Secret sharing based reversible data hiding in encrypted images with multiple data-hiders

Bing Chen, Wei Lu, *Member, IEEE*, Jiwu Huang, *Fellow, IEEE*, Jian Weng, *Member, IEEE*, Yicong Zhou, *Senior Member, IEEE*

Abstract—The existing models of reversible data hiding in encrypted images (RDH-EI) are based on single data-hider, where the original image cannot be reconstructed when the data-hider is damaged. To address this issue, this paper proposes a novel model with multiple data-hiders for RDH-EI based on secret sharing. It divides the original image into multiple different encrypted images with the same size of the original image and distributes them to multiple different data-hiders for data hiding. Each data-hider can independently embed data into the encrypted image to obtain the corresponding marked encrypted image. The original image can be losslessly recovered by collecting sufficient marked encrypted images from undamaged data-hiders when individual data-hiders are subjected to potential damage. This further protects the security of the original image. We provide four cases of the proposed model, namely, two joint cases and two separable cases. From the proposed model, we derive a separable RDH-EI method with high-capacity. Experimental results are presented to illustrate the effectiveness of the proposed method.

Index Terms—Secret sharing, reversible data hiding, encrypted images, multiple data-hiders.

1 Introduction

Reversible data hiding (RDH) is a significant research topics in multimedia security that can provide copyright identification and integrity certification for multimedia in third-party platforms, such as outsourced storage in the cloud. This technique ensures that the cover can be losslessly recovered after the extraction of the embedded data. Due to its important reversibility, the RDH technique is also used for distortion-unacceptable covers, such as military, medical, and legal forensic images. The existing RDH methods are mainly based on three fundamental strategies: lossless compression [1], difference expansion [2], and histogram shifting [3]. Among them, the histogram shifting based methods have been widely investigated and developed in difference histogram shifting [4], [5] and prediction-error histogram shifting [6], [7]. Almost all recent RDH methods first construct a prediction-error histogram with high peak and then reversibly embed data into it by using histogram shifting.

Due to the need for image privacy-preserving, content-owner is reluctant to display images to data-hider, especially those images that contain sensitive information. Encryption is an effective and common technique for protecting image privacy, as it converts a meaningful image into a meaningless image that is difficult for any unauthorized user to recognize. Therefore, it is necessary to design reversible data hiding in encrypted images (RDH-EI). For instance, in cloud storage, the content-owner desires to store the image in the cloud with privacy-preserving by encrypting the image before uploading it to the cloud. For management purposes, the cloud service provider will embed some additional data into the encrypted image without accessing the content. On the receiver side, the authorized receiver can perfectly restore the original image after decryption and data extraction as needed.

The pixel correlation of an image is broken after encryption; however if the image is encrypted by an encryption algorithm that does not change pixel location or pixel block location, the pixel correlation still exists, such as the advanced encryption standard (AES) and stream cipher. By utilizing the correlation, data extraction and image recovery can be implemented. In [8], Puech et al. introduced an RDH-EI method that encrypts the original image by the AES and embeds one message bit into an AES encrypted block by bit-plane replacement. The embedded data can be extracted by directly reading the bit of replaced location. The original image is restored by calculating the local standard deviation of decrypted blocks. The pixel flipping based method was proposed by Zhang [9], in which the original image is encrypted by stream cipher, and one message bit is concealed into one stream-ciphered block by flipping the three least significant bits (LSBs) of pixels. The embedded data and the original image are jointly restored by estimating the texture

This work is supported by the National Natural Science Foundation of China (No. U1736118, No. U19B2022, and U1636202), the Key Areas R&D Program of Guangdong (No. 2019B010136002 and No. 2019B010139003), the Key Project of Scientific Research Plan of Guangzhou (No. 201804020068), and Shenzhen R&D Program (GJHZ20180928155814437). (Corresponding author: Wei Lu)

Bing Chen and Wei Lu are with the School of Data and Computer Science, Guangdong Key Laboratory of Information Security Technology, Ministry of Education Key Laboratory of Machine Intelligence and Advanced Computing, Sun Yat-sen University, Guangzhou 510006, China (e-mail: chenbing75@mail.sysu.edu.cn; luwei3@mail.sysu.edu.cn).

Jiwu Huang is with the Guangdong Key Laboratory of Intelligent Information Processing, Key Laboratory of Media Security, and Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, China, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China. (e-mail: jwhuang@szu.edu.cn).

Jian Weng is with the College of Information Science and Technology and the College of Cyber Security, Jinan University, Guangzhou 510632, China (e-mail: cryptjweng@gmail.com).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@um.edu.mo).

complexity of each block. Since then, many improvements have been made, and these methods focused on enhancing data extraction accuracy [10], [11], separating data extraction and image recovery [12], [13], vacating room before encryption [14], [15], [16], [17], [18], [19], [20], [21], and the implementation of other covers [22], [23], [24], [25]. All of them were designed to increase the embedding capacity as high as possible while maintaining low distortion.

In recent years, homomorphism based methods were proposed that can directly operate on encrypted image to achieve desired result. In general, the original image is first encrypted by homomorphic encryption (e.g., Paillier encryption [26]), and then, a secret message is embedded into the encrypted image by homomorphic property. Finally, data extraction and image recovery are realized by judging a certain relationship between the directly decrypted image and original image. Chen et al. [27] proposed that one message bit is concealed into one pixel pair via modifying the encrypted parity-check bits. By comparing the decrypted parity-check bits, the embedded message bit can be extracted and the original image can be reconstructed. Shiu et al. [28] improved the pixel overflow in [27] by using difference expansion and introducing a location map to record the location of unsuitable pixel pairs. In [29], to deal with pixel overflow, the LSBs of the reference pixel are set to zero, and then the encrypted reference pixel is mirrored to the host pixel. For achieving better rate-distortion performance, a method using histogram shifting into homomorphic encryption was proposed [30]. Moreover, some methods [31], [32], [33], [34] can even obtain lossless recovery of the decrypted image, i.e., the directly decrypted image is identical to the original image. However, in order to ensure the scheme security, homomorphic encryption must be performed with long-bit key, which suffers from severe data expansion and high computational complexity.

To reduce data expansion and computational complexity, secret sharing based RDH-EI methods were introduced. In [35], Wu et al. proposed a secret sharing based RDH-EI method that encrypts the original image to generate encrypted images by using Shamir's threshold secret sharing [36], and distributes the encrypted images to a data-hider for data hiding. In the best case, the size of the encrypted images is only twice as large than that of the original image. Another secret sharing based RDH-EI method was presented in [37], where an original image is converted into an encrypted image and the encrypted image is distributed to a data-hider for data hiding. Since the size of generated encrypted image is same as that of the original image, data expansion does not occur, and computational complexity is improved.

However, all the existing RDH-EI methods work on single data-hider based model. Once the data-hider is subjected to potential damage, such as improper management by itself or malicious attack by an adversary, the original image cannot be reconstructed from the marked encrypted images. In this paper, we propose a novel model with multiple data-hiders for RDH-EI by adopting secret sharing. Different from previous models, the proposed model divides the original image into multiple encrypted images with the same size of the original image and distributes the encrypted images to multiple different data-hiders for data hiding. Each data-hider can independently embed data into the encrypted image to obtain the corresponding marked encrypted image. Since the proposed model contains multiple data-hiders, the original image can be restored by collecting sufficient marked encrypted images from undamaged data-hiders even if some of the data-hiders are subjected to potential damage. We present four cases of the proposed model, that is, two joint cases and two separable cases. We implement a separable case called content-owner-independent and data-hider-independent. The proposed separable RDH-EI method can achieve higher embedding capacity compared with the state-of-the-art methods.

The remainder of this paper is organized as follows. In Section 2, two previous models with single data-hider for RDH-EI are described. The proposed model with multiple data-hiders based on secret sharing is given in Section 3. Section 4 elaborates on a separable RDH-EI method with high-capacity using the proposed model. Experimental results and discussions are illustrated in Section 5. Finally, Section 6 gives some concluding remarks.

2 Previous Models

Until now, the existing RDH-EI methods work on the two models with single data-hider: the traditional model and the secret sharing based model, as shown in Fig. 1. Both of them contain three participants: the content-owner, the data-hider, and the receiver, which perform the image encryption phase, the data hiding phase, and the data extraction and image recovery phase, respectively. For the traditional model, an original image is converted into an encrypted image, and the encrypted image is distributed to a data-hider for data hiding. Alternatively, by the secret sharing based model, the original image is converted into multiple encrypted images, and the encrypted images are distributed to a data-hider for data hiding.

2.1 Traditional model

The flowchart of the traditional model is depicted in Fig. 1(a). In the image encryption phase, the content-owner utilizes an encryption key ke to encrypt the original image by an encryption algorithm (e.g., stream cipher or homomorphic encryption), and distributes the encrypted image to a data-hider for data hiding. In the data hiding phase, with a data hiding key kh, the data-hider embeds data into an encrypted image to generate a marked encrypted image. In the data extraction and image recovery phase, an authorized receiver can perform data extraction and image recovery with the data hiding key kh and the decryption key kd.

