into non-overlapping blocks with n pixels. read the LSBs of the first 32 encrypted pixels to identify the blocks A_E' with the server's data.

Step-2. With *Key* 2, the authenticated receiver can extract the LSB-planes of A_E' and decrypt these data to get the hidden server's data.

Step-3. With *Encryption key*, the authenticated receiver decrypts the image.

Step-4. In reverse order, read the first (n-1) LSBs of each block until the JBIG end-of-stream (EOS) symbol to get *MLM'*. Decompress *MLM'* to reconstruct the multi-level location map *MLM*. Then, the value $m^{(j)}$ for each block is determined.

Step-5. With *MLM*, extract the embedded LSB_A and *ES*1. Then, *ES*1 is decrypted to get the owner's secret data with *Key* 1.

Step-6. In *part A*, the LSB-planes of blocks modified by the remote server are recovered with the extracted LSB_A. In *part B*, with $m^{(j)}$, each $\mathbf{x}^{(j)}$ is recovered exactly from $\mathbf{y}^{(j)}$ by using equation (5).

Step-7. With the *FLM* of *MLM*, rearrange all recovered blocks to get the original image.

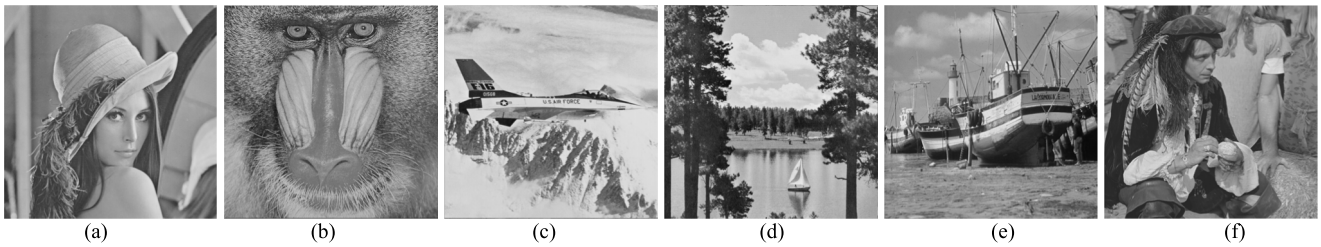


FIGURE 7. Test images. (a) Lena; (b) Baboon; (c) Airplane; (d) Lake; (e) Boat; (f) Man.

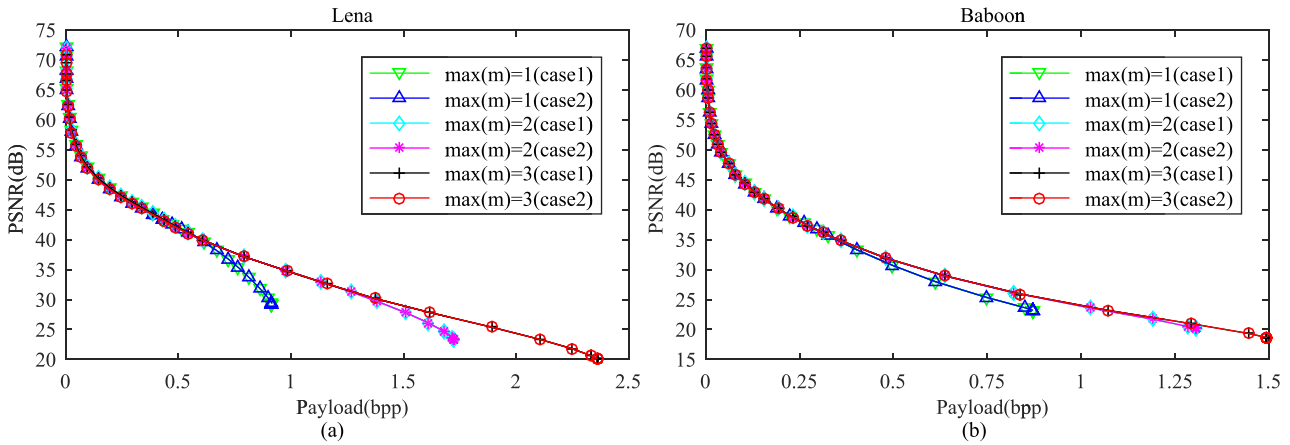


FIGURE 8. Performance of embedding capacity versus the decrypted marked image quality with different max(m).

