Hindawi
Security and Communication Networks
Volume 2018, Article ID 1734961, 14 pages
https://doi.org/10.1155/2018/1734961

WILEY | Hindawi

*Research Article*

# Separable Reversible Data Hiding in Encrypted Images Based on Two-Dimensional Histogram Modification

**Dawen Xu**,[1] **Kai Chen**,[1] **Rangding Wang**,[2] **and Shubing Su**[1]

[1]*School of Electronics and Information Engineering, Ningbo University of Technology, Ningbo 315211, China*
[2]*CKC Software Lab, Ningbo University, Ningbo 315211, China*

Correspondence should be addressed to Dawen Xu; dawenxu@126.com

Received 29 September 2017; Revised 30 November 2017; Accepted 21 December 2017; Published 7 February 2018

Academic Editor: Xinpeng Zhang



An efficient method of completely separable reversible data hiding in encrypted images is proposed. The cover image is first partitioned into nonoverlapping blocks and specific encryption is applied to obtain the encrypted image. Then, image difference in the encrypted domain can be calculated based on the homomorphic property of the cryptosystem. The data hider, who does not know the original image content, may reversibly embed secret data into image difference based on two-dimensional difference histogram modification. Data extraction is completely separable from image decryption; that is, data extraction can be done either in the encrypted domain or in the decrypted domain, so that it can be applied to different application scenarios. In addition, data extraction and image recovery are free of any error. Experimental results demonstrate the feasibility and efficiency of the proposed scheme.

## 1. Introduction

With the rapid developments occurring in mobile internet and cloud storage, privacy and security of personal data have gained significant attention nowadays. There are no guarantees that stored data will not be accessed by unauthorized entities, such as the cloud provider itself or malicious attackers. Under these specific circumstances, sensitive images, such as medical and personal images, need to be encrypted before outsourcing for privacy-preserving purposes [1, 2]. In other words, the consumers would like to give the untrusted cloud server only an encrypted version of the data instead of the original content. The cloud service provider (who stores the data) is not authorized to access the original content (i.e., plaintext). However, in some application scenarios, the cloud servers or database managers need to embed some additional messages, such as authentication or notation data, directly into an encrypted data for tamper detection or ownership declaration purposes. For example, patient's information can be embedded into his/her encrypted medical image to avoid unwanted exposure of confidential information.

To address this problem, researchers have been studying the possibility of hiding data directly in the encrypted domain. Over the past few years, a considerable amount of schemes about data hiding in encrypted images or videos has been reported in the literature [3–10]. However, within these schemes, the host image/video is permanently distorted caused by data embedding. In general, the cloud service provider has no right to introduce permanent distortion. This implies that, for a legal receiver, the original plaintext content should be recovered without any error after image decryption and data extraction. To solve this problem, reversible data hiding (RDH) in the encrypted domain is preferred.

RDH is a technique that slightly alters digital media (e.g., images or videos) to embed secret data while the original digital media can be recovered without any error after the hidden messages have been extracted [11]. This specific data hiding technique has been found to be useful in some important and sensitive areas, that is, military communication, medical science, law-enforcement, and error concealment [12, 13], where the original media is required to be reconstructed without any distortion. So far, three major approaches, that is, lossless compression [14], histogram modification [11, 15], and difference expansion [16], have already been developed for RDH. For more details of these methods and other RDH methods, refer to the latest review of recent research [17].

Although RDH techniques have been studied extensively, these techniques are suitable for plaintext instead of ciphertext.

RDH in the encrypted domain has emerged as a new and challenging research field. In recent years, some RDH methods for encrypted images have been proposed. In general, these methods can be divided into three categories, that is, methods by vacating room after encryption (VRAE) [18–24], methods by reserving room before encryption (RRBE) [25–28], and methods based on homomorphic encryption [29–34]. In VRAE framework, the original signal is encrypted directly by the content owner, and the data hider embeds the additional bits by modifying some bits of the encrypted data. The advantage of this framework is that the operation of the end user is simple and efficient. However, as the entropy of an encrypted image has been maximized, the embedding capacity is limited. Moreover, the accuracy of data extraction and the quality of restored image are not satisfactory. In RRBE framework, the embedding room is created in the plaintext domain, that is, vacating room before encryption. The advantages of this framework are mainly reflected in two aspects; namely, embedding capacity is relatively large and pure reversibility is achieved. But this framework might be impractical because it requires the content owner to perform an extra preprocessing before content encryption [17]. In general, the content owner expects to send only an encrypted image to the manager without extra information. In addition to VRAE and RRBE, another type of method has recently been proposed by using homomorphic encryption. With the additive homomorphic property of Paillier cryptosystem, Chen et al. [29] firstly proposed a homomorphic encryption based RDH approach. Shiu et al. [31] improved Chen et al.'s method [29] by adopting the concept of difference expansion into homomorphic encryption. Moreover, RDH in the homomorphic encrypted domain has also been investigated in [32, 33]. However, the used public-key cryptosystems lead to data expansion after image encryption. In [30, 34], the additive homomorphic property of modulo operation is utilized to realize the RDH in the encrypted domain. The advantage is that encryption does not cause data expansion.

In this paper, we develop an effective and reliable framework for RDH in the encrypted domain. In fact, the proposed method belongs to the third category. Its main contribution is the combination of the modular addition and two-dimensional (2D) histogram modification. Its advantages are mainly manifested in four aspects. First of all, room for data hiding does not need to be vacated before encryption, which is more reasonable compared with the methods in [25–28]. Secondly, completely separable and completely reversible can be achieved, which is more reliable than the methods in [18–21]. Thirdly, the modular arithmetic addition operation, which has additive homomorphism, is utilized for image encryption. It does not cause data expansion, unlike the public-key cryptosystems in [29, 31–33]. Finally, since data embedding in encrypted domain is accomplished by using pairwise coefficient modification, embedded capacity has been greatly improved compared with the methods in [30, 34]. The rest of the paper is organized as follows. In Section 2, we describe the proposed scheme, which includes image encryption, data embedding in encrypted image, data extraction, and original image recovery. Experimental results and analysis are presented in Section 3. Finally, in Section 4, conclusions and future work are drawn.

## 2. Proposed Scheme

In this section, a RDH method in encrypted images is illustrated. It is composed of three parts, that is, generation of the encrypted image, generation of the marked encrypted image, data extraction, and image recovery. First, the content owner encrypts the original image with encryption key to produce an encrypted image. Then, the data hider without knowing the actual content of the original image can embed some additional data into the encrypted image. Here, the data hider can be a third party, for example, a database manager or a cloud provider, who is not authorized to access the original content of the signal (i.e., plaintext). At the receiving end, maybe the content owner himself or an authorized third party can extract the hidden data either in encrypted or decrypted image. For illustrative purposes, the framework of the proposed scheme is given in Figure 1.

*2.1. Image Encryption.* Assume the original image $X$ is an 8-bit gray-scale image with size $M \times N$ and pixels $x(i, j) \in [0, 255]$, $0 \le i \le M-1$, $0 \le j \le N-1$. As we know, in the plaintext image, the correlation will gradually decrease with the increase of the distance between two pixels. In order to make good use of the correlation among pixels for RDH, the cover image is divided into a number of nonoverlapping blocks of size $3 \times 3$ as shown in Figure 2. If both $M$ and $N$ can be divisible by 3, the number of nonoverlapping blocks is $(M/3)\times(N/3)$. If $M$ or $N$ cannot be divisible by 3, the image is divided into $\lceil M/3 \rceil \times \lceil N/3 \rceil$ blocks, including $\lfloor M/3 \rfloor \times \lfloor N/3 \rfloor$ blocks of size $3 \times 3$. Here, $\lceil M/3 \rceil$ denotes the smallest integer greater than or equal to $M/3$, and $\lfloor M/3 \rfloor$ denotes the greatest integer less than or equal to $M/3$.

