International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Reversible Data Hiding in Encrypted Images

Chaple Gopal[1], G Balram[2]

[1]M.Tech Sudent, Department of CSE, Anurag Group of Institutions, Hyderabad, India

[2]Assistant Professor, Department of CSE, Anurag Group of Institutions, Hyderabad, India

Abstract: ***In recent times, more attention is paid to reversible data hiding (RDH) in encrypted images, since it preserves the excellent property that the original cover can be losslessly improved after embedded data is extracted while protecting the image content's privacy. All earlier methods embed data by reversibly vacating room from the encrypted images, which may be raise some errors on data extraction and image restoration. In this paper, we proposed a novel method by reserving room before encryption with a traditional RDH algorithm, and thus it is easier for the data hider to reversibly embed data in the encrypted image. Proposed method that can achieve the real reversibility, data extraction and image recovery are free of any error. This Experiments show that this novel method can embed more than 10 times as large payloads for the same image quality as the earlier methods, such as for PSNR dB.***

Keywords: Reversible data hiding, image encryption, privacy protection, histogram shift

1. Introduction

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

In the theoretical part, Kalker and Willems [1] established a rate-distortion method of RDH, through which they proved the rate-distortion bounds of RDH for memory-less covers and proposed a recursive code construction which, however it does not approach the bound. Zhang *et al.* [2], [3] was 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 the entropy, that establishes the equivalence between the both data compression and RDH for binary covers.

In practical aspect, a lot of RDH techniques have emerged in recently. Fridrich*et al.* [4] constructed a general framework for RDH. In this first extracting compressible features of original cover and then compressing them lossless and spare space can be saved for embedding auxiliary data. A most popular method is based on difference expansion (DE) [5], in which the difference of each pixel group is expanded, e.g., is multiplied by 2, and thus the least significant bits (LSBs) of the difference are all-zero and can be used for embedding messages. Another hopeful strategy for RDH is histogram shift (HS) [6], in which space is saved for data embedding by shifting the bins of histogram of grey values. The state-of-art methods [7]–[11] usually combined the both DE and HS to residuals of the image, e.g., the expected 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. Al-though few RDH techniques in encrypted images have been published yet, we have some promising applications if RDH can be applied to encrypted images. 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 that data encryption and coloring offer possibilities for upholding the content owner's privacy and data integrity.

The cloud service provider has no right to introduce permanent distortion during data coloring into encrypted data. Hence, a reversible data coloring technique based on encrypted data is preferred. Suppose a medical image data-base is stored in a data centre, and a server in the data centre can embed notations into an encrypted version of a medical image through a RDH technique. With this, the server can manage the image or verify its integrity without having the knowledge of the original content, with that he patient's privacy is protected. On the further hand, a doctor is having the cryptographic key can be decrypt and restore the image in a reversible manner for the purpose of further diagnosing.

Some efforts on RDH in encrypted images have been made. Zhang divided the encrypted image into several blocks by flipping the 3 LSBs of the half of pixels in each block and room can be vacated for the embedded bit. With that the data extraction and image recovery proceed by finding which part has been flipped in one block. This method can be realized with the help of spatial correlation in decrypted image. In Hong *et al.* [17] ameliorated Zhang's method at the decoder side by further exploiting the spatial correlation using a different approximation equation and side match technique to achieve much lower error rate. These two methods are mentioned above rely on spatial correlation of original image to extract data. It remains the encrypted image should be decrypted first before data extraction.

To separate the data extraction from image decryption for data embedding following the idea of compressing encrypted images [14], [15]. The compression of encrypted data can be formulated as source coding with side information at the decoder [14], with that the typical method is to generate the compressed data in lossless manner by exploiting the patterns of parity- check matrix of channel codes. In the [18] method compressed the encrypted LSBs to vacate room for

additional data by finding syndromes of a parity -check matrix and the side information used at the receiver side is also the spatial correlation of decrypted images.

In all the three methods try to vacate room from the encrypted images directly. However, the entropy of encrypted images has been maximized, with these techniques we can only achieve small payloads [16], [17] or generate marked image with poor quality for large payload [18] and all of them are subject to some error rates on data extraction and image restoration. Even though the methods in [16], [17] can eliminate errors by error correcting codes and also the pure payloads will be further consumed.

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 recommended 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 that the positions of these LSBs in the encrypted image can be used to embed data. It was 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 the data extraction and image recovery are free of any error.
- For particular 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.

2. Previous Arts

The methods proposed in [16]–[18] can be summarized as the framework, Fig. 1(a) illustrates "vacating room after encryption (VRAE)".

