Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption

Dr. Sudhir S Kanade[1], Durga S. Patil[2]

[1] *Head of Department, Electronics and Telecommunication Engineering Dept., T.P.C.T's College of Engineering, Osmanabad, Maharashtra, India*
[2] *M.E. Electronics & Telecommunication Student, Electronics and Telecommunication Engineering Dept., T.P.C.T's College of Engineering, Osmanabad, Maharashtra, India*

ABSTRACT

Now a day, more attention is to reversible data hiding (RDH) in encrypted images as well as in audio and video, by using RDH method excellent property that the original image (cover) can be receives as it is recovered after embedded data is extracted also protecting the image content's confidentiality. All previous methods embedding data into image by reversibly vacating room in the encrypted images, which may be result as some errors on data extraction and/or image restoration. That mean some secrete information is loss in data extraction also degraded quality of image. In this paper, we propose a new method by reserving room before encryption .By using the new RHD method improver's efficiency of image. The proposed method improves efficiency & quality encrypted image usually used in medical area, aromatic etc. The new Algorithm Used in novel RDH is reducing noise Effect. Furthermore, this novel method can achieve real reversibility, separate data extraction and greatly improvement on the quality of marked decrypted images. The proposed method uses different techniques like Generation of Encrypted Images, Data hiding in encrypted images, data extraction and image recovery, and last but not the least data extraction and image restoration. For getting better results, we have compared the proposed method with the state-of the-art works. The proposed method can embed more than 10 times as large payloads for the same acceptable PSNR (e.g., PSNR = 40 dB) which implies a very good potential for practical applications. With the help of above mentioned techniques, Reversible Data Hiding in encrypted images by reserving room before encryption is possible.

Keyword: - *Reversible Data Hiding, Image Encryption, Novel Method of RDH, Encryption Techniques, Difference Expansion, Histogram Shift*

1. INTRODUCTION

Reversible data hiding (RDH) is a technique in image processing area for encryption, by which the original cover can be losslessly recovered after the embedded message, is extracted. The RDH approach is widely used in medical science, defense field and forensic lab, where there is no degradation of the original content is allowed. Since more research RDH method in recently. In theoretical aspect rate-distortion model for RDH Kalker and Willems [2], 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. 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. Many RDH techniques have emerged in recent years. Fridrich et al [3] constructed a general framework for RDH for method. By first extracting compressible features of original cover and then compressing them lossless, spare space can be saved for embedding auxiliary data.

A various RDH method is more popular is based on difference expansion (DE) [4], in which the difference of each pixel group is expanded by various method or technique. Example, multiplied by 2, and thus the least significant bits (LSBs) of the difference are all-zero and can be used for embedding messages. Another reliable strategy for RDH is histogram shift (HS), in which space is saved for data embedding by shifting the bins of histogram of gray values. With respective to providing confidentiality for images, encryption is an effective and popular means as it converts the original and meaningful content to non-readable one. Although there are few RDH techniques in encrypted images have been published yet, there are some promising applications if RDH can be

applied to encrypted images. 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.[5]In our system we provide the high quality image to the users. It also provides the more security of the data. The proposed system is reduces the time as well as cost as compared to previous system.

2. PREVIOUS ARTS

The methods proposed in [6]-[8] can be summarized as the framework, "vacating room after encryption (VRAE)", as illustrated in Fig. 1(a).

In this framework, a content owner encrypts the original image using a standard cipher with an encryption key. After producing the encrypted image, the content owner hands over 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. Then 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 according to the encryption key.

In all methods of [6]-[8], the encrypted 8-bit gray-scale images are generated by encrypting every bit-planes with a stream cipher. The method in [6] segments the encrypted image into a number of non-overlapping blocks sized by *a X a*; each block is used to carry one additional bit. To do this, pixels in each block are pseudo-randomly divided into two sets *S1* and *S2* according to a data hiding key. If the additional bit to be embedded is 0, flip the 3 LSBs of each encrypted pixel in *S1*, otherwise flip the 3 encrypted LSBs of pixels in *S2*. For data extraction and image recovery, the receiver flips all the three LSBs of pixels in *S1* to form a new decrypted block, and flips all the three LSBs of pixels in *S2* 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. However, there is a risk of defeat of bit extraction and image recovery when divided block is relatively small (e.g. a=8) or has much fine-detailed textures.