In general, according to the domain of data extraction, the RDH-EI methods can be divided into two categories: joint methods and separable methods. For the joint methods [9], [11], [32], the embedded data is extracted in the plaintext domain, in which the marked encrypted image is decrypted by using the decryption key kd before data extraction. For the separable methods [12], [13], [29], [31], the embedded data is extracted in the encrypted domain. That is to say, the embedded data can be directly extracted from the marked encrypted image by using the data hiding key kh.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3011923, IEEE Transactions on Dependable and Secure Computing

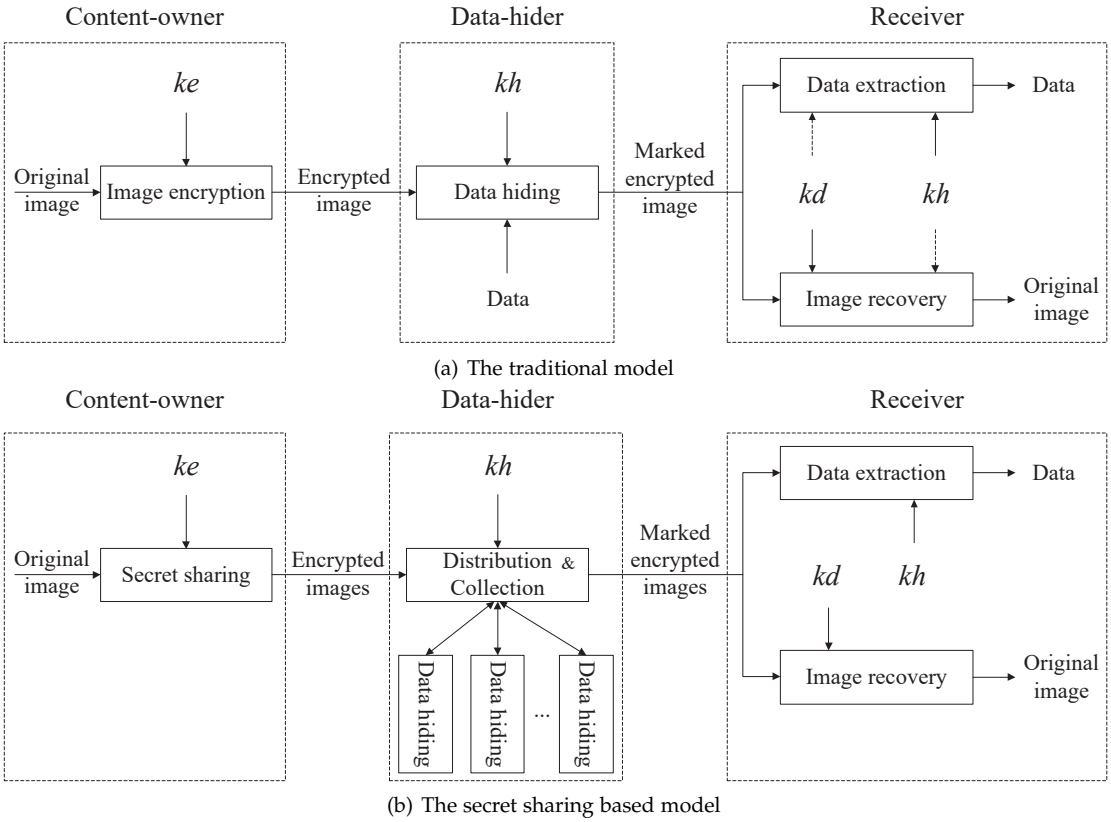


Fig. 1. Two existing RDH-EI models with single data-hider, where the dotted arrow indicates optional.

2.2 Secret sharing based model with single data-hider

A secret sharing based model with single data-hider for RDH-EI is proposed by Wu et al. [35], as shown in Fig. 1(b). The content-owner encrypts an original image into multiple encrypted images by Shamir's k-out-of-n threshold secret sharing with encryption key ke. Here k and n are integers satisfying $2 \leq k \leq n$. With Shamir's secret sharing, a prime p (e.g., 251 for an 8-bit gray-scale image) is picked and a $k-1$ degree polynomial $F(x)$ over finite field F_p is constructed by

$$F(x) = \left(a^{(0)} + \sum_{\alpha=1}^{k-1} a^{(\alpha)} x^{\alpha} \right) \bmod p, \tag{1}$$

where $a^{(0)} \in F_p$ is a constant term that is used to share pixel and for $\alpha = 1, 2, \cdots, k-1, a^{(\alpha)} \in F_p$ are randomly chosen integers. Substitute n different nonzero integers into x in Eq. (1), respectively, to obtain n encrypted images. Then the generated encrypted images are distributed to a data-hider for data hiding. In this model, the data-hider has a control center and multiple storage and processing centers. The control center performs the distribution and collection of encrypted images, while the storage and processing centers perform data hiding. It was assumed that no outside adversary can access the control center, which means that the security is based on the control center. In other words, once the control center is damaged, the original image cannot be reconstructed, which is same as the traditional model.

For the above models, there is only single data-hider. If this data-hider is subjected to potential damage, the original image cannot be reconstructed. To address this issue, the next section provides a novel model with multiple data-hiders for RDH-EI by adopting secret sharing.

3 Secret sharing based model with multiple data-hiders

As depicted in Fig. 2, the proposed secret sharing based model with multiple data-hiders is comprised of three phases: the image encryption phase, the data hiding phase, and the data extraction and image recovery phase. Different from previous models that only involve single data-hider, the proposed model involves multiple data-hiders. In the proposed model, the original image is converted into multiple encrypted images of the same size as the original image, and the encrypted images are distributed to multiple different data-hiders for data hiding. Each data-hider can independently embed data into the encrypted image to obtain the corresponding marked encrypted image. On the receiver side, the original image is reconstructed from a certain number of marked encrypted images, as well as the embedded data.

Next, we formally introduce each phase of the proposed model. In the image encryption phase, the original image I is divided into n encrypted images by secret sharing with an encryption key ke, and each encrypted image is distributed to the associated data-hider. The image encryption procedure is formulated by

$$\left(E^{(1)}, E^{(2)}, \cdots, E^{(n)} \right) = Enc_{ke}\left(I \right), \tag{2}$$

where $Enc_{ke}(*)$ is the image encryption algorithm with encryption key ke and $E^{(t)}, 1 \leq t \leq n$ is the t-th encrypted

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3011923, IEEE Transactions on Dependable and Secure Computing

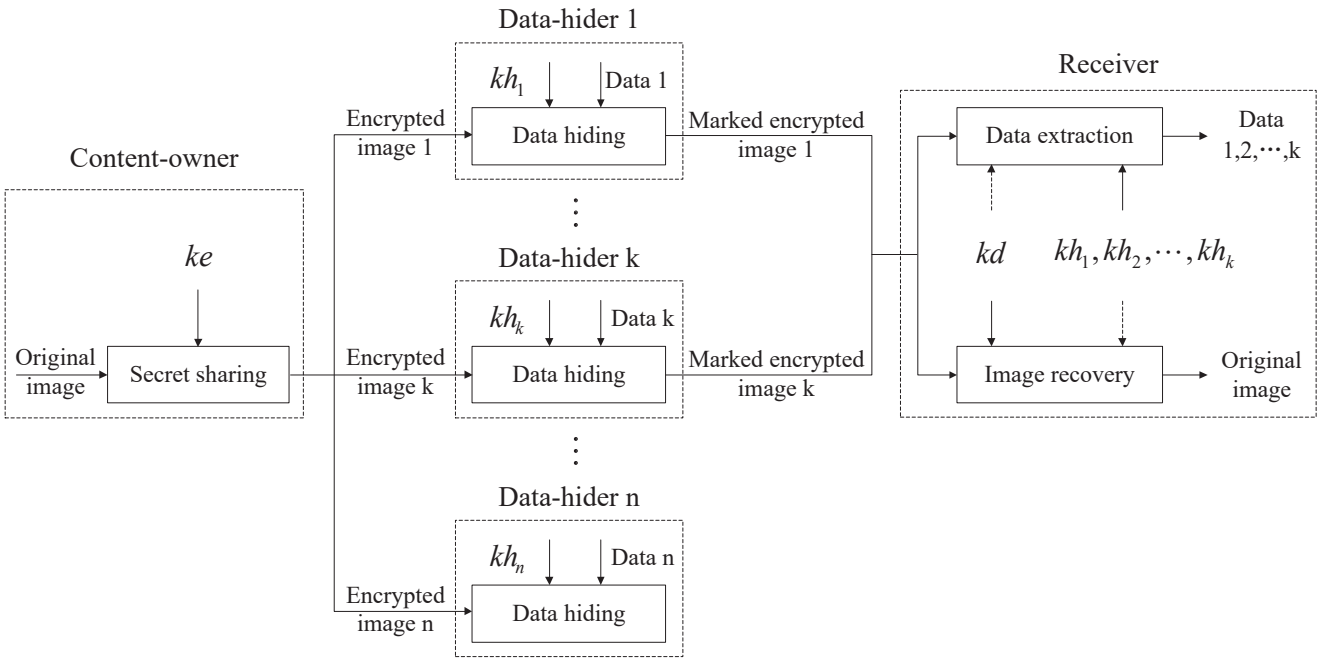


Fig. 2. The proposed secret sharing based model with multiple data-hiders, where the dotted arrow indicates optional.

image. Then $E^{(t)}$ is distributed to the t-th data-hider for data hiding. For encrypted image $E^{(t)}$, the data can be embedded into it by the t-th data-hider with data hiding key kh_t. The associated marked encrypted image is obtained by

$$EM^{(t)} = Emb_{kh_t}\left(E^{(t)}, D^{(t)}\right), \tag{3}$$

where $Emb_{kh_t}(*)$ is the data hiding algorithm with data hiding key kh_t, $D^{(t)}$ is the t-th embedded data, and $EM^{(t)}$ is the t-th marked encrypted image. When having any k $(2 \leq k \leq n)$ or more marked encrypted images, the receiver can reconstruct the original image, as well as the k or more embedded data. Let any k or more marked encrypted images be $EM^{(t_1)}, EM^{(t_2)}, \cdots, EM^{(t_r)}$, where $\{t_1, t_2, \cdots, t_r\} \subseteq \{1, 2, \cdots, n\}, k \leq r \leq n$. Similar to the traditional model, the RDH-EI methods using the proposed model can also be divided into joint methods and separable methods.

