Separable Reversible Data hiding in Encrypted Images by *n*-nary Histogram Modification

Zhenxing Qian[1] **Xiyu Han** **Xinpeng Zhang**

Abstract. This work proposes a novel scheme of separable reversible data hiding for encrypted image using histogram modification. After the original image is encrypted by the sender, the data-hider embeds the secret message into the encrypted image by a histogram modification and *n*-nary data hiding scheme. On the receiver side, the secret message can be extracted by the embedding key, and the original image can be totally recovered using both the embedding key and the encryption key. If the embedding key is unavailable for the receiver, the original image can still be approximately recovered using the encryption-key. This separable method does not require the sender to reserve embedding room by himself, and can perfectly recover the original image. Compared to the existing reversible data hiding methods for encrypted images, the proposed method largely improves the embedding capacity.

Keywords: Reversible data hiding. Information hiding. Image recovery

1.1 Introduction

Reversible data hiding (RDH) is a technique that embeds secret data into the cover, like military or medical images, by slightly modifying the data of the cover; and on the receiver side, the original image can be losslessly recovered [1,2]. Many RDH methods have been proposed for digital images, such as the general RDH framework by redundancy compression [3], the difference expansion (DE) method [4], the histogram shifting (HS) method [5], and so on [6-8]. Traditionally, these methods are useful for embedding data into the images that are open to the data-hider. However, in some scenarios, the owner of the cover is unwilling to show the information of the original image to the data-hider. For example, to protect the patient's privacy, content of the medical image might be unavailable for the technician who embeds the information into the medical image. These applica-

[1] Z. Qian (✉)
School of Communication and Information Engineering, Shanghai University, 200072, China
e-mail: zxqian@shu.edu.cn

tions require RDH embedding data into the encrypted version of the original image.

Some useful RDH in encrypted images have been proposed. In [9], Zhang divides the encrypted image into blocks, and embeds one bit into each block by flipping 3 LSBs of the half of the pixels in the block. Hong et al. [10] provided an improved version of Zhang's method, by exploiting the correlation of the border of neighboring blocks, and using the side-match scheme to achieve a lower error rate. These two methods can be realized because of the spatial correlation in the decrypted image. Thus, data extraction can only be done after image decryption, which means these methods are not separable. To resolve this problem, Zhang further proposed a separable RDH scheme for encrypted image by compressing the encrypted data using source coding with side information [11], which guarantees the data extraction independent from image encryption. In [12], Ma et al. provided a RDH idea in encrypted images by reserving room before encryption. This method empties out room by embedding LSBs of some pixels into other pixels with the traditional RDH method and then encrypts the image on the sender side, and as a result, positions of these LSBs in the encrypted image can be used for data hiding by the data-hider. Although this method greatly improved the embedding capacity, an additional RDH has to be implemented by the sender, which might be impossible for the users, because RDH in encrypted image always requires the sender to do nothing except encryption and the embedding tasks are always supposed to be accomplished by the data-hider.

This work proposes a new separable RDH scheme for encrypted images. We make full use of the disordered histogram and the encrypted image, and define an n-nary data hiding scheme for data hiding. The proposed is separable, in which the secret bits can be extracted independently from the stego. The original image can be perfectly recovered using the embedding key and the encryption key, and can be approximately recovered if only the encryption key is available. Compared to the existing RDH methods for encrypted images, the proposed method largely improves the embedding capacity than the methods in [9]~[11], and requires no room-reserving operation like [12] for the senders.

1.2 Proposed Scheme

Sketch of the proposed method is shown in Fig. 1. The original image is encrypted into the encrypted data using an embedding key, and the encrypted data are sent to the data hider who embeds the secret bits into the encrypted data using an embedding key to generate a stego. There are three cases for the receiver to extract secret bits or recover the image. If the receiver has only the embedding key, he can extract the secret bits independently. If he has only the encryption key, he can approximately recover the original image with some "pepper & salt" noises. If both the embedding key and the encryption keys are available for the receiver, the se-

cret bits can be extracted and the original image can be perfectly recovered. Details of the procedures are as follows.

