ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Reversible Data Hiding In Encrypted Images By Reserving Room before Encryption

G.Raj Kumar
M.Tech student
Krishna Murthy Institute of Technology & Engineering

Mr. Srikanth Ch, M.Tech
Guide,
Krishna Murthy Institute of Technology & Engineering

ABSTRACT

Recently, a high attention is paid to reversible data hiding (RDH) in encrypted images, since it does maintain the excellent property that the original cover can be lossless recovered after embedded data is extracted while protecting the image content's confidentiality. All legacy 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. Here, we propose 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. Experiments show that this novel method can embed more than 10 times as large payloads for the same image quality as the previous methods, such as for PSNR dB.

Index Terms ***Reversible data hiding, image encryption, privacy protection, histogram shift.***

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 technique is widely used in medical imagery, military imagery and law forensics, where no distortion of the original cover is allowed.

In practical aspect, many RDH techniques have emerged in recent years. Fridrich *et al.* 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), 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) , in which space is saved for data embedding by shifting the bins of histogram of gray values. The state-of-art methods usually combined DE or HS to residuals of the image, e.g., the predicted errors, to achieve good performance.

Regarding to provide the confidentiality for images, encryption is an effective and popular means as it converts the original and meaningful content to incomprehensible one. Few RDH techniques in encrypted images have been published yet, there are some promising applications if RDH can be applied to encrypted images. In , 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. Suppose a medical image database is stored in a data center, and a server in the data center can embed notations into an encrypted version of a medical image through a RDH technique. With the notations, the server can manage the image or verify

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

its integrity without having the knowledge of the original content, and thus the patient's privacy is protected. On the other hand, a doctor, having the cryptographic key, can decrypt and restore the image in a reversible manner for the purpose of further diagnosing.

Here, in this paper, we introduce a novel method for RDH in encrypted images, for which we do not "vacate room after encryption", 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

PREVIOUS ARTS

The methods proposed in [1]–[3] can be summarized as the framework, "vacating room after encryption (VRAE)", as illustrated in Fig. 1(a).

In this, a content owner encrypts the original image using a standard cipher with an encryption key. After getting the encrypted image, the content gives it to a data hider (e.g., a database manager) and the data hider can embed some auxiliary data into the encrypted image by losslessly vacating some room according to a data hiding key. Finally, a receiver, maybe the content owner himself or an authorized third party can extract the embedded data with the data hiding key and further recover the original image from the encrypted version.

The encrypted 8-bit gray-scale images are generated by encrypting every bit-planes with a stream cipher in all methods of [1]-[3]. The method in [1] segments the encrypted image into a number of nonoverlapping blocks sized by $a \times a$; each block is used to carry one additional bit. For this, pixels in each block are pseudo-randomly divided into two sets S_1 and S_2 according to a data hiding key. If the additional bit to be embedded is 0, flip the 3 LSBs of each encrypted pixel in S_1, otherwise flip the 3 encrypted LSBs of pixels in S_2. For data extraction and image recovery, the receiver flips all the three LSBs of pixels in S_1 to form a new decrypted block, and flips all the three LSBs of pixels in S_1 to form another new block; one of them will be decrypted to the original block. Due to spatial correlation in natural images, original block is presumed to be much smoother than interfered block and embedded bit can be extracted correspondingly.

PROPOSED METHOD

As shown in Fig. 1(b), the content owner first reserves enough space on original image and then converts the image into its encrypted version with the encryption key. Now, the data embed 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. Clearly, standard RDH algorithms are the ideal operator for reserving room before encryption and can be easily applied to Framework RRBE to achieve better performance compared with techniques from Framework VRAE.

Generation of Encrypted Image

In order to genereate the encrypted image, the first stage can be divided into three steps: image partition, self reversible embedding followed by image encryption. At the beginning, image partition step divides original image into two parts and ; then, the LSBs of are reversibly embedded into with a standard RDH algorithm.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Image Partition: here we use a standard RDH technique, so the goal of image partition is to construct a smoother area **B**, on which standard RDH algorithms such as [10], [11] 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 ***MxN*** and pixels $\mathbf{C}_{i,j} \in [0, 255], 1 \le i \le M, 1 \le j \le N$. 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 . In detail, every block consists of ***m*** rows, where , $m = \lceil l/N \rceil$, 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 subsequential blocks along the rows. For each block, define a function to measure its first-order smoothness

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

Higher relates to blockswhich contain relativelymore complex textures. The content owner, therefore, selects the particular block with the highest to be , and puts it to the front of the image concatenated by the rest part with fewer textured areas, as shown in Fig. 2.

