ISSN: 2321-7782 (Online)
Volume 2, Issue 1, January 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Paper
Available online at: www.ijarcsms.com

Reversible Encryption and Data Hiding

Meenal V. Jagdale[1]
M.Tech
Department of Electronics and Telecommunication
G.H.Raisoni College of Engineering and Technology For Women
Nagpur - India

Dr. Shubhalaxmi P. Hingway[2]
Professor
Department of Electronics and Telecommunication
G.H.Raisoni College of Engineering and Technology For Women
Nagpur - India

Sheeja S. Suresh[3]
Asst. Prof
Department of Electronics and Telecommunication
G.H.Raisoni College of Engineering and Technology For Women
Nagpur - India

Abstract: ***Recently, more and more attention is paid to reversible data hiding (RDH) in encrypted images, since it maintains the excellent property that the original image cover can be losslessly recovered after which is embedded is extracted while protecting the image content's as confidential. All methods used previously embed data by reversibly vacating room from the images which are been encrypted, which may cause some errors on data extraction and/or image restoration. In this paper, we propose a novel method by reserving room before encryption with a traditional RDH algorithm, and thus it becomes 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 larger payloads for the same image quality as the previously used methods, such as for PSNR dB.***

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

I. INTRODUCTION

Data hiding is a technique that conceals secret data into a carrier to convey messages. Digital images are one of the media that is suitable to convey messages due to several reasons. Firstly, digital images are often transmitted over the Internet which would arouse little suspicion. Secondly, the high correlation between pixels provides rich space for data embedding. When a digital image is used as a carrier, the image used to embed data is known as the cover image, and the image with data embedded is called the stego image. In the embedding process, the pixels of the cover image are modified and thus, distortion occurs. In general, the more the cover image is distorted, the more vulnerable is the stego image to steganalytic attempts. To prevent the stego image from being suspicious and detected, either visually statistically, the distortions caused by embedding the data should be as minimum as possible, which imply that a high quality embedded image is demanded. For most of the existing data hiding techniques, the distortions caused by data embedding are permanent i.e. the stego image cannot be restored to its original state. However, for some applications, such as medical or military images, it is desired that the original cover image can be completely recovered because of the requirements for legal considerations or high precision nature. To fulfill these requirements, the reversible data hiding scheme for high quality images is introduced.

In 2003, Tian proposed a reversible data hiding method using the difference expansion technique. In his method, one bit can be embedded into two consecutive pixels; therefore, the maximum embedding capacity is 0.5 bpp. Alattar generalized the difference expansion technique so that n-1 bits can be embedded into n pixels, resulting the maximum embedding capacity (n-1)/n bpp. However, the difference expansion based reversible data hiding methods have to double the differences between pixels; therefore, a larger distortion occurs and may not be suitable for applications where high quality images are demanded. In

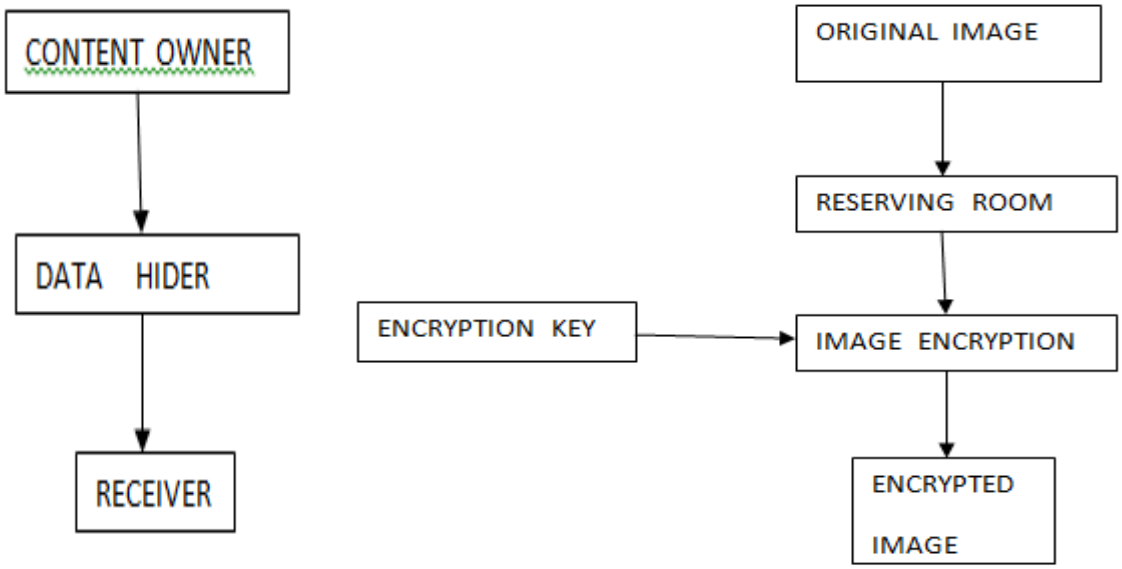
2006, Ni et al. proposed a novel histogram-shifting reversible data hiding technique. In Ni et al.'s method, pixel values are modified one grayscale value at most and thus, a high quality stego image can be achieved.