IV. EXPERIMENTAL RESULTS

To verify the proposed method, we have conducted many experiments using six typical grayscale test images sized 512×512, as shown in Figure 7. We use the parameter $n = 4 \times 4$ as the amount of image block for GIT in the experiments.

Figure 8 shows the performances of Lena and Baboon in terms of payload versus PSNR. The payload, i.e. embedding capacity, is expressed with bit per pixel (*bpp*), and PSNR (dB) reflects the quality between decrypted marked image and original image. *Case*1 represents equal distribution of the total embedding capacity for both image owner and remote server, while *Case* 2 stands for vacating embedding capacity for encrypted image as much as possible. With fixed $max(m)$, i.e., $max(m) \in \{1, 2, 3\}$, the performance is almost the same between these two cases. With different $max(m)$, along with the increased payload, the larger $max(m)$, the better performance and the larger maximal payload can be achieved.

In terms of quality of the decrypted marked image versus embedding capacity, Figure 9 (a) $\sim$ (d) shows the composition of embedding capacity for Lena and Baboon, with $max(m) = 1$ and $max(m) = 3$ respectively. The curves of total embedding capacity (EC) indicates the achieved payload of two parts of secret data with different image quality. The curves of embedding capacity for remote serves shows the achieved payload of the server's secret data, and the rest is for the owner's secret data. When the total embedding capacity is low, i.e. the threshold T is low, the distribution of total embedding capacity adheres to the principle of *Case*1 or *Case*2. As the threshold increases, total embedding capacity increases accordingly. And when the blocks of *part A* decreases, the embedding capacity for remote server will reach the maximum, and then decrease with limited embedding capacity. The reserved room of *part A* is limited to three LSB planes in the experiment. Releasing the limitation to higher bit-planes will enlarge the limited embedding capacity, but the distortion will be significantly higher.

Next, we test whether the proposed method is able to determine a proper threshold T to meet special requirement of embedding capacity. Table 3 shows the real achieved embedding capacity when given different required embedding capacities. By using the proposed algorithm of proper threshold determination, we compare the results with that of the traditional RDH method in image clear domain, namely, adaptive reversible data hiding by extending the generalized integer transformation (ARDHGIT) [6]. It is obvious that the real achieved embedding capacity is larger than the required embedding capacity. It indicates we are able to find a proper threshold by the proposed algorithm. With almost the same embedding capacity, the quality of decrypted marked image of the proposed method is close to the quality of marked image produced by ARDHGIT.

We also compare the proposed method with four state-of-the-art VRBE-based RDH-EI methods, as shown

