Reversible Data Hiding For Embedding Data Securely in Encrypted Image by Reserving Room Before Encryption

[1] Sunita.G.J, [2] Syeda Asra
[1]M.Tech. Scholar, [2]Assistnat Professor
Department of Computer Science and Engineering
Appa Institute of Engineering and Technology, Gulbarga, India

***Abstract*—Reversible data hiding with image encryption (RDH), since it maintains the excellent property that the original cover can be losslessly recovered after embedded data is extracted while protecting the image content's confidentiality. In previous work vacating room after encrypted image, which subject to some errors on data extraction and/or image restoration. In this paper, a novel method by reserving room before encryption with a traditional RDH algorithm, and thus it is easy for the data hider to reversibly embed data in the encrypted image. The proposed method can achieve real reversibility, that is, data extraction and image recovery are free of any error.**

***Index Terms* - Reversible data hiding, Image encryption, Privacy protection, Histogram shifting, LSB technique**

I. INTRODUCTION

Reversible data hiding (RDH) in images is a technique, by which the original cover can be losslessly recovered after the embedded message is extracted. This important technique is widely used in medical imagery, military imagery and law forensics, where no distortion of the original cover is allowed. Since first introduced, RDH has attracted considerable research interest.

In theoretical aspect, Kalker and Willems [1] established a rate-distortion model for RDH, through which they proved the rate-distortion bounds of RDH for memory less covers and proposed a recursive code construction which, however, does not approach the bound. Zhang *et al.* [2], [3] improved the recursive code construction for binary covers and proved that this construction can achieve the rate-distortion bound as long as the compression algorithm reaches entropy, which establishes the equivalence between data compression and RDH for binary covers.

In practical aspect, many RDH techniques have emerged in recent years. Fridrich *et al.* [4] constructed a general framework for RDH. By first extracting compressible features of original cover and then compressing them losslessly, spare space can be saved for embedding auxiliary data. A more popular method is based on difference expansion (DE) [5], in which the difference of each pixel group is expanded, e.g., multiplied by 2, and thus the least significant bits (LSBs) of the difference are all-zero and can be used for embedding messages. Another promising strategy for RDH is histogram shift (HS) [6], in which space is saved for data embedding by shifting the bins of histogram of gray values. The state-of-art methods [7]–[11] usually combined DE or HS to residuals of the image, e.g., the predicted errors, to achieve better performance.

With regard to providing confidentiality for images, encryption [12] is an effective and popular means as it converts the original and meaningful content to incomprehensible one. In [13], Hwang *et al.* advocated a reputation-based trust-management scheme enhanced with data coloring (a way of embedding data into covers) and software watermarking, in which data encryption and coloring offer possibilities for upholding the content owner's privacy and data integrity. Obviously, the cloud service provider has no right to introduce permanent distortion during data coloring into encrypted data. Thus, a reversible data coloring technique based on encrypted data is preferred. In [16], Zhang divided the encrypted image into several blocks. By flipping 3 LSBs of the half of pixels in each block, room can be vacated for the embedded bit. The data extraction and image recovery proceed by finding which part has been flipped in one block. This process can be realized with the help of spatial correlation in decrypted image. Hong *et al.* [17] ameliorated Zhang's method at the decoder side by further exploiting the spatial correlation using a different estimation equation and side match technique to achieve much lower error rate.

In the present paper, we propose a novel method for RDH in encrypted images, for which we do not "vacate room after encryption" as done in [16]–[18], but "reserve room before encryption". In the proposed method, we first empty out room by embedding LSBs of some pixels into other pixels with a traditional RDH method and then encrypt the image, so the positions of these LSBs in the encrypted image can be used to embed data. Not only does the proposed method separate data extraction from image decryption but also achieves excellent performance in two different prospects:

- Real reversibility is realized, that is, data extraction and image recovery are free of any error.
- For given embedding rates, the PSNRs of decrypted image containing the embedded data are significantly improved; and for the acceptable PSNR, the range of embedding rates is greatly enlarged.