However, the maximum payload is limited by the peak height of the image histogram; therefore, the payload of their method is relatively low. Hwang et al.also proposed a reversible data hiding method based on histogram-shifting and had better embedding efficiency compared to Ni et al.'s work.

In 2007, Thodi and Rodriguez proposed a very different method by expanding the prediction errors. Because the prediction error is usually smaller than the difference between two consecutive pixel values, the stego image quality obtained by their method is better than that of Tian's method. However, Thodi and Rodriguez's method is also based on expansion-embedding technique, a larger distortion may occur; therefore, their method is not suitable for applications requiring high quality images.

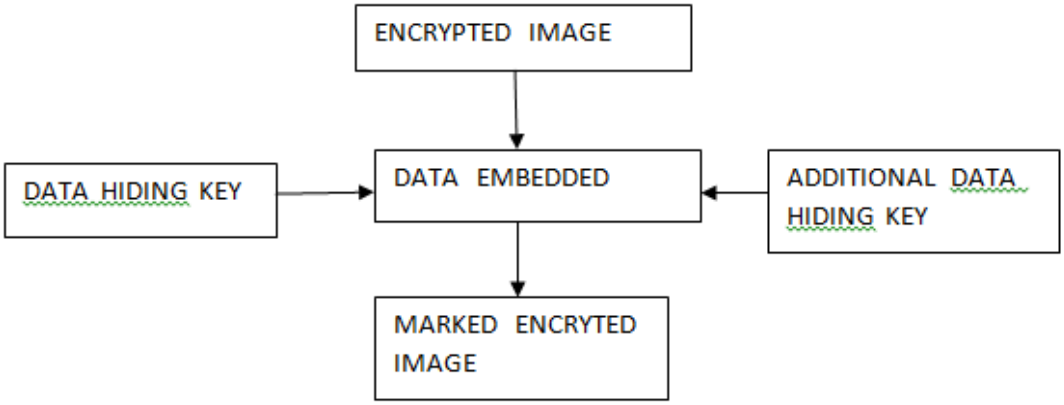
II. PROPOSED METHOD

Vacating room from the encrypted images losslessly is relatively difficult and also sometimes inefficien t.Thus, 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 are much easier which leads us to the framework, "reserving room before encryption (RRBE)".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. The data embeding 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 the Framework VRAE. Where standard RDH algorithms are the ideal operator for reserving room before encryption and can be easily applied to Framework RRBE to provide better performance as 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 encrypts it with respect to protecting the 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.