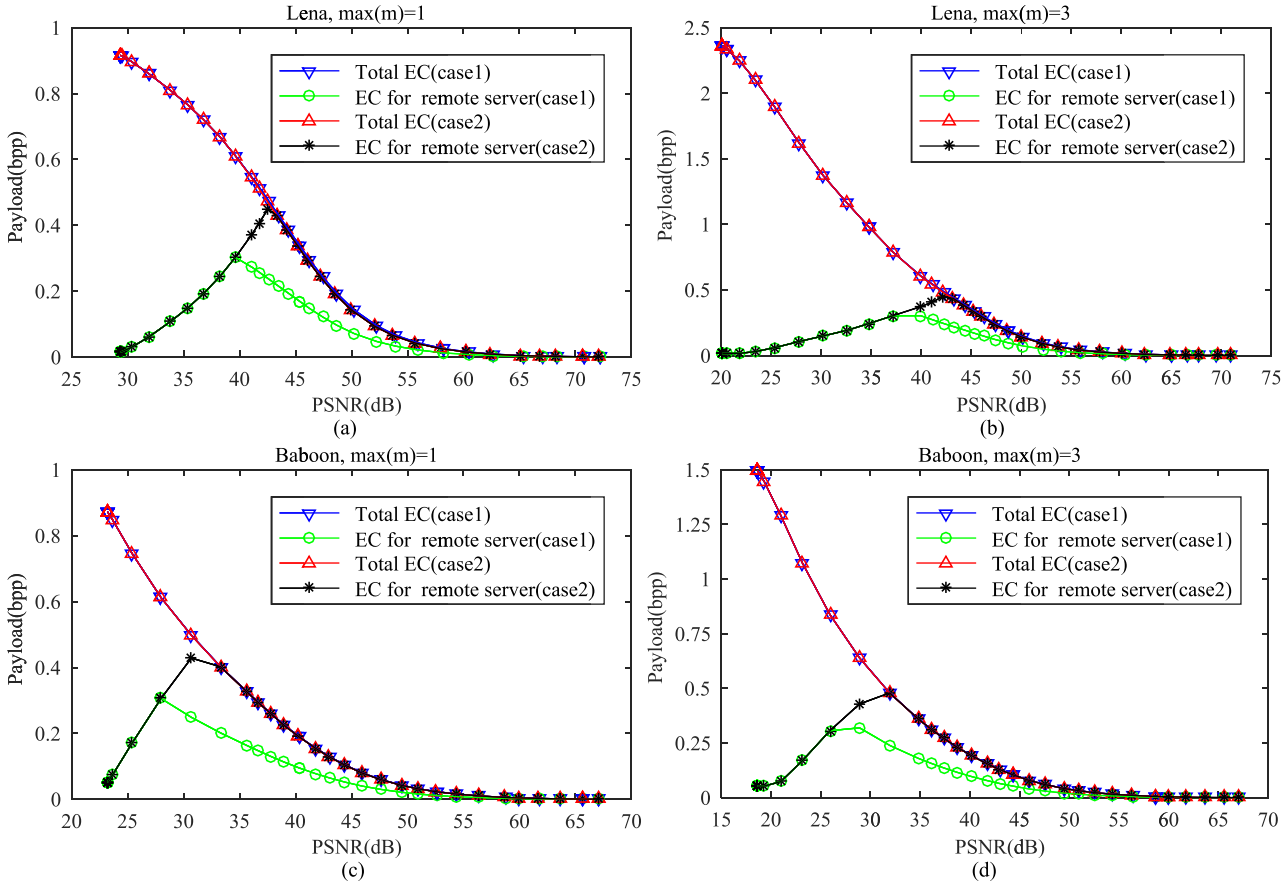


FIGURE 9. Composition of embedding capacity.

TABLE 3. The achieved embedding capacity/psnr with given embedding capacity (Bits/dB).

Image	Method	Given embedding capacity (Bits)					
		10,000	20,000	40,000	80,000	160,000	320,000
Lena	[6]	10,128/56.83	20,970/53.46	40,477/50.19	80,446/46.33	160,699/40.12	320,096/31.99
	Proposed	10,407/56.34	21,007/53.08	40,189/49.86	80,333/46.01	160,054/39.89	320,113/31.72
Baboon	[6]	10,067/50.67	20,052/46.71	40,135/42.18	80,075/36.56	160,054/29.44	320,056/21.67
	Proposed	10,072/50.06	20,156/46.21	40,082/41.89	80,124/36.37	160,051/29.41	320,181/21.65
Airplane	[6]	10,416/60.35	20,982/56.79	40,313/53.06	80,617/48.09	160,398/42.01	321,487/33.12
	Proposed	11,015/59.52	21,377/56.07	41,538/52.25	80,651/47.62	160,690/41.70	321,421/32.79
Lake	[6]	10,109/57.31	20,058/52.62	40,251/47.56	80,101/42.09	160,095/35.57	320,055/26.67
	Proposed	10,269/56.60	20,163/52.11	40,475/47.22	80,129/41.82	160,045/35.33	320,081/26.55
Boat	[6]	10,139/55.03	20,265/50.54	40,041/46.02	80,042/41.21	160,071/35.05	320,042/26.69
	Proposed	10,065/54.49	20,196/50.11	40,144/45.63	80,137/40.92	160,140/34.84	320,057/26.60
Man	[6]	10,188/58.82	20,100/54.43	40,414/49.29	80,115/43.78	160,194/36.72	320,046/28.33
	Proposed	10,126/57.90	20,354/53.48	40,148/48.66	80,088/43.30	160,290/36.45	320,065/28.19
Average	[6]	10,175/56.50	20,405/52.42	40,272/48.05	80,233/43.01	160,252/36.48	320,297/28.08
	Proposed	10,326/55.82	20,542/51.84	40,429/47.59	80,244/42.67	160,212/36.27	320,320/27.92