To ensure that pixels in the same block are encrypted with the same random value, the encryption matrix $R_a = \{r_a(i, j) \mid r_a(i, j) \in [0, 255]\}$ is obtained using the following equation:

$$r_a(i, j) = c\left(\left\lfloor \frac{i}{3} \right\rfloor, \left\lfloor \frac{j}{3} \right\rfloor\right), \tag{1}$$

where $C = \{c(p, q) | c(p, q) \in [0, 255], 0 \le p \le \lfloor M/3 \rfloor, 0 \le q \le \lfloor N/3 \rfloor\}$ is a pseudo-random matrix generated with the encryption key $\mathrm{En}_{\mathrm{key}}$. After getting the encryption matrix $R$, image encryption is done as follows.

$$S = E(X, R) = \left(x(i, j) + r_a(i, j)\right) \bmod 256 = s(i, j) \quad \forall i = 0, 1, \ldots, M-1, \; j = 0, 1, \ldots, N-1, \tag{2}$$

where $S$ represents an encrypted image. The corresponding decryption can be done in the following manner:

$$X = D(S, R) = \left(s(i, j) - r_a(i, j)\right) \bmod 256 = x(i, j) \quad \forall i = 0, 1, \ldots, M-1, \; j = 0, 1, \ldots, N-1. \tag{3}$$

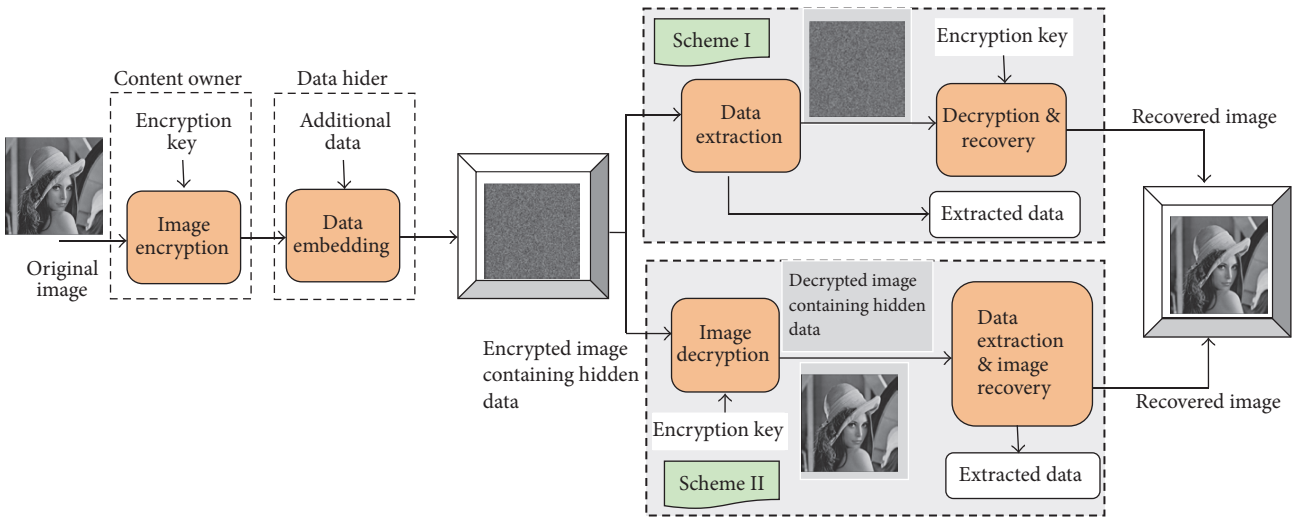



Figure 1: The framework of proposed scheme.

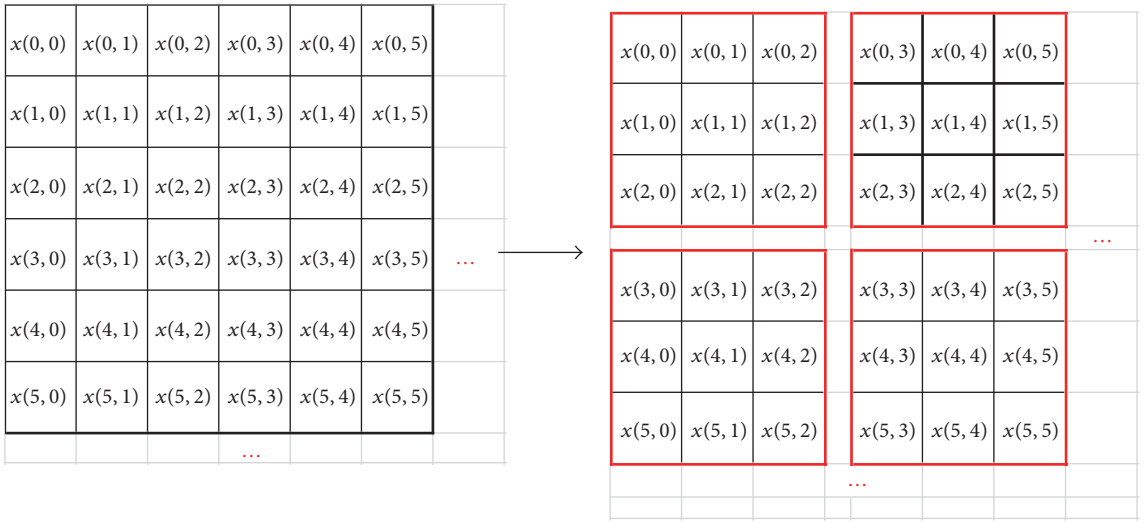



Figure 2: Example of image partition.

*2.2. Data Embedding in Encrypted Image.* After receiving the encrypted image, the data hider can embed some additional information into it for the purpose of media notation or integrity authentication. In order to achieve reversibility, the idea of histogram shifting is introduced in ciphertext based on homomorphic encryption. The whole process consists of two parts, namely, difference histogram generation and difference histogram modification.

*(1) Difference Histogram Generation.* Before performing the data embedding operation, a two-dimensional difference histogram of the encrypted image needs to be generated. The detailed procedure can be described as follows.

*Step 1.* Divide the encrypted image into nonoverlapping $3 \times 3$ blocks, which is the same as Figure 2. If the width or height of the image is not a multiple of 3, then the edge block will be ignored during the data embedding process.

*Step 2.* Calculate the difference between the basic pixel and the remaining pixels in each 3×3 block. Here, the pixel located in the center coordinate $(m, n)$ is taken as the basic pixel for prediction. Then the difference can be calculated by using the following equation:

$$f(m+u, n+v) = (s(m+u, n+v) - s(m, n)) \bmod 256, \tag{4}$$

where $u, v \in \{-1, 0, 1\}$. Note that the values of $u$ and $v$ cannot be zero at the same time. Obviously, eight differences can be obtained in each $3 \times 3$ block.

Although $s(m+u, n+v)$ is the encrypted value, it is easy to prove the following equation:

$$(s(m+u, n+v) - s(m, n)) \bmod 256 = (x(m+u, n+v) - x(m, n)) \bmod 256. \tag{5}$$

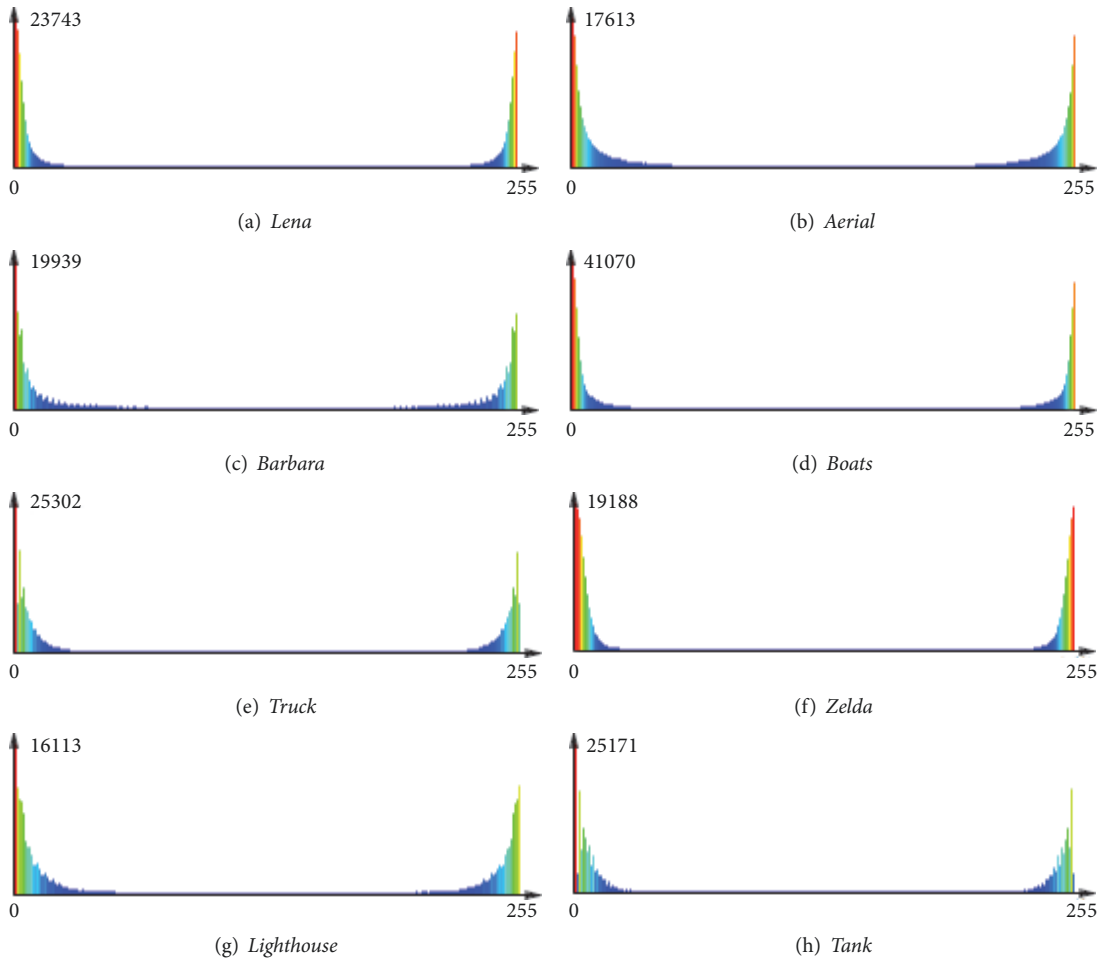



Figure 3: *1D* histogram of image difference.