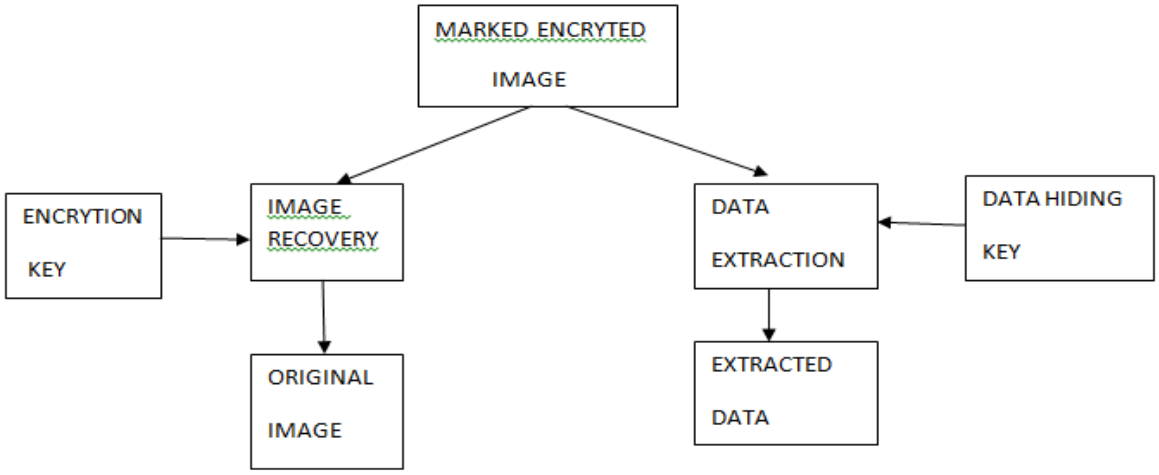


(a)Block diagram for RRBE

(b) block diagram for content owner



(c) Block diagram for data hider



(d) Block Diagram for Reciever

Fig (1). Framework RRBE

A .Encrypted image generation:

To construct the encrypted image, the very first stage is being divided into three steps: image partition, self reversible embedding followed by 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.

1) 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 imageC is an 8 bits gray-scale image with its size M x N and pixels C_{ij}€ [0,255], 1≤ i ≤ M, 1≤ j ≤ 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, everyblock consists of rows, where m = [*l*/N] and the number of blocks can be computed throughn = M – m + 1 . An importantpoint here is that each block is overlapped by pervious and/or sub-sequential blocks along the rows. For each block, define afunction to measure its first-order smoothness

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

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 Fig. 2. It is obvious 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 decreases significantly in terms of PSNR, after embedding the data in the second stage with growing bit-planes exploited. Hence, 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.

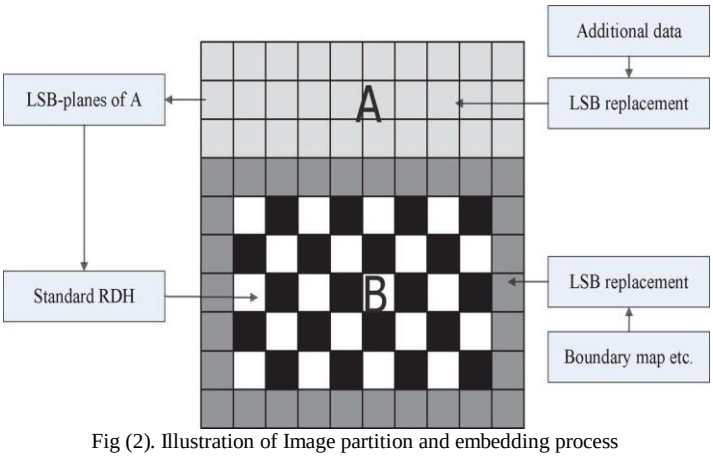


Fig (2). Illustration of Image partition and embedding process

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

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

Where the weight w_i, $1 \le i \le 4$, Then the estimating error is calculated via $e_{ij} = B_{i,j} - B'_{i,j}$ along with embedding some data into the estimating error sequence with histogram shift. Then, we further calculate the estimating errors of black pixels with the help of surrounding white pixels that may have been modified. Then another estimating error sequence is produced that can accommodate messages. Thus we summarize that, to exploit all pixels of B, two estimating error sequences are constructed for embedding messages in every single-layer of embedding process. Using bidirectional histogram shift, some messages can be embedded on each error sequence i.e. firstly we divide the histogram of estimating errors into two parts namely the left part and the right part, and search for the highest point in each part, denoted by LM and RM, respectively. For typical images, LM = -1and RM=0. Further, look for the zero point in each part, denoted by LN and RN. 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 towards right, and then, we can represent the bit 0 with RM and the bit 1with 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. In RDH algorithms, there occurs the overflow and underflow problem when the natural boundary pixels change from 255to 256 or from 0 to . For its avoidance, we just embed data into estimating error with its corresponding pixel that are valued from 1 to254. However, problems still arise when non-boundary pixels are changed from 1 to 0 or from 254 to 255 during the embedding process. These created boundary pixels are defined as pseudo-boundary pixels in the embedding process. Hence, a boundary map is introduced to indicate whether boundary pixels in marked image are natural or pseudo in extracting process.

3) Image Encryption: After the rearranged self-embedded image which is denoted by X is generated, we encrypt X to construct the encrypted image denoted by E .Using stream cipher; the encryption version of X can be easily obtained. For example, agray value $X_{i,j}$ ranging from 0 to 255 can be represented by8 bits, $X_{i,j}(0), X_{i,j}(1), \ldots, X_{i,j}(7)$, such that

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

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

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

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 of A to indicate data hider the total number of rows and the bit-planes he can embed information into. Since after image encryption, none of the data hider and the third party access the content of original image without the encryption key, hence privacy of the content owner is protected.