in Figure 10. For [35], three LSB-planes of certain selected pixels are reserved for data embedding after image encryption. Set the percentages of pixels $p \in \{1\%, 5\%, 10\%, 15\%, 20\%\}$ for estimation with [36]. For [39], we set host pixels' number of the reference pixel $T = 4$ and the parameter of hidden bits $n = 3$. Block size $n = 4 \times 4$ is used for [49], and set $n = 4 \times 4$ and $max(m) = 4$ for our method. Given low payloads, the performances of these methods except [39] are close where high PSNR can be ensured. However, the highest payload provided by [36] is less than 0.05 *bpp*, and that of [35] less than 0.75 *bpp*, that of [49] no more than 0.99 *bpp*. With dual data embedding as [49], the proposed method achieves much higher payload with good rate-distortion performance.

Some property comparisons are summarized in Table 4. Methods in [19]–[22], [24]–[27] and [33], [34] are based

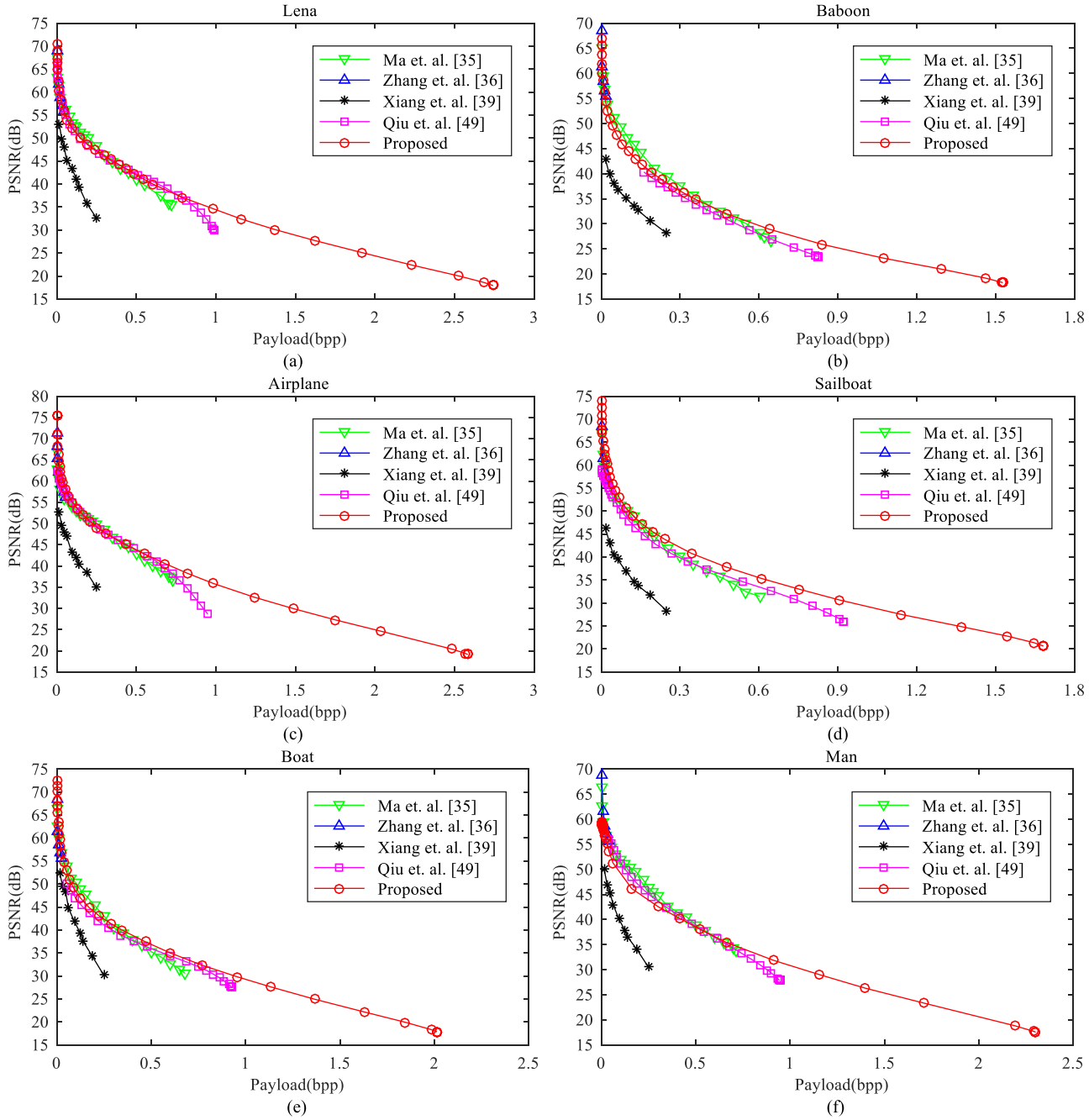


FIGURE 10. Performance comparisons with Ma et al. [35], Zhang et al. [36], Xiang et al. [39], and Qiu et al. [49].

on VRAE, while the proposed method and methods in [35]–[49] are based on VRBE. Since methods in [19]–[22] achieve data extraction and image recovery jointly, there is error in data extraction or image recovery in some cases. In the other methods, data extraction and image recovery are carried out separately, the embedded data can be extracted exactly, and the original image can be recovered losslessly except the CPE method in [42]. In terms of the achieved maximal payload, the levels are represented by *Low* (< 0.5 *bpp*), *Medium* (≥ 0.5 *bpp* and < 1.0 *bpp*), *High* (≥ 1.0 *bpp* and < 2.0 *bpp*), and *Huge* (≥ 2.0 *bpp*). Therefore, the reported maximal payload of most VRAE methods [19]–[22], [24]–[27], [34] are relatively low except method in [33], which can achieve medium payload. In VRBE methods, the maximal payload of methods in [36], [37], [39] is low, that of methods in [35], [38], [41], [49] and the EPE method in [42] is medium, high payload can be achieved in [40], CPE method [42] and [44], [45], while the proposed method and methods in [43]–[48] can achieve huge payload more than 2.0 *bpp*. It is obvious that the VRBE methods can achieve higher payload than VRAE methods. Then, we compare the distortion-rate performance with payload versus PSNR of the