Self-Reversible Embedding:

The goal of self-reversible embedding is to embed the LSB-planes of ***A*** into ***B*** by employing traditional RDH algorithms. For illustration, we simplify the method in [10] to demonstrate the process of self-embedding.

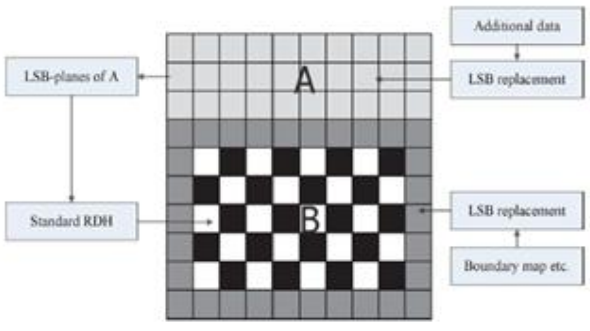

Fig. 2. Illustration of image partition and embedding process.

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. 2. Then, each white pixel $\mathbf{B}_{ij}$, is estimated by the interpolation value obtained with the four black pixels surrounding it as follows

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

Fig. 3 illustrates the idea of selecting proper points. Generally speaking, two solutions can gain significantly improvement in terms of PSNR when the length of data is relatively short.

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 $\mathbf{X}_{i,j}$ ranging from 0 to 255 can be represented by

8 bits, $\mathbf{X}_{i,j}(0), \mathbf{X}_{i,j}(1), \ldots, \mathbf{X}_{i,j}(7)$, such that

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

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

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

Data Hiding in Encrypted Image

After getting the encrypted image, the data hider 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 $\mathbf{A_E}$. Since $\mathbf{A_E}$ is rearranged to the top of , 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 . Finally,he sets a label following ***m*** to point out the end position of embedding process and

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

further encrypts ***m*** according to the data hiding key to formulate marked encrypted image denoted by **E'**.

IMPLEMENTATION ISSUES

Choice of LSB-Plane Number

When original image **C** is divided into **A** and **B**, the size of **A** is determined not only by the length of to-be-embedded messages but also by the number of LSB-planes embedded reversibly in **B**. The use of multiple LSB-planes takes into account the fact that the size of **B** can be enlarged with an in crease in embedding capability. It is more likely that **B** only need to implement embedding scheme once to accommodate LSB-planes of **A**, thus leading to distortion reduction. In other words, **A** shares part of distortion happens in **B.**

The comparison results measured by PSNR for three different choices of LSB-planes are there in table 1, where the embedding rate is measured by bits per pixel (bpp). The choice of single LSB-plane outperforms the other two at low embedding rate levels (less than 0.2 bpp). Utilizing multiple LSB-planes can only introduce average distortion from 0.5 to 1.75 (case of two LSB-planes) in , calculated by mean squared error (MSE). With a growing embedding rate, the gain by choosing two LSB-planes is espe cially significant, where the improvement can be as high as 2 to 4 dB over selecting single LSB-plane.

Choice of Embedding Strategy

In the single-layer embedding we introduce two solutions for embedding only a small portion of messages: 1) embedding data into peak points by making use of part error sequence; and 2) searching for proper points in the histogram of all estimating errors. The comparison results are listed in Table II. The first solution performs better than the other when cover image is relatively smooth with little fine-detail regions, therefore resulting in a sharper representation in error histogram. The improvement can be as high as 2 to 4 dB at low embedding rate levels. As for textured images such as Baboon with rather flat error histogram, the second solution has a better performance of 1 to 2 dB.