Hong *et al.* [7] reduced the error rate of Zhang's method [6] by fully exploiting the pixels in calculating the smoothness of each block and using side match. The extraction and recovery of blocks are performed according to the descending order of the absolute smoothness difference between two candidate blocks and recovered blocks can further be used to evaluate the smoothness of unrecovered blocks, which is referred to as side match.

Zhang's method in [8] pseudo-randomly permuted and divided encrypted image into a number of groups with size of L. The P LSB-planes of each group are compressed with a parity-check matrix and the vacated room is used to embed data. For instance, denote the pixels of one group by $x_1,\ldots,x_L$, and its encrypted P LSB-planes by c that consists of $\boldsymbol{P \cdot L}$ bits. The data hider generates a parity-check matrix $\boldsymbol{G}$ sized, $\boldsymbol{(P \cdot L - S) X P.L}$ and compresses $\mathbf{c}$ as its syndrome $\mathbf{s}$ such that $\mathbf{s = G \cdot c}$. Because the length of $\mathbf{s}$ is $\boldsymbol{(P \cdot L - S)}$, $\boldsymbol{S}$ bits are available for data accommodation. At the receiver side, $8 - \boldsymbol{P}$ the most significant bits (MSB) of pixels are obtained by decryption directly. The receiver then estimates x_i $(1 \le I \le L)$ by the MSBs of neighboring pixels, and gets an estimated version of $\boldsymbol{c}$ denoted by $\boldsymbol{c}'$. On the other hand, the receiver tests each vector belonging to the coset $\boldsymbol{\Omega(s)}$ of syndrome $\mathbf{s}$, where $\boldsymbol{\Omega(s) = \{u \,|G.u = s\}}$. From each vector of $\boldsymbol{\Omega(s)}$, the receiver can get a restored version of $\boldsymbol{c}$, and select the one most similar to the estimated version $\boldsymbol{c}'$ as the restored LSBs.

3. REVERSIBLE DATA HIDING (RDH)

Reversible data hiding in images is a technique that hides data in digital images for secret communication. It is a technique to hide additional message into cover media with a reversible manner so that the original cover content can be perfectly restored after extraction of the hidden message. Traditionally, data hiding is used for secret communication. In some applications, the embedded carriers are further encrypted to prevent the carrier from being analyzed to reveal the presence of the embedment. Other applications could be for when the owner of the carrier might not want the other person, including data hider, to know the content of the carrier before data hiding is actually performed, such as military images or confidential medical images. In this case, the content owner has to encrypt the content before passing to the data hider for data embedment. The receiver side can extract the embedded message and recover the original image. Many reversible data hiding methods have been proposed recently .As is well known, encryption is an effective and popular means of privacy protection. In order to securely share a secret image with other person, a content owner may encrypt the image before transmission. In some application scenarios, an inferior assistant or a channel administrator hopes to append some additional message, such as the origin information, image notation or authentication data, within the encrypted image though he does not know the original

image content. For example, when medical images have been encrypted for protecting the patient privacy, a database administrator may aim to embed the personal information into the corresponding encrypted images.

It may be also hopeful that the original content can be recovered without any error after decryption and retrieve of additional message at receiver side. Traditionally, data hiding is used for secret communication. In some applications, the embedded carriers are further encrypted to prevent the carrier from being analyzed to reveal the presence of the embedment. Other applications could be for when the owner of the carrier might not want the other person, including data hider, to know the content of the carrier before data hiding is actually performed, such as military images or confidential medical images. In this case, the content owner has to encrypt the content before passing to the data hider for data embedment. The receiver side can extract the embedded message and recover the original image. A major recent trend is to minimize the computational requirements for secure multimedia distribution by selective encryption where only parts of the data are encrypted. There are two levels of security for digital image encryption: low level and high-level security encryption. In low-level security encryption, the encrypted image has degraded visual quality compared to that of the original one, but the content of the image is still visible and understandable to the viewers. In the high-level security case, the content is completely scrambled and the image just looks like random noise. In this case, the image is not understandable to the viewers at all. Selective encryption aims at avoiding the encryption of all bits of a digital image and yet ensuring a secure encryption.