TABLE 4. Property comparisons of related schemes with our method.

Methods	Properties						
	VRBE /VRAE	Separable / Joint	Extraction error	Image recovery error	Maximal payload	Performance in payload versus PSNR	Data embedding for content owner
Zhang et al. [19]	*VRAE*	*Joint*	*Yes*	*Yes*	*Low*	*Medium*	*No*
Hong et al. [20]	*VRAE*	*Joint*	*Yes*	*Yes*	*Low*	*Medium*	*No*
Liao et.al. [21]	*VRAE*	*Joint*	*Yes*	*Yes*	*Low*	*Medium*	*No*
Qin et.al. [22]	*VRAE*	*Joint*	*Yes*	*Yes*	*Low*	*Medium*	*No*
Zhang et al. [24]	*VRAE*	*Separable*	*No*	*No*	*Low*	*Medium*	*No*
Zhang et al. [25]	*VRAE*	*Separable*	*No*	*No*	*Low*	*Medium*	*No*
Qian et al. [26]	*VRAE*	*Separable*	*No*	*No*	*Low*	*Medium*	*No*
Zheng et al. [27]	*VRAE*	*Separable*	*No*	*No*	*Low*	*High*	*No*
Li et al. [33]	*VRAE*	*Separable*	*No*	*No*	*Medium*	*High*	*No*
Zhang et al. [34]	*VRAE*	*Separable*	*No*	*No*	*Low*	*Medium*	*No*
Ma et al. [35]	*VRBE*	*Separable*	*No*	*No*	*Medium*	*High*	*No*
Zhang et al. [36]	*VRBE*	*Separable*	*No*	*No*	*Low*	*High*	*No*
Xu et al. [37]	*VRBE*	*Separable*	*No*	*No*	*Low*	*High*	*No*
Cao et al. [38]	*VRBE*	*Separable*	*No*	*No*	*Medium*	*High*	*No*
Xiang et al. [39]	*VRBE*	*Separable*	*No*	*No*	*Low*	*Medium*	*No*
Wu et al. [40]	*VRBE*	*Separable*	*No*	*No*	*High*	*High*	*No*
Yi et al. [41]	*VRBE*	*Separable*	*No*	*No*	*Medium*	*Low*	*No*
Puteaux et al.'s CPE [42]	*VRBE*	*Separable*	*No*	*Yes*	*High*	*Low*	*No*
Puteaux et al.'s EPE [42]	*VRBE*	*Separable*	*No*	*No*	*Medium*	*Low*	*No*
Puteaux et al. [43]	*VRBE*	*Separable*	*No*	*No*	*Huge*	*Low*	*No*
Yi et al. [44]	*VRBE*	*Separable*	*No*	*No*	*High*	*Low*	*No*
Yi et al. [45]	*VRBE*	*Separable*	*No*	*No*	*High*	*Low*	*No*
Wu et al. [46]	*VRBE*	*Separable*	*No*	*No*	*Huge*	*Low*	*No*
Chen et al. [47]	*VRBE*	*Separable*	*No*	*No*	*Huge*	*Low*	*No*
Yin et al. [48]	*VRBE*	*Separable*	*No*	*No*	*Huge*	*Low*	*No*
Qiu et al. [49]	*VRBE*	*Separable*	*No*	*No*	*Medium*	*High*	*Yes*
Proposed	*VRBE*	*Separable*	*No*	*No*	*Huge*	*High*	*Yes*