*Proof.* One has

$$\begin{aligned}
&(s(m+u,n+v)-s(m,m)) \bmod 256\\
&\quad=\left(\left(x(m+u,n+v)+r_a(m+u,n+v)\right) \bmod 256\right.\\
&\quad\left.-\left(x(m,n)+r_a(m,n)\right) \bmod 256\right) \bmod 256\\
&\quad=\left(\left(x(m+u,n+v)+r_a(m+u,n+v)\right)\right.\\
&\quad\left.-\left(x(m,n)+r_a(m,n)\right)\right) \bmod 256,\\
&r_a(m+u,n+v)=r_a(m,n)\quad(\text{Refer to Eq. (1)}),\\
&\therefore (s(m+u,n+v)-s(m,n)) \bmod 256\\
&\quad=(x(m+u,n+v)-x(m,n)) \bmod 256.
\end{aligned} \tag{6}$$

□

According to the above proof, the correlation between the neighboring pixels in the local area of the plaintext image is preserved; that is, the difference remains unchanged even after encryption. All other 3 × 3 blocks can be processed in the same manner.

*Step 3.* Generate the difference histogram using differences in each 3 × 3 block. There is a high degree of correlation between adjacent pixels in a local region of an image. That is, they have similar gray values, or even the same gray value. Thus, the resulting difference histogram has a higher peak than the histogram of the original image. To demonstrate the distribution of the image difference, the histograms of some residual images are shown in Figure 3. It is clearly seen that the distribution is approximately symmetrical. The methods in [30, 34] mainly focus on exploiting one-dimensional (1D) coefficient histogram for RDH. The 1D coefficient histogram is usually defined as

$$h(r)=\#\left\{f_k(m+u,n+v)\mid f_k(m+u,n+v)=r\right\}, \tag{7}$$

where # denotes the cardinal number of a set, $r$ is an integer, and $k$ represents the block number. By considering every two differences together, the associated two-dimensional (2D) histogram can be defined as

$$\begin{aligned}
h(r_1,r_2)=\#\left\{\left(f_k\left(M_j\right),f_k\left(M_{j+1}\right)\right)\mid f_k\left(M_j\right)\right.\\
\left.=r_1,f_k\left(M_{j+1}\right)=r_2\right\},
\end{aligned} \tag{8}$$

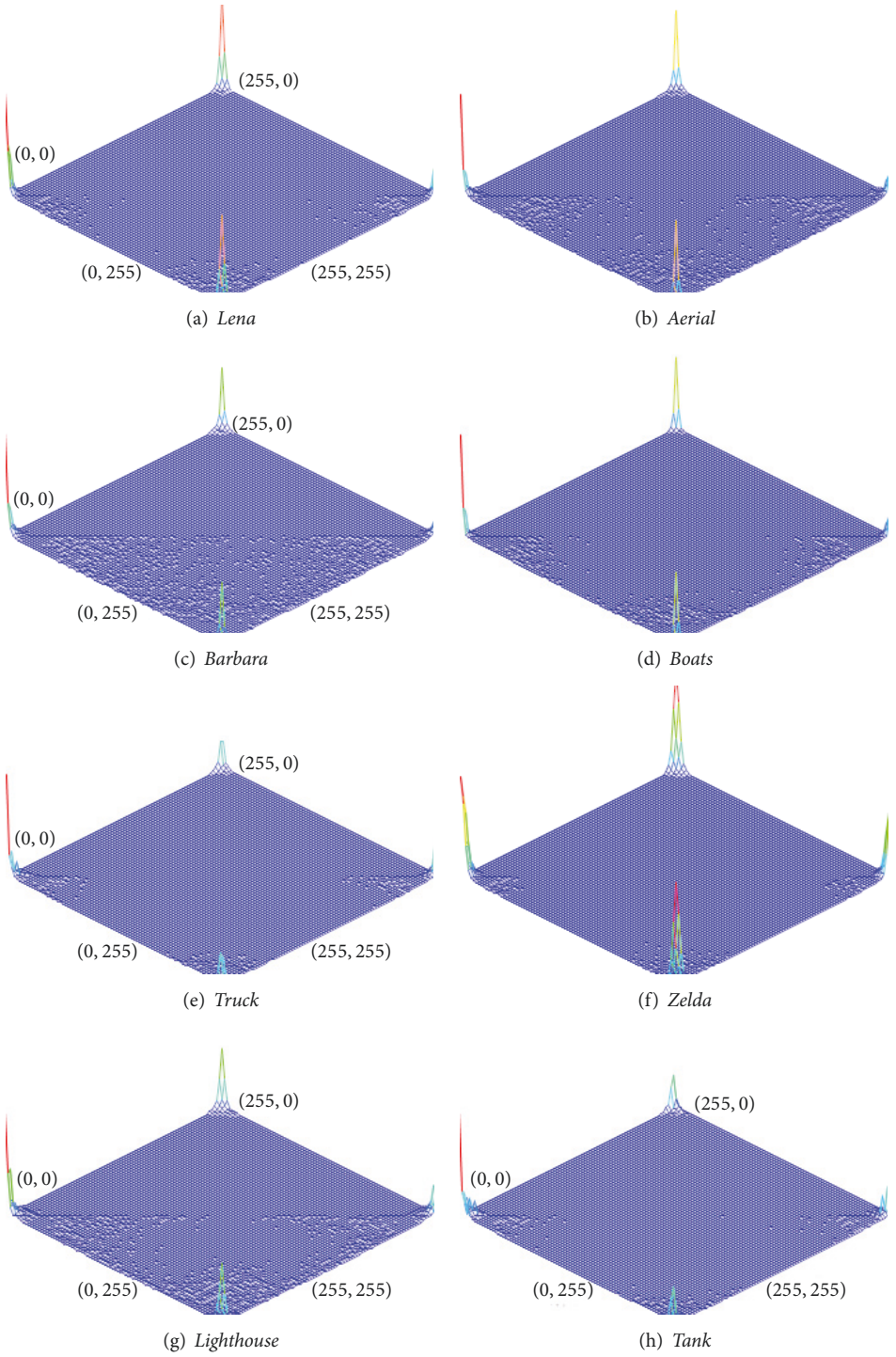



Figure 4: *2D* histogram of image difference.

where $f_k(M_j)$ denotes the $j$th difference in the $k$th $3 \times 3$ block. More specifically, $(f_k(M_1), f_k(M_2)) = (f_k(m-1, n-1), f_k(m, n-1))$, $(f_k(M_3), f_k(M_4)) = (f_k(m+1, n-1), f_k(m+1, n))$, $(f_k(M_5), f_k(M_6)) = (f_k(m-1, n), f_k(m-1, n+1))$, and $(f_k(M_7), f_k(M_8)) = (f_k(m, n+1), f_k(m+1, n+1))$. The distribution of the two-dimensional histogram is presented in Figure 4.

*(2) Difference Histogram Modification*. When the difference histogram is generated, reversible data hiding can be accomplished by using histogram shifting method. In [30], the conventional 1D histogram shifting technique is adopted. If the highest bin $T_p$ is located in the left side of the difference histogram, for example, $T_p = 0$, the graphical representation of data embedding is shown in Figure 5(a).

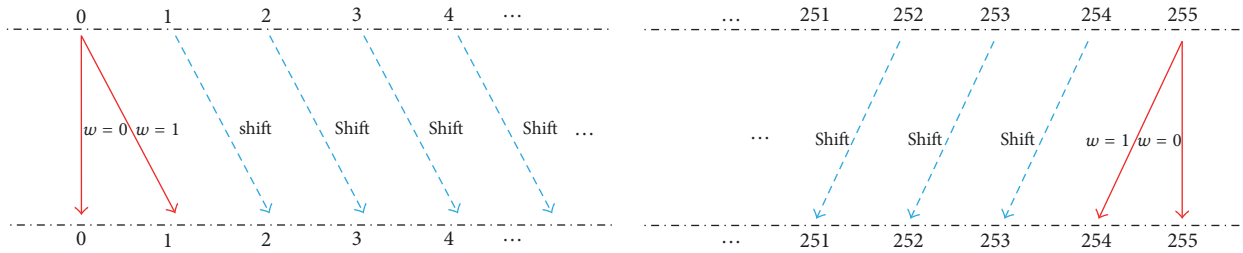



Figure 5: Illustration of the 1D histogram modification.