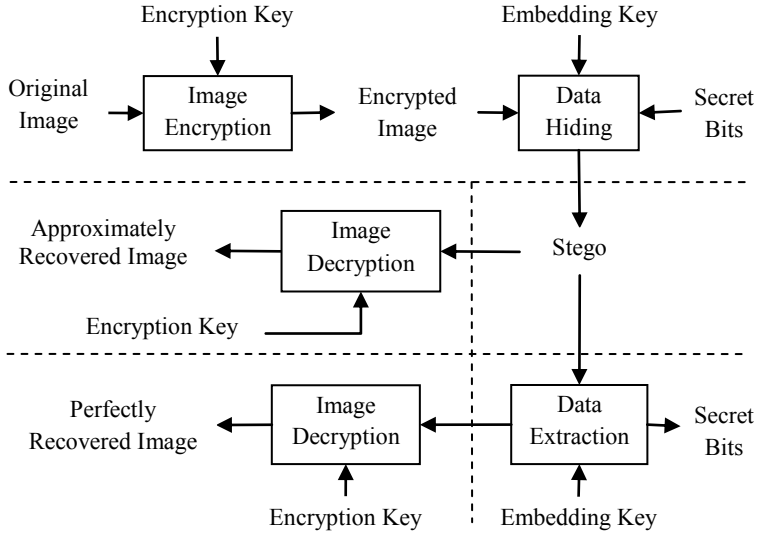

Fig. 1.1 Sketch of the propose method

1.2.1 Image Encryption

Assume the gray images with values belonging to [0, 255] for each pixel are used in the scheme. Denote the original image as **I** that sized $M \times N$. With a pre-defined key K_p, we first pseudo-randomly realign all the pixels into a disordered image **I'** which is semantically meaningless.

However, histogram of the realigned image retains the same as the original. To avoid the information leakage of the histogram, we further define a mapping function associate with a new key K_h for pseudo-randomly reordering the indices of the histogram,

$$y=ENC(x, K_h),\ \ x,y \in [0, 255] \tag{1}$$

where the $ENC()$ is the mapping function to turn the intensity x into another intensity y by the key K_h. Thus, an encrypted image **E** is finally generated by the mapping, $\mathbf{E}(a, b)=ENC(\mathbf{I}'(a, b)\ , K_h)$, where $a=1,\ldots,M$ and $b=1,\ldots,N$.

After the encryption, the information of the contents and the histogram of the image are concealed. There are totally $(MN)!\cdot 256!$ possible permutations for the original image, it would be difficult for the data-hider or the adversaries to break the encryption. Record the keys K_p and K_h as the encryption key $\mathbf{K}_{enc}=\{K_p, K_h\}$.

1.2.2 Data Hiding

After receiving the encrypted data **E**, the data-hider embeds the secret bits by modifying a small portion of the encrypted data. Firstly, the data-hider constructs the histogram of the encrypted image. Denote the histogram values for the encrypted image as h_i, i=0,1,…255. Find the intensity set that has maximum quantity in the histogram,

$$\mathbf{P} = \{\arg\max_i(h_i),\ i \in [0,255]\} \tag{2}$$

If **P** contains more than one member, discard the intensities except the first one, and generate a new set **P**=$\{p_0\}$ with only one element. Then the data-hider calculates

$$\mathbf{Q} = \{\arg\min_i(h_i),\ i \in [0,255]\} \tag{3}$$

where **Q**=$\{q_0, q_1, \ldots, q_{r-1}\}$ is the set of intensities with minimum quantities. By merging the two sets, the data-hider generates a new set $\mathbf{T}=\mathbf{P}\cup\mathbf{Q}$. Intensities in the new set **T** are available for data hiding.

Accordingly, the data-hider uses an n-nary data hiding algorithm for information embedding. The value n is determined by the cardinality of the set **T**, that is n=| **T** |. Define the intensities in **T** as $\{t_0, t_1, \ldots, t_{n-1}\}$, where $t_0=p_0$, $t_i=q_{i-1}$, i=1,2,…,n–1.

Generally, $h(q_i)$, the number of pixels that equals q_i, is equal zero or a small positive integer. If $h(q_i)$ is not equal to zero, the data-hider records the positions of the intensities that are equal to $h(q_i)$, and changes the corresponding values to p_0 in the encrypted data. Denote these positions and the corresponding intensities as a sequence **W**.

After pseudo-randomly realign the secret bits by a new key K_s, the hider obtains the encrypted secret data that contains U bits $\mathbf{S}=b_0\ b_1\ \ldots\ b_{U-1}$ where $U = \lfloor h(p_0)\cdot\log_2(n)\rfloor$. Turn these bits into an n-nary array $\mathbf{S}^{(n)}=s_0s_1\ldots s_{V-1}$ by

$$s_k = \left\lfloor \sum_{i=0}^{U-1} b_i \cdot 2^i \Big/ n^k \right\rfloor \bmod n, \quad k = 0,1,...,V \tag{4}$$

where $V=h(p_0)$ and $s_k \in [0, n-1]$.

Subsequently, modify all the intensities that are equal to p_0 in the encrypted image **E** to generate the stego **F** using equation (5). For example, to embed the n-nary value that equals k into the encrypted data, we find the intensity that equals p_0 and modify it to t_k.

$$\mathbf{F}(i,j) = \begin{cases} t_k & \text{if } \mathbf{E}(i,j) = p_0 \ \ \&\ \ s_l = k \\ \mathbf{E}(i,j) & \text{if } \mathbf{E}(i,j) \neq p_0 \end{cases} \tag{5}$$

Concatenating K_s, **T** and **W**, and compressing these data by a lossless compression algorithm, the embedding key $K_{emb}=\{K_s, \mathbf{T}, \mathbf{W}\}$ is generated.