Reversible data hiding is a technique to embed additional message into some distortion-unacceptable cover media, such as military or medical images, with a reversible manner so that the original cover content can be perfectly restored after extraction of the hidden message. As an effective and popular means for privacy protection, encryption converts the ordinary signal into in comprehensible data, so that the general signal processing typically takes place before encryption or after decryption. However, in some circumstances that a content owner does not trust the service provider, the ability to manipulate the encrypted data when keeping the plain content secret is desired. When the secret data to be transmitted are encrypted, a channel provider without any knowledge of the cryptographic key may compress the encrypted data due to the limited channel resource. Encryption is an effective means of privacy protection. To share a secret image with other person, a content owner may encrypt the image before transmission. In some cases, a channel administrator needs to add some additional message, such as the origin information, image notation or authentication data, within the encrypted image however he does not know the original image content. It may be also expected that the original content can be recovered without any error after decryption and retrieve of additional message at receiver side. That means a reversible data hiding scheme for encrypted image is desirable. Data hiding is referred to as a process to hide data (representing some information) into cover media. That is, the data hiding process links two sets of data, a set of the embedded data and another set of the cover media data.

In most cases of data hiding, the cover media becomes distorted due to data hiding and cannot be inverted back to the original media. That is, cover media has permanent distortion even after the hidden data have been removed. In some applications, such as medical diagnosis and law enforcement it is desired that the original cover media can be recovered efficiently with no loss. The marking techniques satisfying this requirement are referred to as reversible, lossless, distortion-free or invertible data hiding techniques. Performance of a reversible data-embedding algorithm Reversible data embedding, which is also called lossless data embedding, embeds invisible data (which is called a payload) into a digital image in a reversible fashion. As a basic requirement, the quality degradation on the image after data embedding should be low. An exciting feature of reversible data embedding is the reversibility, that is, one can remove the embedded data to restore the original image. Reversible data embedding hides some information in a digital image in such a way that an authorized party could decode the hidden information and also restore the image to its original state. The performance of a reversible data-embedding algorithm can be measured by the following

- Payload capacity limit
- Visual quality
- Complexity

The distortion- free data embedding is the motivation of reversible data embedding. Data will certainly change the original content by embedding some data into it. Even a very slight change in pixel values may not be desirable, especially in sensitive imagery, such as military data and medical data. In such a scenario, every bit of information is important. From the application point of view, since the difference between the embedded image and original image is almost unnoticeable from human eyes, reversible data embedding could be thought as a secret communication channel since reversible data embedding can be used as an information carrier.

4. PROPOSED METHOD

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(b), the content owner first reserve enough space on original image and then converts the image into its encrypted version with the encryption key. Now, the data embedding 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. Obviously, 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. This is because in this new framework, we follow the customary idea that first losslessly compresses the redundant image content (e.g., using excellent RDH techniques) and then encrypts it with respect to protecting privacy.

Next, we elaborate a practical method based on the Framework "RRBE", which primarily consists of four stages: generation of encrypted image, data hiding in encrypted image, data extraction and image recovery. Note that the reserving operation we adopt in the proposed method is a traditional RDH approach.