In this framework, a content owner encrypts the original image with an encryption key using a standard cipher. After producing the encrypted image, owner of the content 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. After that a receiver, maybe the content owner himself or an authorized third party can extract the embedded data with the data hiding key and finally recover the original image from the encrypted version according to the encryption key.

In all methods of [16]–[18], the encrypted 8-bit grayscale images are generated by encrypting every bit-planes with a stream cipher. The method in [16] segments the encrypted image into a number of non-overlapping 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 0 is embedded, 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 will flips all the three LSBs of pixels in S_1 to form a new decrypted block. Also the receiver flips all the three LSBs of pixels in S_2 to form another new block; one of them will be decrypted to the original block. Because of spatial correlation in natural images, the original block is presumed to be much smoother than interfered block and embedded bit can be extracted correspondingly. But, there is a risk of defeat of bit extraction and image recovery when divided block is relatively small (e.g., $a=8$) which has much fine-detailed textures.

Hong *et al.* [17] reduced the error rate of Zhang's method [16] by fully exploiting the pixels in calculating the smoothness of each block and using side match. Side match is referred to as 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 smooth-ness of unrecovered blocks referred as side match.

Zhang's method in [18] pseudo-randomly permuted and divided encrypted image into a number of groups with size **L**. The each groupof ***P*** LSB-planes are compressed with a parity-check matrix and the vacated room is used to embed data. Further instance we denote the pixels of one group by x_1 ,…… $x_L$and its encrypted ***P*** LSB-planes by ***c***that consists of ***P . L*** bits. The data hider generates a parity-check matrix ***G*** sized ***(P . L - S)*** x***P . L***, and compresses ***C*** as its syndrome **s** such that ***S = G. C.*** Because the length of it is***(P . L - S), S*** bits are available for data accommodation.

At the receiver side, the ***8 – P*** 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 gets an estimated version of C denoted by C' . On the other hand, the receiver tests each vector belonging to the cosetΩ(s) of syndrome, where Ω(s) = { u | G.u= s }. From each vector of Ω(s), the receiver can get a restored version of ***c*** , and select the one most similar to the estimated version ***c'*** as the restored LSBs.

3. Proposed Method

Since losslessly vacating room from the encrypted images is relatively difficult and sometimes inefficient, so it is obsessed to find novel RDH techniques working directly for encrypted images. In this RDH technique if we reverse the order of encryption and vacating room, which is reserving room prior to image encryption at content owner side "reserving room before encryption (RRBE)"is framework used when the RDH tasks in encrypted images would be more natural and much easier.

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. Then, 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. In which data extraction and image recovery are identical to that of Framework VRAE. Particularly, standard RDH algorithms are the ideal operator for reserving room be-fore encryption and can be easily applied to Framework RRBE to achieve

better performance compared with techniques from Framework VRAE. Because in this new framework, 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 Frame-work "RRBE", which primarily consists of four stages: 1) generation of encrypted image, 2) data hiding in encrypted image, 3) data extraction and 4) image recovery. Here, the reserving operation we adopt in the proposed method is a traditional RDH approach.

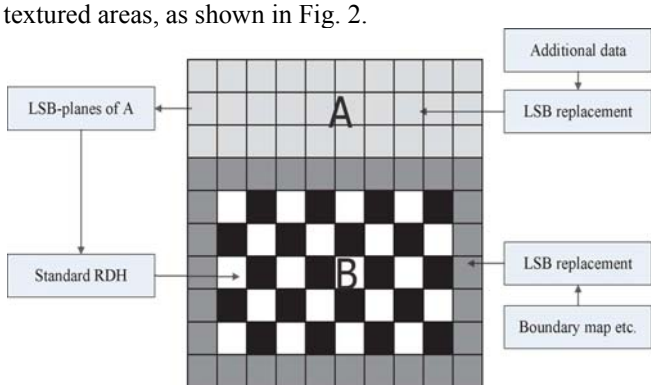
3.1 Generation of Encrypted Image

Actually, to construct the encrypted image, first stage can be divided into three steps: image partition and self-reversible embedding followed by image encryption. 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; latter, encrypt the rearranged image to generate its final version.