Depending on whether the data hiding key $kh_{t_e}, 1 \leq e \leq r$ is required in the image recovery procedure, the data extraction and image recovery of the joint methods can be summarized by the following two cases,

$$\begin{cases} D^{(t_e)} = Ext_{kh_{t_e}}\left(EM^{(t_e)}, kd\right), 1 \leq e \leq r, \\ I = Rec_{kd}\left(EM^{(t_1)}, \cdots, EM^{(t_r)}\right), \end{cases} \tag{4}$$

and

$$\begin{cases} D^{(t_e)} = Ext_{kh_{t_e}}\left(EM^{(t_e)}, kd\right), 1 \leq e \leq r, \\ I = Rec_{kd}\left((EM^{(t_1)}, \cdots, EM^{(t_r)}), (kh_{t_1}, \cdots, kh_{t_r})\right), \end{cases} \tag{5}$$

where $Ext_{kh_{t_e}}(*)$ is the data extraction algorithm with data hiding key kh_{t_e}, $Rec_{kd}(*)$ is the image recovery algorithm with decryption key kd, and $D^{(t_e)}$ is the t_e-th extracted data. In joint methods, data extraction cannot be performed in the encrypted domain. Marked encrypted images need to be decrypted with decryption key kd before data extraction, which indicates that the data extraction is related to the content-owner. In this scenario, the receiver needs to obtain permission from content-owner when verifying the integrity of the marked encrypted image. For the joint method shown in Eq. (4), the data hiding key kh_{t_e} is not required in image recovery, which means that image recovery is independent of the data-hider. Therefore, the method shown in Eq. (4) is content-owner-related and data-hider-independent. For the joint method shown in Eq. (5), the data hiding key kh_{t_e} is required, which means that image recovery is related to the data-hider. Therefore, the method shown in Eq. (5) is content-owner-related and data-hider-related.

In the light of this idea, two cases of data extraction and image recovery in separable methods are considered, there are

$$\begin{cases} D^{(t_e)} = Ext_{kh_{t_e}}\left(EM^{(t_e)}\right), 1 \leq e \leq r, \\ I = Rec_{kd}\left(EM^{(t_1)}, \cdots, EM^{(t_r)}\right), \end{cases} \tag{6}$$

and

$$\begin{cases} D^{(t_e)} = Ext_{kh_{t_e}}\left(EM^{(t_e)}\right), 1 \leq e \leq r, \\ I = Rec_{kd}\left((EM^{(t_1)}, \cdots, EM^{(t_r)}), (kh_{t_1}, \cdots, kh_{t_r})\right). \end{cases} \tag{7}$$

In separable methods, the embedded data is directly extracted from the marked encrypted image without decryption key kd, which indicates that the data extraction is independent of the content-owner. In this scenario, the data-hider can update embedded data with a data hiding key as needed. For the separable method shown in Eq. (6), the data hiding key kh_{t_e} is not required in image recovery, which means that image recovery is independent of the data-hider. Therefore, the method represented by Eq. (6) is content-owner-independent and data-hider-independent. Similarly, it is known that the method represented by Eq. (7) is content-owner-independent and data-hider-related.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3011923, IEEE Transactions on Dependable and Secure Computing

Based on the above analysis, the characteristics of this novel model are summarized as follows.

1) The idea of secret sharing is employed in RDH-EI, by which the content-owner can distribute the encrypted images to multiple different data-hiders for data hiding. Thus, even if a portion of the data-hiders are damaged, the original image can still be reconstructed by collecting sufficient marked encrypted images from undamaged data-hiders, which further protects the security of the original image.
2) The model involves multiple data-hiders, each of which can independently manage the corresponding encrypted image and independently communicate with the receiver.
3) Although the model generates multiple encrypted images, the size of each generated encrypted images is same as that of original image, so no data expansion occurs for each data-hider.

4 Separable RDH-EI method with high-capacity using the proposed model

In this section, a separable RDH-EI method with high-capacity based on the proposed model is designed, in which the data extraction and image recovery in Eq. (6) are implemented. That is, the embedded data is extracted in the encrypted domain, and the image recovery does not require the data hiding key. The proposed method consists of three phases: the pixel shrink and image encryption phase, the data hiding phase, and the data extraction and image recovery phase. In the pixel shrink and image encryption phase, the pixels that are not suitable for secret sharing are shrunk to generate a shrunk image. The shrunk image is then converted into multiple encrypted images with the same size of the original image by threshold secret sharing. Finally, the encrypted images are distributed to multiple different data-hiders for data hiding. In the data hiding phase, each data-hider embeds a secret message into the encrypted image to obtain the associated marked encrypted image by bit-plane replacement. On the receiver side, when sufficient marked encrypted images are collected, the original image can be restored, as well as the embedded secret message.

4.1 Pixel shrink and image encryption

Assume that the original image I is an 8-bit gray-scale image with size $M_1 \times N_1$ and $I_{i,j}$ is the pixel value at location (i,j), where $I_{i,j} \in [0,255], 1 \le i \le M_1, 1 \le j \le N_1$. To encrypt the image, Shamir's k-out-of-n threshold secret sharing with simple modification is introduced. Because Shamir's secret sharing is constructed over finite field F_p, the pixels with $I_{i,j} \ge 251$ are not suitable for secret sharing. To this end, we need to shrink the original image before encryption. Specifically, the pixel shrink is described as follows.

1) The original image I is divided into two parts A and B, where A is composed of the first ten pixels and B is the other pixels, as shown in Fig. 3. The first two pixels of A are used to carry parameters for data hiding, and the rest of A is used to carry parameters for the inverse process of histogram shifting.

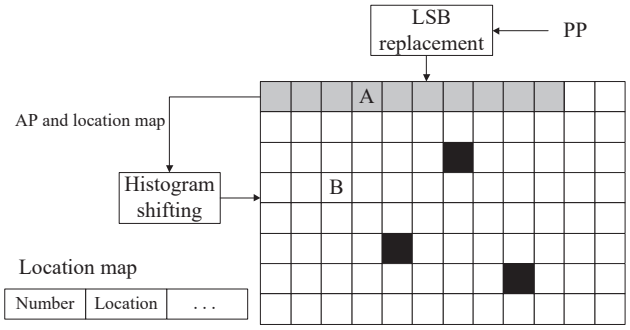


Fig. 3. Image segmentation and self-embedding in pixel shrink, where black pixels represent that their pixel values satisfy $I_{i,j} \ge 250$.

2) Extract all bits of the first two pixels of A and three LSBs of the last eight pixels of A, denoted by AP. In order to avoid overflow in secret sharing, the original three LSBs of A are set to zero.
3) Scan B along the rows, and record the number and location of pixels with $I_{i,j} \ge 250$ by using a location map. Then, set the associated pixel value to 249.
4) Embed AP and the location map into B using histogram shifting based RDH. That is, construct the histogram of B, and search for a proper embedding point, denoted by PP, whose size is closest to the size of AP and the location map. Then, shift all values between $PP+1$ and 249 with one step toward the right. Finally, by representing the bit 0 with PP and the bit 1 with $PP+1$, AP and the location map can be embedded into PP and $PP+1$. Fig. 4 shows the illustration of the parameter embedding.
5) Embed PP into the LSB of the last eight pixels of A using LSB replacement.

As a result, a shrunk image I' of the original image I is obtained and its pixels are constrained in the range of $[0,250]$, which is suitable for secret sharing.

After that, shrunk image I' is encrypted. For the pixel value $I'_{i,j}$ of the shrunk image I' at location (i,j), the following polynomial is constructed,

$$F_{i,j}(x) = \begin{cases} \left(T_{i,j}^{(0)} + I'_{i,j}x\right) \bmod p, & \text{if } k=2, \\ \left(T_{i,j}^{(0)} + I'_{i,j}x + \sum_{\alpha=2}^{k-1} a_{i,j}^{(\alpha)} x^{\alpha}\right) \bmod p, & \text{if } 2<k\le n, \end{cases} \tag{8}$$

where $T_{i,j}^{(0)} \in F_p$ and $a_{i,j}^{(\alpha)} \in F_p$ are a constant term and a randomly chosen integer, respectively. Constant term $T_{i,j}^{(0)}$ is determined by encryption key ke. Different from the method in [35] where pixels are shared by using the constant term, we here use one term to share pixels. The constant term $T_{i,j}^{(0)}$ is used for image recovery in the proposed method, which will be described in detail in Section 4.3. For location (i,j), according to the encryption key ke, the content-owner randomly generates n nonzero integers $x_{i,j}^{(t)} \in F_p, t=1,2,\cdots,n$, that are distinct from one another. Then the nonzero integer $x_{i,j}^{(t)}$ is substituted into Eq. (8) to obtain the associated encrypted result $F_{i,j}(x_{i,j}^{(t)})$. When all pixels are encrypted, n encrypted images are obtained. In addition, to tell the data-hider where the data can be embedded, parameters t and n are embedded into the encrypted

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3011923, IEEE Transactions on Dependable and Secure Computing

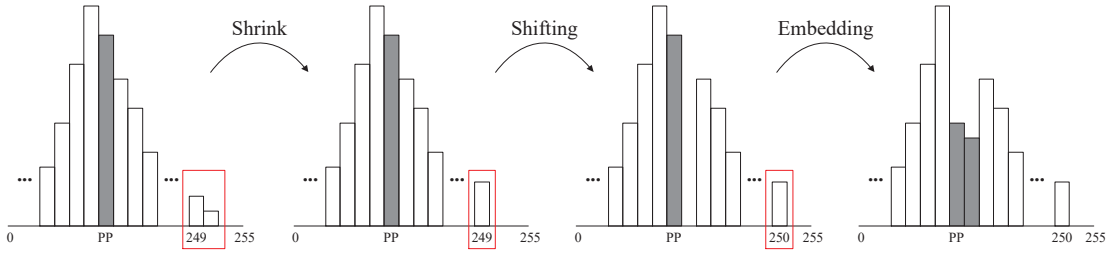


Fig. 4. The illustration of the parameter embedding in pixel shrink.

images. That is, the first two pixels of the t-th encrypted image are replaced by t and n, respectively. Thus, the final n encrypted images of the original image are generated and distributed to n different data-hiders.

4.2 Data hiding

When the t-th data-hider obtains the associated encrypted image $E^{(t)}$, the secret message can be independently embedded into it to generate the t-th marked encrypted image $EM^{(t)}$. First of all, the t-th data-hider reads the first two pixels of encrypted image $E^{(t)}$ to obtain the parameters t and n. According to n, the rest of the encrypted image $E^{(t)}$ is divided into a number of non-overlapping groups in raster-scan order, each of which contains n pixels. Then, with a bit-plane replacement, data hiding can be implemented. That is, the l $(1 \leq l \leq 7)$ LSBs of the t-th pixel in each group are replaced with the secret message bits, where the secret message bits are obtained by encrypting the original message bits with data hiding key kh_t. When all the bits of the secret message are embedded, the associated marked encrypted image $EM^{(t)}$ is obtained. Since l bits are embedded into each group, the total $\left\lfloor \frac{M_1 N_1 - 2}{n} \right\rfloor \cdot l$ bits can be accommodated in all groups, where $\lfloor * \rfloor$ is the floor function. Therefore, the embedding rate can be calculated by

$$\text{Embedding rate} = \frac{\left\lfloor \frac{M_1 N_1 - 2}{n} \right\rfloor \cdot l}{M_1 N_1} \approx \frac{l}{n}. \tag{9}$$

Similarly, when all n data-hiders accomplish their data hiding, n marked encrypted images are generated.

