Vector-based Efficient Data Hiding in Encrypted Images via Multi-MSB Replacement

Yike Zhang, *Student Member, IEEE,* and Wenbin Luo, *Senior Member, IEEE*

Abstract—As an essential technique for data privacy protection, reversible data hiding in encrypted images (RDHEI) methods have drawn intensive research interest in recent years. In response to the increasing demand for protecting data privacy, novel methods that perform RDHEI are continually being developed. We propose two effective multi-MSB (most significant bit) replacement-based approaches that yield comparably high data embedding capacity, improve overall processing speed, and enhance reconstructed images' quality. Our first method, Efficient Multi-MSB Replacement-RDHEI (EMR-RDHEI), obtains higher data embedding rates (DERs, also known as payloads) and better visual quality in reconstructed images when compared with many other state-of-the-art methods. Our second method, Lossless Multi-MSB Replacement-RDHEI (LMR-RDHEI), can losslessly recover original images after an information embedding process is performed. To verify the accuracy of our methods, we compared them with other recent RDHEI techniques and performed extensive experiments using the widely accepted BOWS-2 dataset. Our experimental results showed that the DER of our EMR-RDHEI method ranged from 1.2087 bit per pixel (bpp) to 6.2682 bpp with an average of 3.2457 bpp. For the LMR-RDHEI method, the average DER was 2.5325 bpp, with a range between 0.2129 bpp and 6.0168 bpp. Our results demonstrate that these methods outperform many other state-of-the-art RDHEI algorithms. Additionally, the multi-MSB replacement-based approach provides a clean design and efficient vectorized implementation.

Index Terms—Reversible data hiding, image encryption, multi-MSB replacement, location map

I. INTRODUCTION

Reversible data hiding in encrypted images (RDHEI) methods aim to protect data privacy and integrity. Digital image security plays an essential role in protecting data privacy in the military and medical industries. With the rapid development of cloud computing, more and more content owners need effective techniques to secure the data transfer process. RDHEI methods allow two separate entities to store and share information inside an image without knowing each other's identity. For this reason, RDHEI technology is urgently sought for cloud-based storage services. A cloud-based data provider (the data hider) can use RDHEI to embed information without knowing the original content. Overall, RDHEI provides both privacy and security for the content owner and the receiver.

The demand for using this technology with cloud-based storage services has made RDHEI a critical research field worldwide. Many of the current Reversible Data Hiding (RDH) methods are suitable only when the data hider is also

Yike Zhang and Wenbin Luo are with the Engineering Department, St. Mary's University, San Antonio, TX, 78228, USA (e-mail: yike.zhang@vanderbilt.edu; wluo@stmarytx.edu). *Source code is available upon request.*

the content owner. Despite the features of RDH, using them in a cloud-based environment may not be optimal, as the content owner and the data hider are two different entities. With the use of RDHEI, after the content owner uploads encrypted images to the cloud, the data hider can embed additional information into the encrypted images for various purposes such as tagging and inserting personal information. Over the years, researchers have developed more state-of-the-art techniques to protect data privacy, security, and integrity.

The main challenge in this research lies in finding a better trade-off between embedding capacity and maintaining the quality of the reconstructed images [18]. A growing number of approaches have come to prominence for increasing information hiding capacity in encrypted images while preserving great visual quality during recovery. Generally, RDHEI methods can be classified into two groups: Reserving Room Before Encryption (RRBE) [40, 3, 13, 21] and Vacating Room After Encryption (VRAE) [41, 23, 6, 21]. The RRBE schema is shown in Fig. 1. These techniques can be further divided into the following categories: prediction error [35, 7], location map employment [39, 14], data compression [20, 33], histogram modification [37, 34], and difference expansion [4, 26]. Moreover, some methods [13, 3] can perform data extraction and image decryption separately, while other approaches [6, 31] require simultaneous extraction and decryption.

We propose two novel RRBE-based schemas to address image recovery in both lossy and lossless RDHEI, which we have shown to outperform many state-of-the-art techniques. For our first method, Efficient Multi-MSB Replacement-RDHEI (EMR-RDHEI), the assistant information is embedded in the least significant bits (LSBs) of a given image to lower the original image's bit loss and retain the high visual quality of the reconstructed image. The other proposed method, Lossless Multi-MSB Replacement-RDHEI (LMR-RDHEI), safely inserts the assistant information into the most significant bits (MSBs) of an image. The LMR-RDHEI method recovers an image without any error; i.e., the PSNR (Peak signal-to-noise ratio) of a restored image reaches infinity. The JBIG-KIT image compression library [15] is used in the LMR-RDHEI method to realize lossless image recovery while achieving a comparably high data embedding capacity. Both of our multi-MSB replacement-based approaches achieve high DER and PSNR for restored images when testing with the BOWS-2 dataset.

In summary, the main contributions of our work are presented below:

- We propose two novel methods for lossy and lossless RDHEI, named EMR-RDHEI and LMR-RDHEI, respec-
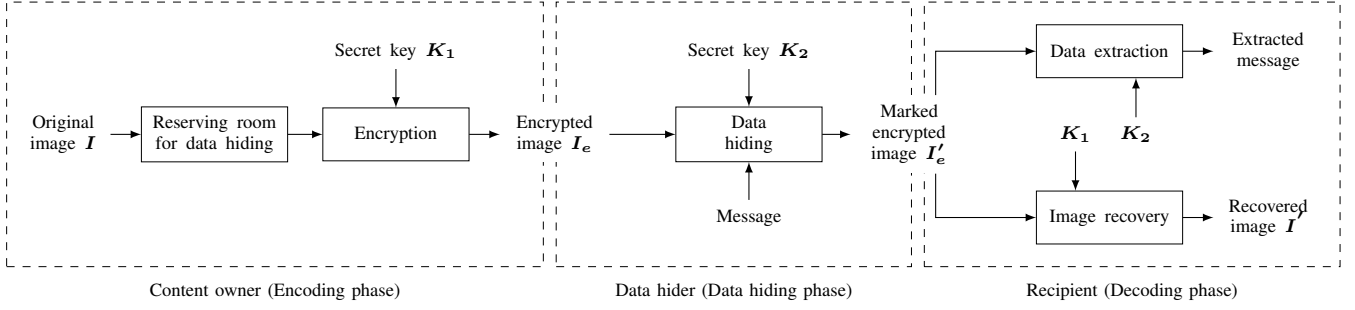
Fig. 1: Reserving Room Before Encryption (RRBE) Schema.

tively. The methods utilize a multi-MSB replacement-based approach, allowing large volumes of data to be embedded while retaining high/lossless reconstructed images' visual quality.

- Compared with other state-of-the-art algorithms, the design and implementation of our methods involve lower complexity computations and do not require extra files/overhead during the transfer process, as all data is embedded directly into images.
- We demonstrate through the benchmark BOWS-2 dataset that our methods achieve one of the highest reported DER numbers (3.2457 bpp and 2.5325 bpp for EMR-RDHEI and LMR-RDHEI, respectively). Moreover, the reconstructed image quality is similar to or better than many other state-of-the-art techniques.

The rest of our paper is organized as follows: Section II discusses other state-of-the-art approaches in this field. Section III presents our proposed methods. Section IV demonstrates experiments, security analysis, and comparison details. Finally, Section V summarizes our methods' limitations and discusses future work.

II. RELATED WORK

RDHEI is particularly useful for securing data transfer, including information authentication and protection. The RRBE pipelines, shown in Fig. 1, provide a general schema for the safe delivery and authentication of data. Initially, in the RRBE schema, the content owner reserves spare room in an image and encrypts it. Then, the data hider embeds secret information inside the encrypted image, and this is known as the marking process. After obtaining the marked encrypted image, the receiver decrypts the image and extracts the secret information simultaneously [13, 3] or separately [6, 31] depending on the available keys.

The first prediction-based method was proposed by Puteaux and Puech [18], with the aim of embedding secret message in the MSB of each pixel. In their approach, predicting the MSB values without errors is a high priority. During the image reconstruction phase, they use both top and left neighboring pixels to predict the current pixel value. Considering the current pixel $p(i,j)$, with $0 \leq i \leq M-1$ and $0 \leq j \leq N-1$, for an image P of size $M \times N$, the current pixel's inverse value is computed by:

$$inv(i,j) = (p(i,j) + 128) \mod 256 \qquad (1)$$

The computed value $pred(i,j)$ is considered as a predictor in the final decoding phase. In order to predict the current pixel $pred(i,j)$, the authors utilized the upper pixel $p(i,j-1)$ and left pixel $p(i-1,j)$ to calculate it:

$$pred(i,j) = (p(i-1,j) + p(i,j-1))/2 \qquad (2)$$

Next, the absolute difference between $pred(i,j)$, $p(i,j)$ and between $pred(i,j)$, $inv(i,j)$ is stored into Δ and Δ^{inv}, respectively:

$$\begin{cases} \Delta = |pred(i,j) - p(i,j)| \\ \Delta^{inv} = |pred(i,j) - inv(i,j)| \end{cases} \qquad (3)$$

If $\Delta \leq \Delta^{inv}$, there is no prediction error since the value of $pred(i,j)$ is closer to its predictor than the inverse value $inv(i,j)$. Otherwise, there can be an error, and the corresponding information is stored in an error location binary map.

In their paper, they applied the prediction-based technique in two different approaches: the first approach, CPE-HCRDH, yielded reconstructed images with slight distortions from the original ones, and the second approach, EPE-HCRDH, used flags to highlight incorrectly predicted pixels in order to realize lossless image recovery. According to their results, the prediction-based methods significantly increased the DER to a maximum of 1 bpp. Previously, none of the existing methods succeeded in combining high embedding capacity (near 1 bpp) and high restored images' visual quality (greater than 50 dB) [18].

However, a shortcoming exists in this EPE-HCRDH method. There is a small probability (1/256) of identifying an embedded 8-bit data string "11111111" as a flag, which will jeopardize the entire data recovery process. Our proposed schemas do not employ any fixed sentinel values as flags, and the proposed LMR-RDHEI method recovers embedded data and restores the original image perfectly during the decoding phase.