1) Image Partition*:* The operator here for reserving roombefore encryption is a standard RDH technique, so the main goal of image partition is to construct a smoother area **B** , on which standard RDH algorithms such to achieve better performance. To ensure that, without loss of generality to assume the original image is an 8 bits gray- **C** scale image with its size $M\times N$ and $\leq$ pixels $\mathbf{C}_{i,j}\in[0,255], 1\leq i\leq M, 1\leq j\leq N$. Initially, the content owner extracts from the original image, along with the rows, several overlapping blocks whose number is determined by the size of embedded messages, represented by l. 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 sub sequential blocks along the rows. Each block is define a function to measure its first-order smoothness

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

Higher f relates to blocks which contain relatively more complex textures. The content owner, therefore, selects the particular block with the highest f to be **A**, and puts it to the front of the image concatenated by the rest part **B** with fewer textured areas, as shown in Fig. 2.



The above discussion implicitly relies on the fact that only single LSB -plane of A is recorded. It is straightforward that the content owner can also embed two or more LSB-planes of **A** is recorded. It is straightforward that the content owner can also embed two or more LSB-planes of **A into B** which leads to half or more than half reduction in size of **A.** However the performance of **A,** in terms of PSNR after data embedding in the second stage decreases significantly with growing bit-planes exploited. Hence, in this paper we investigate situations that at most three LSB-planes of **A** are employed and determine the number of bit-plane with regard to different payloads experimentally in the next section.

2) Self-Reversible Embedding: The goal of self-reversibleembedding is to embed the LSB-planes of **A** into **B** by employing traditional RDH algorithms. We simplify the method in [10] to demonstrate the process of self-embedding. This step does not rely on any specific RDH algorithm pixels in the rest of image **B** are first categorized into two sets: white pixels with its indices i and j satisfying (i + j) mod2 = 0 and black pixels whose indices meet (i + j) mod2 = 1, as shown in Fig. 2. Then, for the each white pixel, $\mathbf{B}_{i,j}$ which is estimated by the interpolation value obtained with the four black pixels surrounding it as follows

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

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 $e_{i,j}=\mathbf{B}_{i,j}-\mathbf{B}'_{i,j}$ and then some data can be embedded into the estimating error sequence with histogram shift, which will be defined later. Further we calculate the estimating errors of black pixels with the help of surrounding white pixels that may have been modified. The other estimating error sequence is generated which can accommodate messages as well. Also, we can implement multilayer embed-ding scheme by considering the modified **B** as "original" one when needed. In summary, to exploit all pixels of**B**, two estimating error sequences are constructed for embedding messages in every single-layer embedding process.

Through bidirectional histogram shift, some of the messages can be embedded on each error sequence. That remains, first divide the histogram of estimating errors into two parts left part and the right part and search for the highest point in each part, de-noted by LM and RM, respectively. For typical images, $LM=-1$and $RM=0$. Furthermore, search for the zero point in each part, denoted by LNandRN. To embed messages into positions with an estimating error that is equal to RM, shift all error values between $RM+1$ and $RN-1$ with one step toward right and then after we can represent the bit 0 with RM and the bit 1 with $RM+1$. The embedding process in the left part is similar except that the shifting direction is left and the shift is realized by subtracting 1 from the corresponding pixel values.

The same with other RDH algorithms is overflow or underflow problem occurs when natural boundary pixels change from 255 to 256 or from 0 to -1 . To avoid it, we only embed data into estimating error with its corresponding pixel valued from 1 to 254. Although ambiguities still arise when nonboundary pixels are changed from 1 to 0 or from

254 to 255 during the embedding process. These created boundary pixels in the embed-ding process are defined as pseudo-boundary pixels. Boundary map is introduced to tell whether boundary pixels in marked image are natural or pseudo in extracting process. It is a binary sequence with bit "0" for natural boundary pixel, bit "1" for pseudo-boundary pixel. Later estimating errors of marginal area of $\mathbf{B}$ cannot be calculated via (2), to make the best use of $\mathbf{B}$ we choose its marginal area shown to place the boundary map we use LSB replacement to embed it and original LSBs of marginal area is assembled with messages, i.e., LSB-planes of $\mathbf{A}$, and reversibly embedded into $\mathbf{B}$. In most cases, even with a large embedding rate of the length of boundary map is very short the marginal area of $\mathbf{B}$ is enough to accommodate it.

Numerous parameters such as $LNRNLMRMLPRP$, payloads are embedded into the some estimating errors of black pixels Rb , total embedding rounds$x$, start row SR and end row ER of $\mathbf{A}$ in original image is embedded into marginal area in a similar way these parameters play an important role in data extraction and image recovery process.