4.3 Data extraction and image recovery

When any k marked encrypted images are collected, the original image can be restored, as well as the embedded data. Data extraction is separated from the image restoration, that is, the embedded data is extracted in the encrypted domain. Without loss of generality, assume k marked encrypted images $EM^{(t_z)}, z = 1, 2, \cdots, k, \{t_1, t_2, \cdots, t_k\} \subseteq \{1, 2, \cdots, n\}$, are collected. To achieve data extraction, for $z = 1, 2, \cdots, k$, the receiver first obtains the parameters t_z and n from the first two pixels of the t_z-th marked encrypted images. After knowing the group size n, the receiver divides k marked encrypted images beyond the first two pixels into a number of groups with n pixels and extracts the secret message from the l LSBs of the t_z-th pixel in each group. Finally, the original message is restored by decrypting the extracted secret message with data hiding key kh_{t_z}.

On the other hand, if the receiver holds the decryption key kd, the original image can be recovered from the k marked encrypted images $EM^{(t_z)}, z = 1, 2, \cdots, k$. According to the decryption of Shamir's secret sharing, the associated decryption of Eq. (8) is to reconstruct a $k-1$ degree polynomial $F_{i,j}(x)$ by

$$F_{i,j}(x) = \sum_{\beta=1}^{k} \left(F_{i,j}(x_{i,j}^{(t_\beta)}) \prod_{\alpha=1, \alpha \neq \beta}^{k} \frac{x - x_{i,j}^{(t_\alpha)}}{x_{i,j}^{(t_\beta)} - x_{i,j}^{(t_\alpha)}} \right) \bmod p, \tag{10}$$

where $t_\alpha, t_\beta \in \{1, 2, \cdots, n\}, F_{i,j}(x_{i,j}^{(t_\beta)})$ are the t_β-th encrypted result, and nonzero integers $x_{i,j}^{(t_\alpha)}$ and $x_{i,j}^{(t_\beta)}$ are generated by encryption key ke. Note that the encryption key ke here is equal to the decryption key kd. Then, $I'_{i,j}$ is recovered from the coefficient of the one term in $F_{i,j}(x)$. For each group of $EM^{(t_z)}$, the l LSBs of the t_z-th pixel are replaced and the $8-l$ most significant bits (MSBs) of the t_z-th pixel are unchanged in the data hiding phase. Thus the value of the t_z-th encrypted pixel must be one of $Y_{i,j}^{(t_z)}(\gamma)$, for $\gamma = 0, 1, \cdots, 2^l - 1$, which satisfies

$$Y_{i,j}^{(t_z)}(\gamma) = \sum_{w=l}^{7} \left(\lfloor EM_{i,j}^{(t_z)} / 2^w \rfloor \bmod 2 \right) \cdot 2^w + \gamma, \tag{11}$$

where the first part is the unchanged $8-l$ MSBs, the rest part γ is the replaced l LSBs, and $EM_{i,j}^{(t_z)}$ is the pixel value of $EM^{(t_z)}$ at location (i, j). In fact, the value of $Y_{i,j}^{(t_z)}(\gamma)$ may exceed 250 due to data hiding. In this case, we only need to exclude the value of $Y_{i,j}^{(t_z)}(\gamma)$ exceeding 250. For simplicity, we consider that the value of $Y_{i,j}^{(t_z)}(\gamma)$ does not exceed 250.

Besides, from Eqs. (8) and (10), we have

$$\begin{aligned} T_{i,j}^{(0)} &= F_{i,j}(0) \\ &= \sum_{\beta=1}^{k} \left(F_{i,j}(x_{i,j}^{(t_\beta)}) \prod_{\alpha=1, \alpha \neq \beta}^{k} \frac{x_{i,j}^{(t_\alpha)}}{x_{i,j}^{(t_\alpha)} - x_{i,j}^{(t_\beta)}} \right) \bmod p. \end{aligned} \tag{12}$$

Let $X_{i,j}^{(t_\beta)} = \prod_{\alpha=1, \alpha \neq \beta}^{k} \frac{x_{i,j}^{(t_\alpha)}}{x_{i,j}^{(t_\alpha)} - x_{i,j}^{(t_\beta)}} \bmod p$, Eq. (12) can be

reduced to

$$\begin{aligned} T_{i,j}^{(0)} &= \sum_{\beta=1}^{k} \Bigg(\Big(F_{i,j}(x_{i,j}^{(t_\beta)}) \bmod p \Big) \\ &\quad \cdot \Big(\prod_{\alpha=1,\alpha\neq\beta}^{k} \frac{x_{i,j}^{(t_\alpha)}}{x_{i,j}^{(t_\alpha)} - x_{i,j}^{(t_\beta)}} \bmod p \Big) \Bigg) \bmod p \\ &= \sum_{\beta=1}^{k} \Big(F_{i,j}(x_{i,j}^{(t_\beta)}) X_{i,j}^{(t_\beta)} \Big) \bmod p, \end{aligned} \tag{13}$$

where $F_{i,j}(x_{i,j}^{(t_\beta)}) \in F_p$. As defined in Eq. (11), the encrypted result $F_{i,j}(x_{i,j}^{(t_z)}), 1 \le t_z \le k$, must be one of $Y_{i,j}^{(t_z)}(\gamma)$. For $\gamma = 0, 1, \cdots, 2^l - 1$, applying Eq. (13), at least one of $T_{i,j}^{(0)}(\gamma)$ is equal to $T_{i,j}^{(0)}$, where

$$\begin{aligned} T_{i,j}^{(0)}(\gamma) = &\Big(F_{i,j}(x_{i,j}^{(t_1)}) X_{i,j}^{(t_1)} + \cdots + Y_{i,j}^{(t_z)}(\gamma) X_{i,j}^{(t_z)} + \cdots \\ &+ F_{i,j}(x_{i,j}^{(t_k)}) X_{i,j}^{(t_k)} \Big) \bmod p. \end{aligned} \tag{14}$$

Clearly, Eq. (14) can be conducted as

$$\begin{cases} T_{i,j}^{(0)}(1) = \Big(T_{i,j}^{(0)}(0) + X_{i,j}^{(t_z)} \Big) \bmod p, \\ T_{i,j}^{(0)}(2) = \Big(T_{i,j}^{(0)}(0) + 2X_{i,j}^{(t_z)} \Big) \bmod p, \\ \cdots \\ T_{i,j}^{(0)}(2^l - 1) = \Big(T_{i,j}^{(0)}(0) + (2^l - 1) X_{i,j}^{(t_z)} \Big) \bmod p, \end{cases} \tag{15}$$

which implies for $\gamma = 0, 1, \cdots, 2^l - 1$, $T_{i,j}^{(0)}(\gamma) = \Big(T_{i,j}^{(0)}(0) + \gamma X_{i,j}^{(t_z)} \Big) \bmod p$.

In the following, we will demonstrate that $T_{i,j}^{(0)}(\gamma)$ are distinct from one another for $\gamma = 0, 1, \cdots, 2^l - 1$. Assume there exists two values γ_1 and γ_2 such that $T_{i,j}^{(0)}(\gamma_1) = T_{i,j}^{(0)}(\gamma_2)$, that is,

$$\Big(T_{i,j}^{(0)}(0) + \gamma_1 X_{i,j}^{(t_z)} \Big) \bmod p = \Big(T_{i,j}^{(0)}(0) + \gamma_2 X_{i,j}^{(t_z)} \Big) \bmod p, \tag{16}$$

where $\gamma_1 \neq \gamma_2$ and $\gamma_1, \gamma_2 \in 0, 1, \cdots, 2^l - 1$. From Eq. (16), it is obtained that

$$T_{i,j}^{(0)}(0) + \gamma_1 X_{i,j}^{(t_z)} + k_1 p = T_{i,j}^{(0)}(0) + \gamma_2 X_{i,j}^{(t_z)} + k_2 p, \tag{17}$$

where $k_1, k_2 = 0, \pm 1, \pm 2, \cdots$. It follows from Eq. (17) that

$$(\gamma_1 - \gamma_2)\, X_{i,j}^{(t_z)} = (k_2 - k_1)\, p. \tag{18}$$

Since $X_{i,j}^{t_z} \in F_p$, we have

$$\begin{cases} \gamma_1 - \gamma_2 = p, \\ X_{i,j}^{(t_z)} = k_2 - k_1. \end{cases} \tag{19}$$

In fact, $\gamma_1, \gamma_2 \in 0, 1, \cdots, 2^l - 1, 1 \le l \le 7$. Thus we have $\gamma_1 - \gamma_2 \neq 251 = p$, which contradicts Eq. (19). This implies that $T_{i,j}^{(0)}(\gamma)$ are distinct from one another for $\gamma = 0, 1, \cdots, 2^l - 1$.

In other words, only one of $T_{i,j}^{(0)}(\gamma)$ is equal to $T_{i,j}^{(0)}$. Assume there is γ' $(0 \le \gamma' \le 2^l - 1)$ such that $T_{i,j}^{(0)}(\gamma') = T_{i,j}^{(0)}$, then the encrypted result $F_{i,j}(x_{i,j}^{(t_z)})$ can be determined by $F_{i,j}(x_{i,j}^{(t_z)}) = Y_{i,j}^{(t_z)}(\gamma')$. According to Eq. (10), the associated polynomial $F_{i,j}(x)$ can also be reconstructed. As a result, the pixel value of the shrunk image $I'_{i,j}$ is recovered from the coefficient of the one term in $F_{i,j}(x)$ as shown in Eq. (8).

Finally, the original image can be reconstructed from the shrunk image described below.