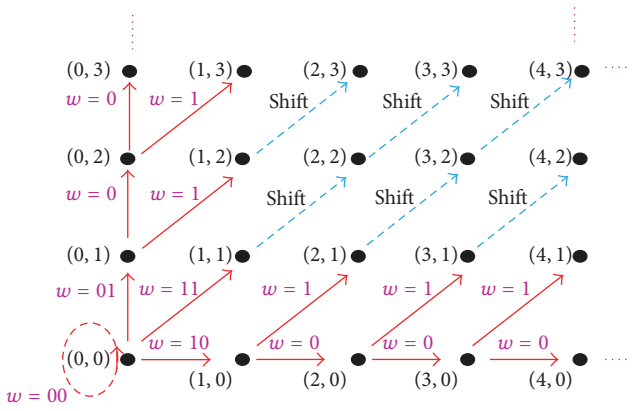



Figure 6: Illustration of the 2D histogram modification.

Otherwise, if the highest bin is located in the right side of the difference histogram, for example, $T_p = 255$, then its graphical representation is shown in Figure 5(b). Specifically, the conventional 1D RDH [30] can also be implemented in an equivalent way by modifying the 2D histogram [35].

For example, histogram modification in Figure 5(a) is in fact equivalent to the one shown in Figure 6. To further illustrate this case, some examples are provided below.

(i) For the coefficient pair $(f_k(M_j), f_k(M_{j+1})) = (0,0)$, in the method of 1D RDH shown in Figure 5(a), $f_k(M_j)$ is expanded to 0 or 1 for embedding a data bit $w \in \{0,1\}$, and $f_k(M_{j+1})$ is expanded similarly. Consequently, in the method of 2D RDH shown in Figure 6, the coefficient pair $(0,0)$ will be expanded to $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$ when the to-be-embedded bits are $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$, respectively.

(ii) For $(f_k(M_j), f_k(M_{j+1})) = (0,1)$, in the method of 1D RDH, $f_k(M_j)$ is expanded to 0 or 1 for embedding a data bit $w \in \{0,1\}$, and $f_k(M_{j+1})$ is shifted to 2. Correspondingly, in the method of 2D RDH, the pair $(0,1)$ is expanded to $(0,2)$ if $w = 0$, and $(1,2)$ if $w = 1$.

(iii) For $(f_k(M_j), f_k(M_{j+1})) = (2,1)$, in the method of 1D RDH, $f_k(M_j)$ and $f_k(M_{j+1})$ are shifted to 3 and 2, respectively. Accordingly, in the method of 2D RDH, the pair $(2,1)$ is shifted to $(3,2)$.

In particular, various histogram modification strategies can be designed based on 2D histogram. A reasonable histogram modification strategy directly contributes to the superior performance. The purpose of our design is to provide high embedding capacity while maintaining good visual quality. According to the statistical distribution of the difference histogram in Figure 4, we find that the probability of occurrence is larger when the difference is closer to 0 or 255. Based on this, a novel RDH technology is presented as shown in Figure 7.

Suppose the message to be embedded is a binary sequence denoted as $B = \{b(l) \mid l = 1,2,\ldots,K,\ b(l) \in \{0,1\}\}$. In order to enhance the security, a stream cipher is used to encrypt the message according to the data-hiding key $\mathrm{Dh}_{\mathrm{key}}$. Thus, the to-be-embedded binary information, that is, $W = \{w(l) \mid l = 1,2,\ldots,K, w(l) \in \{0,1\}\}$, is an encrypted version of $B$. It is difficult for anyone who does not retain the data hiding key to recover the message. The 2D histogram modification in the encrypted domain can be described as follows. According to the symmetry in Figure 4, only the modification in the lower-left quadrant is described for simplicity.

(1) If $(f_k(M_j), f_k(M_{j+1})) = (0,0)$, it has eight candidate directions for modification. In this case, three bits can be embedded. Specifically, the marked coefficient pair $(f'_k(M_j), f'_k(M_{j+1}))$ is determined as follows:

$$\left(f'_k\left(M_j\right), f'_k\left(M_{j+1}\right)\right) = \begin{cases} (0,0) & \text{if } w = '000' \\ (0,1) & \text{if } w = '001' \\ (1,0) & \text{if } w = '010' \\ (1,1) & \text{if } w = '011' \\ (0,2) & \text{if } w = '100' \\ (2,0) & \text{if } w = '101' \\ (1,2) & \text{if } w = '110' \\ (2,1) & \text{if } w = '111' \end{cases} \tag{9}$$

(2) If $(f_k(M_j), f_k(M_{j+1})) = (y,0)$, 1 bit can be embedded in each coefficient pair. Then, the marked coefficient pair $(f'_k(M_j), f'_k(M_{j+1}))$ is determined as follows

$$\left(f'_k\left(M_j\right), f'_k\left(M_{j+1}\right)\right) = \begin{cases} \left(f_k\left(M_j\right) + 2, f_k\left(M_{j+1}\right)\right) & \text{if } w = '0' \\ \left(f_k\left(M_j\right) + 2, f_k\left(M_{j+1}\right) + 1\right) & \text{if } w = '1', \end{cases} \tag{10}$$

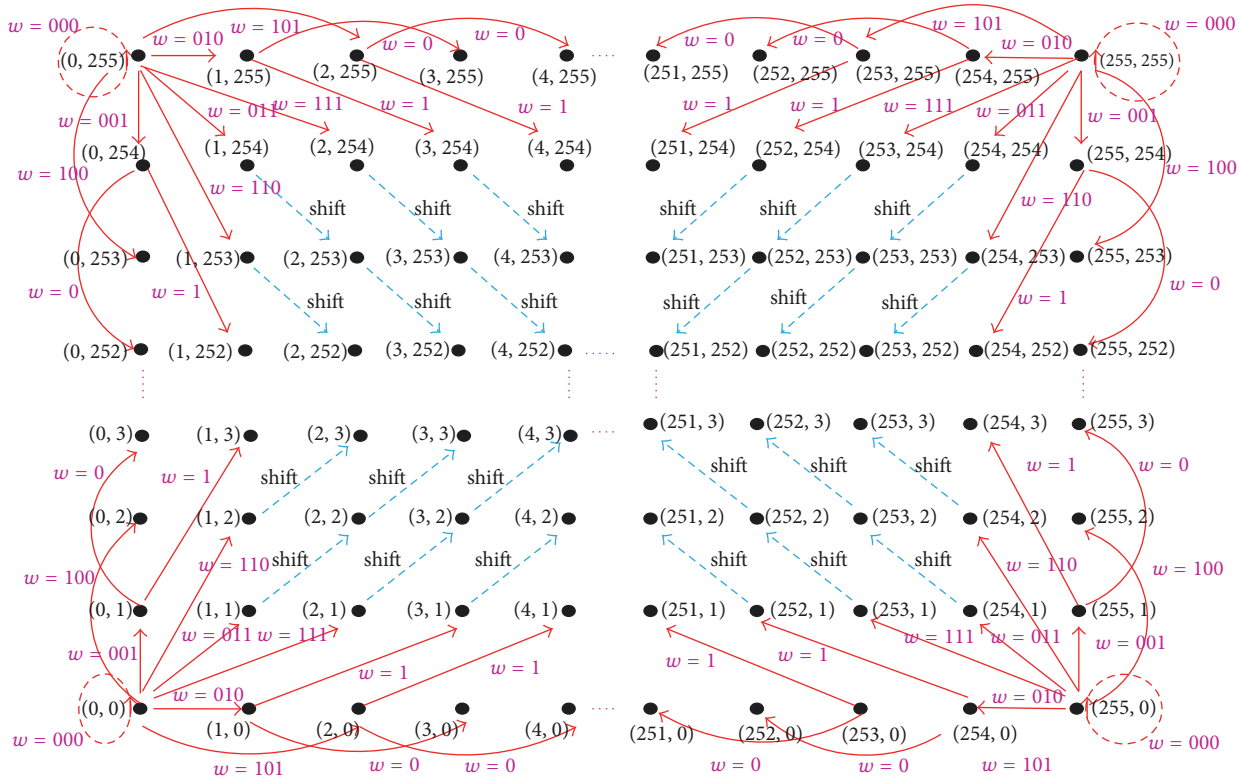



Figure 7: Illustration of the proposed 2D histogram modification.

where $1 \le y \le 125$. Although the middle part of the difference histogram is usually empty, ambiguities arise when the bins from two sides overlapped in the middle after expansion. To avoid it, the differences of 126 and 127 will not be expanded. However, ambiguities still arise when difference is changed from 125 to 127 or from 124 to 126 during the embedding process. The overlapping problem can be resolved by using a location map. It is a binary array with its every element corresponding to 126 and 127, 0 for genuine, and 1 for pseudo. The location map and the secret information will be embedded together in the encrypted domain.