Inspired by Puteaux and Puech [18], Khan et al. [10] presented a Prediction Error Estimation technique that leveraged how adjacent pixels in the original image are closely related and thus easier to be predicted using their neighboring pixels. In their approach, Khan et al. [10] also use the upper pixel $p(i,j-1)$ and left pixel $p(i-1,j)$ to predict the current pixel $p(i,j)$. With the help of a binary location map, the available pixel locations are stored for embedding secret message's bits. Unlike Puteaux and Puech [18]'s approach, Khan et al.

[10] opted for Arithmetic Coding [29] to compress the binary location map, embedding it into the encrypted image's LSBs. The original LSBs are stored in those MSBs that could be predicted without errors in the decoding procedure.

Their results demonstrated complete reversibility of the original image while retrieving error-free data. However, the resulting DER is still comparably low. For example, when applying the method to the grey-level image Lena of size 512×512, they only achieved the embedding rate of 0.995 bpp.

Puteaux and Puech [18]'s prediction-based method was improved by Puyang et al. [22] when they published a two-MSB prediction model. They introduced the Median Edge Detection (MED) predictor that generates predictions by using three neighboring pixels to acquire the prediction value $pred(i,j)$, as illustrated in the following:

$$pred(i,j) = \begin{cases} \max(p(i-1,j), p(i,j-1)), p(i-1,j-1) < \min(p(i-1,j), p(i,j-1)) \\ \min(p(i-1,j), p(i,j-1)), p(i-1,j-1) > \min(p(i-1,j), p(i,j-1)) \\ p(i-1,j) + p(i,j-1) - p(i-1,j-1) \quad \text{otherwise} \end{cases} \quad (4)$$

Their experimental results showed an average DER of 1.346 bpp, which is a higher embedding rate compared with Puteaux and Puech [18]'s work. However, their DER is still not optimal with the two-MSB prediction model and limits the DER to a maximum of 2 bpp.

In 2020, Puteaux and Puech [20] proposed a new recursive schema to label encrypted images by using the LOCO-I compression kit to embed bits in each bit-plane of an image. Their approach uses a grey image with size $M \times N$ pixels as a stack of 8 bit-planes $I^{[k]}$, with $0 \leq k \leq 7$. $I_k^{[k,7]}$ indicates the original images after k adaptations. Then, the prediction error technique is used to predict each pixel value $p_k^{[k,7]}(i,j)$ of the image $I_k^{[k,7]}$, using the $7-k$ least significant bit-planes $I_k^{[k+1,7]}$ and previously scanned pixels. During this process, the first-bit value $p_k^{[0,0]}$ cannot be predicted. It is kept unmodified and serves to initialize the prediction. A pixel $p_k^{[k,7]}(i,j)$ from $I_k^{[k+1,7]}$ is made of $8-k$ bits, and is defined as:

$$p_k^{[k,7]}(i,j) = \sum_{l=k}^{7} p_k^l(i,j) \times 2^{7-l} \quad (5)$$

Where $p_k^l(i,j)$ is the bit at index l.

For their prediction-based schema, Puteaux and Puech [20] combine the Median Edge Detection predictor (also known as LOCO-I [16]) with the predictor $pred(i,j)$ of the pixel $p_k^{[k,7]}(i,j)$. $pred(i,j)$ is defined in the following:

$$pred(i,j) = \text{MED}(p_k^{[k,7]}(i,j)) = \begin{cases} \min(A,B) & \text{if } C \geq \max(A,B) \\ \max(A,B) & \text{if } C \leq \min(A,B) \\ A+B-C & \text{otherwise} \end{cases} \quad (6)$$

$C = p_k^{[k,7]}(i-1,j-1)$	$B = p_k^{[k,7]}(i,j-1)$
$A = p_k^{[k,7]}(i-1,j)$	$p_k^{[k,7]}(i,j)$

Fig. 2: Prediction of the pixel $p_k^{[k,7]}(i,j)$.

After the predictor calculation, the inverse of $p_k^{[k,7]}(i,j)$ is computed as:

$$inv(i,j) = (p_k^{[k,7]}(i,j) + 2^{(7-k)}) \mod 2^{8-k} \quad (7)$$

Δ and Δ^{inv} are used to store the final results:

$$\begin{cases} \Delta = \left| pred(i,j) - p_k^{[k,7]}(i,j) \right| \\ \Delta^{inv} = \left| pred(i,j) - inv(i,j) \right| \end{cases} \quad (8)$$

If $\Delta \leq \Delta^{inv}$, the original bit value can be predicted correctly. Otherwise, there is an error during the prediction of the current pixel — this is noted by highlighting the PE location map $\mathbf{L_{loc}^k}$. In the paper, Puteaux and Puech [20] successfully processed all the bit-planes of an image recursively. The average payload of their recursive approach was approximately 2.4586 bpp when tested on the BOWS-2 dataset. To compare the results with our latest methods, we applied our schemas to the same benchmark dataset and achieved an average payload of 3.2457 bpp and 2.5325 bpp for the EMR-RDHEI and LMR-RDHEI, respectively. Along with the improved data embedding rate, our methods benefit from a vectorized processing and clean design.

Huang and Wang [9] proposed an RDHEI approach based on a specific encryption process. This technique is not based on the VRAE or RRBE schema. In the encryption process, they instead compute the prediction error using the MED predictor, which is able to predict its adjacent pixel values. Assuming the original image is an 8-bit grey-level image with size $M \times N$, with $p(i,j)$ representing the original image pixels, the MED prediction process can be summarized as follows:

$$pred(i,j) = \begin{cases} \min(p(i-1,j), p(i,j-1)) & \text{if case 1} \\ \max(p(i-1,j), p(i,j-1)) & \text{if case 2} \\ p(i-1,j) + p(i,j-1) - p(i-1,j-1) & \text{else} \end{cases} \quad (9)$$

Where $2 \leq i \leq M$ and $2 \leq j \leq N$.

Additionally, case 1 and case 2 are defined as the following:

$$\begin{cases} \text{case 1:} & p(i-1,j-1) \geq \max(p(i-1,j), p(i,j-1)) \\ \text{case 2:} & p(i-1,j-1) \leq \min(p(i-1,j), p(i,j-1)) \end{cases} \quad (10)$$

Thus, the prediction error e can be calculated as:

$$e(i,j) = p(i,j) - pred(i,j) \quad (11)$$

Overall, their approach achieved a lossless image recovery, yet their DER is suboptimal. The average DER tested on the UCID [25] dataset was 0.9392 bpp.

Malik et al. [14] employed a prediction-error estimation method to vacate space before embedding a message into an image. Let I be the original $M \times N$ grey-level image. They first decompose the image into four sub-images (I_1, I_2, I_3, I_4) by using Eq. 12.

$$\begin{cases} I_1(i,j) = I(2i-1, 2j-1) \\ I_2(i,j) = I(2i-1, 2j) \\ I_3(i,j) = I(2i, 2j-1) \\ I_4(i,j) = I(2i, 2j) \end{cases} \quad (12)$$

Where $i = [1, \lfloor M/2 \rfloor]$ and $j = [1, \lfloor N/2 \rfloor]$.

Next, $I_1(i,j)$ is used to predict the rest of the three sub-images noted as $P_{12}(i,j)$, $P_{13}(i,j)$, and $P_{14}(i,j)$:

$$P_{12}(i,j) = \begin{cases} \lceil (I_1(i,j) + I_1(i,j+1))/2 \rceil & \text{if } i < N/2 \\ I_1(i,j) & \text{if } i = N/2 \end{cases} \quad (13)$$

$$P_{13}(i,j) = \begin{cases} \lceil (I_1(i,j) + I_1(i+1,j))/2 \rceil & \text{if } i < M/2 \\ I_1(i,j) & \text{if } i = M/2 \end{cases} \quad (14)$$

$$P_{14}(i,j) = \begin{cases} \lceil (I_1(i,j) + I_1(i+1,j+1))/2 \rceil & \text{if } H_1(i,j) < H_2(i,j) \\ \lceil (I_1(i,j+1) + I_1(i+1,j))/2 \rceil & \text{if } H_1(i,j) \geq H_2(i,j) \\ \lceil (I_1(i,j) + I_1(i+1,j))/2 \rceil & \text{if } i < M/2, j = N/2 \\ \lceil (I_1(i,j) + I_1(i,j+1))/2 \rceil & \text{if } i = M/2, j < N/2 \\ I_1(i,j) & \text{if } i = M/2, j = N/2 \end{cases} \quad (15)$$

Then $H_1(i,j)$ and $H_2(i,j)$ are defined as:

$$\begin{cases} H_1(i,j) = \left| I_1(i,j) - I_1(i+1,j+1) \right| & (i < M/2, j < N/2) \\ H_2(i,j) = \left| I_1(i,j+1) - I_1(i+1,j) \right| & (i < M/2, j < N/2) \end{cases} \quad (16)$$

Afterwards, the prediction-error values of the sub-images are estimated as:

$$\begin{cases} PE_1(i,j) = I_1(i,j) \\ PE_2(i,j) = I_2(i,j) - PE_{12}(i,j) & i = 1,2,...,M/2 \\ PE_3(i,j) = I_3(i,j) - PE_{13}(i,j) & j = 1,2,...,N/2 \\ PE_4(i,j) = I_4(i,j) - PE_{14}(i,j) \end{cases} \quad (17)$$

After that, a location map is created to store prediction-error values' places to determine whether a particular pixel can be safely used for data embedding.

In Malik et al. [14]'s approach, they are able to recover an image losslessly. However, the DER of their approach was less than 0.75 bpp.

Our proposed schema employs a location map as an indicator to help us track redundancy in a given image. However, our location map generation process is performed in a completely different manner (via multi-MSB replacement). With our proposed location map, our methods achieve significantly higher embedding rates and high visual quality in restored images.