1) Extract PP from the LSB of the last eight pixels of A.
2) Construct the histogram of B and perform the inverse of histogram shifting using PP to get AP and the location map.
3) Replace the first two pixels of A and the three LSBs of the last eight pixels of A with AP.
4) Restore the original pixel values of overflow pixels in B according to the location map.

In the proposed method, the receiver can restore the original image by collecting any k or more marked encrypted images. If the receiver does not receive the marked encrypted image or the received marked encrypted image is incompetent, we assert that the data-hider is damaged. In the following, the details of how the proposed method works when facing data-hider damage are given.

Step 1. The receiver sends a request to the t_z-th data-hider to authorize the t_z-th marked encrypted image. If the receiver receives the marked encrypted image, go to Step 2; otherwise, go to Step 6.
Step 2. The receiver extracts the secret message from the marked encrypted image, and decrypts the secret message with data hiding key kh_{t_z} to obtain the message.
Step 3. The receiver verifies the message integrity of the marked encrypted image. If the verification is successful, go to Step 4; otherwise, go to Step 6.
Step 4. Determine whether the number of received marked encrypted images reaches k. If yes, the receiver stops sending requests to the data-hider; otherwise, go to Step 6.
Step 5. According to the proposed method, the original image is restored with k marked encrypted images.
Step 6. Let $z = z + 1$, and repeat Step 1 to Step 5.

4.4 An example of the proposed method

To better demonstrate the proposed method, an example is given in Fig. 5, in which 3-out-of-4 threshold secret sharing is used. Suppose a group with four original pixels is $(I_{1,1}, I_{1,2}, I_{1,3}, I_{1,4}) = (251, 248, 247, 250)$. Since there are pixels that are not suitable for secret sharing, the original pixels are modified to $(I'_{1,1}, I'_{1,2}, I'_{1,3}, I'_{1,4}) = (249, 248, 247, 249)$ by pixel shrink. According to the encryption key, constant term $T^{(0)} = (T_{1,1}^{(0)}, T_{1,2}^{(0)}, T_{1,3}^{(0)}, T_{1,4}^{(0)}) = (165, 159, 58, 46)$ and nonzero integers

$$\begin{cases} x^{(1)} = (x_{1,1}^{(1)}, x_{1,2}^{(1)}, x_{1,3}^{(1)}, x_{1,4}^{(1)}) = (39, 21, 113, 8) \\ x^{(2)} = (x_{1,1}^{(2)}, x_{1,2}^{(2)}, x_{1,3}^{(2)}, x_{1,4}^{(2)}) = (205, 47, 73, 201) \\ x^{(3)} = (x_{1,1}^{(3)}, x_{1,2}^{(3)}, x_{1,3}^{(3)}, x_{1,4}^{(3)}) = (14, 32, 192, 52) \\ x^{(4)} = (x_{1,1}^{(4)}, x_{1,2}^{(4)}, x_{1,3}^{(4)}, x_{1,4}^{(4)}) = (101, 48, 63, 145) \end{cases} \tag{20}$$

are generated. Then according to Eq. (8), the content-owner constructs four 2-degree polynomials with constant term

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3011923, IEEE Transactions on Dependable and Secure Computing

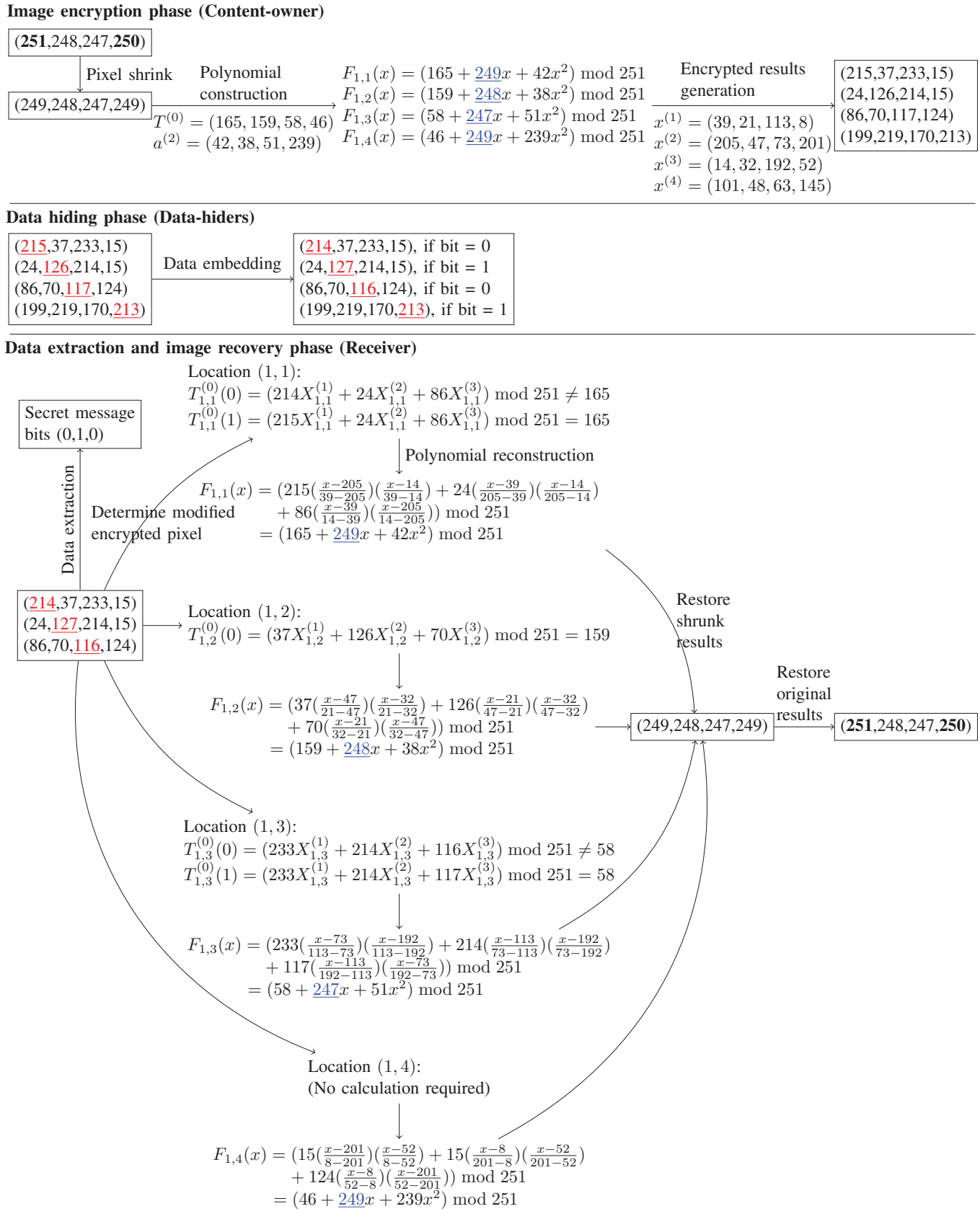


Fig. 5. Sketch of an RDH-EI example with 3-out-of-4 threshold secret sharing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3011923, IEEE Transactions on Dependable and Secure Computing

$T^{(0)}$ and random integer $a^{(2)} = (a_{1,1}^{(2)}, a_{1,2}^{(2)}, a_{1,3}^{(2)}, a_{1,4}^{(2)}) = (42, 38, 51, 239)$ over finite field F_{251}, that is,

$$\begin{cases} F_{1,1}(x) = (165 + 249x + 42x^2) \bmod 251, \\ F_{1,2}(x) = (159 + 248x + 38x^2) \bmod 251, \\ F_{1,3}(x) = (58 + 247x + 51x^2) \bmod 251, \\ F_{1,4}(x) = (46 + 249x + 239x^2) \bmod 251. \end{cases} \tag{21}$$

By substituting Eq. (20) into the generated polynomials from Eq. (21), the associated four encrypted results

$$\begin{cases} (F_{1,1}(x_{1,1}^{(1)}), F_{1,2}(x_{1,2}^{(1)}), F_{1,3}(x_{1,3}^{(1)}), F_{1,4}(x_{1,4}^{(1)})) \\ = (215, 37, 233, 15) \\ (F_{1,1}(x_{1,1}^{(2)}), F_{1,2}(x_{1,2}^{(2)}), F_{1,3}(x_{1,3}^{(2)}), F_{1,4}(x_{1,4}^{(2)})) \\ = (24, 126, 214, 15) \\ (F_{1,1}(x_{1,1}^{(3)}), F_{1,2}(x_{1,2}^{(3)}), F_{1,3}(x_{1,3}^{(3)}), F_{1,4}(x_{1,4}^{(3)})) \\ = (86, 70, 117, 124) \\ (F_{1,1}(x_{1,1}^{(4)}), F_{1,2}(x_{1,2}^{(4)}), F_{1,3}(x_{1,3}^{(4)}), F_{1,4}(x_{1,4}^{(4)})) \\ = (199, 219, 170, 213) \end{cases} \tag{22}$$

are obtained and respectively distributed to four different data-hiders for data hiding.

In the data hiding phase, a secret message can be embedded into the encrypted results by bit-plane replacement. For $t = 1, 2, 3, 4$, the t-th data-hider embeds the secret message bit into the t-th pixel of the t-th encrypted result. For example, the embedded bits are 0, 1, 0, and 1 for the 1st, 2nd, 3th, and 4th data-hiders, respectively. The encrypted results $F_{1,1}(x_{1,1}^{(1)})$, $F_{1,2}(x_{1,2}^{(2)})$, $F_{1,3}(x_{1,3}^{(3)})$, and $F_{1,4}(x_{1,4}^{(4)})$ are used to carry the embedded bits 0, 1, 0, and 1, respectively. Then the marked encrypted results

$$\begin{cases} (EM_{1,1}^{(1)}, EM_{1,2}^{(1)}, EM_{1,3}^{(1)}, EM_{1,4}^{(1)}) \\ = (214, 37, 233, 15) & \text{(23a)} \\ (EM_{1,1}^{(2)}, EM_{1,2}^{(2)}, EM_{1,3}^{(2)}, EM_{1,4}^{(2)}) \\ = (24, 127, 214, 15) & \text{(23b)} \\ (EM_{1,1}^{(3)}, EM_{1,2}^{(3)}, EM_{1,3}^{(3)}, EM_{1,4}^{(3)}) \\ = (86, 70, 116, 124) & \text{(23c)} \\ (EM_{1,1}^{(4)}, EM_{1,2}^{(4)}, EM_{1,3}^{(4)}, EM_{1,4}^{(4)}) \\ = (199, 219, 170, 213) & \text{(23d)} \end{cases}$$

are obtained from the corresponding encrypted results.