(3) If $(f_k(M_j), f_k(M_{j+1})) = (0, z)$, the marked coefficient pair $(f_k'(M_j), f_k'(M_{j+1}))$ is determined as follows:

$$\left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = \begin{cases} \left(f_k\left(M_j\right), f_k\left(M_{j+1}\right) + 2\right) & \text{if } w = 0 \\ \left(f_k\left(M_j\right) + 1, f_k\left(M_{j+1}\right) + 2\right) & \text{if } w = 1, \end{cases} \tag{11}$$

where $1 \le z \le 125$.

(4) If $1 \le f_k(M_j) \le 126$ and $1 \le f_k(M_{j+1}) \le 126$, the coefficient pair $(f_k(M_j), f_k(M_{j+1}))$ is shifted to $(f_k'(M_j), f_k'(M_{j+1}))$ as follows:

$$\left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = \left(f_k\left(M_j\right) + 1, f_k\left(M_{j+1}\right) + 1\right). \tag{12}$$

According to the characteristic of modulus function, the following equation can be established.

$$\begin{aligned} & f\left(m+u, n+v\right) \pm 1 \\ & \quad = \left(s\left(m+u, n+v\right) - s\left(m, n\right) \bmod 256 \pm 1\right) \\ & \quad \cdot \bmod 256 \\ & \quad = \left(\left(s\left(m+u, n+v\right) \pm 1\right) \bmod 256 - s\left(m, n\right)\right) \\ & \quad \cdot \bmod 256. \end{aligned} \tag{13}$$

According to (13), the operation of $f(m+u, n+v) \pm 1$ can be accomplished by replacing $s(m+u, n+v)$ with $(s(m+u, n+v) \pm 1) \bmod 256$. Thus, in (10)~(12), the modification of the difference is equivalent to the modification of the pixel value. Then the marked and encrypted image $S' = \{s'(i, j) \mid s'(i, j) \in [0, 255]\}$ of the proposed scheme is obtained. The embedding capacity in the lower-left quadrant denoted as $EC_2$ can be computed by

$$EC_2 = 3 \cdot h\left(0, 0\right) + \sum_{1 \le y \le 125} h\left(y, 0\right) + \sum_{1 \le z \le 125} h\left(0, z\right). \tag{14}$$

*2.3. Data Extraction and Original Image Recovery.* In this scheme, data extraction and image decryption are completely separable. In other words, the hidden data can be extracted either in encrypted or in decrypted domain. Furthermore, our method is also reversible, where the hidden data could be removed to obtain the original image. We will first discuss the extraction in the encrypted domain followed by the decrypted domain.

*(1) Scheme I: Data Extraction in the Encrypted Domain*. In order to protect the users' privacy, the database manager (e.g., a cloud server) does not have sufficient permissions to access original video content due to the absence of encryption key. But the manager sometimes need to note and mark the personal information in corresponding encrypted images as well as verify their integrity. In this case, both data embedding and extraction should be manipulated in the encrypted domain. In the encrypted domain, the hidden data extraction can be accomplished by the following steps. According to the symmetry in Figure 4, only the extraction in the lower-left quadrant is described for simplicity.

*Step 1.* Divide the encrypted image into nonoverlapping $3 \times 3$ blocks, which is the same as Figure 2. The center pixel in each block is selected as the basic pixel for prediction.

*Step 2.* Calculate the difference between the basic pixel and the remaining pixels in each $3 \times 3$ block by using the following equation:

$$\begin{aligned} &f'(m+u, n+v) \\ &\quad = \left(s'(m+u, n+v) - s'(m, n)\right) \bmod 256. \end{aligned} \tag{15}$$

*Step 3.* The associated 2D histogram can be generated in the same way as in Section 2.2

$$\begin{aligned} h(r_1, r_2) = \#\left\{\left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) \mid f_k'\left(M_j\right)\right. \\ \left. = r_1, f_k'\left(M_{j+1}\right) = r_2\right\}. \end{aligned} \tag{16}$$

*Step 4.* According to the previous embedding rules, the hidden data can be extracted as

$$\widetilde{w} = \begin{cases} '000' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (0,0) \\ '001' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (0,1) \\ '010' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (1,0) \\ '011' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (1,1) \\ '100' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (0,2) \\ '101' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (2,0) \\ '110' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (1,2) \\ '111' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (2,1) \\ '0' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (y,0),\ 2 < y \le 127 \\ '1' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (y,1),\ 2 < y \le 127 \\ '0' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (0,z),\ 2 < z \le 127 \\ '1' & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (1,z),\ 2 < z \le 127, \end{cases} \tag{17}$$

where $\widetilde{w}$ denotes the extracted message bits. Since the whole process is entirely operated in encrypted domain, it effectively avoids the leakage of original content.

*Step 5.* With the data-hiding key, $\mathrm{Dh}_{\mathrm{key}}$, the extracted hidden bits could be further decrypted to obtain the original message. In addition, the image difference value can be further restored as follows:

$$f_k\left(M_j\right), f_k\left(M_{j+1}\right) = \begin{cases} (0,0) & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) \in \{(1,0),(0,1),(1,1)\} \\ \left(f_k'\left(M_j\right) - 2, f_k'\left(M_{j+1}\right)\right) & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (y,0),\ 2 \le y \le 127 \\ \left(f_k'\left(M_j\right) - 2, f_k'\left(M_{j+1}\right) - 1\right) & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (y,1),\ 2 \le y \le 127 \\ \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right) - 2\right) & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (0,z),\ 2 \le z \le 127 \\ \left(f_k'\left(M_j\right) - 1, f_k'\left(M_{j+1}\right) - 2\right) & \text{if } \left(f_k'\left(M_j\right), f_k'\left(M_{j+1}\right)\right) = (1,z),\ 2 \le z \le 127 \\ \left(f_k'\left(M_j\right) - 1, f_k'\left(M_{j+1}\right) - 1\right) & \text{if } 2 \le f_k'\left(M_j\right) \le 127,\ 2 \le f_k'\left(M_{j+1}\right) \le 127. \end{cases} \tag{18}$$

It should be noted that the boundary difference can be restored according to the location map. Similarly, according to (13), the operation of $f'(m+u, n+v) \pm 1$ can be accomplished by replacing $s'(m+u, n+v)$ with $(s'(m+u, n+v) \pm 1) \bmod 256$. Thus the encrypted image without the hidden data, that is, $S = \{s(i,j) \mid s(i,j) \in [0,255]\}$, is obtained.

*Step 6.* With the encryption key, $\mathrm{En}_{\mathrm{key}}$, the original cover image can be accurately restored by performing the decryption operation as in (3).

*(2) Scheme II: Data Extraction in the Decrypted Domain*. In scheme I, both data embedding and extraction are performed in the encrypted domain. However, in some cases, users want to decrypt the image first and then extract the hidden data from the decrypted image when it is needed. For example, with the encryption key, an authorized user wants to achieve the decrypted image containing the hidden data, which can be used to trace the source of the data. In this case, data extraction after image decryption is suitable. The whole process of decryption and data extraction comprised the following steps.

*Step 1.* Image decryption can be accomplished according to the following equation:

$$\begin{aligned} \widetilde{X} = \left(s'(i,j) - r_a(i,j)\right) \bmod 256 = \widetilde{x}(i,j) \\ \forall i = 0, 1, \ldots, M-1,\ \ j = 0, 1, \ldots, N-1. \end{aligned} \tag{19}$$