II. RELATED WORK

Lots of research has been done in the area of reversible data hiding. In last few years various efficient methods have been proposed for reversible data hiding. Some noticeable work in area of reversible data hiding is as follows:

In [1] T. Kalker and F.M.Willems has proposed a framework of protection system for secret data communication through encrypted data concealment in encrypted images. The image is then separated into number of blocks locally and lifting wavelet will be used to detect approximation and detailed coefficients.

In [2] W. Zhang, B. Chen, and N. Yu has proposed a reversible data hiding (RDH), the original cover can be losslessly restored after the embedded information is extracted. Kalker and Willems established a rate–distortion model for RDH, in which they proved out the rate–distortion bound and proposed a recursive code construction.

In [3] J. Fridrich and M. Goljan has proposed lossless data embedding has the property that the distortion due to embedding can be completely removed from the watermarked image without accessing any side channel. This can be a very important property whenever serious concerns over the image quality and artifacts visibility arise, such as for medical images, due to legal reasons, for military images or images used as evidence in court that may be viewed after enhancement and zooming.

In [4] J. Tian, has introduced a difference expansion technique which discovers extra storage space by exploring the redundancy in the image content. Both the secret data holding capacity limit and the visual quality of embedded images of the DE method are among the best in the literature, along with a low computational complexity.

In [5] Z. Ni, Y. Shi, N. Ansari, and S. Wei they proposed the reversible data hiding algorithm, which can recover the original image without any distortion from the marked image after the hidden data have been extracted, is presented in this paper.

In [6] D.M. Thodi and J. J. Rodriguez has proposed the framework reversible watermarking enables the embedding of useful information in a host signal without any loss of host information. Tian's difference-expansion technique is a high-capacity, reversible method for data embedding. However, the method suffers from undesirable distortion at low embedding capacities and lack of capacity control due to the need for embedding a location map. They propose a histogram shifting technique as an alternative to embedding the location map. The proposed technique improves the distortion performance at low embedding capacities and mitigates the capacity control problem.

In [7] X. L. Li, B. Yang, and T. Y. Zeng has proposed the prediction-error expansion (PEE) is an important technique of reversible watermarking which can embed large payloads into digital images with low distortion. In this paper, the PEE technique is further investigated and an efficient reversible watermarking scheme is proposed, by incorporating in PEE two new strategies, namely, adaptive embedding and pixel selection.

III. PROBLEM STATEMENT

The reversible data hiding (RDH) in encrypted images, all previous methods embed data by reversibly vacating room from the encrypted images, which may be subject to some errors on data extraction and/or image restoration. So the proposed method by reserving room before encryption with a traditional RDH algorithm, and thus it is easy for the data hider to reversibly embed data in the encrypted image.

IV. OBJECTIVE OF THE PROJECT

The aim or objective of the project is to implement a reversible data hiding (RDH) technique in encrypted images. The proposed technique or method can achieve real reversibility, that is, data extraction and image recovery are free of any errors.

V. SCOPE OF THE PROJECT

The work proposes a reversible data hiding scheme for encrypted image, which is made up of image encryption, data embedding and data-extraction/image-recovery phases. The data of original cover are entirely encrypted, and the additional message is embedded by modifying a part of encrypted data. At receiver side, with the aid of spatial correlation in natural image, the embedded data are successfully extracted while the original image is perfectly recovered.

VI. PROPOSED SYSTEM

Since losslessly vacating room from the encrypted images is relatively difficult and sometimes inefficient, why are we still so obsessed to find novel RDH techniques working directly for encrypted images? If we reverse the order of encryption and vacating room, i.e., reserving room prior to image encryption at content owner side, the RDH tasks in encrypted images would be more natural and much easier which leads us to the novel framework, “reserving room before encryption (RRBE)”. As shown in Fig. 1, the content owner first reserves enough space on original image and then convert the image into its encrypted version with the encryption key. Now, the data embedding ding process in encrypted images is inherently reversible for the data hider only needs to accommodate data into the spare space previous emptied out. The data extraction and image recovery are identical to that of Framework VRAE.