1.2.3 Data Extraction and Image Recovery

On the receiver side, the secret data can be extracted from the stego **F** with the embedding key K_{emb}. After decompressing the embedding key, data of K_s, **T** and **W** are recovered. According to **T**, the n-nary data can be extracted by

$$s_i' = k, \quad \text{if } \mathbf{F}(i, j) = t_k \tag{6}$$

where $i=0,1,\ldots V$ and $k \in [0, n\text{-}1]$. Then calculate the secret bits,

$$b_j' = \left\lfloor \sum_{i=0}^{V-1} s_i' \cdot n^i \Big/ 2^j \right\rfloor \bmod 2, \quad j = 0,1,...,U \tag{7}$$

By the key K_s, the receiver reorders these bits of b_k' to recover the original message.

Considering the receiver has both of the embedding key and the encryption key, he can further decrypt the received data and perfectly recovers the original image. The modified data of the encrypted image **E'** can be first recovered by

$$\mathbf{E}'(i, j) = p_0, \quad \text{if } F(i, j) \in \mathbf{T} \tag{8}$$

Then, replace all the intensities on the positions recorded by **W** with the original values. This way, the encrypted data is totally restored. Next, recover the original image according to the encryption key $\mathbf{K}_{enc}=\{K_p, K_h\}$. With the key K_h, the mapping relationships are constructed again to change each data into the original intensity,

$$\mathbf{R}'(a, b)=ENC^{-1}(\mathbf{E}'(a, b), K_h), \; a=1,\ldots,M; \; b=1,\ldots,N \tag{9}$$

where $ENC^{-1}()$ is the reverse of the mapping function $ENC()$. Pseudo-randomly reorder **R'** using the key K_p to finally recover the original image **R**.

Additionally, if the receiver has only the encryption-key, content of the original image can also be approximately recovered. The receiver can use the key $\mathbf{K}_{enc}=\{K_p, K_h\}$ to turn the stego **F** directly into an spatial image which is close to the original image. Although some "pepper & salt" noises will appear in the approximately decrypted image, main contents of the original image are well preserved.

1.3 Proposed Scheme

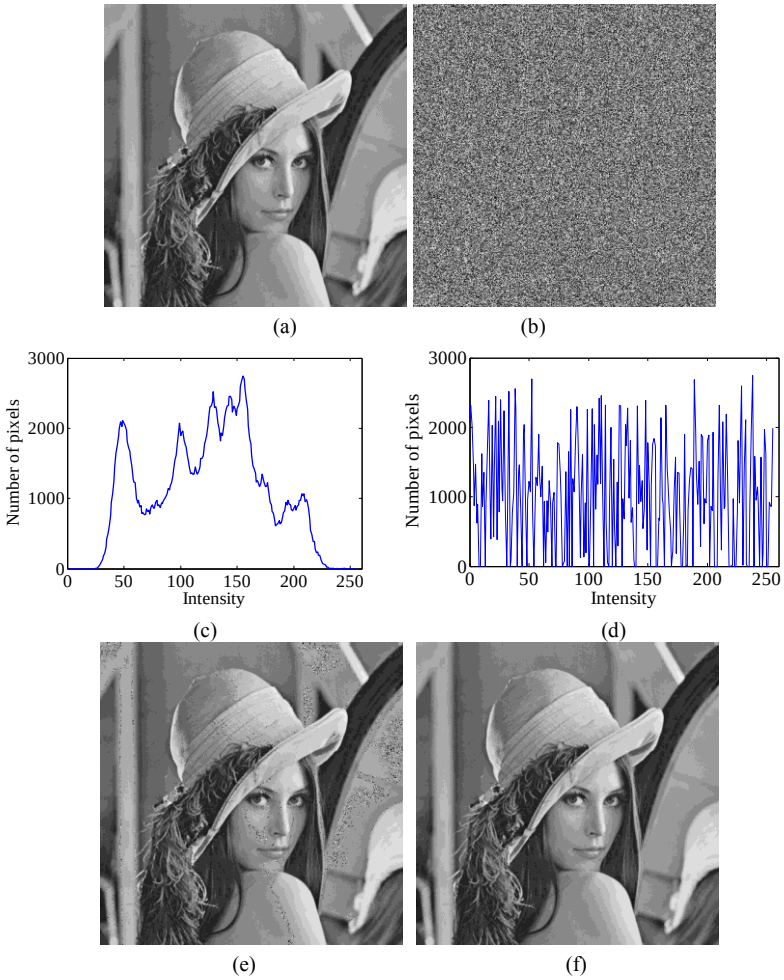

Fig. 1.2 RDH in encrypted image "Lenna", (a) the original image, (b) the histogram of (a), (c) the encrypted image, (d) the histogram of (c), (e) the approximately recovered image, (f) the perfectly recovered image.