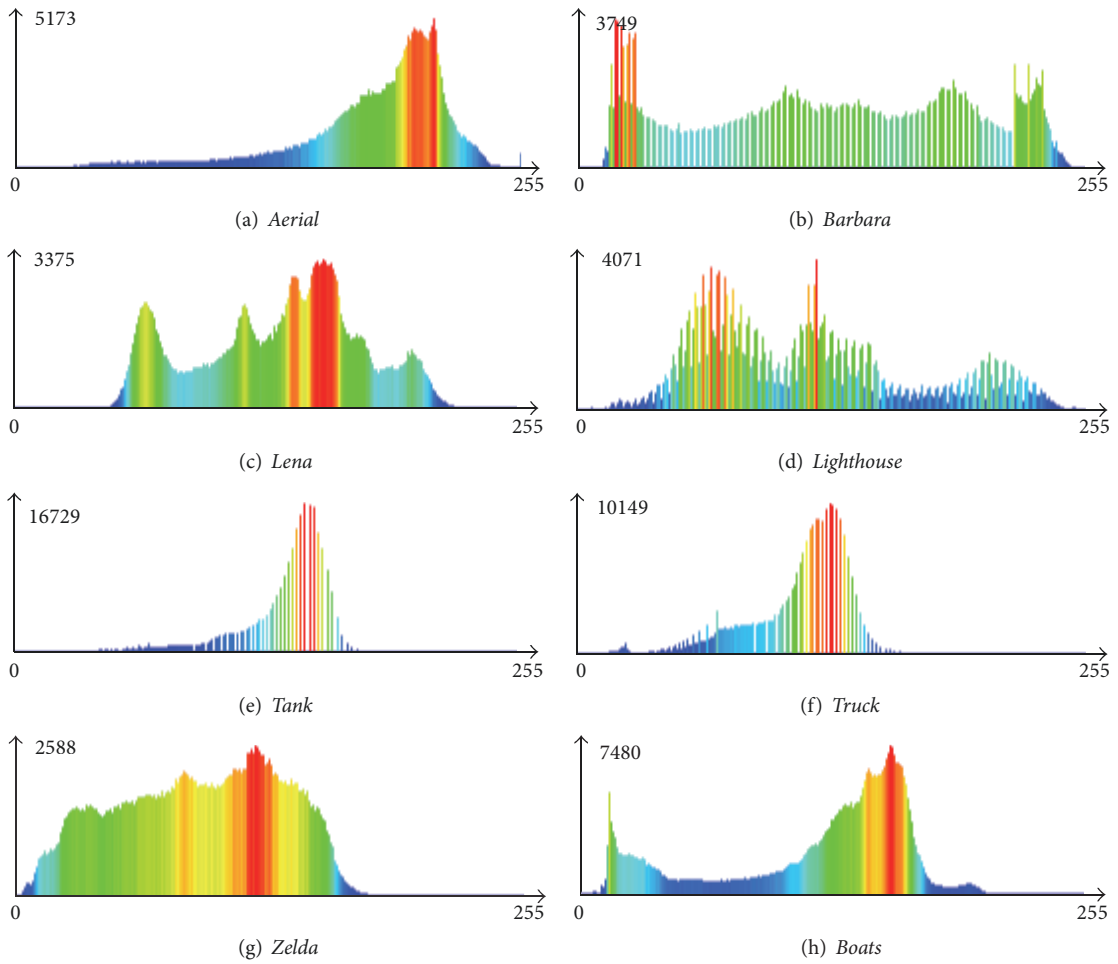



Figure 8: Histogram of the original image.

No visible distortions can be observed in the marked and decrypted images, as will be shown in later experimental results.

*Step 2.* Divide the marked and decrypted image $\widetilde{X} = \{\tilde{x}(i, j) \mid \tilde{x}(i, j) \in [0, 255]\}$ into nonoverlapping $3 \times 3$ blocks, which is the same as Figure 2. The center pixel in each block is selected as the basic pixel for prediction.

*Step 3.* Calculate the difference between the basic pixel and the remaining pixels to form the prediction error

$$f''(m+u, n+v) = (\tilde{x}(m+u, n+v) - \tilde{x}(m, n)) \bmod 256. \quad (20)$$

According to (5), the following equation is established:

$$f''(m+u, n+v) = f'(m+u, n+v). \quad (21)$$

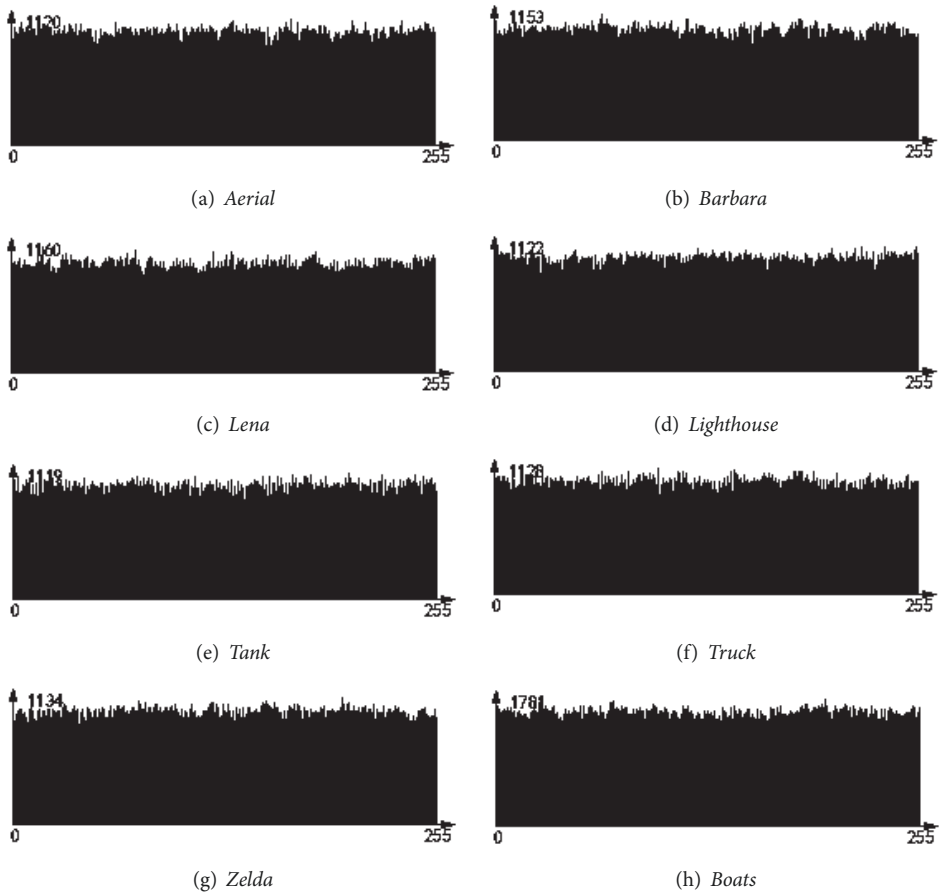
*Step 4.* The hidden data $\widetilde{w}$ can be extracted in a manner similar to (17). That is, it is only necessary to replace $(f'_k(M_j), f'_k(M_{j+1}))$ in (17) with $(f''_k(M_j), f''_k(M_{j+1}))$.

*Step 5.* The image difference can also be restored in the same manner as in (18). The only thing that needs to be adjusted is to replace $f'_k(M_j)$ and $f'_k(M_{j+1})$ with $f''_k(M_j)$ and $f''_k(M_{j+1})$, respectively. Similarly, the operation of $f''(m+u, n+v) \pm 1$ can be accomplished by replacing $\tilde{x}(m+u, n+v)$ with $(\tilde{x}(m+u, n+v) \pm 1) \bmod 256$. Therefore, the original image, that is, $X = \{x(i, j)\}$, is successfully restored.

## 3. Experimental Results and Analysis

Eight well-known standard gray images, that is, *Aerial*, *Barbara*, *Lena*, *Lighthouse*, *Tank*, *Truck*, *Zelda*, and *boats* [36], are considered for experimental purposes. The size of first 7 images is $512 \times 512 \times 8$, and the size of "Boats" is $720 \times 576 \times 8$. The secret data is a binary sequence created by pseudo-random number generator.

*3.1. Scrambling Effect and Security Analysis.* For an image encryption scheme, the security depends on cryptographic security and perceptual security. Cryptographic security denotes the security against cryptographic attacks, which relies on the underlying cipher. In the proposed scheme, pseudo-random sequence $r_a(i, j)$ is used to encrypt image. Figure 8 illustrates the histogram of the original image. After encryption, the corresponding histogram is shown in Figure 9. By comparing Figures 8 and 9, it can be observed




(a) *Aerial* (b) *Barbara*

(c) *Lena* (d) *Lighthouse*

(e) *Tank* (f) *Truck*

(g) *Zelda* (h) *Boats*

Figure 9: Histogram of the corresponding encrypted image.

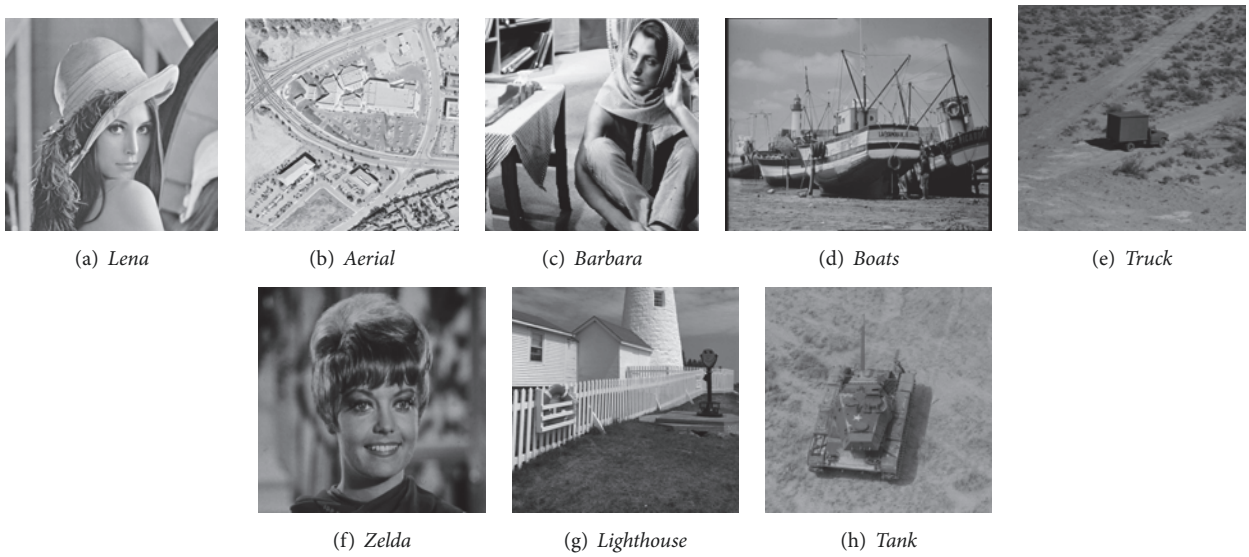
that the modified distribution appears to be uniform, which suggests that a statistical analysis would not be effective for evaluating the original content.