In the data extraction and image recovery phase, we assume that the three marked encrypted results Eqs. (23a), (23b), and (23c) are authorized. Obviously, the embedded secret message bits can be directly extracted by reading the LSB of the modified pixel in each marked encrypted result. To achieve image recovery, the receiver generates constant term $T^{(0)}$ and nonzero integers $x^{(1)}$, $x^{(2)}$, and $x^{(3)}$ by the encryption key, and calculates the corresponding $(X_{1,1}^{(1)}, X_{1,2}^{(1)}, X_{1,3}^{(1)}, X_{1,4}^{(1)})$, $(X_{1,1}^{(2)}, X_{1,2}^{(2)}, X_{1,3}^{(2)}, X_{1,4}^{(2)})$, and $(X_{1,1}^{(3)}, X_{1,2}^{(3)}, X_{1,3}^{(3)}, X_{1,4}^{(3)})$. For instance, to restore the original pixel at location $(1, 1)$, a constant term $T_{1,1}^{(0)}(0)$ is first generated by Eq. (14) with possible encrypted pixel $Y_{1,1}^{(1)}(0) = 214$. Because the generated constant term is not equal to the original constant term $T_{1,1}^{(0)} = 165$, another constant term $T_{1,1}^{(0)}(1)$ is generated with another possible encrypted pixel $Y_{1,1}^{(1)}(1) = 215$, which is equal to the original constant term. It is implied that the original encrypted pixel at location $(1, 1)$ is 215. Then, according to Eqs. (10) and (20), the polynomial $F_{1,1}(x)$ is reconstructed as

$$\begin{aligned} F_{1,1}(x) = \Big(&215(\frac{x-205}{39-205})(\frac{x-14}{39-14}) \\ &+ 24(\frac{x-39}{205-39})(\frac{x-14}{205-14}) \\ &+ 86(\frac{x-39}{14-39})(\frac{x-205}{14-205})\Big) \bmod 251 \\ = (&165 + 249x + 42x^2) \bmod 251. \end{aligned} \tag{24}$$

Thus, from the structure of the polynomial defined in Eq. (8), the shrunk pixel $I'_{1,1} = 249$ is obtained from the coefficient of the one term in $F_{1,1}(x)$. Similarly, the other three shrunk pixels at different locations can also be obtained. Finally, the original pixels $(I_{1,1}, I_{1,2}, I_{1,3}, I_{1,4}) = (251, 248, 247, 250)$ can be restored by the inverse of the pixel shrink.

TABLE 1
Size of $I_{i,j} \geq 250$ and peak point for different images.

Image	Pixel value						
	250	251	252	253	254	255	Peak point
Lena	0	0	0	0	0	0	2747
F-16	0	0	0	0	0	0	8312
Peppers	0	0	0	0	0	0	2749
Boat	3	5	3	1	0	2	5796
Sailboat	0	0	0	0	0	0	3707
Baboon	0	0	0	0	0	0	2762

5 Experimental Results and Discussions

Six test images including "Lena", "F-16", "Peppers", "Boat", "Sailboat", and "Baboon" from the standard dataset [38] are used for the experiments. All the test images are gray-scale images sized by 512×512.

5.1 Discussion on location map

The location map is used to record the number and location of pixels with $I_{i,j} \geq 250$, and is reversibly embedded into part B of the original image using histogram shifting. The size of the location map determines the effectiveness of the histogram shifting, which is also critical to the secret sharing of the proposed method. Table 1 illustrates the number of pixels with $I_{i,j} \geq 250$ and peak point in the six standard images. As can be seen, even for the image Boat, the location map is 300 bits (8 bits are used for number and 18 bits are used for location), and the peak point is enough to accommodate it. In the pixel shrink procedure, we only use the original histogram shifting to embed the location map. In fact, there are many improved methods for embedding the location map, such as prediction-error histogram shifting [6], [7], which can vacate more embedding room.

5.2 Feasibility of the proposed scheme

Example 1: In this example, 3-out-of-4 threshold secret sharing is adopted to encrypt test image Boat. The experiment result is shown in Fig. 6. Fig. 6(a) shows the original image, Fig. 6(b)–(e) show the four generated encrypted images. For each encrypted image, 65535 bits are embedded and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3011923, IEEE Transactions on Dependable and Secure Computing

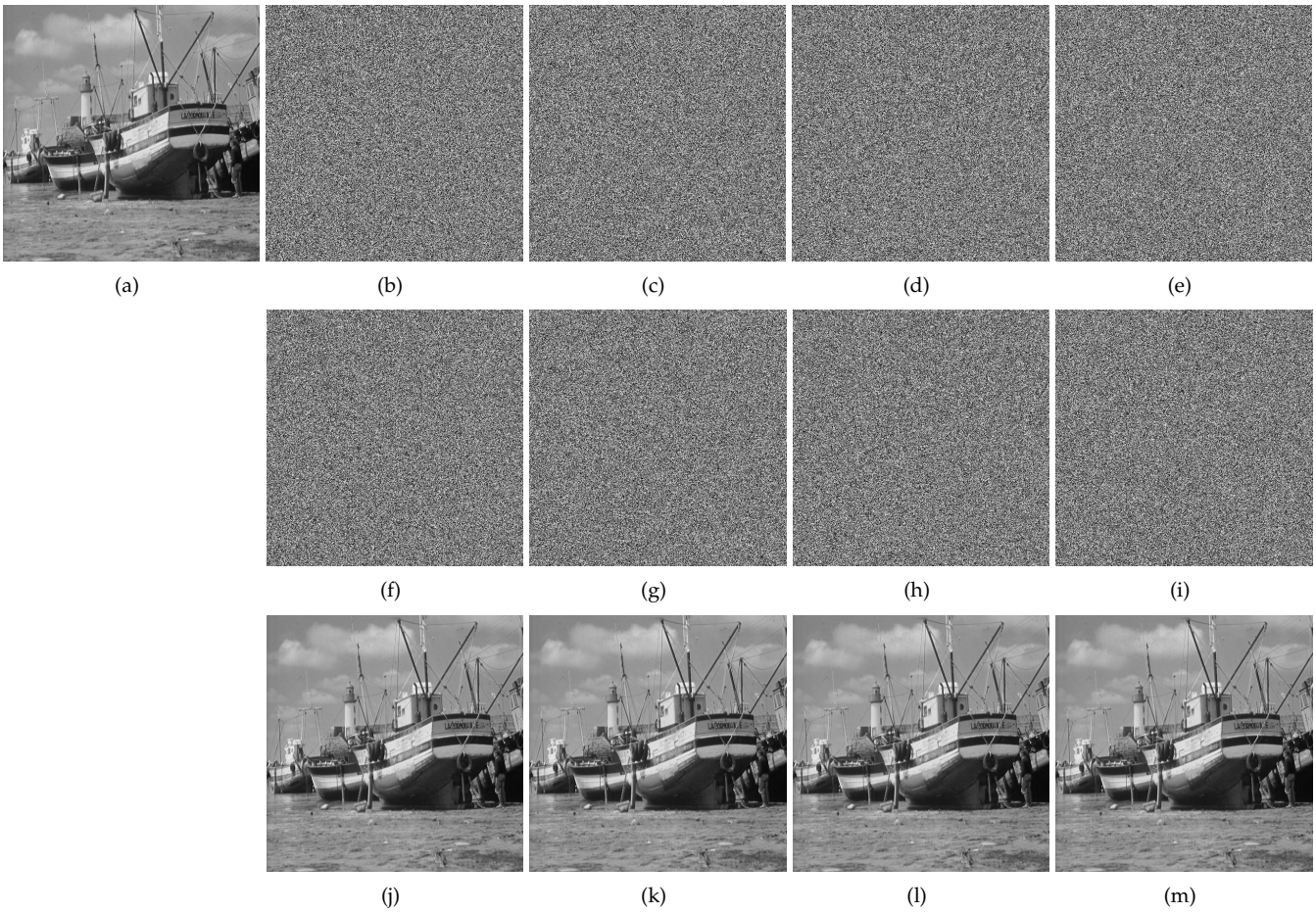


Fig. 6. Adopting 3-out-of-4 threshold secret sharing for the experiment. (a) Original image Boat, (b) 1st encrypted image, (c) 2nd encrypted image, (d) 3rd encrypted image, (e) 4th encrypted image, (f) 1st marked encrypted image with 65535 bits embedded, (g) 2nd marked encrypted image with 65535 bits embedded, (h) 3rd marked encrypted image with 65535 bits embedded, (i) 4th marked encrypted image with 65535 bits embedded, (j) Reconstructed image with PSNR = $+\infty$ dB from (f), (g), and (h), (k) Reconstructed image with PSNR = $+\infty$ dB from (f), (g), and (i), (l) Reconstructed image with PSNR = $+\infty$ dB from (f), (h), and (i), (m) Reconstructed image with PSNR = $+\infty$ dB from (g), (h), and (i).