Yi and Zhou [35] proposed a method based on prediction-error encoding (PE-RDHEI) that uses a weighted checkerboard prediction (WCBP) to predict $3/4$ of the pixels in an original image with the help of the remaining $1/4$ of the pixels. Given an 8-bit $M \times N$ grey-level image I, they first separate the pixels into two categories I_1 and I_2. I_1 consists of the $MN/4$ pixels and I_2 contains the remaining $3MN/4$ pixels. The first step is to predict the pixels (Eq. 18) based on its four diagonal pixels (Fig. 3(a)). The second step is to predict the pixels (Eq. 19) by its four neighboring pixels (Fig. 3(b)):

$$X_p = \begin{cases} \lfloor 0.35 \times (X_{NW} + X_{SE}) + 0.15 \times (X_{NE} + X_{SW}) \rceil & \text{if } |X_{NW} - X_{SE}| < |X_{NE} - X_{SW}| \\ \lfloor 0.15 \times (X_{NW} + X_{SE}) + 0.35 \times (X_{NE} + X_{SW}) \rceil & \text{otherwise} \end{cases} \quad (18)$$

$$X_p = \begin{cases} \lfloor 0.35 \times (X_N + X_S) + 0.15 \times (X_W + X_E) \rceil & \text{if } |X_N - X_S| < |X_W - X_E| \\ \lfloor 0.15 \times (X_N + X_S) + 0.35 \times (X_W + X_E) \rceil & \text{otherwise} \end{cases} \quad (19)$$

X_{NW}		X_{NE}
	X_p	
X_{SW}		X_{SE}

(a) Diagonal pixels

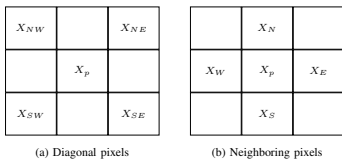|	X_N	
X_W	X_p	X_E
	X_S	

(b) Neighboring pixels

Fig. 3: WCBP examples.

Using the WCBP method, they realized a lossless image recovery with an average DER of 1.907 bpp. We compared our LMR-RDHEI with their method and obtained an average DER of 2.5325 when testing on a large dataset of 10,000 images.

In 2019, Liu et al. [11] published a fully reversible RD-HEI schema based on pixel-prediction techniques. They use gradient-adjust predictions to detect spare space in the original image and hide secret information inside it. Given the original image I of size $M \times N$, the predicted pixel value $p'(i,j)$ can be obtained from the original pixels $p(i,j)$, as explained in the following:

$$p'(i,j) = \begin{cases} p(i,j-1) & \text{if } d_v - d_h > 80 \\ p(i-1,j) & \text{if } d_v - d_h < -80 \\ (p(i-1,j) + p(i,j-1))/2 + \\ (p(i-1,j+1) + p(i-1,j-1))/4 & \text{otherwise} \end{cases}, \quad (20)$$

where d_v and d_h denotes the gradient change in the vertical and horizontal directions. They are defined in Eq. 21.

$$\begin{cases} d_v = |p(i,j-1) - p(i,j-2)| + |p(i-1,j) - p(i-1,j-1)| + \\ \quad |p(i-1,j) - p(i-1,j+1)| \\ d_h = |p(i,j-1) - p(i-1,j-1)| + |p(i-1,j) - p(i-2,j)| + \\ \quad |p(i-1,j+1) - p(i-2,j+1)| \end{cases}, \quad (21)$$

where $3 \leq i \leq M$ and $3 \leq j \leq N-1$.[1] The final results showed that their method could achieve an average DER of 0.95 bpp. One of the merits of their schema is realizing a lossless recovery among restored images.

Most recently, Wang and He [30] proposed an RDHEI method based on adaptive MSB prediction. The original image is first encrypted in a block-wise manner, so the pixels' correlation within the block can be preserved. In their prediction, the targeted image is divided into 2×2 non-overlapping blocks. Finally, they achieved a lossless original image recovery at an average DER of 2.26 bpp when testing on a selected set of 100 grey-level images from the BOWS-2 dataset.

In summary, the recently published methods discussed above have common shortcomings, such as sacrificing lower DER for higher reconstructed images' visual quality and vice versa. Our proposed methods achieved both higher DER and higher recovered images' visual quality, as illustrated in Section III and evaluated in Section IV.

III. PROPOSED MULTI-MSB REPLACEMENT METHODS

In our paper, we present two methods: EMR-RDHEI and LMR-RDHEI. With EMR-RDHEI, we are able to achieve high DER and high visual quality in restored images. With our LMR-RDHEI method, images are restored perfectly without any loss. The proposed multi-MSB replacement methods differ from the prediction-based approach employed by Puteaux and Puech [18, 20] and Puyang et al. [22].

We employ a one-to-one binary location map that tracks redundant information in the original image. Each pixel in the input image is assigned a label of 0 or 1 in the associated

---

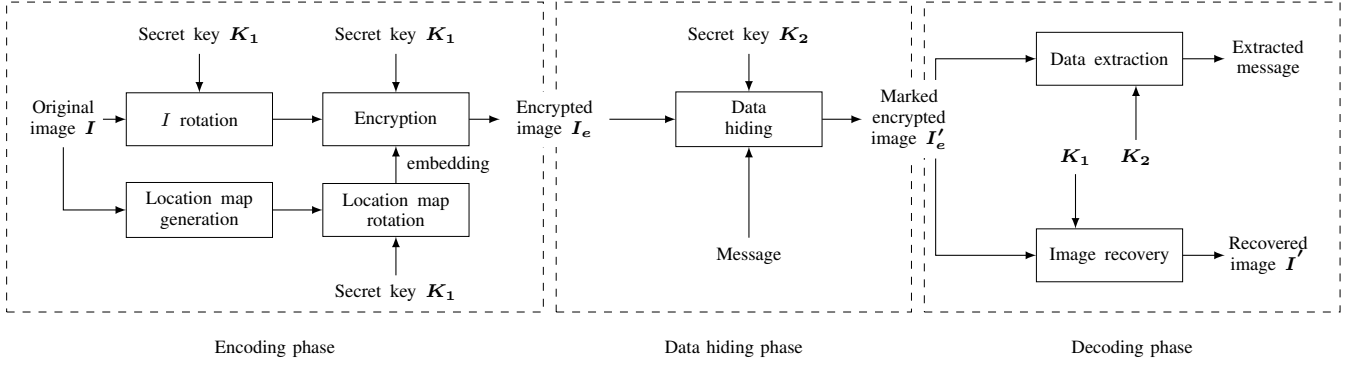[1]This is due to the prediction limitations as there are not enough surrounding pixels.

Fig. 4: General Schema of EMR-RDHEI Method.

binary location map depending on the multiple MSBs comparison. Instead of performing expensive numerical operations to predict the MSB value [18, 22, 20], we directly compare the MSBs of adjacent pixels to determine if they are identical and thereby consider them as redundant. By detecting and using these redundancies within the images' MSBs, we can embed large amounts of information with near-zero or zero loss of original data — depending on the method we apply.

In the proposed methods, we directly embed our assistant information into an encrypted image. In EMR, the assistant information is the location map. In LMR, it contains both the location map and the first MSB map[2]. Once the assistant information is generated, it may contain partial textures from the original image. Thus, we apply the rotation procedure to the original image and assistant information to break potential patterns. This process further enhances the overall data security. More details are presented in Section IV-A.

Since modern computers can perform bit comparison operations much faster than other numerical computations, e.g., division and multiplication, our algorithms can be executed more efficiently than many other state-of-the-art methods. The number, b, of MSBs (b-MSBs) used in our methods is heavily dependent on an image's texture. In our proposals, two or more MSBs ($b \geq 2$) of a pixel are used for storing data through redundancy detection when considering adjacent pixels. This way, our methods achieve high embedding capacity and low/no data loss.

A. EMR-RDHEI method

The general schema of our EMR-RDHEI method, shown in Fig. 4, includes three phases: encoding, data hiding, and decoding. The encoding phase in EMR-RDHEI is composed of five steps: location map generation, location map rotation, original image rotation, original image encryption, and location map embedding. The rotation and encryption subprocesses use the secret key K_1 to rotate the original image/location map and encrypt the original image. In the data hiding phase, we employ the multi-MSB replacement technique to embed a secret message into the encrypted image's redundant pixel bits. After obtaining the marked encrypted image (encrypted image with the secret message) in the

decoding phase, the hidden message is extracted and recovered without any errors, and the original image is reconstructed. Moreover, the EMR-RDHEI method is separable, meaning that we can perform the data extraction and image reconstruction independently. During all phases, the images are processed in a vectorized sliding window manner.

1) Encoding phase:

a) Optimal location map generation: An essential part of our proposed methods is generating an optimal location map for an original image I that finds the best trade-off between parameter b and redundant pixels quantity. Before embedding a secret message into an image, we identify the redundant pixels whose multiple MSBs (b-MSBs) can be replaced and used for data embedding.

As shown in Fig. 5, the original image I of size $M \times N$ is expanded via a new dimension (noted as B), resulting in the expanded image Φ of size $(M \times N \times B)$. This dimensional expansion allows the parallel comparison of multiple b parameters by gliding matrix Ω ($B \times M$) along the N dimension. Ω, a matrix that tracks the last non-redundant MSBs, is used to identify changes in MSBs from pixel to pixel. The detected redundancies via the MSBs comparisons are then stored in the location maps $\mathcal{L}$ for all possible b parameters ($b \in [2, 7], b \in \mathbf{Z}$). Once the entire image is processed, the DER of each generated location map is individually calculated; thus, we can select the location map with the highest DER for the input image.
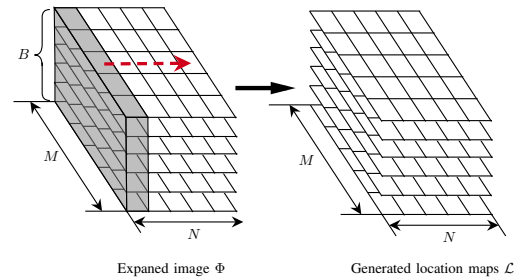

Fig. 5: Location map generation process.