Experiments are conducted to verify the proposed method. The test images are standard gray images size 512×512. Fig. 2(a) is the original image "Lena", and Fig. 2(c) the corresponding histogram. After encrypting the original image with the encryption key, both the image and histogram are randomly distributed, which are shown in Fig. 2(b) and Fig. 2(d). Then, 14284 secret bits are embedded into the encrypted image by 42-nary data hiding using an embedding key. On the receiver side, the secret bits can be extracted without any errors if the embedding

key is available. When the receiver has only the encryption-key, the original image can be approximately recovered, as shown in Fig. 2(e), with some salt & paper noises. If the receiver has both the embedding key and the encryption key, the original image can be perfectly recovered without any errors, which is shown in Fig. 2(f).

Embedding capacities of some images are listed in Table 1. We compare the proposed method to the methods in [9]~[11]. Results show that the proposed method largely improves the embedding capacity. We did not compare the embedding capacity of the proposed method to the method in [12], because the method in [12] is in fact a pre-processing RDH embedding which reserves enough capacity on the sender side. Actually, the sender should not do any other operations except image encryption.

Table 1.1 Embedding Capacity Comparison

Images	Method [9] (bits)	Method [10] (bits)	Method [11] (bits)	Proposed	
				(bits)	*n*-nary
Lena	1024	1024	8650	14284	42
Man	655	1024	6554	46557	57
Lake	655	1024	3408	14828	19
Baboon	256	334	1966	13056	31

Some features of the proposed method are also compared to the methods [9]~[12], and the results are shown in Table 2. The proposed method is separable for data extraction and image recovery, the same as the methods [11] and [12]. Errors may happen in the recovered image depending on the block size when using the methods [9] and [10], while the proposed method can perfectly recover the original image on the receiver side. Another important feature in Table 2 is whether the RDH method needs pre-processing for reserving embedding room for data hiding in encrypted image. The proposed method requires no other operations except encryption, the same as the methods in [9]~[11].

Table 1.2 Comparison of Features for Different Methods

Features	Method [9]	Method [10]	Method [11]	Method [12]	Proposed
Separable	No	No	Yes	Yes	Yes
Errors in recovery	Yes	Yes	No	No	No
Pre-reserving room for data-hiding	No	No	No	Yes	No

1.4 Conclusions

This work proposed a new separable RDH method for data hiding in encrypted images. The original image is first encrypted to disorder the principle content and the histogram information. The data hider makes full use of the histogram of the encrypted image, and embeds secret message into these data by *n*-nary data hiding. On the receiver side, the secret information can be extracted without any error, and the original image can be perfectly recovered with both the embedding-key and the encryption-key. If the receiver has only the encryption-key, the original image can also be approximately recovered. Compared to the existing RDH methods for encrypted images, the embedding capacity is largely improved.

1.5 References

1. Kalker T, Willems F M (2002) Capacity bounds and code constructions for reversible data-hiding, Proc. 14th Int. Conf. Digital Signal Processing (DSP2002), 71–76
2. Sachnev V, Kim H J, Nam J, et al. (2009) Reversible watermarking algorithm using sorting and prediction, IEEE Trans. Circuits Syst. Video Technol. 19(7): 989–999
3. Fridrich J, Goljan M (2002) Lossless data embedding for all image formats, Proc. SPIE Proc. Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents, 4675: 572–583, San Jose, CA, USA
4. Tian J (2003) Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Tech. 13(8): 890–896
5. Ni Z, Shi Y, Ansari N, Wei S (2006) Reversible data hiding, IEEE Trans. Circuits Syst. Video Tech. 16(3):354–362
6. Thodi D M, Rodríguez J J (2007). Expansion embedding techniques for reversible watermarking, IEEE Transactions on Image Processing, 16(3):721-730
7. Luo L et al. (2010) Reversible image watermarking using interpolation technique, IEEE Trans. Inf. Forensics Security, 5(1):187–193
8. Li X L, Yang B, Zeng T Y (2011) Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection, IEEE Trans. Image Processing, 20(12):3524–3533
9. Zhang X (2011) Reversible data hiding in encrypted images, IEEE Signal Process. Lett., 18(4):255–258
10. Hong W, Chen T, Wu H (2012) An improved reversible data hiding in encrypted images using side match, IEEE Signal Process. Lett., 19(4):199–202
11. Zhang X (2012) Separable reversible data hiding in encrypted image, IEEE Trans. Inf. Forensics Security, 7(2):826–832
12. Ma K, Zhang W, et al. (2013) Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption, IEEE Trans. Inf. Forensics Security, 8(3):553-562