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

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

The encrypted bits $\mathbf{E}_{i,j}(k)$ calculated through exclusive-or operation

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

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

B. Data Hiding in Encrypted Image

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

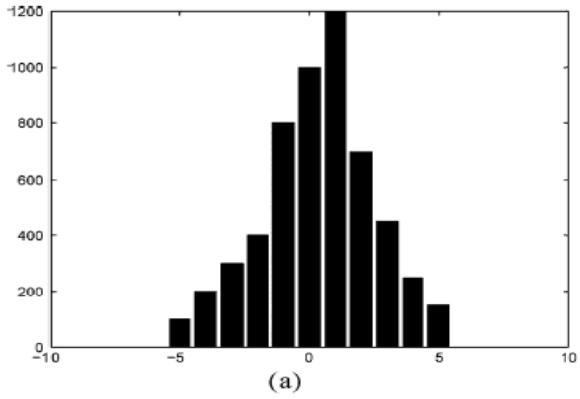


Figure 1: selection of proper points (a). Original Histogram

C. Data Extraction and Image Recovery

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

Case 1: Extracting Data from Encrypted Images: Tomanage and update personal information of images which are encrypted for protecting clients' privacy, a database manager may only get access to the data hiding key and have to manipulate data in encrypted domain. The data extraction before image decryption guarantees the feasibility of our work in this case. When the database manager gets the data hiding key he can decrypt the LSB-planes of $\mathbf{A}_E$ and extract the additional data $\mathbf{m}$ by directly reading the decrypted version. Later requesting for updating information of encrypted images the database manager will updates information through LSB replacement and encrypts updated information according to the data hiding key all over again. The whole process entirely operated on encrypted domain will avoids the leakage of original content.

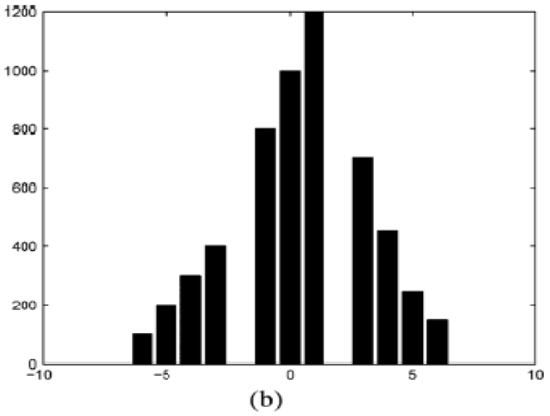


Figure 2: selection of proper points (b). Shifted Histogram

(In this figure, length of messages is 1000 bits, LP = -2 and RP = 2)

Case 2: Extracting Data from Decrypted Images: In the first Case, both embedding and extraction of the data are manipulated in encrypted domain. Further, 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. We see the following example is an application for such scenario. Assume that Alice was out-sourced her images to a cloud server, and those images are encrypted to protect their contents. That encrypted images the cloud server marks the images by embedding some notation

includes the identity of the images' owner the identity of the cloud server and time stamps, to manage the encrypted images. The cloud server doesn't have any right to do any permanent damage to the images. Therefore authorized user Bob who has been shared the encryption key and the data hiding key, down-loaded and decrypted the images. Bob is hoped to get the marked decrypted images those decrypted images still including the notation, which that it can be used to trace the source and history of the data. The direction of image decryption before/without data extraction is perfectly suitable for this case. Afterward, we describe how to generate a marked decrypted image..

a) Generate Marked Decrypted Image: To form themarked decrypted image $\mathbf{X}$" which is made up of $\mathbf{A}$" and $\mathbf{B}$" the content owner should do following two steps.

Step 1. With the help of encryption key, the content owner will decrypts the image except the LSB-planes of $\mathbf{A}_E$. The decrypted version of $\mathbf{E}$' containing the embedded data can be calculated by

$$\mathbf{X}''_{i,j}(k) = \mathbf{E}'_{i,j}(k) \oplus r_{i,j}(k) \quad (5)$$

and

$$\mathbf{X}''_{i,j} = \sum_{k=0}^{7} \mathbf{X}''_{i,j}(k) \times 2^k, \quad (6)$$

Step 2. Extract SR and ER in marginal area of $\mathbf{B}$". By rearranging A" and B" to its original state that the plain image containing embedded data is obtained.

As can be seen, the marked decrypted image $\mathbf{X}$" is identical to rearranged $\mathbf{X}$ except LSB-planes of**A**. At the meantime, it keeps perceptual transparency compared with original image**C**. More specifically, the distortion is introduced via two separate ways: the embedding process by modifying the LSB-planes of **A** and self-reversible embedding process by embedding LSB-planes of **A**into**B**. The first part distortion is well controlled via exploiting the LSB -planes of $\mathbf{A}$ only and the second part can benefit from excellent performance of current RDH techniques.

b) Data Extraction and Image Restoration: After generating the marked decrypted image the content owner further extract the data and recover original image. This process is similar to that of traditional RDH methods [10], [11]. The following outlines the specific steps:

Step 1. Record and decrypt the LSB-planes of **AE** according to the data hiding key; extract the data until the end label is reached.