At the beginning of the generation process, Ω is initialized based on the first column of Φ, and the first column of $\mathcal{L}$ is

---

[2]This is extracted from the first MSB bit-plane of an image.

set with all 1s, defined as follows:

$$\Omega(i,k) = \Phi(i,0,k) \tag{22}$$

$$\mathcal{L}(i,0,k) = 1 \tag{23}$$

As Ω glides along the N dimension, $\mathcal{L}$ is generated based on the values of Φ and Ω. If their $[1, k+1]$ MSBs are equal, then $\mathcal{L}(i,j,k)$ is marked as 0 (redundant); otherwise, $\mathcal{L}(i,j,k)$ is marked as 1 (non-redundant). After each Ω and Φ comparison, Ω is consistently updated based on the newest non-redundant MSBs. Eq. 24 and Eq. 25 elaborate on the updating operations executed while gliding Ω along the N dimension.

$$\mathcal{L}(i,j,k) = \mathbb{I}[\Phi^{[1,k+1]}(i,j,k) = \Omega^{[1,k+1]}(i,k)], j \in [1, N-1], j \in \mathbf{Z} \tag{24}$$

$$\Omega(i,k) := \begin{cases} \Phi(i,j,k) & \text{if } \mathbb{I}[\mathcal{L}(i,j,k) = 1], j \in [1, N-1], j \in \mathbf{Z} \\ \Omega(i,k) & \text{otherwise} \end{cases} \tag{25}$$

Where $\mathbb{I}$ is the indicator function; i, j, and k are the indices of the tensor M, N, and B, respectively.

To find the most optimal location map, we calculate the DER of each location map stacked in $\mathcal{L}$. We select the location map along with the parameter b with the highest DER. The process of selecting the optimal b is interpreted as Eq. 26:

$$b = [\operatorname*{argmax}_{k} \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \mathbb{I}(\mathcal{L}(i,j,k) = 0) \times b] + 2 \tag{26}$$

Where $k \in [0,5]$, $b \in [2,7]$, $k \in \mathbf{Z}$, $b \in \mathbf{Z}$, and $b = k + 2$.

To further elaborate on the process of generating a location map, we go through the procedure of constructing only one location map l with b set to 4 (shown in Fig. 6). As mentioned in the initialization process, the first column of l is set to all 1s, and ω is set to the first column of I. As the ω vector (red arrow in Fig. 6) glides along the image width N, comparisons with the b-MSBs of ω and I are performed. If redundancy is detected, i.e., no change in MSBs, we mark that pixel as a 0 in l; otherwise, we mark it as a 1. When the location map is complete, the image's redundancy is used to embed a secret message by replacing their tracked redundant MSBs.
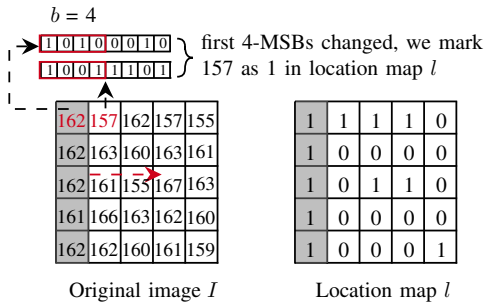

Fig. 6: Location map l generation ($b = 4$).

b) Secret key generation: After obtaining the most optimal location map, the content owner needs to select or generate an encryption key (K_1) to encrypt their original image. Either 2D-LSCM (2D Logistic-Sine-coupling map [8]) function or chaotic generator (based on the Piecewise Linear Chaotic Map [38]) can be used to generate a sequence of pseudo-random bytes. By using the keystream generator, a long sequence of generated bytes $s(i,j)$ is constructed and reshaped into

the input image's size. In our proposed encryption schema, the only requirement is to use a cryptographically secure stream cipher while encrypting. This high-quality cipher can be acquired not only by using the methods mentioned above but through RC4 and its variants suggested in [24, 27, 1, 28].

c) Rotation and image encryption: As mentioned earlier, the generated keystream s is the same size as the original image. Before image encryption, s, I, and l are divided into square blocks from sizes $\{2^1 \times 2^1, ..., 2^n \times 2^n\}$ - where $n = \min(\lfloor M/2 \rfloor, \lfloor N/2 \rfloor)$. If the image size and the generated sequence s cannot be fully divided into the excepted blocks, we retain the remaining pixels and do not rotate them. After partitioning the image into square blocks, the summation of each individual square in s is calculated. Afterwards, modulus 4 is applied to the summation value to determine the rotation angle for each block. The mapping of each resulting value and the corresponding rotation angle is listed in the following: $\{0 : 0°, 1 : 90°, 2 : 180°, 3 : 270°\}$. The rotation process is expressed in Eq. 27 and is demonstrated in Fig. 7:

$$\sum_{i=C_1 2^n}^{(C_1+1)2^n} \sum_{j=C_2 2^n}^{(C_2+1)2^n} s(i,j) \mod 4 = \begin{cases} 0, & \text{no rotation on } I \text{ and } l \\ 1, & \text{rotate } I \text{ and } l \text{ 90 degree} \\ 2, & \text{rotate } I \text{ and } l \text{ 180 degree} \\ 3, & \text{rotate } I \text{ and } l \text{ 270 degree} \end{cases} \tag{27}$$

Where $C_1 \in [0, \lfloor N/2^n \rfloor], C_1 \in \mathbf{Z}, C_2 \in [0, \lfloor M/2^n \rfloor], C_2 \in \mathbf{Z}$, and $n \in \mathbf{Z}$.
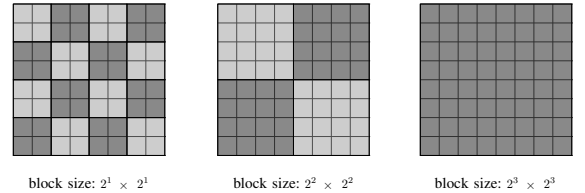

Fig. 7: Block sizes for an image of size 8×8.

After the rotation, we encrypt the rotated original image using the stream cipher s through exclusive-or (XOR) operation, and then we obtain the encrypted image I_e:

$$I_e(i,j) = s(i,j) \oplus I(i,j) \tag{28}$$

d) Location map embedding: The next step is to embed the rotated location map directly into the LSBs of I_e. Among all those altered LSBs, there is a 50% chance of a pixel's LSB being altered since the rotated location map and LSBs only contain 0s and 1s. As a result, the theoretically expected PSNR value of a reconstructed image using EMR-RDHEI is approximated as follows:

$$\text{PSNR} \approx 10 \times \log_{10} \left(\frac{255^2}{1/2} \right) \approx 51.1411 \, dB \tag{29}$$

2) Data hiding phase: In the data hiding phase, the data hider shall not access the original image content I nor the secret key K_1. First, the data hider extracts the rotated location map using the LSBs from the encrypted image I_e. After that, the data hider uses the secret key K_2 to encrypt the secret information. Then, the data hider's encrypted secret message

is embedded using the pixels marked as 0 (redundant) by the location map into I_e. Thus, those redundant pixels' b-MSBs are replaced (shown in Fig. 8) and used for embedding data bits. Note that the pixels marked with a label of 1 in the location map cannot be changed. Since they remain unchanged in this process, they can be used as indicators during the reconstruction of the altered pixels' b-MSBs in the decoding phase.
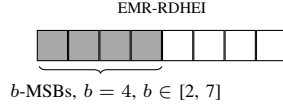
EMR-RDHEI

b-MSBs, $b = 4$, $b \in [2, 7]$

Fig. 8: Example of EMR-RDHEI data hiding allocation.

3) Decoding phase: In the decoding phase, the receiver obtains the marked encrypted image I_e' and scans it in the same sliding fashion as previously mentioned. The receiver reconstructs the original image and/or extracts the hidden message depending on the available keys. Since our EMR-RDHEI method is separable, data extraction and image reconstruction can be performed independently. Using this method, we are able to recover the error-free secret message using the secret key K_2 and/or reconstruct the image I' with low loss while using secret key K_1. There are three scenarios based on the key(s) that the receiver owns, which are discussed in the following.

a) The receiver has only the image encryption key K_1: In this situation, the receiver can only recover the original image without deciphering the embedded secret message. With EMR-RDHEI, the receiver first extracts the location map l from the LSBs of the marked encrypted image. Afterwards, the receiver decrypts the image using an XOR operation. Since the original image and location map are rotated before encryption, the next step is to perform the same sequence of rotations but in the opposite order. Similar to the encoding phase, we partition the 2D sequence s, marked decrypted image, and location map l into certain blocks (depending on the block sizes used in the encoding phase). Unlike in Section III-A1c, we use reverse order block partitioning and apply the opposite rotation manner to de-rotate the previously rotated images. After calculating the summation of the blocks and applying modulus 4 to them, the rotation angles to undo the initial rotations are determined. The mapping of each remainder and inverse rotation angles is defined as: $\{0 : 0°, 1 : 270°, 2 : 180°, 3 : 90°\}$. The de-rotation process equation is the reverse of Eq. 30 and is expressed as follows:

$$\sum_{i=C_1 2^n}^{(C_1+1)2^n} \sum_{j=C_2 2^n}^{(C_2+1)2^n} s(i,j) \mod 4 = \begin{cases} 0, & \text{no rotation on } I \text{ and } l \\ 1, & \text{rotate } I \text{ and } l \text{ 270 degree} \\ 2, & \text{rotate } I \text{ and } l \text{ 180 degree} \\ 3, & \text{rotate } I \text{ and } l \text{ 90 degree} \end{cases}$$

(30)

Where $C_1 \in [0, \lfloor N/2^n \rfloor], C_1 \in \mathbf{Z}, C_2 \in [0, \lfloor M/2^n \rfloor], C_2 \in \mathbf{Z}$, and $n \in \mathbf{Z}$.

After rotating the marked decrypted image and location map l, the original image I is reconstructed by using l. The reconstruction process starts with initializing the ω vector as it copies the image's first column pixels' b-MSBs. This is done because the first column of the image is always labeled as 1 (non-redundant) in l. ω keeps track of the last non-redundant b-MSBs. The 1s (non-redundant) in the location map act as markers that help restore the b-MSBs of its following columns' pixels labeled as 0 (redundant) in l. By gliding ω along the width of l and updating its value, we can quickly reconstruct I. This process is similar to the l location map generation; however, this time, instead of identifying and annotating redundant pixels' b-MSBs, we reconstruct the redundant pixels' b-MSBs through the use of non-redundant pixels' b-MSBs. Following the procedures mentioned above, the original image can be reconstructed.

b) The receiver has only the data hiding key K_2: For this case, the receiver can only decipher the embedded secret message without reconstructing the original image. In order to decipher the embedded secret message, the receiver restores the location map l from the LSBs of I_e'. There is no need to undo the rotations (Eq. 30) in this scenario because the secret message is embedded after the rotations. Then, the receiver concatenates all the b-MSBs of the pixels labeled with a 0 in l and decrypts the concatenated bits using the secret key K_2 to obtain the plain message. With the proposed EMR-RDHEI method, we achieve error-free plain message recovery.

c) The receiver has both keys, K_1 and K_2: In the last scenario, the original image and plain message can be recovered by following the procedures described above.