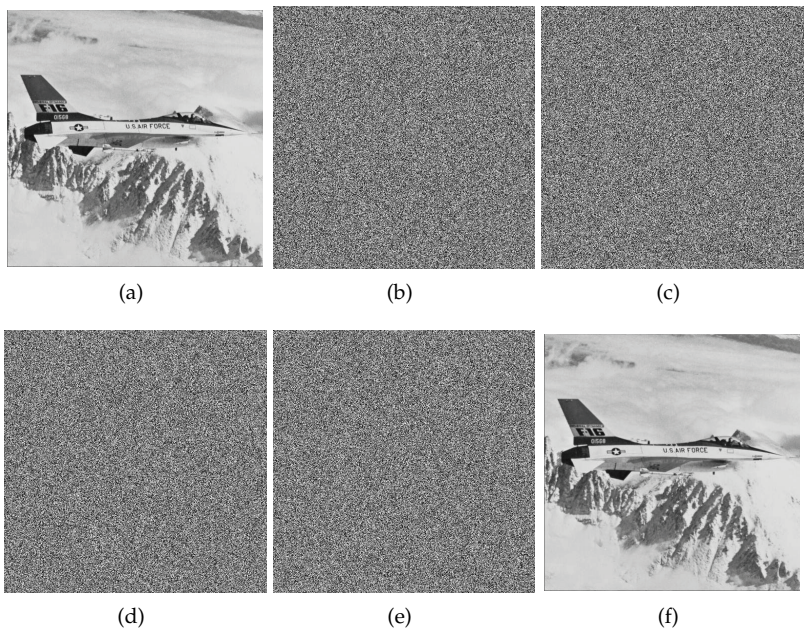


Fig. 7. Adopting 2-out-of-2 threshold secret sharing for the experiment. (a) Original image F-16, (b) 1st encrypted image, (c) 2nd encrypted image, (d) 1st marked encrypted image with $131,071 \times 7$ bits embedded (i.e., embedding rate $\approx$ 3.5 bpp), (e) 2nd marked encrypted image with $131,071 \times 7$ bits embedded, (f) Reconstructed image with PSNR = $+\infty$ dB.

TABLE 2
The expansion rates for the content-owner, the data-hiders, and the receiver under different secret sharing strategies.

	n=2	n=3	n=4	n=5
k=2	(2,1,2)	(3,1,2)	(4,1,2)	(5,1,2)
k=3	/	(3,1,3)	(4,1,3)	(5,1,3)
k=4	/	/	(4,1,4)	(5,1,4)
k=5	/	/	/	(5,1,5)

the associated marked encrypted images are obtained, as depicted in Fig. 6(f)–(i). When any three marked encrypted images are collected, the reconstructed image Boat with PSNR (peak signal to noise ratio) = $+\infty$ dB can be obtained, as shown in Fig. 6(j)–(m).

Example 2: In this example, 2-out-of-2 threshold secret sharing is applied for test image F-16, and thus two encrypted images are generated. The experiment result is shown in Fig. 7. By replacing a portion of the encrypted pixels, secret messages are embedded into the encrypted images shown in Fig. 7(b)–(c). The corresponding marked encrypted images generated are shown in Fig. 7(d)–(e). In each encrypted image beyond the first two pixels, two encrypted pixels are used to carry seven bits, and thus the corresponding embedding rate is appropriate 3.5 bpp (bit per pixel). When collecting the two encrypted images, a reconstructed image the same as the original image is obtained.

5.3 Data expansion analysis

Data expansion means that the encrypted image or the marked encrypted image is larger in size than the original image. As defined in [35], the data expansion is evaluated by the expansion rate, as denoted by

$$\text{Expansion rate} = \frac{\text{Total bits of the (marked) encrypted image}}{\text{Total bits of the original image}}.$$

In the proposed method, the content-owner converts an original image into n different encrypted images of the same size as the original image and distributes the encrypted images to n different data-hiders for data hiding. Each data-hider holds an encrypted image and embeds a secret message into it to obtain the marked encrypted image with the same size of the encrypted image. The receiver can restore the original image by collecting any k marked encrypted images. Table 2 shows the expansion rates for the content-owner, the data-hiders and the receiver under different secret sharing strategies. For the content-owner and the receiver, the expansion rates increase linearly with the values of n and k, respectively. For the data-hiders, no data expansion occurs.

In the methods using traditional encryption [16], [17], [18], data expansion does not occur. In the methods using homomorphic encryption [31], [32], [33], [34], data expansion is serious, that is, the encrypted image is hundreds times of the original image. Compared with the homomorphism based methods, the data expansion of the proposed method is acceptable. When the data-hider is damaged, all the existing methods cannot reconstruct the original image which can be addressed by the proposed method. It is guaranteed that even if $n-k$ data-hiders are damaged, the original image can still be reconstructed. In addition, each data-hider holds a different marked encrypted image, and the receiver needs k marked encrypted images to restore the original image, which implies that even if the marked encrypted images in $k-1$ data-hiders are leaked, the original image will not be leaked. Thus, the proposed method can effectively reduce the probability of leakage of the original image. In short, with proper data expansion, the proposed method can not only address the potential data-hider damage problem, but also reduce the probability of leakage of the original image.

5.4 Performance comparison

The embedding rate of the proposed method is compared with several state-of-the-art methods [16], [17], [18], [37]. As shown in Fig. 8, the embedding rate of the proposed method is significantly higher than these state-of-the-art methods. Meanwhile, the embedding rate of the proposed method is a constant value, which means that it is not affected by the image distribution. In these state-of-the-art methods, the embedding rates depend on image distribution. Images with smooth textures can achieve higher embedding rates, while images with complex textures have lower embedding rates, which can be avoided in the proposed method. This is because they vacate embedding room before encryption by the correlation of natural images, such as MSB prediction [16], [17], adaptive coding [18], and difference expansion [37]. Instead, the proposed method vacates embedding room after encryption and embeds data into encrypted images with bit-plane replacement.

In Table 3, we give the running time of the proposed method and the compared methods for the embedding rate of approximately 0.5 bpp. All the programs are developed by Java Eclipse SDK and run on 64-bit Windows 7 SP1 with Intel Core i3-2310M CPU @2.1GHz, 8GB RAM, and 120GB SSD. For the sake of fairness, $(2,2)$ secret sharing is considered in the proposed method since $(2,2,4,4)$-multi-secret sharing is adopted in [37]. Among them, [16] and [17] have a fast processing speed in encryption and decryption as they use a simple exclusive-or operation. Yi and Zhou's method [18] consumes more time than [16] and [17] due to the block permutation for scrambling pixels. However, the keys in these methods are the same size as plaintext because the keys cannot be reused for encryption. In other words, information leakage may occur in these methods if the key is reused. Therefore, the methods in [16], [17], [18] are difficult in terms of key management. Instead, the method of [37] and the proposed method can reuse the key since the semantic security of the methods is guaranteed through probability. In the proposed method, an original image is converted into two encrypted images by secret sharing, so the running time of the proposed method is twice that of the method of [37] in theory. However, it can be seen that the encryption time of the method of [37] is more than half of the proposed method, because there exists a pixel pair determination for secret sharing. In the proposed method, pixel shrink is introduced to ensure that all pixels can be used for secret sharing. For the running time of decryption, the proposed method needs to switch operations on different marked encrypted images, and hence it takes more time in decryption. The decryption operation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2020.3011923, IEEE Transactions on Dependable and Secure Computing

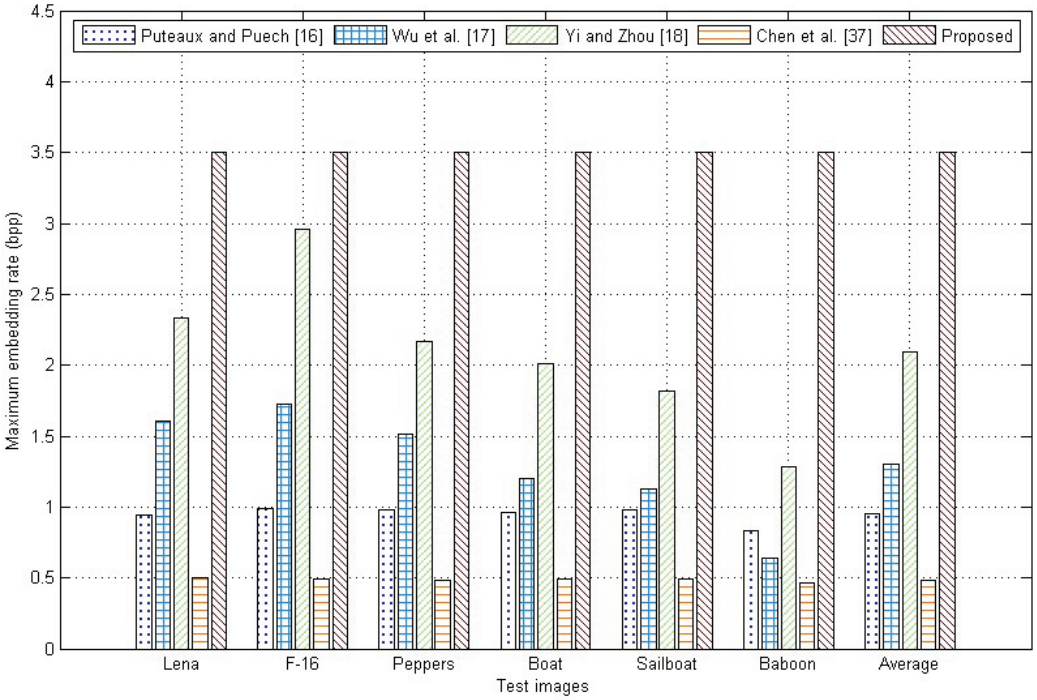


Fig. 8. Maximum embedding rate comparison among the proposed method and state-of-the-art methods.

in [37] is performed in a marked encrypted image, so it consumes less running time. However, the variable x in [37] is shared among the participants, which can be avoided in the proposed method. Sharing variable will increase extra computational consumption.

5.5 Feature comparison

The feature comparison among the proposed method and state-of-the-art methods is given in Table 4. It can be seen that the proposed method is separable, that is, the embedded secret message is extracted from the marked encrypted image. The method in [37] is joint, where data extraction requires decryption of the marked encrypted image. Different from these methods [16], [17], [18], [37], in which the embedded room is vacated to hide data before encryption, the proposed method vacates the embedded room after encryption. Both the method in [37] and the proposed method utilize secret sharing for image encryption. However, the method in [37] still works on the traditional model with single data-hider. The original image cannot be reconstructed when the data-hider is subjected to potential damage in the traditional model, which can be solved by the proposed model. Based on the proposed model, multiple data-hiders are active in the proposed method.