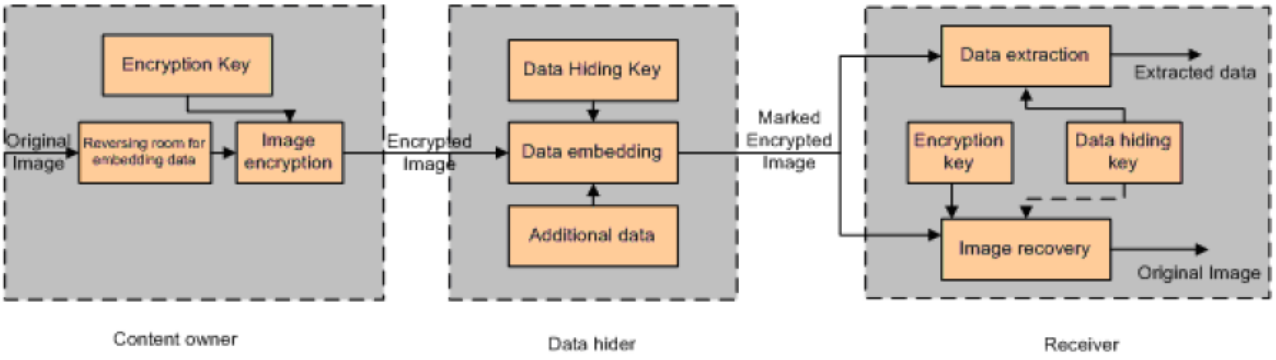


Fig 1 Framework of Reserving Room Before Encryption (RRBE)

Next, we elaborate a practical method based on the Framework "RRBE", which primarily consists of three stages:

1. Generation of encrypted image.
2. Data hiding in encrypted image.
3. Data extraction and image recovery.

1. Generation of encrypted image

To construct the encrypted image, the very first stage is being divided into three steps:

a. Image partition,.
b. Self- Reversible embedding.
c. Image encryption.

Initially, image partition step divides original image into two parts A and B then, the LSBs of A are reversibly embedded into B with a standard RDH algorithm so that LSBs of A can be used for accommodating messages; at last, encrypt the rearranged image to generate its final version.

a. **Image Partition**: The operator here for reserving room before encryption is a standard RDH technique, so the goal of image partition is to construct a smoother area B , on which standard RDH algorithms can achieve better performance. To do that, without loss of generality, assume the original image C is an 8 bits gray-scale image with its size M x N and pixels $C_{i,j} \in [0,255]$, $1 \le i \le M$, $1 \le j \le N$. First, the content owner extracts from the original image, along the rows, several overlapping blocks whose number is determined by the size of to-be-embedded messages, denoted by *l*. In detail, every block consists of rows, where $m = [l/N]$ and the number of blocks can be computed through $n = M - m + 1$. An important point here is that each block is overlapped by pervious and/or sub-sequential blocks along the rows. For each block, define a function to measure its first-order smoothness.

$$f = \sum_{u=2}^{m} \sum_{v=2}^{N-1} \left| C_{u,v} - \frac{C_{u-1,v} + C_{u+1,v} + C_{u,v-1} + C_{u,v+1}}{4} \right| \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Higher *f* relates to blocks which contain relatively more complex textures. The content owner, therefore, selects the particular block with the highest to *f* be A, and puts it to the front of the image concatenated by the rest part B with fewer textured areas, as shown in Fig1.

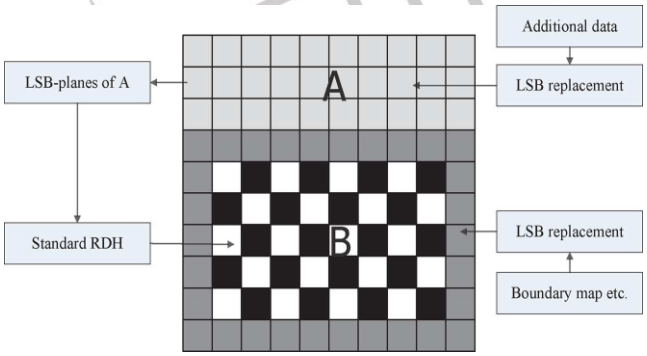


Fig 2 Illustration of image partition and embedding process.