B. LMR-RDHEI method

The general schema of our proposed LMR-RDHEI method is demonstrated in Fig. 9. The encoding phase in LMR-RDHEI includes the following procedures: (1) location map and first MSB map generation, (2) location map and first MSB map compression, (3) original image, location map, and MSB map rotation, (4) original image encryption, and (5) compressed maps embedding. In both the first MSB map and location map's compression processes, we employ the JBIG-KIT lossless image compression library [15]. The original image, first MSB map, and location map use secret key K_1 to rotate. The encryption of the original image also uses secret key K_1 to encrypt.

In the data hiding phase, we apply the multi-MSB replacement technique to embed a secret message into the encrypted image's redundant pixels. Since the first MSB bit-plane stores the bits of the compressed rotated first MSB map and compressed rotated location map, we do not insert any other bit information into the first MSB bit-plane. Thus, compared with the EMR-RDHEI method, the overall embedding rate slightly decreases. At the final decoding phase, the hidden message is extracted from the marked encrypted image without any errors, and the original image is recovered losslessly. Similar to the EMR-RDHEI method, the LMR-RDHEI method is separable, meaning that the data extraction and image recovery processes can be done independently. The LMR-RDHEI method is also processed in a vectorized sliding window manner.
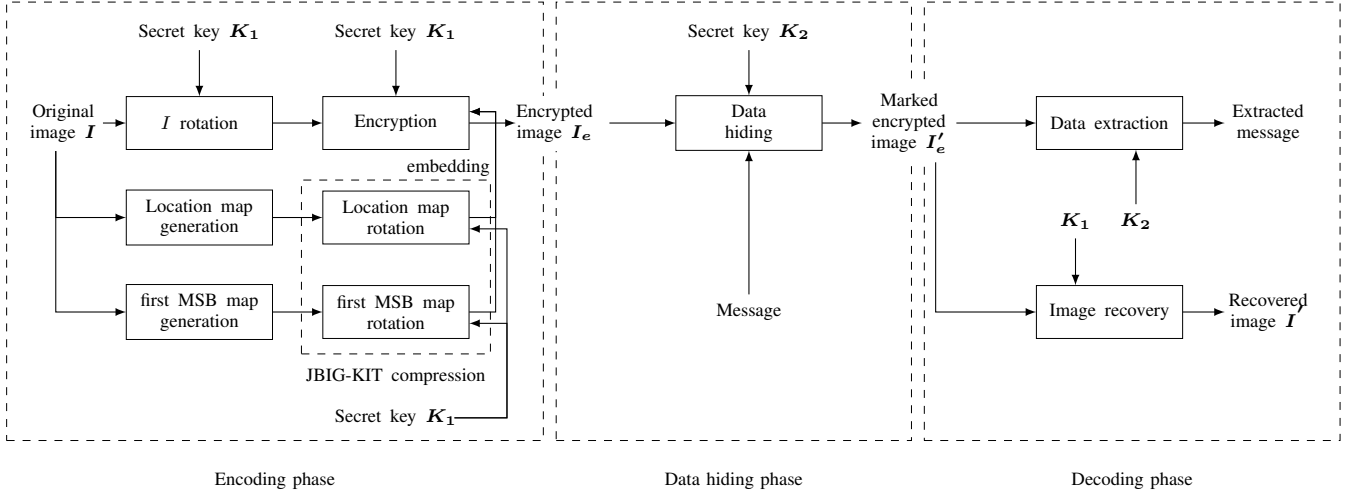
1) Encoding phase:

Fig. 9: General Schema of LMR-RDHEI Method.

a) Maps generation, rotation, and compression: First, the most optimal location map l is generated with a process similar[3] to the procedure we described in Section III-A1a. Then, the first MSB map, which is extracted from the first MSB bit-plane, can be expressed as the following:

$$\eta(i,j) = (I(i,j) \wedge 128) \mod 127 \qquad (31)$$

After generating the most optimal location map l and the first MSB map η, we rotate them (along with the original image) according to the secret keystream s generated by K_1. Note that the secret key generation process is the same one illustrated in Section III-A1b. The overall rotation procedure is also similar to the steps we specified in Section III-A1c. For the LMR-RDHEI method, we take the JBIG-KIT compression efficiency and future data embedding rate into consideration when choosing block sizes. The rotation procedure can negatively impact the JBIG-KIT compression efficiency because the shuffling lowers the correlation between new neighboring pixels. To further explain, the following example is provided. For images of size 512×512, instead of using the block sizes of $\{2^1 \times 2^1, 2^2 \times 2^2, ... , 2^9 \times 2^9\}$, block sizes of $\{2^4 \times 2^4, 2^5 \times 2^5, ... , 2^9 \times 2^9\}$ are adopted primarily to gain a better compression rate by JBIG-KIT. By using these larger block sizes, a comparably good compression rate can be achieved, a high message embedding rate is maintained, and the images remain secure. For more details, experiments regarding different block sizes will be demonstrated in Section IV-A. In this rotation process, if the compressed maps' size exceeds the size of the first MSB bit-plane (512×512 bits), we generate the next most optimal location map l until both maps can be sufficiently compressed to fit in the first MSB bit-plane collectively. We choose to alter the location map because the compressed first MSB map is fixed.

b) Maps embedding and image encryption: In contrast to the previously mentioned EMR-RDHEI method, LMR-RDHEI can realize a lossless recovery of the original image. EMR-RDHEI stores the location map into LSBs of an encrypted

---

[3]$b \in [2, 7]$ in EMR-RDHEI, and $b \in [2, 8]$ in LMR-RDHEI.

image, and it will cause a slight bit-level loss in the restored image. LMR-RDHEI concatenates the compressed first MSB and location maps then inserts them back into the first MSB bit-plane of an encrypted image to prevent any potential bit losses. After the rotated image is encrypted using the generated secret key K_1 with Eq. 28, we embed both maps into the encrypted image.

2) Data hiding phase: The first MSB bit-plane cannot be modified because it holds essential information of the compressed first MSB map and location map; therefore, it is preserved for the decoding phase. The data hider extracts the compressed location map from the first MSB bit-plane and then decompresses it using the JBIG-KIT. The data hider then hides the secret bits into the pixels labeled with a 0 in the location map. Like EMR-RDHEI, the pixels labeled with a 1 cannot be used for embedding any secret information. These pixels serve as indicators that are essential for recovering the original $(b\text{-}1)$ MSB bits in those altered pixels. As for pixels marked with a label of 0 in the location map, $(b\text{-}1)$ out of b bits can be replaced and used for embedding a secret message. For example, if $b = 4$, only 3 of 4 bits (excluding the first bit) in a pixel can hide secret information (shown in Fig. 10).
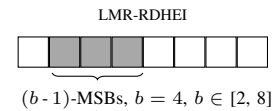

Fig. 10: Example of LMR-RDHEI data hiding allocation.

3) Decoding phase: Similar to the aforementioned EMR-RDHEI method, LMR-RDHEI is separable — the data extraction and image recovery processes can be performed individually. A receiver can extract the hidden message and/or reconstruct the original image depending on the available secret key(s). Unlike EMR-RDHEI, not only can LMR-RDHEI recover error-free secret message with secret key K_2, but it can also realize a lossless image recovery. There are three scenarios in the decoding phase, which are presented as follows.

a) The receiver has only the image encryption key K_1: For recovering the original image losslessly only, the receiver can first extract the compressed location map l and the compressed first MSB map η from the first MSB bit-plane of the marked encrypted image. After decompressing both maps with JBIG-KIT, the receiver uses the secret key K_1 to decrypt the image with an XOR operation and obtain the decrypted marked image. Since the two maps and the original image are rotated before encryption, the next step is to de-rotate both maps and the decrypted marked image with the secret key K_1. The process for de-rotating these maps and images is performed in the same manner as the one described in Section III-A3a with the only minor difference of adding the de-rotation process to the first MSB map in LMR-RDHEI. After obtaining the de-rotated marked decrypted image, l, and η, the decompressed de-rotated η is used to recover the original first MSB bit-plane. In the end, the original image I is restored losslessly by decompressing the de-rotated l and later recovering the $(b$-1$)$ MSBs with its guidance. The $(b$-1$)$ MSBs recovery is similar to EMR-RDHEI's b MSBs recovery process described in the latter part of Section III-A3a. The only difference is that the location map l is used to restore the $(b$-1$)$ MSBs instead of the b MSBs.

b) The receiver has only the data hiding key K_2: For recovering the plain message only, the receiver first extracts the compressed location map from the first MSB bit-plane. Afterwards, JBIG-KIT is applied to decompress the location map. Similar to EMR-RDHEI, there is no de-rotation (Eq. 30) in this procedure. The next step is to concatenate all the $(b$-1$)$-MSBs of pixels labeled as 0 in the corresponding location map. Finally, the concatenated bits are decrypted using the data hiding key K_2. Thus, the plain message is obtained. Note that our LMR-RDHEI method can realize error-free plain message recovery.

c) The receiver has both keys, K_1 and K_2: For the last case, the receiver can follow the steps described above for recovering the original image and the hidden message, resulting in a lossless image and a error-free plain message.

IV. EXPERIMENTAL RESULTS

In this section, we present experimental results to demonstrate and support our proposed methods. In Section IV-A, a detailed example is provided: the grey-level image Lena of size 512×512 using EMR-RDHEI and LMR-RDHEI. After that, 10,000 images with various textures from the BOWS-2 database [2] are used to measure the general efficiency of our methods. The security analyses of our methods are presented in Section IV-B. Finally, a comparison of our methods with many other state-of-the-art schemas is shown in Section IV-C. In Sections IV-B and IV-C2, PSNR and SSIM are used to measure the similarity between target samples.