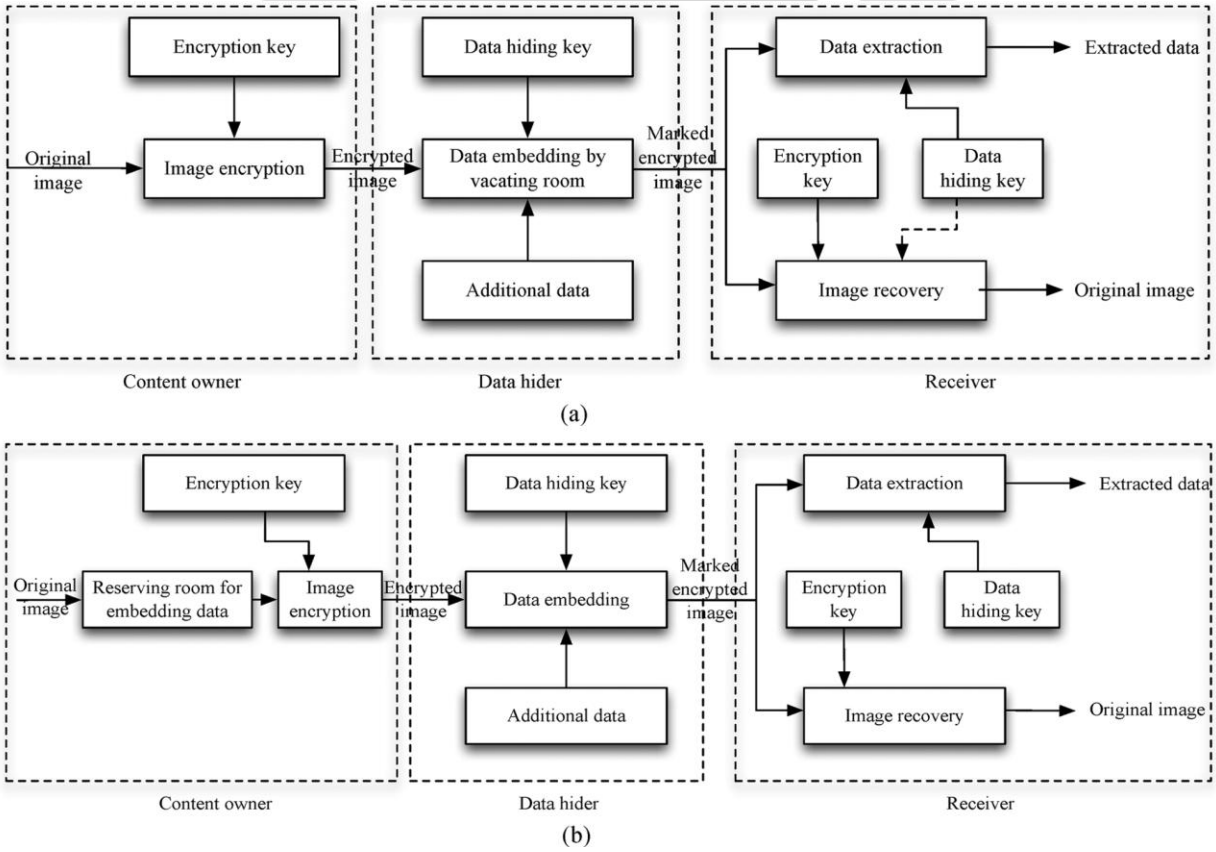


Fig-1: Framework: "vacating room after encryption (VRAE)" versus framework: "reserving room before encryption (RRBE)." (Dashed line in (a) states that the need of data hiding key in image recovery varies in different practical methods). (a) Framework VRAE. (b) Framework RRBE.

4.1 Encrypted Image Generation

In this module, to construct the encrypted image, the first stage can be divided into two steps. Image Partition and Self Reversible Embedding followed by image encryption. At the beginning, 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.

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.

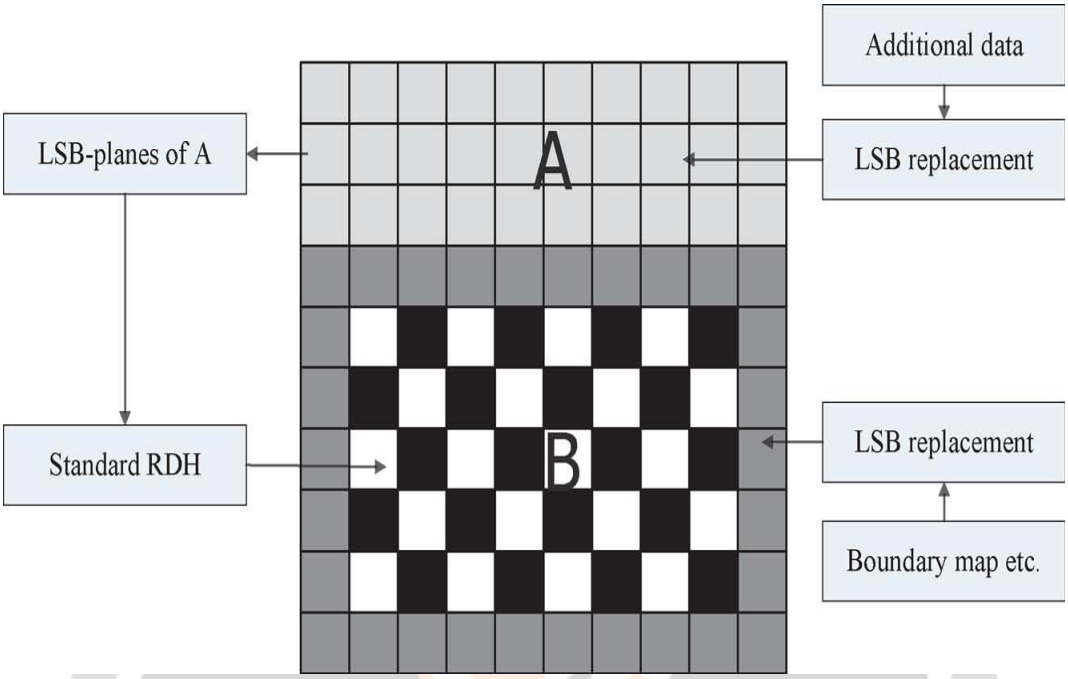


Fig - 2: Illustration of image partition and embedding process.