b) Self-Reversible Embedding: The goal of self-reversible embedding is to embed the LSB-planes of A into B by employing traditional RDH algorithms. The method in [10] to demonstrate the process of self-embedding. Pixels in the rest of image B are first categorized into two sets: white pixels with its indices i and j satisfying (i+j) mod 2 = 0 and black pixels whose indices meet (i+j) mod 2 = 1, as shown in Fig.1.2. Then, each white pixel, $B_{i,j}$ is estimated by the interpolation value obtained with the four black pixels surrounding it as follows:

$$B'_{i,j} = w_1 B_{i-1,j} + w_2 B_{i+1,j} + w_3 B_{i,j-1} + w_4 B_{i,j+1} \quad \ldots\ldots\ldots\ldots\ldots (2)$$

where the weight w_i $1 \leq i \leq 4$, is determined by the same method as proposed in[10]. The estimating error is calculated via $e_{i,j} = B_{i,j} - B'_{i,j}$ and then some data can be embedded into the estimating error sequence with histogram shift, After that, then calculate the estimating errors of black pixels with the help of surrounding white pixels that may have been modified.

c) Image Encryption: After rearranged self-embedded image, denoted by X , is generated, we can encrypts X to construct the encrypted image, denoted by E .With a stream cipher, the encryption version of X is easily obtained. For example, a gray value $X_{i,j}$ ranging from 0 to 255 can be represented by 8 bits, $X_{i,i}(0), X_{i,i}(1)\_ \_ \_ \_ X_{i,j}(7)$ such that

$$\mathbf{X}_{i,j}(k) = \left\lfloor \frac{\mathbf{X}_{i,j}}{2^k} \right\rfloor \bmod 2, \quad k = 0, 1, \ldots, 7. \qquad \text{..... (3)}$$

The encrypted bits $E_{i,j}(k)$ can be calculated through exclusive- or operation

$$\mathbf{E}_{i,j}(k) = \mathbf{X}_{i,j}(k) \oplus r_{i,j}(k), \qquad \text{..... (4)}$$

Where $r_{i,j}$ (k) is generated via a standard stream cipher determined by the encryption key. Finally, we embed 10 bits information into LSBs of first 10 pixels in encrypted version A of to tell data hider the number of rows and the number of bit-planes he can embed information into. After the image encryption, the data hider or a third party cannot access the content of original image without the encryption key, thus privacy of the content owner being protected.

2. Data Hiding in Encrypted Image

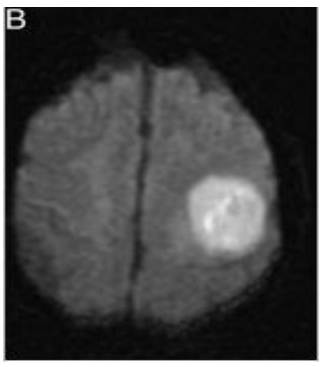
Once the data hider acquires the encrypted image E, he can embed some data into it, although he does not get access to the original image. The embedding process starts with locating the encrypted version of A, denoted by A_E . Since A_E been rearranged to the top of E, it is effortless for the data hider to read 10 bits information in LSBs of first 10 encrypted pixels. After knowing how many bit-planes and rows of pixels he can modify, the data hider simply adopts LSB replacement to substitute the available bit-planes with additional data m . Finally, the data hider sets a label following m to point out the end position of embedding process and further encrypts according to the data hiding key to formulate marked encrypted image denoted by E' . Anyone who does not possess the data hiding key could not extract the additional data.