Throughout our experiments, the samples used are standard grey-level images of size 512×512, which are listed in Fig. 11. Additionally, for testing these images, the chosen block sizes for EMR-RDHEI are $\{2^1 \times 2^1, 2^2 \times 2^2, ..., 2^9 \times 2^9\}$. As mentioned in Section III-A1a, for images of size 512×512, the block sizes of $\{2^4 \times 2^4, 2^5 \times 2^5, ... , 2^9 \times 2^9\}$ are chosen for LMR-RDHEI to ensure JBIG-KIT's compression efficiency, maintain a high embedding rate, and secure the overall information.
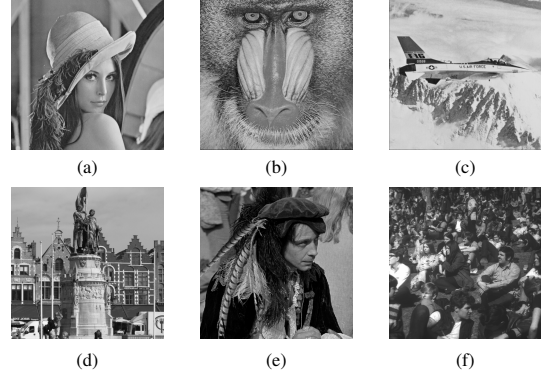

Fig. 11: Standard grey-level images of size 512×512: a) Lena; b) Baboon; c) Airplane; d) Knight; e) Man; f) Crowd.

A. Detailed examples

We thoroughly demonstrate the performance of EMR-RDHEI and LMR-RDHEI by applying them to the grey-level image Lena of size 512×512. Results are shown in Fig. 12 and Fig. 13. Additionally, to measure the general efficiency of both EMR-RDHEI and LMR-RDHEI, our methods are tested using 10,000 grey-level images with various textures from BOWS-2 database [2]. In our experiments, we compute the minimum, maximum, and average values of DER, PSNR, and SSIM. Furthermore, as mentioned earlier in Section III-B1, we test LMR-RDHEI with different block sizes. The results are summarized in Table I and Table II. In Table II, DER indicates the average embedding rate. When testing LMR-RDHEI, good cases mean both the first MSB map and location map are successfully compressed to fit in the first MSB bit-plane. In contrast, bad cases imply that both maps cannot be compressed sufficiently; therefore, the image cannot be used to hide any secret message using LMR-RDHEI.

	DER (bpp)	PSNR (dB)	SSIM
maximum	6.2682	51.1723	0.9997
average	3.2457	51.1409	0.9958
minimum	1.2087	51.1108	0.9746

TABLE I: Results base on EMR-RDHEI method.

block sizes	DER (bpp)	good cases	bad cases
$\{2^4 \times 2^4, ..., 2^9 \times 2^9\}$	2.5325	99.99%	0.01%
$\{2^3 \times 2^3, ..., 2^9 \times 2^9\}$	2.5118	99.98%	0.02%
$\{2^2 \times 2^2, ..., 2^9 \times 2^9\}$	2.4737	99.93%	0.07%
$\{2^1 \times 2^1, ..., 2^9 \times 2^9\}$	2.4402	99.86%	0.14%

TABLE II: Testing different block sizes on the BOWS-2.

To demonstrate the dynamic range of embedding rates using both methods, we sample 10,000 images from the BOWS-2
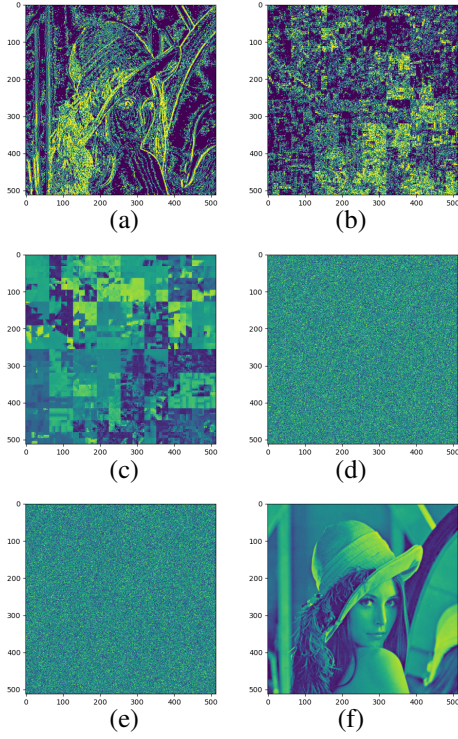
Fig. 12: With our EMR-RDHEI method, the embedding rate was 2.6566 bpp. a) The most optimal generated location map l, when $b = 4$; b) The location map after rotation; c) The original image Lena I after rotation; d) Encrypted image Lena I_e; e) Marked encrypted image Lena $I_e^{'}$; f) Reconstructed image $I^{'}$, PSNR = 51.1356 dB, SSIM = 0.9928.

dataset and plot the DER histograms in Fig. 14(a) and Fig. 14(b) for EMR-RDHEI and LMR-RDHEI. The average embedding rates were 3.2457 bpp and 2.5325 bpp, respectively.
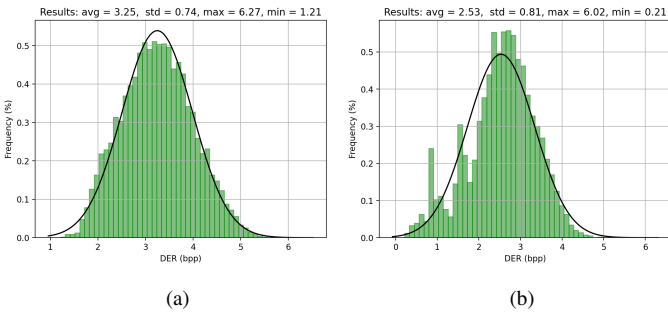

Fig. 14: DER results of the total 10,000 samples from the BOWS-2 database. DER analysis for (a) EMR-RDHEI and (b) LMR-RDHEI methods.

B. Security analysis

In this subsection, we use some well-known statistical metrics, including Shannon entropy, χ^2 test, number of changing pixel rate (NPCR), and unified averaged changed intensity (UACI) tests to analyze the security level of our proposed
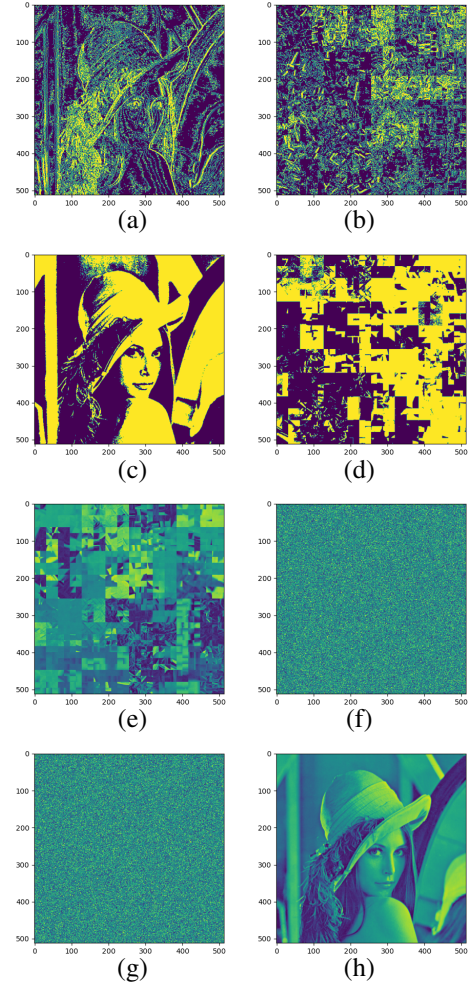

Fig. 13: With our LMR-RDHEI method, the embedding rate was 1.9925 bpp. a) The most optimal generated location map l, when $b = 4$; b) The location map after rotation; c) The extracted first MSB map before rotation; d) The first MSB map after rotation; e) The original image Lena I after rotation; f) Encrypted image Lena I_e; g) Marked encrypted image Lena $I_e^{'}$; h) Reconstructed image $I^{'}$, PSNR $\rightarrow +\infty$, SSIM = 1.

methods. Formal definitions for each metric are given below, followed by testing results.

a) Shannon entropy: This measures the probability distribution of different pixel values in an image. A higher Shannon entropy value indicates that the pixel values in an image are distributed more uniformly within an allowable range. The Shannon entropy, $H(I)$, of an image I is defined as follows:

$$H(I) = -\sum_{i=0}^{255} P(\alpha_i) \log_2(P(\alpha_i)) \tag{32}$$

$P(\alpha_i)$ represents the probability of a pixel value α_i ($0 \leq \alpha_i \leq 255$) in each grey level.

b) χ^2 test: It is an indicator of divergence in a grey-level image I from its theoretical counterpart in which all