decrypted marked image, from which, the embedded data and the original image should be able to be recovered. Since it is considered that there is no need to preserve the high quality of the encrypted image compared to the clear domain [42], these methods in [41]–[48] vacate the embedding room by modifying higher bit-planes or MSBs-planes to achieve higher payload, the distortion-rate performance is theoretically not good. With low payloads, the PSNR by the methods in [19]–[22], [24]–[26], [34], [39] range from 50dB to 30dB approximately, so the corresponding distortion-rate performance is medium. Methods in [27], [33], [35]–[38], [40], [49] and the proposed method can achieve good distortion-rate performance, some are showed in Figure 10. Among of them, just the proposed method can achieve huge maximal payload. The last column indicates that most RDH-EI method ignoring the requirement of data embedding for the image owner. To our knowledge, dual data embedding for both image owner and remote server was first presented in [49]. The proposed method achieves much higher payload than method [49], meeting the requirement of embedding capacity adaptively.

V. CONCLUSION

In this paper, we propose a new VRBE-based RDH-EI method which not only enables the remote server to embed data in image encrypted domain, but also allows the image owner to embed data before image encryption. Using the

technique of GIT, the image owner embeds some additional data and vacates room for data embedding for encrypted domain before image encryption. After image encryption and uploading, it is easy for the remote server to embed data into the encrypted image by LSB replacement. On the recipient side, the embedded data by the image owner and the remote server can be extracted exactly, and the original image be recovered free of error after image decryption. Experimental results show that the proposed method is efficient in data embedding and effective in data security.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their constructive suggestions in improving this articles.

REFERENCES

[1] M. Celik, G. Sharma, A. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[2] M. Celik, G. Sharma, and A. Tekalp, "Lossless watermarking for image authentication: A new framework and an implementation," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 1042–1049, Apr. 2006.

[3] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[4] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.

[5] X. Wang, X. Li, B. Yang, and Z. Guo, "Efficient generalized integer transform for reversible watermarking," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 567–570, Jun. 2010.

[6] Y. Qiu, Z. Qian, and L. Yu, "Adaptive reversible data hiding by extending the generalized integer transformation," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 130–134, Jan. 2016.

[7] Z. Ni, Y. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[8] X. Li, W. Zhang, X. Gui, and B. Yang, "Efficient reversible data hiding based on multiple histograms modification," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 2016–2027, Sep. 2015.

[9] J. Wang, J. Ni, X. Zhang, and Y.-Q. Shi, "Rate and distortion optimization for reversible data hiding using multiple histogram shifting," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 315–326, Feb. 2017.

[10] J. Qin and F. Huang, "Reversible data hiding based on multiple two-dimensional histograms modification," *IEEE Signal Process. Lett.*, vol. 26, no. 6, pp. 843–847, Jun. 2019.

[11] X. Zhang, "Reversible data hiding with optimal value transfer," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 316–325, Feb. 2013.