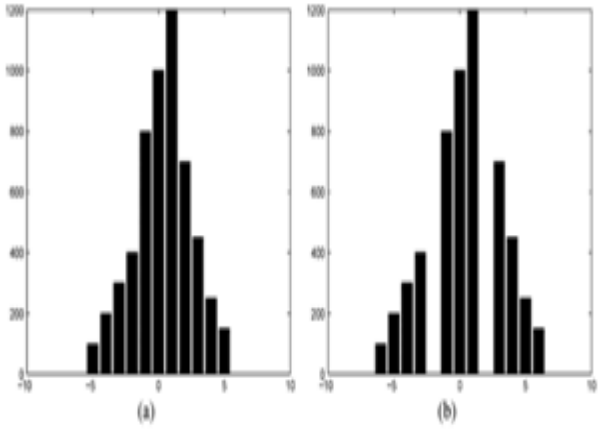

Fig. 3. Selection of proper points. (a) original histogram, (b) shifted histogram. (In this figure, length of messages is 1000 bits, $LP = -2$ and $RP = 2$.)

Fig. 4. Emergence of "Cut" artifact of Baboon image (embedding rate = 0.5 bpp for resolution): (a) Single LSB-plane applied (rectangle area), (b) two LSB-planes applied.

TABLE II
EMBEDDING STRATEGIES ANALYSIS UNDER VARIOUS EMBEDDING RATES

		PSNR results (dB)							
embedding rate (bpp)		0.005	0.01	0.05	0.1	0.2	0.3	0.4	0.5
Lena	peak points	67.16	63.44	55.46	52.33	49.07	45.00	40.65	35.84
	proper points	64.53	62.05	55.90	51.64	48.99	44.83	40.54	36.08
Airplane	peak points	65.94	63.18	57.02	54.20	50.98	48.26	44.67	40.78
	proper points	63.89	62.74	57.46	53.98	51.09	48.48	44.91	40.52
Barbara	peak points	65.39	62.56	55.56	51.46	47.68	43.56	39.24	34.80
	proper points	59.62	58.08	53.63	51.04	47.10	43.02	39.24	34.88
Baboon	peak points	57.493	55.71	50.19	46.17	40.68	35.87	31.16	25.92
	proper points	59.61	56.80	50.49	46.26	40.51	35.91	31.07	25.94
Peppers	peak points	63.77	61.30	54.17	51.02	46.00	42.08	36.91	—
	proper points	64.71	62.31	51.20	51.23	46.11	42.20	37.10	—
Boat	peak points	67.22	64.13	56.75	52.62	49.10	45.21	41.24	35.99
	proper points	63.28	60.75	55.53	51.62	49.03	45.29	41.36	35.99

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

TABLE III
LENGTH OF BOUNDARY MAP UNDER DIFFERENT EMBEDDING RATES

	Boundary map size (bits)							
embedding rate (bpp)	0.005	0.01	0.05	0.1	0.2	0.3	0.4	0.5
Lena	0	0	0	0	0	0	0	0
Airplane	0	0	0	0	0	0	0	0
Barbara	0	0	0	0	0	0	0	0
Baboon	0	0	0	0	0	2	18	109
Peppers	0	1	43	92	291	797	1741	—
Boat	0	0	0	0	0	0	0	0

(a) (b) (c) (d)

Fig. 5. (a) Original image, (b) encrypted image, (c) decrypted image containing messages (embedding rate 0.1 bpp), (d) recovery version.

Discussion on Boundary Map

In this paper, the Boundary Map is used for distinguishing between natural and pseudo boundary pixels and its size is critical to practical applicability of proposed approach. The boundary map size of six standard images is represented in table III. In most cases, no boundary map is needed. Even for Peppers image, the largest size is 1741 bits (with a large embedding rate 0.4 bpp by adopting embedding scheme 4 rounds) and the marginal area (**512 x 4 x 4 = 8912** bits) is large enough to accommodate it.

EXPERIMENTS AND COMPARISONS

Here we consider an image Lena, shown in Fig. 5(a), to describe the feasibility of proposed method. Fig. 5(b) is the encrypted image containing embedded messages and the decrypted version with messages is illustrated in Fig. 5(c). Fig. 5(d) depicts the recovery version which is identical to original image.