B. 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 initiates by locating the encrypted version of A , denoted by A_E . Since A_E has been rearranged to the top of E, it is easy for the data hider to read 10 bits information in LSBs of first 10 encrypted pixels. After knowing the number of 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 . Here, the data hider sets a label to point out the end position of embedding process and further encrypts according to the data hiding key to formulate marked encrypted image E'.

C. Data Extraction and Image Recovery:

Extraction of data is completely independent from image decryption and thus its order implies two different practical applications.

1) Case 1: Extracting Data From Encrypted Images: In order to update and manage personal information of images that 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. The order of data extraction before image decryption guarantees the feasibility. The database manager can decrypt the LSB-planes of A_E and extract the additional data m by directly reading the decrypted version once provided with the data hiding key. Since the whole process is operated on encrypted domain, the leakage of original content is avoided.

2) Case 2: Extracting Data From Decrypted Images:

In the previous case both embedding and extraction of the data are manipulated in encrypted domain. While there exists a different situation where the user wants to decrypt the image first and extract the data from the decrypted image when it is needed.

(a) Generating the Marked Decrypted Image: To acquire the marked decrypted image X" which is made up of A" and B", the content owner should do following two steps.

• Step 1. With the encryption key, the content owner decrypts the image except the LSB-planes of A_E. The decrypted version of E' containing the embedded data can be calculated by,

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

And

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

Where $E'_{i,j}(k)$ and $X''_{i,j}(k)$ are the binary bits of $E'_{i,j}$ and $X''_{i,j}$ obtained via (3) respectively.

• Step 2. Extract SR and ER in marginal area of B". Rearranging A" and B" to its original state ,we can obtain the plain image containing embedded data. As the marked decrypted image X" is identical to rearranged 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 exploiting the LSB-planes of A only and the second part can benefit from excellent performance of current RDH techniques.

(b) Data Extraction and Image Restoration: The content owner can further extract the data and recover original image after generating the marked decrypted image. The process is similar to the traditional RDH methods. The following outlines the specific steps:

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

• Step 2. Extract LN, RN, LM, RM, LP, RP, Rb, x and boundary map from the LSB of marginal area of B". Then, scan B" to - undertake the following steps.

• Step 3. If R_b is equal to 0, which means no black pixels participate in embedding process, go to Step5.

• Step 4. Calculate estimating errors $e'_{i,j}$ of the black pixels $B''_{i,j}$. If $B''_{i,j}$ belongs to [1, 254], 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 $B''_{i,j} \in \{ 0, 255 \}$, refer to the corresponding bit *b* in boundary map. If $b = 0$, skip this one, else operate like B"i,j € [1, 254] . Repeat this step until the part of payload R_b is extracted. If extracted bits are LSBs of pixels in marginal area then it restores them immediately.

• Step 5. Calculate estimating errors $e'_{i,j}$ of the white pixels $B''_{i,j}$,and extract embedded bits and recover white pixels in the same manner with Step 4. If extracted bits are LSBs of pixels in marginal area, restore 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, the procedures are exactly identical to that in Case 2. Thus, it is omitted in Case 1 for simplicity.

III. RESULTS

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

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

IV. CONCLUSION

Reversible data hiding in encrypted images is a new topic drawing attention because of the privacy-preserving requirements from cloud data management. In previous methods RDH process was implemented in encrypted images by vacating room after encrypting the image, as opposed to which we proposed by reserving room before encrypting the image. Hence, the data hider can have advantage of the extra space emptied out in previous stage will make the data hiding process much effortless. This 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 gain separate data extraction, real reversibility and greatly improvement on the quality of marked decrypted images.

References

1. Kede Ma,Weiming Zhang, Xeianfeng Zhao,"Reversible data hiding in encrypted images by reserving room before encryption" IEEE Trans. On information forensic and security,VOL.8NO.3 March 2013.
2. 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.
3. 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.
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. 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.
6. L. Luo et al., "Reversible image watermarking using interpolation technique," IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187–193,Mar. 2010.
7. 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.
8. 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.
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. J. Tian, "Reversible data embedding using a difference expansion," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896 Aug. 2003.
11. J. Fridrich and M. Goljan, "Lossless data embedding for all image formats," in Proc. SPIE Proc. Photonics West, Electronic Imaging, Securityand Watermarking of Multimedia Contents, San Jose, CA, USA, Jan. 2002, vol. 4675, pp. 572–583.