3. Data Extraction and Image Recovery

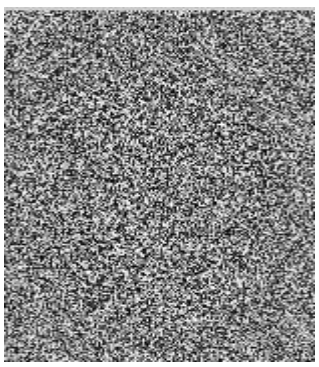
Since data extraction is completely independent from image decryption, the order of them implies two different practical applications.

i. **Case 1: Extracting Data From Encrypted Images**: When the database manager gets the data hiding key, he can decrypt the LSB-planes of A_E and extract the additional data m by directly reading the decrypted version. When requesting for updating information of encrypted images, the database manager, then, updates information through LSB replacement and encrypts updated information according to the data hiding key all over again. As the whole process is entirely operated on encrypted domain, it avoids the leakage of original content.

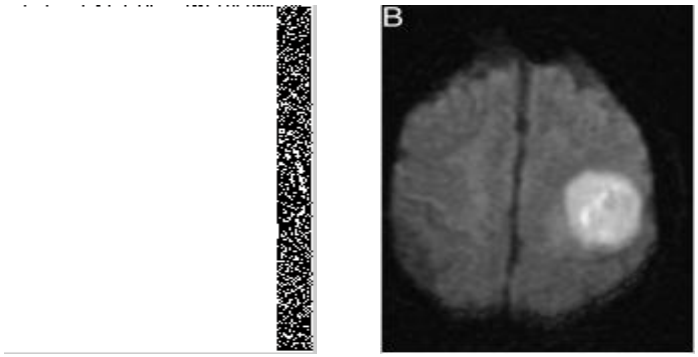
ii. **Case 2: Extracting Data From Decrypted Images***:* In Case 1, both embedding and extraction of the data are manipulated in encrypted domain. On the other hand, there is a different situation that the user wants to decrypt the image first and extracts the data from the decrypted image when it is needed. The following example is an application for such scenario. Assume Alice outsourced her images to a cloud server and the images are encrypted to protect their contents. Into the encrypted images, the cloud server marks the images by embedding some notation, including the identity of the images' owner, the identity of the cloud server and time stamps, to manage the encrypted images. The cloud server has no right to do any permanent damage to the images. Now an authorized user, Bob who has been shared the encryption key and the data hiding key, downloaded and decrypted the images. Bob hoped to get marked decrypt images, i.e., decrypted images still including the notation, which can be used to trace the source and history of the data. The order of image decryption before/without data extraction is perfectly recovered.



(a)



(b)



(c) (d)

Fig 3(a) Original Image (b) Encrypted Image (c) Encrypted image with data hiding (d) Recovered original image as well as data.

VII. CONCLUSION

Reversible data hiding in encrypted images is a new topic drawing attention because of the privacy-preserving requirements from cloud data management. Previous methods implement RDH in encrypted images by vacating room after encryption, as opposed to which we proposed by reserving room before encryption. Thus the data hider can benefit from the extra space emptied out in previous stage to make data hiding process effortless. The proposed method can take advantage of all traditional RDH techniques for plain images and achieve excellent performance without loss of perfect secrecy. Furthermore, this novel method can achieve real reversibility, separate data extraction and greatly improvement on the quality of marked decrypted images.

REFERENCES

[1] T. Kalker and F.M.Willems, "Capacity bounds and code constructions for reversible data-hiding," in Proc. 14th Int. Conf. Digital Signal Processing (DSP2002), 2002, pp. 71–76.

[2] W. Zhang, B. Chen, and N. Yu, "Capacity-approaching codes for Reversible data hiding," in Proc 13th Information Hiding (IH'2011), LNCS 6958, 2011, pp. 255–269, Springer-Verlag.

[3] J. Fridrich and M. Goljan, "Lossless data embedding for all image formats," In Proc. SPIE Proc. Photonics West, Electronic Imaging, Security And Watermarking of Multimedia Contents, San Jose, CA, USA, Jan. 2002, vol. 4675, pp. 572–583.

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

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

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

[7] X. L. Li, B. Yang, and T. Y. Zeng, "Efficient reversible watermarking Based on adaptive prediction-error expansion and pixel selection," IEEE Trans. Image Process. vol. 20, no. 12, pp. 3524–3533, Dec. 2011.