6 Conclusion and Future Work

This paper proposes a novel model with multiple data-hides for RDH-EI based on secret sharing, in which the original image is converted into multiple encrypted images with the same size of the original image, and the encrypted images are distributed to multiple data-hiders for data hiding. In such a way, even if a portion of the data-hiders are subject to potential damage, the original image can still be reconstructed by collecting sufficient marked encrypted images from undamaged data-hiders. We first review two previous models with single data-hider, and then, a secret sharing based model with multiple data-hiders and its associated four cases are provided. Based on the proposed model, a separable RDH-EI method with high-capacity is proposed in which data extraction is performed in the encrypted domain and the data hiding key is not required in image recovery; that is, the proposed method is content-owner-independent and data-hider-independent. Finally, the experimental results illustrate the effectiveness of the proposed method. Future work will consider how to implement other cases of the proposed model, such as a joint RDH-EI method. In addition, it is an interesting direction to extend the idea of secret sharing to other multimedia such as audio and video.

References

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

[2] 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, Aug. 2003.

[3] 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. 354–362, Mar. 2006.

[4] W.-L. Tai, C.-M. Yeh, and C.-C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906–910, Jun. 2009.

[5] X. Li, W. Zhang, X. Gui, and B. Yang, "A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1091–1100, Jul. 2013.

[6] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, Mar. 2010.

TABLE 3
Running time comparison (in milliseconds) of the proposed method and the state-of-the-art methods for various images.

Image		Lena	F-16	Peppers	Boat	Sailboat	Baboon	Average
Puteaux and Puech [16]	Encryption	0.15	0.13	0.17	0.15	0.16	0.17	0.16
	Decryption	0.14	0.14	0.17	0.14	0.17	0.17	0.16
Wu et al. [17]	Encryption	0.14	0.13	0.16	0.15	0.16	0.17	0.15
	Decryption	0.13	0.14	0.17	0.14	0.16	0.16	0.15
Yi and Zhou [18]	Encryption	2.51	2.50	2.52	2.52	2.62	2.54	2.54
	Decryption	3.86	3.85	3.87	3.85	3.84	4.01	3.88
Chen et al. [37]	Encryption	2.53	2.55	2.58	2.69	2.71	2.63	2.62
	Decryption	8.14	8.18	8.19	8.37	8.38	8.34	8.27
Proposed	Encryption	4.12	4.09	4.15	4.13	4.11	4.05	4.11
	Decryption	29.16	29.15	28.89	28.77	28.94	28.84	28.96

TABLE 4
Feature comparison among the proposed scheme and state-of-the-art schemes.

Scheme	Separable	Vacating room before encryption	Encryption strategy	Working model	Participant (Data-hider)
Puteaux and Puech [16]	Yes	Yes	Stream cipher	Traditional model	Single
Wu et al. [17]	Yes	Yes	Stream cipher	Traditional model	Single
Yi and Zhou [18]	Yes	Yes	Block permutation and Block modulation	Traditional model	Single
Chen et al. [37]	No	Yes	Secret sharing	Traditional model	Single
Proposed	Yes	No	Secret sharing	Secret sharing based model	Multiple

[7] H.-T. Wu and J. Huang, "Reversible image watermarking on prediction errors by efficient histogram modification," *Signal Processing*, vol. 92, no. 12, pp. 3000–3009, Dec. 2012.

[8] W. Puech, M. Chaumont, and O. Strauss, "A reversible data hiding method for encrypted images," in *Proc. SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, vol. 6819, Jan. 2008, pp. 68 191E–1–68 191E–9.

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

[10] 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, Apr. 2012.

[11] X. Liao and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," *Journal of Visual Communication and Image Representation*, vol. 28, pp. 21–27, Apr. 2015.

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

[13] S. Yi and Y. Zhou, "Parametric reversible data hiding in encrypted images using adaptive bit-level data embedding and checkerboard based prediction," *Signal Processing*, vol. 150, pp. 171–182, Sept. 2018.

[14] 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, Mar. 2013.

[15] 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, May 2016.

[16] 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*, vol. 13, no. 7, pp. 1670–1681, Jul. 2018.

[17] H.-T. Wu, Z. Yang, Y.-M. Cheung, L. Xu, and S. Tang, "High-capacity reversible data hiding in encrypted images by bit plane partition and MSB prediction," *IEEE Access*, vol. 7, pp. 62 361–62 371, May 2019.

[18] S. Yi and Y. Zhou, "Adaptive code embedding for reversible data hiding in encrypted images," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sept. 2017, pp. 4322–4326.

[19] Y. Wu, Y. Xiang, Y. Guo, J. Tang, and Z. Yin, "An improved reversible data hiding in encrypted images using parametric binary tree labeling," *IEEE Transactions on Multimedia*, 2019.

[20] C. Qin, X. Qian, W. Hong, and X. Zhang, "An efficient coding scheme for reversible data hiding in encrypted image with redundancy transfer," *Information Sciences*, vol. 487, pp. 176 – 192, Jun. 2019.

[21] Z. Yin, Y. Xiang, and X. Zhang, "Reversible data hiding in encrypted images based on multi-MSB prediction and huffman coding," *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 874–884, Apr. 2020.

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

[23] H.-Z. Wu, Y.-Q. Shi, H.-X. Wang, and L.-N. Zhou, "Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1620–1631, Aug. 2017.

[24] H. Ren, W. Lu, and B. Chen, "Reversible data hiding in encrypted binary images by pixel prediction," *Signal Processing*, vol. 165, pp. 268–277, Dec. 2019.

[25] Z. Qian, H. Xu, X. Luo, and X. Zhang, "New framework of reversible data hiding in encrypted jpeg bitstreams," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 351–362, Feb. 2019.

[26] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology – EUROCRYPT'99*, vol. 1592, 1999, pp. 223–238.

[27] 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, Jul. 2014.

[28] 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, Nov. 2015.

[29] S. Xiang and X. Luo, "Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 11, pp. 3099–3110, Nov. 2018.

[30] M. Li and Y. Li, "Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding," *Signal Processing*, vol. 130, pp. 190–196, Jan. 2017.

[31] 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, Sept. 2016.

[32] B. Chen, X. Wu, and Y.-S. Wei, "Reversible data hiding in encrypted images with private-key homomorphism and public-

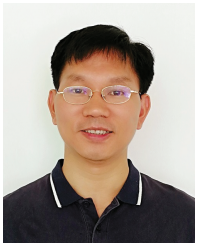
key homomorphism," *Journal of Visual Communication and Image Representation*, vol. 57, pp. 272–282, Nov. 2018.
[33] B. Chen, X. Wu, W. Lu, and H. Ren, "Reversible data hiding in encrypted images with additive and multiplicative public-key homomorphism," *Signal Processing*, vol. 164, pp. 48–57, Nov. 2019.
[34] S. Zheng, Y. Wang, and D. Hu, "Lossless data hiding based on homomorphic cryptosystem," *IEEE Transactions on Dependable and Secure Computing*, 2019.
[35] X. Wu, J. Weng, and W. Yan, "Adopting secret sharing for reversible data hiding in encrypted images," *Signal Processing*, vol. 143, pp. 269–281, Feb. 2018.
[36] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.
[37] Y.-C. Chen, T.-H. Hung, S.-H. Hsieh, and C.-W. Shiu, "A new reversible data hiding in encrypted image based on multi-secret sharing and lightweight cryptographic algorithms," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3332–3343, Dec. 2019.
[38] "The usc-sipi image database." [Online]. Available: http://sipi.usc.edu/database/.

Bing Chen received the Ph.D. degree in computer science and technology from the Sun Yat-sen University, Guangzhou, China, in 2020. His research interests include multimedia security, information hiding, and secret sharing.

Wei Lu (M'18) received the B.S. degree in automation from Northeast University, China in 2002, the M.S. degree and the Ph.D. degree in computer science from Shanghai Jiao Tong University, China in 2005 and 2007 respectively. He was a research assistant at Hong Kong Polytechnic University from 2006 to 2007. He is currently a Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His research interests include multimedia forensics and security, data hiding and watermarking, privacy protection. He has published over 100 papers in security conferences and journals, such as TIFS, TDSC, TCSVT, TPAMI, TNNLS, TCYB, etc. He is an Associate Editor for the Signal Processing and the Journal of Visual Communication and Image Representation. He is a member of the IEEE.

Jiwu Huang (M'98-SM'00-F'16) received the B.S. degree from Xidian University, Xian, China, in 1982, the M.S. degree from Tsinghua University, Beijing, China, in 1987, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1998. He is currently a Professor with the College of Information Engineering, Shenzhen University, Shenzhen, China. His current research interests include multimedia forensics and security.

He is a member of the IEEE Signal Processing Society Information Forensics and Security Technical Committee. He was a General Co-Chair of the IEEE Workshop on Information Forensics and Security in 2013 and a TPC Co-Chair of the IEEE Workshop on Information Forensics and Security in 2018. He is an Associate Editor for the IEEE Transactions on Information Forensics and Security. He is a fellow of the IEEE.



Jian Weng (M'17) received the B.S. and the M.S. degrees in computer science and engineering from the South China University of Technology in 2000 and 2004 respectively, and the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University, in 2008. From 2008 to 2010, he held a post-doctoral position with the School of Information Systems, Singapore Management University. He is currently a Professor and Vice President of Jinan University. His research interests include public key cryptography, cloud security, blockchain, etc. He has published over 100 papers in cryptography and security conferences and journals, such as CRYPTO, EUROCRYPT, ASIACRYPT, TCC, PKC, TPAMI, TIFS, and TDSC. He served as a PC co-chairs or PC member for more than 30 international conferences. He also serves as associate editor of IEEE Transactions on Vehicular Technology.

Yicong Zhou (M'07-SM'14) received the B.S. degree from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees from Tufts University, Massachusetts, USA, all in electrical engineering. He is currently an Associate Professor and Director of the Vision and Image Processing Laboratory in the Department of Computer and Information Science at University of Macau. His research interests include image processing, computer vision, machine learning, and multimedia security.

Dr. Zhou is a Senior Member of the International Society for Optical Engineering (SPIE). He was a recipient of the Third Price of Macao Natural Science Award in 2014 and 2020. He is a Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. He serves as an Associate Editor for IEEE Transactions on Neutral Networks and Learning Systems, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Geoscience and Remote Sensing, and four other journals.