Images	Methods		Entropy	χ^2 test	NPCR (%)	UACI (%)	PSNR (dB)
Lena	EMR-RDHEI/LMR-RDHEI	I	7.4456	340.6470	/	/	/
		I_e	7.9994/7.9994	15.0197/14.8271	99.6025/99.5853	28.7115/28.7276	9.2217/9.2041
		I_e'	7.8583/7.9993	169.1861/15.4492	99.6250/99.6044	28.6808/28.6693	9.2173/9.2261
	Malik et al. [14]	I	7.4456	340.6470	/	/	/
		I_e	7.9994	14.6164	99.6223	28.6375	9.2247
		I_e'	7.9993	15.7695	99.6262	28.6367	9.2265
Baboon	EMR-RDHEI/LMR-RDHEI	I	7.3579	396.0794	/	/	/
		I_e	7.9992/7.9993	16.8684/16.2893	99.6181/99.6216	27.8875/27.9002	9.5132/9.5074
		I_e'	7.8588/7.9994	100.6018/14.2764	99.6204/99.6162	28.0922/27.8928	9.4575/9.5106
	Malik et al. [14]	I	7.3579	396.0794	/	/	/
		I_e	7.9992	16.1595	99.6155	27.8212	9.5270
		I_e'	7.9993	15.7457	99.6143	27.8899	9.5194
Airplane	EMR-RDHEI/LMR-RDHEI	I	6.6776	826.2035	/	/	/
		I_e	7.9993/7.9993	16.3967/16.4838	99.5983/99.5991	32.3403/32.3001	8.0536/8.0641
		I_e'	7.9351/7.9993	115.1221/15.9512	99.6048/99.6201	32.3150/32.3569	8.0593/8.0504
	Malik et al. [14]	I	6.6776	826.2035	/	/	/
		I_e	7.9993	16.4092	99.6101	32.2851	8.0674
		I_e'	7.9993	16.1689	99.6132	32.3564	8.0516
Knight	EMR-RDHEI/LMR-RDHEI	I	7.3227	661.7007	/	/	/
		I_e	7.9992/7.9992	16.1288/16.6209	99.6021/99.6212	30.1702/30.1601	8.7040/8.7064
		I_e'	7.9574/7.9993	76.7231/15.4173	99.6216/99.5953	30.1915/30.1196	8.7002/8.7175
	Puteaux and Puech [18] (CPE-HCRDH/EPE-HCRDH)	I	7.3227	668.628	/	/	/
		I_e	7.9994/7.9994	14.8342/14.8806	99.6143/99.6136	30.1338/30.1344	8.7081/8.7081
		I_e'	7.9994/7.9994	15.1188/14.8299	99.6082/99.6059	30.1521/30.1569	8.7069/8.7039

TABLE III: Security evaluation on classic images with our proposed methods.

pixels occur with an equal probability of $1/256$. χ^2 can be computed as follows:

$$\chi^2 = 256 \cdot (h \times w) \sum_{i=0}^{255} \left(P(\alpha_i) - \frac{1}{256} \right)^2 \quad (33)$$

c) NPCR and UACI analysis: These two tests are typical quantities used to evaluate the strength of an image encryption algorithm against differential attacks. Conventionally, a higher NPCR/UACI score is interpreted as a stronger resistance to differential attacks [32].

$$NPCR = \frac{1}{h \times w} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \sigma(i,j) \times 100\%, \quad (34)$$

where:

$$\sigma(i,j) = \begin{cases} 1, & \text{if } I(i,j) = I'(i,j) \\ 0, & \text{otherwise} \end{cases} \quad (35)$$

and

$$UACI = \frac{1}{h \times w} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \frac{|I(i,j) - I'(i,j)|}{255} \times 100\%, \quad (36)$$

where $I(i,j)$ represents the pixels in the original grey-level image and $I'(i,j)$ represents the pixels in the reconstructed image.
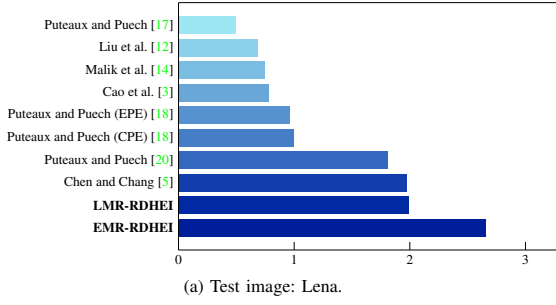
In Table III, I, I_e, and I_e' represent the original image, encrypted image, and marked encrypted image, respectively. For the statistical security analysis, our results show that the sensitive information is undetectable in the encrypted image and marked encrypted image using our proposed methods. As shown in Table III, the similarity (in terms of PSNR) between an original image and an encrypted image/marked encrypted image is very low (approximately 9 dB). Furthermore, if part of the secret message is modified by an attacker, as the message is being encrypted with secret key K_2, they cannot be decrypted and exploited for validation in the later decoding phase. If an attacker attempts to modify b-MSBs in the pixels labeled with a 1 in the location map, significant noise will be introduced into the reconstructed image. Note that we also need to de-rotate the original image and the location map in both of our methods to perform image reconstruction. Thus, the original image content cannot be obtained, and these additional processes further secure the confidential information. Moreover, without the encryption key K_1, the marked encrypted image cannot be decrypted, de-rotated, and recovered.
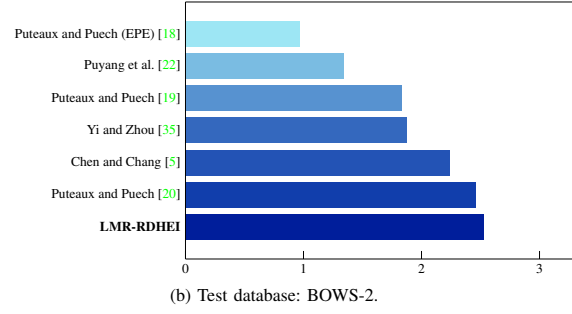
C. Comparison with related approaches

More comparisons of our proposed EMR-RDHEI and LMR-RDHEI methods with many other state-of-the-art approaches are demonstrated in this subsection. These include DER comparisons and performance comparisons. The test samples used in the DER and performance comparisons were previously shown in Fig. 11.

1) DER comparison: In order to demonstrate a straightforward embedding rate comparison between our proposed methods and other recent methods published by [3, 17, 12, 18, 14, 5, 20, 36], we present the maximum DERs of the image Lena and the average DERs of 10,000 images from BOWS-2 image dataset in Fig. 15.

(a) Test image: Lena.

(b) Test database: BOWS-2.

Fig. 15: Comparisons of (maximum/average) DERs with other state-of-the-art methods on image Lena and BOWS-2 database.

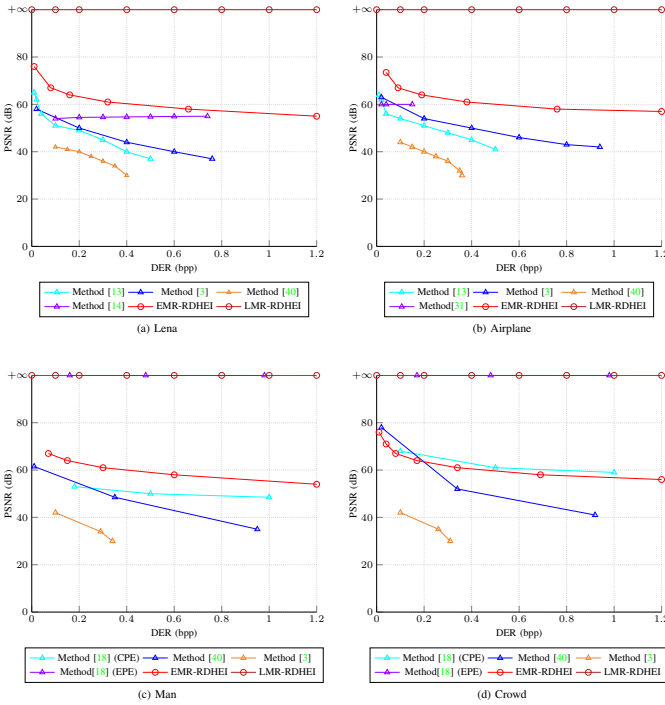
(a) Lena

(b) Airplane

(c) Man

(d) Crowd

Fig. 16: Performance comparison with related works.

The maximum DERs obtained by [3, 17, 12, 18, 14, 5, 20] are shown in Fig. 15(a). Most methods are not able to achieve a maximum embedding rate higher than 2 bpp. In the most recent research published by [14, 5, 20], their best DERs underperform our results: 2.6566 bpp and 1.9925 bpp for EMR-RDHEI and LMR-RDHEI, respectively. As seen in Fig. 15(a), our data hiding rate of EMR-RDHEI is significantly higher than the recently published works by Puteaux and Puech [20], Chen and Chang [5], Puteaux and Puech [18].

In Fig. 15(b), we compare the average embedding rate of LMR-RDHEI with other lossless image recovery methods: [18, 22, 19, 35, 5, 20]. The average DERs produced by [18, 22, 19, 35] are all less than 2 bpp. In 2018, Puteaux and Puech [18] proposed their EPE-HCRDH approach, which yielded an average DER of 0.968 bpp. Later, Puyang et al. [22] employed a two-MSB prediction schema and obtained an average DER of 1.346 bpp. In comparison to recently published works by [5, 20], the average DER result of 2.5325 bpp

from our LMR-RDHEI method is a substantial improvement compared with previously published methods.

2) Performance comparison: In this section, we use the following standard grey-level images of size 512×512: Lena, Airplane, Man, and Crowd to perform the performance comparison. As shown in Fig. 16, the bit-level differences between the reconstructed images and original images are measured using PSNR and used to compare our results with current state-of-the-art methods [13, 3, 40, 14, 31, 18].

As can be seen from our results, our maximum embedding rates outperform previously published schemas. For images Lena, Airplane, Man, and Crowd, the maximum DER using EMR-RDHEI was 2.6566 bpp, 3.0574 bpp, 2.3917 bpp, 2.7504 bpp, respectively; and with LMR-RDHEI, we achieved the maximum DER of 1.9925 bpp, 2.4459 bpp, 1.5118 bpp, 2.2003 bpp, each to each. For LMR-RDHEI, a lossless image recovery can be realized in all testing samples. None of the aforementioned methods achieve the same high embedding rate while maintaining a high PSNR between restored and original images. When we test similar embedding rates (such as 0.5 bpp and 1.2 bpp) compared with other recent approaches, it is clear that the PSNRs obtained by our methods are close to or even higher than state-of-the-art works. In the performance comparisons of images Lena and Man, our LMR-RDHEI method outperforms the lossless recovery EPE-HCRDH method published by Puteaux and Puech [18] in terms of DER.

To summarize, both of our proposed methods simultaneously provide error-free message extraction and, to the best of our knowledge, have one of the highest performances in tests regarding the data embedding rate and image reconstruction quality found in the RDHEI literature. Our proposed methods, EMR-RDHEI and LMR-RDHEI, allow a better trade-off between data hiding rate and visual quality in restored images thanks to the multi-MSB replacement approach.