4.2 Data Hiding In Encrypted Image

In this module, a content owner encrypts the original image using a standard cipher with an encryption key. After producing the encrypted image, the content owner hands over it to a data hider (e.g., a database manager) and the data hider can embed some auxiliary data into the encrypted image by lossless vacating some room according to a data hiding key. Then 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 according to the encryption key.

4.3 Data Extraction and Image Recovery

In this module, the data extraction is completely not dependent on image decryption, hence this order implies two different ways of practical applications such as;

1) Extracting Data From Encrypted Images:-

To manage and update personal information of images which are encrypted for protecting clients' privacy, an inferior database manager may only get access to the data hiding key and have to manipulate data in encrypted domain. When the database manager gets the data hiding key, he can decrypt and extract the additional data 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 up dated 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.

2) Extracting Data From Decrypted Images:-

In this case, 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. Note that 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 decrypted 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 suitable for this case.

More specifically, the distortion is introduced via two separate ways: the embedding process by modifying the LSB-planes of and self-reversible embedding process by embedding LSB planes of into . The first part distortion is well controlled via exploiting the LSB-planes of only and the second part can benefit from excellent performance of current RDH techniques.

4.4 Data Extraction and Image Restoration

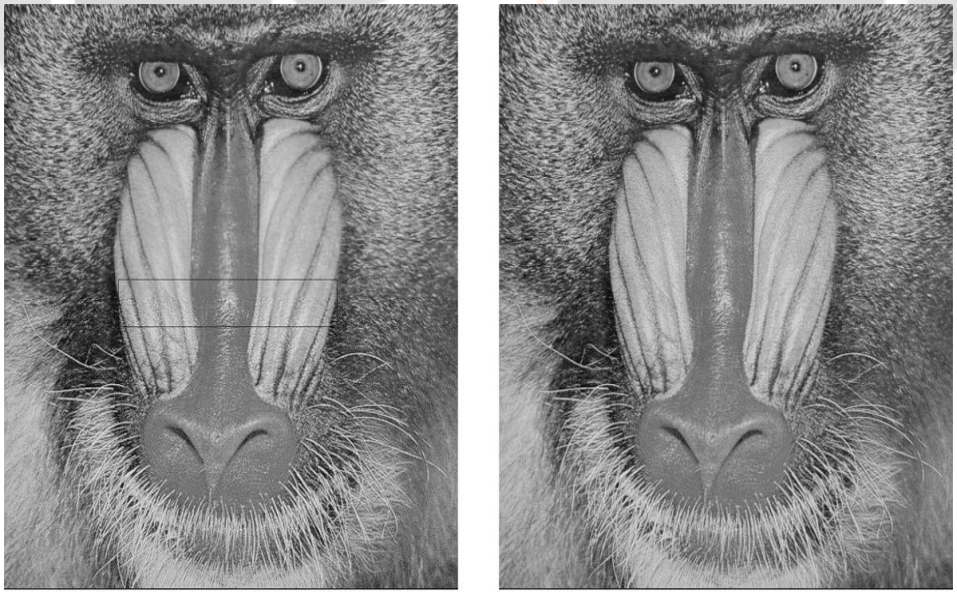
In this module, after generating the marked decrypted image, the content owner can further extract the data and recover original image. Reversible hiding allows extraction of the original host signal and also the embedded message. There are two important requirements for reversible data hiding techniques: the embedding capacity should be large; and distortion should be low. These two requirements conflict with each other. In general, a higher embedding capacity results in a higher degree of distortion. An improved technique embeds the same capacity with lower distortion or vice versa.

5. IMPLEMENTATION ISSUES

The proposed approach will be tested on public available standard images, which include "Lena", "Airplane", "Barbara", "Baboon", "Peppers" and "Boat" [9]. The size of all images is 512 x 512 x 8. The objective criteria PSNR is employed to evaluate the quality of marked decrypted image quantitatively. To achieve high PSNRs, several implement details for the proposed method are discussed first.