[12] X. Hu, W. Zhang, X. Hu, N. Yu, X. Zhao, and F. Li, "Fast estimation of optimal marked-signal distribution for reversible data hiding," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 5, pp. 779–788, May 2013.

[13] W. Zhang, X. Hu, X. Li, and Y. Nenghai, "Optimal transition probability of reversible data hiding for general distortion metrics and its applications," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 294–304, Jan. 2015.

[14] D. Hou, W. Zhang, Y. Yang, and N. Yu, "Reversible data hiding under inconsistent distortion metrics," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5087–5099, Oct. 2018.

[15] F. Huang, X. Qu, H. J. Kim, and J. Huang, "Reversible data hiding in JPEG images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1610–1621, Sep. 2016.

[16] D. Hou, H. Wang, W. Zhang, and N. Yu, "Reversible data hiding in JPEG image based on DCT frequency and block selection," *Signal Process.*, vol. 148, pp. 41–47, Jul. 2018.

[17] Y. Qiu, H. He, Z. Qian, S. Li, and X. Zhang, "Lossless data hiding in JPEG bitstream using alternative embedding," *J. Vis. Commun. Image Represent.*, vol. 52, pp. 86–91, Apr. 2018.

[18] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, and B. Ma, "Reversible data hiding: Advances in the past two decades," *IEEE Access*, vol. 4, pp. 3210–3237, 2016.

[19] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Process. Lett.*, vol. 18, no. 4, pp. 255–258, Apr. 2011.

[20] W. Hong, T.-S. Chen, and H.-Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Process. Lett.*, vol. 19, no. 4, pp. 199–202, Apr. 2012.

[21] X. Liao and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," *J. Vis. Commun. Image Represent.*, vol. 28, pp. 21–27, Apr. 2015.

[22] C. Qin and X. Zhang, "Effective reversible data hiding in encrypted image with privacy protection for image content," *J. Vis. Commun. Image Represent.*, vol. 31, pp. 154–164, Aug. 2015.

[23] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 3, pp. 441–452, Mar. 2016.

[24] X. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 826–832, Apr. 2012.

[25] X. Zhang, Z. Qian, G. Feng, and Y. Ren, "Efficient reversible data hiding in encrypted images," *J. Vis. Commun. Image Represent.*, vol. 25, no. 2, pp. 322–328, Feb. 2014.

[26] Z. Qian and X. Zhang, "Reversible data hiding in encrypted images with distributed source encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 4, pp. 636–646, Apr. 2016.

[27] S. Zheng, D. Li, D. Hu, D. Ye, L. Wang, and J. Wang, "Lossless data hiding algorithm for encrypted images with high capacity," *Multimedia Tools Appl.*, vol. 75, no. 21, pp. 13765–13778, Nov. 2016.

[28] Z. Qian, H. Zhou, X. Zhang, and W. Zhang, "Separable reversible data hiding in encrypted JPEG bitstreams," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 6, pp. 1055–1067, Nov. 2018.

[29] J. He, J. Chen, W. Luo, S. Tang, and J. Huang, "A novel high-capacity reversible data hiding scheme for encrypted JPEG bitstreams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 12, pp. 3501–3515, Dec. 2019.

[30] Z. Qian, H. Xu, X. Luo, and X. Zhang, "New framework of reversible data hiding in encrypted JPEG bitstreams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 2, pp. 351–362, Feb. 2019.

[31] X. Zhang, C. Qin, and G. Sun, "Reversible data hiding in encrypted images using pseudorandom sequence modulation," in *Proc. Int. Workshop Digit.-Forensics Watermarking*, 2012, pp. 358–367.

[32] X. Zhang, "Commutative reversible data hiding and encryption," *Secur. Commun. Netw.*, vol. 6, no. 11, pp. 1396–1403, Nov. 2013.

[33] M. Li, D. Xiao, Y. Zhang, and H. Nan, "Reversible data hiding in encrypted images using cross division and additive homomorphism," *Signal Process., Image Commun.*, vol. 39, pp. 234–248, Nov. 2015.

[34] X. Zhang, J. Long, Z. Wang, and H. Cheng, "Lossless and reversible data hiding in encrypted images with public-key cryptography," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1622–1631, Sep. 2016.

[35] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 3, pp. 553–562, Mar. 2013.