We did compare our proposed method with the state-of the- art works [1]–[3]. As mentioned in Section I, all methods in [1]–[3] maybe introduce some errors on data extraction and/or image restoration, while the proposed method is free of any error for all kinds of images.

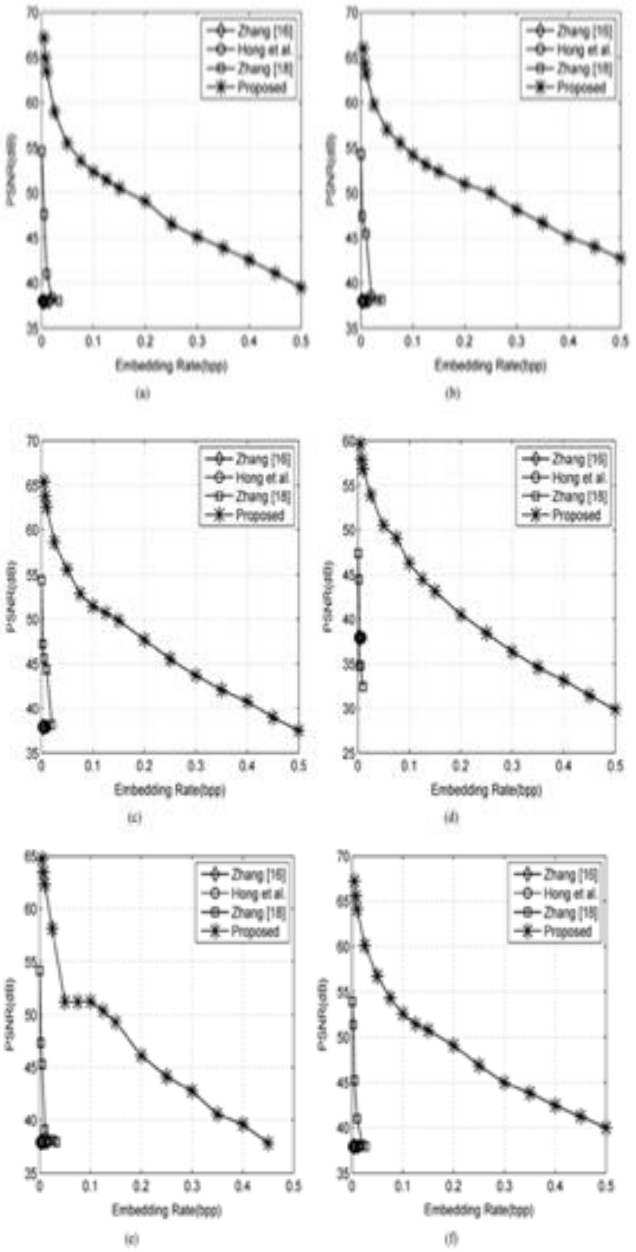

Fig. 6. PSNR comparison with the methods of Zhang [16], Hong [17] and Zhang [18]. (a) Lena, (b) Airplane, (c) Barbara, (d) Baboon, (e) Peppers, (f) Boat

CONCLUSION

Reversible data hiding in encrypted images is a new technique drawing attention because of the privacy-preserving requirements from cloud data management. Previous methods implement RDH in encrypted

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

images, as opposed to which we proposed by reserving room before performing encryption. In this way, the data hider can benefit from the extra space emptied out in previous stage. The proposed method can take advantage of all traditional RDH techniques for plain images and achieve excellent performance without loss. Furthermore, this novel method can achieve real reversibility, separate data extraction and greatly improvement on the quality of marked decrypted images.

REFERENCES

[1] X. Zhang, "Reversible data hiding in encrypted images," *IEEE Signal Process. Lett.*, vol. 18, no. 4, pp. 255–258, Apr. 2011.

[2] W. Hong, T. Chen, and H.Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Process. Lett.*, vol. 19, no. 4, pp. 199–202, Apr. 2012.

[3] X. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 826–832, Apr. 2012.

[4] 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.

[5] 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.

[6] W. Zhang, B. Chen, and N. Yu, "Improving various reversible data hiding schemes via optimal codes for binary covers," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 2991–3003, Jun. 2012.

[7] 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.

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

[9] 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.

[10] 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.