5.1 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 increase in embedding capability. Therefore, it is more likely that **B** only need to implement embedding scheme once to accommodate LSB-planes of **A**, thus leading to distortion reduction.



(a) (b)

Fig – 3: Emergence of "Cut" artifact of Baboon image (embedding rate is 0.5 bpp for visibility). (a) Single LSB-plane applied (rectangle area), (b) two LSB-planes applied.

Table – 1: 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	60.73	55.53	51.62	49.03	45.29	41.36	35.99

Table-2: 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 - 4: (a) Original image, (b) encrypted image, (c) decrypted image containing messages (embedding rate 0.1 bpp), (d) recovery version.

In other words, **A** shares part of distortion happens in **B**. The choice of single LSB-plane outperforms the other two at low embedding rate levels (less than 0.2 bpp). It is consistent with our intuitive understanding: when embedding rate is small, **B** has the capacity to embed LSBs of **A** in a single round without size enlargement. Utilizing multiple LSB-planes can only introduce average distortion from 0.5 to 1.75 (case of two LSB-planes) in **A**, calculated by mean squared error (MSE). With a growing embedding rate, the gain by choosing two LSB-planes is especially significant, where the improvement can be as high as 2 to 4 dB over selecting single LSB-plane.

Furthermore, we prefer using two LSB-planes to single one when their performances are competitive in embedding rate range from 0.2 to 0.3 bpp. This is because by allocating part distortion of **B** into **A**, the "cut" artifact depicted in Fig. 3, can be reduced to a certain degree. Additionally, we cannot expect any significant improvement by exploiting three LSB-planes of **A** from the table. In practice, we utilize single LSB-plane to embed messages when embedding rate is less than 0.25 bpp, and switch to two LSB-planes with embedding rate larger than 0.25bpp.

5.2 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 I. 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. Note that the performance of two solutions gradually approaches the same with little difference at large embedding rate range. In this paper, we propose the first solution when peak points of estimating error sequence of cover image account for more than 20% of the whole errors; otherwise switch to the second.

5.3 Discussion on Boundary Map

Boundary map in this paper is used for distinguishing between natural and pseudo boundary pixels and its size is critical to practical applicability of proposed approach. Table II shows the boundary map size of six standard images. 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 (bits) is large enough to accommodate it.

6. EXPERIMENTS AND COMPARISONS

We take standard image Lena, shown in Fig. 4(a), to demonstrate the feasibility of proposed method. Fig. 4(b) is the encrypted image containing embedded messages and the decrypted version with messages is illustrated in Fig. 4(c). Fig. 4(d) depicts the recovery version which is identical to original image.

We have compared the proposed method with the state-of the-art works [6]-[8]. As mentioned in Section I, all methods in [6]-[8] 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.

The quality of marked decrypted images is compared in the term of PSNR. Out of fairness, we modify the methods in [6], [7] with error-correcting codes to eliminate errors. By introducing error-correcting codes, the pure payload of [6], [7] is reduced from *Cap* to $Cap(1 - H(\rho))$, where $H(\rho)$ is the binary entropy function with error rate ρ. Take test image Baboon for instance. As for the method in [8], we only choose those results with a significantly high probability of successful data extraction and perfect image recovery to draw the curves. The gain in terms of PSNR is significantly high at embedding rate range that the methods in [6]-[8] can achieve. In addition, another advantage of our approach is the much wider range of embedding rate for acceptable PSNRs. In fact, the proposed method can embed more than 10 times as large payloads for the same acceptable PSNR (e.g., PSNR = 40 dB) as the methods in [6]-[8], which implies a very good potential for practical applications.

7. 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.

8. REFERENCE

[1] Wagh Mahesh J, Manish Koul, Murtadak Sona U, Shinde Kavita S, Prof. Bhandare M.G, "RDH (Reversible Data Hiding) in Encrypted Images by Reserving Room Before Encryption", Volume 4, Issue 4, April 2014.

[2] 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.

[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. 2003.2010.Process., vol. 19, no. 4,pp. 1097–1102, Apr. 2010.

[5] 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.

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

[7] 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.

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

[9] Miscelaneous Gray Level Images [Online]. Available:http://decsai.ugr.es/cvg/dbimagenes/g512.php