[36] W. Zhang, K. Ma, and N. Yu, "Reversibility improved data hiding in encrypted images," *Signal Process.*, vol. 94, pp. 118–127, Jan. 2014.

[37] D. Xu and R. Wang, "Separable and error-free reversible data hiding in encrypted images," *Signal Process.*, vol. 123, pp. 9–21, Jun. 2016.

[38] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo, "High capacity reversible data hiding in encrypted images by patch-level sparse representation," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1132–1143, May 2016.

[39] S. Xiang and X. Luo, "Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3099–3110, Nov. 2018.

[40] H.-T. Wu, Y.-M. Cheung, Z. Yang, and S. Tang, "A high-capacity reversible data hiding method for homomorphic encrypted images," *J. Vis. Commun. Image Represent.*, vol. 62, pp. 87–96, Jul. 2019.

[41] S. Yi and Y. Zhou, "Binary-block embedding for reversible data hiding in encrypted images," *Signal Process.*, vol. 133, pp. 40–51, Apr. 2017.

[42] P. Puteaux and W. Puech, "An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1670–1681, Jul. 2018.

[43] P. Puteaux and W. Puech, "EPE-based huge-capacity reversible data hiding in encrypted images," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, Hong Kong, Dec. 2018, pp. 1–7.

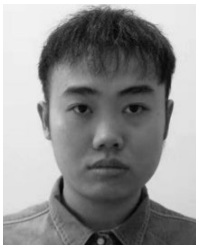
[44] P. Yi, Z. Yin, and Z. Qian, "Reversible data hiding in encrypted images with two-MSB prediction," in *Proc. 10th IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Hong Kong, Dec. 2018, pp. 1–7.

[45] S. Yi and Y. Zhou, "Separable and reversible data hiding in encrypted images using parametric binary tree labeling," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 51–64, Jan. 2019.
[46] Y. Wu, Y. Xiang, Y. Guo, J. Tang, and Z. Yin, "An improved reversible data hiding in encrypted images using parametric binary tree labeling," *IEEE Trans. Multimedia*, to be published, doi: 10.1109/tmm.2019.2952979.
[47] K. Chen and C.-C. Chang, "High-capacity reversible data hiding in encrypted images based on extended run-length coding and block-based MSB plane rearrangement," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 334–344, Jan. 2019.
[48] Z. Yin, Y. Xiang, and X. Zhang, "Reversible data hiding in encrypted images based on multi-MSB prediction and Huffman coding," *IEEE Trans. Multimedia*, to be published, doi: 10.1109/TMM.2019.2936314.
[49] Y. Qiu, H. Wang, Z. Wang, Z. Qian, G. Feng, and X. Zhang, "Reversible contrast mapping based reversible data hiding in encrypted images," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, Hong Kong, Dec. 2018, pp. 1–7.

YINGQIANG QIU received the B.S. and M.S. degrees from the University of Science and Technology of China (USTC), in 2003 and 2006, respectively, and the Ph.D. degree from Fuzhou University, China, in 2017. He is currently an Associate Professor with the College of Information Science and Engineering, Huaqiao University. He has published more than 20 peer-reviewed journal articles and conference papers. His research interests include information hiding, image processing, and multimedia security.

QICHAO YING received the B.S. degree from Shanghai University, China, in 2017, where he is currently pursuing the M.S. degree. His research interests include information hiding, image processing, and multimedia security.

XIAODAN LIN received the B.S. degree from Central South University, China, the M.S. degree from Shanghai Jiaotong University, China, and the Ph.D. degree from Sun Yat-sen University, Guangzhou, China. She has been with the Faculty of the School of Information Science and Engineering, Huaqiao University, since 2008. Her research interests include signal processing, machine learning, and multimedia forensics.



YAOWEN ZHANG received the B.S. degree from the Anhui University of Technology, China, in 2017. He is currently pursuing the M.S. degree with Huaqiao University, China. His research interests include information hiding and multimedia security.

ZHENXING QIAN (Member, IEEE) received the B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC), in 2003 and 2007, respectively. He is currently a Professor with the School of Computer Science, Fudan University. He has published over 100 peer-reviewed articles on international journals and conferences. His research interests include information hiding, image processing, and multimedia security.