Step 2. Extract $LNRNLMRMLPRPR_{bx}$ and boundary map from the LSB of marginal area of $\mathbf{B}$". Then, scan $\mathbf{B}$" to undertake the following steps.

Step 3. If R_b is equal to 0, which means no black pixels participate in embedding process, then go to Step 5.

Step 4.calculate the errors *e'i,j* and estimates black pixels *B"i,,j* which belongs to [1, 254] to recover the estimating Error and original pixel value in a reverse order and extract embedded bits when *e'i,j* is equal to LN,LM (or LP),RM(or RP) and RN. Else, if *B"i,,j* € [1, 254], refer to the corresponding bit *b* in boundary map. If here *b=0*, skip this one or else operate like *B"i,,j* € [1, 254], . Repeat this step until the part of payload *Rb* is extracted. If payload *Rb* is extracted bits are LSBs of pixels in marginal area restores them immediately.

Step 5. Calculate estimating errors *e'i,j* of white pixels *B"i,,j* and extract embedded bits and recover white pixels in the same manner with Step 4. If these extracted bits are LSBs of pixels in marginal area restores them immediately.

Step 6. Continue doing Step 2 to Step 5 x-1 rounds on *B*"and merge all extracted bits to form LSB-planes of A. Until now, we have perfectly recover B.

Step 7. Replace marked LSB-planes of A" with its original bits extracted from B″ to get original cover image C. We note that if the content owner wants to retrieve his image in Case 1, will do the procedures are exactly the same in Case 2. So, it is omitted in Case 1 for simplicity.

4. Conclusion

Reversible data hiding in encrypted images is a new topic drawing attention because of the privacy-preserving requirements from cloud data management. Earlier methods are implement RDH in encrypted images by vacating room after encryption was opposed to which we proposed by reserving room before encryption. With that the data hider can benefit from the extra space emptied out in previous stage to make data hiding process effort-less. The proposed method can take advantage of all traditional RDH techniques for plain images and achieve excellent performance without loss of perfect secrecy. Moreover, this novel method can achieve real reversibility, 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 Pro-cessing (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] 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.

[4] J. Fridrich and M. Goljan, "Lossless data embedding for all image for-mats," in *Proc. SPIE Proc. Photonics West, Electronic Imaging, Secu-rity and Watermarking of Multimedia Contents*, San Jose, CA, USA,Jan. 2002, vol. 4675, pp. 572–583.

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

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

[7] 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.

[8] 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.

[9] P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Process.*, vol. 89, pp. 1129–1143, 2009.

[10] L. Luo *et al.*, "Reversible image watermarking using interpolation tech-nique," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 187–193, Mar. 2010.
[11] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.-Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans.Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 989–999, Jul. 2009.
[12] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook ofApplied Cryptography*. Boca Raton, FL, USA: CRC, 1996.
[13] K. Hwang and D. Li, "Trusted cloud computing with secure resources and data coloring," *IEEE Internet Comput.*, vol. 14, no. 5, pp. 14–22, Sep./Oct. 2010.
[14] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," *IEEE Trans. SignalProcess.*, vol. 52, no. 10, pp. 2992–3006, Oct. 2004.
[15] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of en-crypted grayscale images," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 1097–1102, Apr. 2010.
[16] X. Zhang, "Reversible data hiding in encrypted images," *IEEE SignalProcess. Lett.*, vol. 18, no. 4, pp. 255–258, Apr. 2011.
[17] 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.
[18] X. Zhang, "Separable reversible data hiding in encrypted image," *IEEETrans. Inf. Forensics Security*, vol. 7, no. 2, pp. 826–832, Apr. 2012.
[19] Miscelaneous Gray Level Images [Online]. Available:http://decsai.ugr.es/cvg/dbimagenes/g512.php

Author Profile

Chaple. Gopal received the B.Tech degree in computer science and Engineering from JNTU Hyderabad in 2011 and pursuing M.Tech Degree in Computer Science and Engineering from CVSR College of Engineering from ANURAG Group of Institutions, JNTU Hyderabad, India

G. Balram working as assistant professor in Computer Science Engineering from CVSR College of Engineering from ANURAG Group of Institutions, Venkatapur(V), Ghatkesar (M), RangaReddy District, Hyderabad-500088, Telangana State.