Perceptual security refers to the encrypted image being unintelligible. The original images are given in Figure 10, and their corresponding encrypted results are shown in Figure 11. As can be observed, the marked and encrypted image is a noise-like image. The visual information of the original image is damaged, which means that the data hider has extreme difficulty to obtain any useful information from it. In addition, for standard gray images, that is, *Aerial*, *Barbara*, *Lena*, *Lighthouse*, *Tank*, *Truck*, *Zelda*, and *boats*, PSNR (Peak Signal to Noise Ratio) values are 8.17 dB, 7.87 dB, 9.53 dB, 8.82 dB, 10.17 dB, 9.95 dB, 8.90 dB, and 9.11 dB, respectively. Obviously, scrambling performance of the described encryption system is more than adequate.

*3.2. Visual Quality of Marked and Decrypted Image.* Since the embedding scheme is reversible, the original cover content can be perfectly recovered after extracting the hidden data. In some scenarios, the encrypted image containing the hidden data provided by the server needs to be decrypted by the authorized user. Therefore, the visual quality of the decrypted image containing the hidden data is also expected to be equivalent or very close to that of the original image. In other words, the degradation of the image quality should be maintained at an acceptable range, even if the hidden data has not been removed. In the proposed method, since the maximum change in pixel value is 2, the artifacts introduced will not be perceptible. To verify this, a series of tests have been conducted. The original images and their corresponding decrypted versions containing the hidden data are shown in Figures 10 and 12, respectively. From our subjective examination, it is concluded that the marked content cannot be visually distinguished from nonmarked content. In addition to subjective observation, PSNR values are also given in Figure 8. In addition to *Zelda*, PSNR values of the remaining images are all above 47 dB. Generally, it is almost impossible to detect the degradation in image quality caused by data hiding.

*3.3. Embedding Capacity.* According to the embedding process described in Section 2.2, the embedded capacity can be calculated as follows:

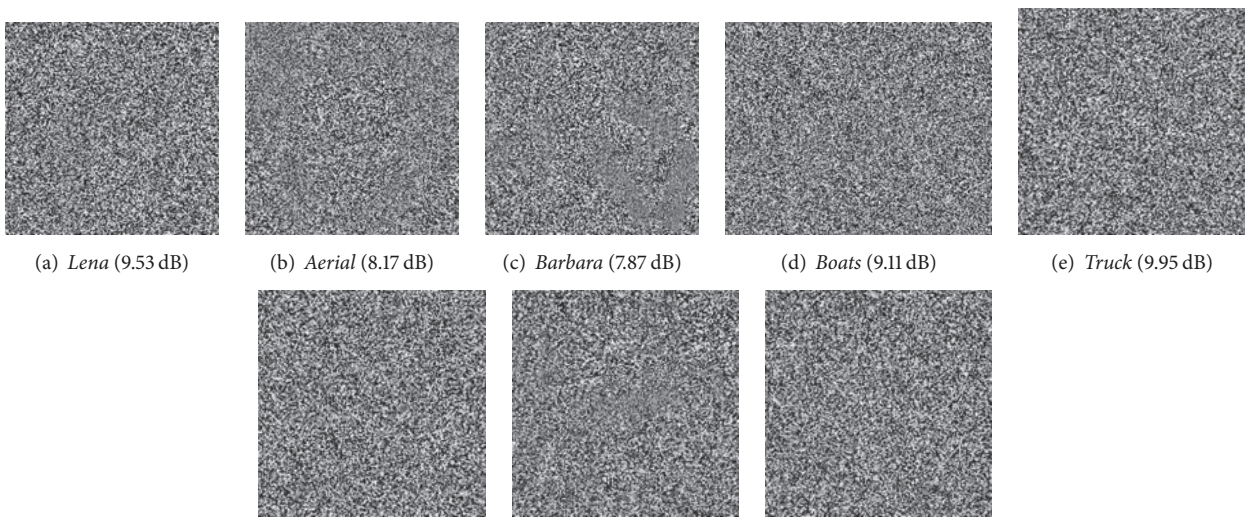
$$\begin{aligned} \mathrm{EC} &= 3 \\ &\quad \cdot \left(h\left(0,0\right) + h\left(0,255\right) + h\left(255,0\right) + h\left(255,255\right)\right) \end{aligned}$$



(a) *Lena* (b) *Aerial* (c) *Barbara* (d) *Boats* (e) *Truck*

(f) *Zelda* (g) *Lighthouse* (h) *Tank*

Figure 10: Original images.



(a) *Lena* (9.53 dB) (b) *Aerial* (8.17 dB) (c) *Barbara* (7.87 dB) (d) *Boats* (9.11 dB) (e) *Truck* (9.95 dB)

(f) *Zelda* (8.90 dB) (g) *Lighthouse* (8.82 dB) (h) *Tank* (10.17 dB)

Figure 11: The corresponding encrypted images.

$$+ \sum_{(1\le y\le 125)\cup(130\le y\le 254)} \left(h\left(y,0\right)+h\left(y,255\right)\right) + \sum_{(1\le z\le 125)\cup(130\le z\le 254)} \left(h\left(0,z\right)+h\left(255,z\right)\right). \tag{22}$$

For standard gray images, that is, *Aerial*, *Barbara*, *Lena*, *Lighthouse*, *Tank*, *Truck*, *Zelda*, and *boats*, the maximal embedding capacities of one-layer embedding strategy are 0.1432 bpp (bit per pixel), 0.1417 bpp, 0.1942 bpp, 0.1164 bpp, 0.1160 bpp, 0.1428 bpp, 0.1570 bpp, and 0.2136 bpp, respectively. It can be observed that the embedding capacity of the proposed scheme depends strongly on the characteristics of the original cover image. As expected, for images with high spatial activity (e.g., *Lighthouse*, *Tank*), low embedding rate is achieved. On the other hand, images with lower spatial activity (e.g., *Lena*, *Boats*) achieve higher embedding rate. The main reason is that most adjacent pixels have similar values in a smooth region. Therefore, they can contribute higher number of differences associated with the peak point compared with those in a complex region.

In our experiments, the size of the encrypted block is set to $3 \times 3$. In general, with the increase of the block size, the embedding capacity will increase whereas the security performance of the encryption algorithm will decrease. According to our analysis in Section 2.2, in any block, the difference of those pixel pairs remains unchanged even after encryption. With the increase of the block size, more correlation between the neighboring pixels may be preserved, and thus the

(a) *Lena* (47.29 dB) (b) *Aerial* (47.67 dB) (c) *Barbara* (47.69 dB) (d) *Boats* (47.27 dB) (e) *Truck* (47.60 dB)

(f) *Zelda* (43.98 dB) (g) *Lighthouse* (47.76 dB) (h) *Tank* (47.70 dB)

FIGURE 12: Decrypted images containing the hidden data.

embedding capacity will increase. On the other hand, the difference value between any pixel pair in each block of the plain image can be recovered in encrypted domain. In addition, higher capacities can be also achieved by applying multiple-layer embedding strategy. However, its cost is the decrease in perceptual quality.

*3.4. Comparison and Discussion.* As mentioned in Section 1, the methods in [18–21] may introduce some errors on data extraction and/or image recovery, while the complete reversibility can be achieved in the proposed method. More importantly, these methods are designed to carry only small payloads. Taking Zhang's method [18], for instance, the embedding rate is 0.0156 bpp associated with block size $8 \times 8$. If error correction mechanism is introduced, the actual embedding rate will be further decreased. It can be observed that our method achieves significantly higher embedding rate. For methods in [25–28], completely error-free data extraction and image recovery can be obtained. But it requires the content owner to perform an extra preprocessing before content encryption, which might be impractical. Instead, the proposed method overcomes these two problems.