V. CONCLUSION

In this paper, we propose two multi-MSB replacement-based RDHEI methods: EMR-RDHEI and LMR-RDHEI. Although the reconstructed images from the EMR-RDHEI method slightly differ from their original contents, in practice, the difference in their visual quality is indistinguishable since only the LSB of some pixels changed. On the other hand, our proposed LMR-RDHEI method is able to recover the

original images perfectly without any loss. Through extensive experiments presented in Section IV, we show that our algorithms outperform many other current state-of-the-art methods in both embedding rate and PSNR. Our novel strategy includes the multi-MSB replacement and the image block division/rotation, in contrast to previously suggested methods such as prediction error, histogram shifting, and other popular techniques. Overall, the multi-MSB replacement technique offers a cleaner design through vectorized bit operations, enabling faster and more efficient computation during image processing. A limitation to our work is that the LMR-RDHEI method has a very low probability ($\leq$ 0.14%) of resulting in a bad case among testing samples, as shown in Table. II. In Section IV-A, we stated that bad cases occur mostly due to some images having high frequency noise and complex textures. It occurs when LMR-RDHEI does not sufficiently compress the assistant data to be stored in the first MSB plane. In the future, we aim to find better solutions to the challenge of achieving a reasonable compression rate with some highly-textured images. Given the high performance of our multi-MSB replacement-based technique and its principle of identifying and utilizing redundant bits to embed a secret message, further investigations on this method are recommended. For example, it is possible to apply a dynamic b parameter to trace and use more redundant bits in an original image to achieve a better performance overall.

REFERENCES

[1] Zoltak Bartosz. Vmpc one-way function and stream cipher. *Fast Software Encryption*, pages 210–225, 2004.

[2] P. Bas and T. Furon. Image database of bows-2, 2008. URL http://bows2.ec-lille.fr.

[3] 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 Transactions on Cybernetics*, 46(5):1132–1143, 2016.

[4] J. Chang, Y. Chou, C. Ni, and H. Wu. Reversible data hiding in pairwisely encrypted images. In *2016 Third International Conference on Computing Measurement Control and Sensor Network (CMCSN)*, pages 60–63, 2016.

[5] K. Chen and C. Chang. High-capacity reversible data hiding in encrypted images based on extended run-length coding and block-based msb plane rearrangement. *Journal of Visual Communication and Image Representation*, 58:334–344, 2019.

[6] W. Hong, T. Chen, and H. Wu. An improved reversible data hiding in encrypted images using side match. *IEEE Signal Processing Letters*, 19(4):199–202, 2012.

[7] X. Hu, W. Zhang, and N. Yu. Optimizing pixel predictors based on self-similarities for reversible data hiding. In *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 481–484, 2014.

[8] Zhongyun Hua, Fan Jin, Binxuan Xu, and Hejiao Huang. 2d logistic-sine-coupling map for image encryption. *Signal Processing*, 149, 03 2018. doi: 10.1016/j.sigpro.2018.03.010.

[9] Delu Huang and Jianjun Wang. High-capacity reversible data hiding in encrypted image based on specific encryption process. *Signal Processing: Image Communication*, 80:115632, 2020. ISSN 0923-5965.

[10] Ahmad Khan, Ming Fan, Asad Malik, Izhar Alam, and Mohammed Husain. A secure reversible data hiding method in encrypted images using prediction error estimation (pee) technique. *IEEE International Conference on Electrical, Electronics and Computer Engineering (UPCON-2019)*, 12 2019.

[11] Jianyi. Liu, Kaifeng. Zhao, and Ru. Zhang. A fully reversible data hiding scheme in encrypted images based on homomorphic encryption and pixel prediction. *Circuits, Systems, and Signal Processing*, 39, 2020.

[12] Z. Liu, Pun, and Chi-Man. Reversible data-hiding in encrypted images by redundant space transfer. *Information Sciences*, 433-434:188–203, 2017.

[13] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Transactions on Information Forensics and Security*, 8(3):553–562, 2013.

[14] Asad. Malik, Hongxia. Wang, Yanli. Chen, and Ahmad Neyaz. Khan. A reversible data hiding in encrypted image based on prediction-error estimation and location map. *Multimedia Tools and Applications*, 79:11591–11614, 2020.

[15] K. Markus. Jbig lossless image compression library, 2014. URL https://www.cl.cam.ac.uk/~mgk25/jbigkit/.

[16] Nasir D. Memon, Xiaolin Wu, V. Sippy, and G. Miller. Interband coding extension of the new lossless JPEG standard. In Jan Biemond and Edward J. Delp III, editors, *Visual Communications and Image Processing '97*, volume 3024, pages 47 – 58. International Society for Optics and Photonics, SPIE, 1997. doi: 10.1117/12.263270. URL https://doi.org/10.1117/12.263270.

[17] P. Puteaux and W. Puech. Reversible data hiding in encrypted images based on adaptive local entropy analysis. In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2017.

[18] P. Puteaux and W. Puech. An efficient msb prediction-based method for high-capacity reversible data hiding in encrypted images. *IEEE Transactions on Information Forensics and Security*, 13(7):1670–1681, 2018.

[19] P. Puteaux and W. Puech. Epe-based huge-capacity reversible data hiding in encrypted images. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7, 2018.

[20] P. Puteaux and W. Puech. A recursive reversible data hiding in encrypted images method with a very high payload. *IEEE Transactions on Multimedia*, pages 1–1, 2020.

[21] Pauline Puteaux, SimYing Ong, KokSheik Wong, and William Puech. A survey of reversible data hiding in encrypted images – the first 12 years. *Journal of Visual Communication and Image Representation*, 77:103085, 2021. doi: https://doi.org/10.1016/j.jvcir.2021.103085.

[22] Y. Puyang, Z. Yin, and Z. Qian. Reversible data hiding in encrypted images with two-msb prediction. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7, 2018.

[23] Z. Qian and X. Zhang. Reversible data hiding in encrypted images with distributed source encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(4):636–646, 2016.

[24] Ronald L. Rivest and Jacob C. N. Schuldt. Spritz - a spongy rc4-like stream cipher and hash function. *IACR Cryptol. ePrint Arch.*, 2016:856, 2016.

[25] Gerald Schaefer and Michal Stich. UCID: an uncompressed color image database. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307, pages 472 – 480. SPIE, 2003. doi: 10.1117/12.525375. URL https://doi.org/10.1117/12.525375.

[26] V. M. Sekhar and C. S. Kumar. Laplacian: Reversible data hiding technique. In *2019 Fifth International Conference on Image Information Processing (ICIIP)*, pages 546–551, 2019.

[27] Paul Souradyuti and Preneel Bart. A new weakness in the rc4 keystream generator and an approach to improve the security of the cipher. *Fast Software Encryption*, pages 245–259, 2004.

[28] Maitra Subhamoy and Paul Goutam. Analysis of rc4 and proposal of additional layers for better security margin. *Progress in Cryptology - INDOCRYPT 2008*, pages 27–39, 2008.

[29] Z. Tang, S. Xu, H. Yao, C. Qin, and X. Zhang. Reversible data hiding with differential compression in encrypted image. *Multimedia Tools and Applications*, 78, 2019. doi: 10.1007/s11042-018-6567-3.

[30] Y. Wang and W. He. High capacity reversible data hiding in encrypted image based on adaptive msb prediction. *IEEE Transactions on Multimedia*, pages 1–1, 2021. doi: 10.1109/TMM.2021.3062699.

[31] X. Wu and W. Sun. High-capacity reversible data hiding in encrypted images by prediction error. *Signal Processing*, 104:387–400, Nov. 2014.

[32] Y. Wu, J.P. Noonan, and S. Agaian. Npcr and uaci randomness tests for image encryption. *Journal of Selected Areas in Telecommunications (JSAT)*, 2011.

[33] X. Xie and C. Chang. Reversible data hiding in encrypted images using reformed jpeg compression. In *2017 5th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–5, 2017.

[34] Dawen Xu and Shubing Su. Separable reversible data hiding in encrypted images based on two-dimensional histogram modification. *Multimedia Security: Novel Steganography and Privacy Preserving*, 2018, 2018.

[35] S. Yi and Y. Zhou. Reversible data hiding in encrypted images using prediction-error encoding. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1789–1793, 2018.

[36] S. Yi and Y. Zhou. Separable and reversible data hiding in encrypted images using parametric binary tree labeling. *IEEE Transactions on Multimedia*, 21(1):51–64, 2019.

[37] Z. Yin, A. Abel, X. Zhang, and B. Luo. Reversible data hiding in encrypted image based on block histogram shifting. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2129–2133, 2016.

[38] Zhijian Wang Yuping Hu, Congxu Zhu. An improved piecewise linear chaotic map based image encryption algorithm. *The Scientific World Journal*, page 7, 2014. doi: https://doi.org/10.1155/2014/275818.

[39] T. Zhang, X. Li, W. Qi, and Z. Guo. Location-based pvo and adaptive pairwise modification for efficient reversible data hiding. *IEEE Transactions on Information Forensics and Security*, 15:2306–2319, 2020.

[40] X. Zhang, J. Long, Z. Wang, and H. Cheng. Lossless and reversible data hiding in encrypted images with public-key cryptography. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(9):1622–1631, 2016.

[41] 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 Transactions on Circuits and Systems for Video Technology*, 26(3):441–452, 2016.

Yike Zhang received an M.S. degree in Computer Engineering from the Engineering Department at St. Mary's University, San Antonio, Texas, USA in Spring 2021. She is currently pursuing her Ph.D. in Computer Science at Vanderbilt University. Her research interests include image processing, multimedia information security, digital forensics, and watermarking. Her work has focused on image processing and analysis in the encrypted domain.

Wenbin Luo received the B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, P.R.China. He received another M.S. degree in statistics and a Ph.D. degree in computer engineering from the University of New Mexico, Albuquerque, NM, USA. Also, he is an Oracle Certified Professional (OCP), a Senior Level Linux Professional (LPIC-3), and a Ubuntu Certified Professional (UCP). He is currently a professor of computer engineering at St. Mary's University of San Antonio, where he has been teaching since 2003. He has authored or co-authored over 40 research papers, a book, and two book chapters. His research interests include digital image processing, computer security, and data structures & algorithms. He was the Publication Chair of IEEE SoSE08, SoSE09, & SoSE13 and was a Program Co-Chair of IEEE SoSE15.