Furthermore, Figure 13 shows the comparison of the embedding capacity between the proposed method and the methods in [30, 34]. Here, the maximum embedding capacity in one-layer embedding strategy is provided. As can be seen, in one-layer embedding strategy, the embedding capacity has been greatly improved. In fact, it can also be seen from Figures 6 and 7 that the embedding capacity of the proposed method can certainly be larger in one-layer embedding strategy. For example, when the coefficient pair is (0, 0), two bits can be embedded in the method of [30]. However, three bits can be embedded in the proposed method. The direct benefit is that a larger capacity can be achieved by one-layer histogram shifting. If two-layer embedding is used, the visual quality reduction is relatively large. Taking *Lena*

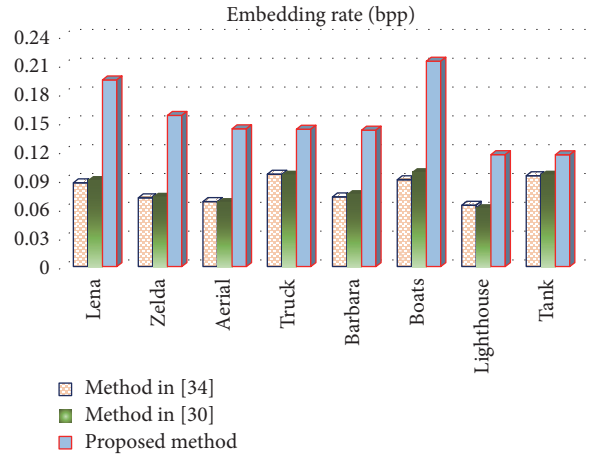
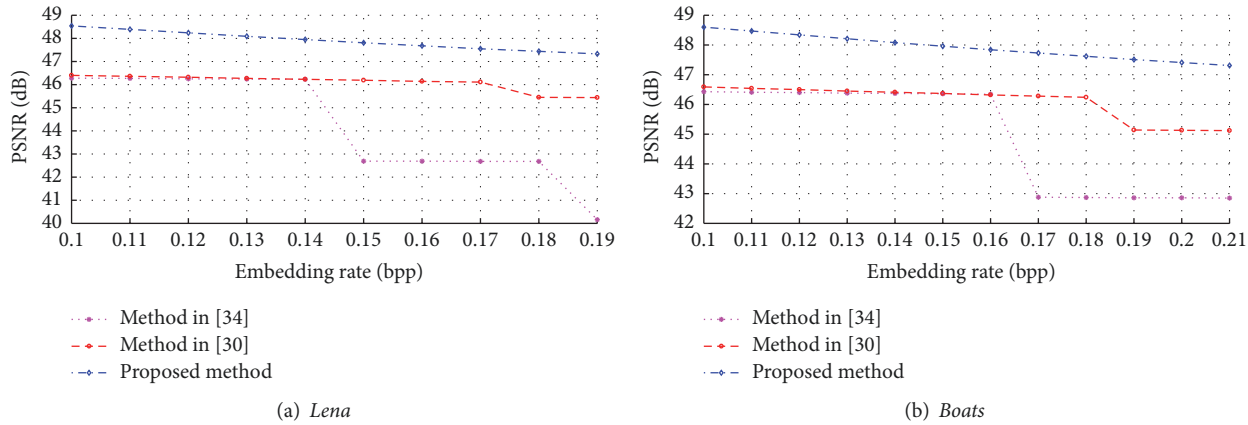



FIGURE 13: The comparison results of embedding capacity.

and *Boats* as an example, the performance comparison of different embedding rates is given in Figure 14. Obviously, the proposed method can provide better performance when the embedding capacity exceeds the maximum capacity of one-layer embedding strategy.

## 4. Conclusions and Future Work

In this paper, an algorithm to reversibly embed secret data in encrypted images is presented. A specific modulo operation is utilized to encrypt the image, which can preserve some correlation between the neighboring pixels. With the preserved correlation, the data hider can embed the secret data into the encrypted image by using 2D histogram modification, even though he does not know the original image content. Since the embedding process is done on encrypted data, our scheme preserves the confidentiality of content. Data




(a) *Lena* (b) *Boats*

Figure 14: The performance comparison of different embedding rates.

extraction is separable from image decryption; that is, the additional data can be extracted either in the encrypted domain or in the decrypted domain. Furthermore, this algorithm can achieve real reversibility and high quality of marked and decrypted images. One of the possible applications of this method is image annotation in cloud computing where high image quality and reversibility are greatly desired.

Although RDH technology and cryptography have been studied extensively, RDH in the encrypted domain is a highly interdisciplinary area of research. Technical research in this field has only just begun, and there is still an open space for research in this interdisciplinary research area. In future, more considerable effort is needed to determine the optimal modification on the histogram for achieving the best rate-distortion performance. Moreover, future work also aims at designing more efficient scheme for RDH in encrypted videos [37].

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (61771270, 61672302), Zhejiang Provincial Natural Science Foundation of China (LY17F020013, LZ15F020002), and Public Welfare Technology Application Research project of Zhejiang Province (2015C33237).

## References

[1] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A Secure and Dynamic Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[2] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[3] T. Bianchi and A. Piva, "Secure watermarking for multimedia content protection: a review of its benefits and open issues," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 87–96, 2013.

[4] B. Zhao, W. Kou, H. Li, L. Dang, and J. Zhang, "Effective watermarking scheme in the encrypted domain for buyer-seller watermarking protocol," *Information Sciences*, vol. 180, no. 23, pp. 4672–4684, 2010.

[5] J. Guo, P. Zheng, and J. Huang, "Secure watermarking scheme against watermark attacks in the encrypted domain," *Journal of Visual Communication and Image Representation*, vol. 30, pp. 125–135, 2015.

[6] A. V. Subramanyam, S. Emmanuel, and M. S. Kankanhalli, "Robust watermarking of compressed and encrypted JPEG2000 images," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 703–716, 2012.

[7] H. Liu, D. Xiao, R. Zhang, Y. Zhang, and S. Bai, "Robust and hierarchical watermarking of encrypted images based on Compressive Sensing," *Signal Processing: Image Communication*, vol. 45, pp. 41–51, 2016.

[8] D. Xu, R. Wang, and Y. Q. Shi, "Data hiding in encrypted H.264/AVC video streams by codeword substitution," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 596–606, 2014.

[9] D. Xu and R. Wang, "Context adaptive binary arithmetic coding-based data hiding in partially encrypted H.264/AVC videos," *Journal of Electronic Imaging*, vol. 24, no. 3, Article ID 033028, 2015.

[10] D. Xu, R. Wang, and Y. Q. Shi, "An improved scheme for data hiding in encrypted H.264/AVC videos," *Journal of Visual Communication and Image Representation*, 2015.

[11] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–361, 2006.

[12] D. Xu, R. Wang, and Y. Q. Shi, "An improved reversible data hiding-based approach for intra-frame error concealment in H.264/AVC," *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 410–422, 2014.

[13] D. Xu and R. Wang, "Two-dimensional reversible data hiding-based approach for intra-frame error concealment in H.264/AVC," *Signal Processing: Image Communication*, vol. 47, pp. 369–379, 2016.

[14] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding-new paradigm in digital watermarking," *EURASIP Journal on Applied Signal Processing*, vol. 2, pp. 185–196, 2002.

[15] X. Li, B. Li, B. Yang, and T. Zeng, "General framework to histogram-shifting-based reversible data hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2181–2191, 2013.

[16] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[17] 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.

[18] X. P. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

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

[20] C. Qin and X. Zhang, "Effective reversible data hiding in encrypted image with privacy protection for image content," *Journal of Visual Communication and Image Representation*, vol. 31, pp. 154–164, 2015.

[21] X. P. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, 2012.

[22] Z. Qian and X. Zhang, "Reversible Data Hiding in Encrypted Images with Distributed Source Encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 636–646, 2016.

[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 Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 441–452, 2016.

[24] F. Huang, J. Huang, and Y.-Q. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2777–2789, 2016.

[25] 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*, vol. 8, no. 3, pp. 553–562, 2013.

[26] W. M. Zhang, K. D. Ma, and N. H. Yu, "Reversibility improved data hiding in encrypted images," *Signal Processing*, vol. 94, no. 1, pp. 118–127, 2014.

[27] 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*, vol. 46, no. 5, pp. 1132–1143, 2016.

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

[29] Y.-C. Chen, C.-W. Shiu, and G. Horng, "Encrypted signal-based reversible data hiding with public key cryptosystem," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1164–1170, 2014.

[30] D. W. Xu, K. Chen, R. D. Wang, and S. B. Su, "Completely separable reversible data hiding in encrypted images," in *International Workshop on Digital-forensics and Watermarking (IWDW 2015), Tokyo, Japan*, vol. 9569 of *LNCS*, pp. 365–377, 2016.

[31] C.-W. Shiu, Y.-C. Chen, and W. Hong, "Encrypted image-based reversible data hiding with public key cryptography from difference expansion," *Signal Processing: Image Communication*, vol. 39, pp. 226–233, 2015.

[32] 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*, vol. 26, no. 9, pp. 1622–1631, 2016.

[33] H.-T. Wu, Y.-M. Cheung, and J. Huang, "Reversible data hiding in Paillier cryptosystem," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 765–771, 2016.

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

[35] B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5010–5021, 2013.

[36] Test Images (Online), Available: http://www.hlevkin.com/TestImages/.

[37] D. Xu and R. Wang, "Efficient reversible data hiding in encrypted H.264/AVC videos," *Journal of Electronic Imaging*, vol. 23, no. 5, Article ID 053022, 2014.