ISSN:0975-9646

Ashwind S et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 1861-1863

Secure Data Transmission Using Reversible Data Hiding

Ashwind S , Ganesh K , Gokul R ,Ranjeeth Kumar C

Department of Information Technology,
Sri Ramakrishna Engineering College, India

Abstract: —Now-a-days, more and more interest is paid to reversible data hiding (RDH) in encrypted images, since it maintains the excellent property that the original image can be losslessly recovered after embedded data is extracted while protecting the image content's privacy. Unlike previous methods, a Novel method is proposed by reserving room before encryption with a traditional RDH algorithm, so that it is easy for the data hider to reversibly embed the secret data in the encrypted image. The proposed method can achieve real reversibility i.e., data extraction and image recovery are free of any error. Our experiments show that this novel method can embed more data for the same image quality as the previous methods, such as for PSNR = 40 dB

Key words: Reversible data hiding, Image encryption, Self-reversible embedding.

1. INTRODUCTION

Image processing is a method to convert an image into digital form and perform some operations on input image to get an enhanced image or to extract some useful information from the given input image. Image processing is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Image processing system includes treating images as two dimensional signals and applying set signal processing methods to them.

Image processing is a rapidly growing technology with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines. Increasing the contrast of small details is the aim of many processing algorithms which all act in the same way they amplify the high frequencies in the image. This is the reason why they are called high-pass filters, and probably the most famous of them is unsharp masking.

The proposed scheme is made up of image encryption, data embedding and data-extraction/image-recovery phases. The content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. Then, the data-hider compresses the least significant bits (LSB) of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data.

At the receiver side, the data embedded in the created space can be easily retrieved from the encrypted image containing additional data according to the data-hiding key. Since the data embedding only affects the LSB, a decryption with the encryption key can result in an image similar to the original version. When using both of the encryption and data-hiding keys, the embedded additional data can be successfully extracted and the original image can be perfectly recovered by exploiting the spatial correlation in natural image.

The first problem occurred here is, when there is homogenous zones, all the blocks in these zones are encrypted in similar manner. The second problem present here is that the block encryption methods are not robust to noise. Due to the large size of blocks, the encryption algorithm per block, symmetric of asymmetric cannot be robust to noise. And data integrity is the third problem. These types of problems can be solved by combining encryption and data hiding hence by using an approach of reversible data hiding for encrypted images. It is able to embed data in encrypted images and then to decrypt image and to rebuild the original image by removing the hidden data.

2. GENERATION OF ENCRYPTED IMAGE

To construct the encrypted image, the first stage can be divided into three steps: image partition, self-reversible embedding followed by image encryption. Before all these the image must be encrypted. For this is purpose we use encryption key. The encryption key is just entered by the user. The same encryption key must be used at the receiver side in order to decrypt the image. Then, a data hiding key is given to hide the secret data into the encrypted image.

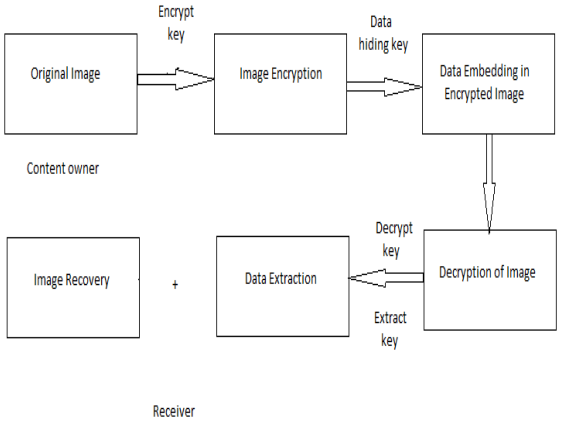


Fig 1 Overview of proposed system

2.1 LSB EMBEDDING ALGORITHM

To construct the encrypted image, the first stage is partition of image, self-reversible embedding and image encryption. At the beginning, image partition step divides original image into two parts and ; 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 , on which standard RDH algorithms that can achieve better performance. To do that, without loss of generality, assume the original image is an 8bit gray-scale image with its size and pixels. 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. In detail, every block consists of rows, where, m and the number of blocks can be computed through n = M – m +1. An important point here is that each block is overlapped by pervious and/or sub sequential blocks along the rows. For each block, define a function to measure its first-order smoothness

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

Equation 1 - To find smoothness

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 with fewer textured areas, as shown in Fig. 2.

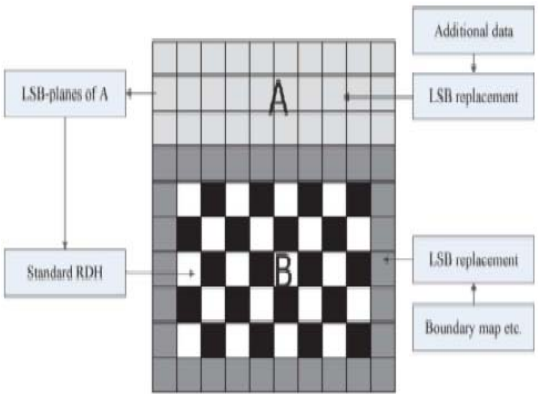


Fig 2. Depiction of image partition and embedding process

The above discussion implicitly relies on the fact that only single LSB-plane of A is recorded. It is straight forward 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. Therefore, 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.

3. Self-Reversible Embedding & Data Hiding

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 to demonstrate the process of self-embedding. Note that this step does not rely on any specific RDH algorithm.

After rearranging self-embedded image, we can encrypt to construct the encrypted image. With a stream cipher, the encryption version is easily obtained. Pixels in the rest of image B are first categorized into two sets: white pixels with its indices i and j satisfying. By bidirectional histogram shift, some messages can be embedded on each error sequence i.e., first divide the histogram of estimating errors into two parts, i.e., 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=-1 and RM=0. Furthermore, search 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 1 with RM+1.

4. Generating Marked Decrypted Image

To form the marked decrypted image which is made up of A and B, the content owner should do following steps.

Step 1: With the encryption key, the content owner decrypts the image except the LSB-planes.

Step 2: By rearranging part 1 and part 2 to its original state, the plain image containing embedded data is obtained.

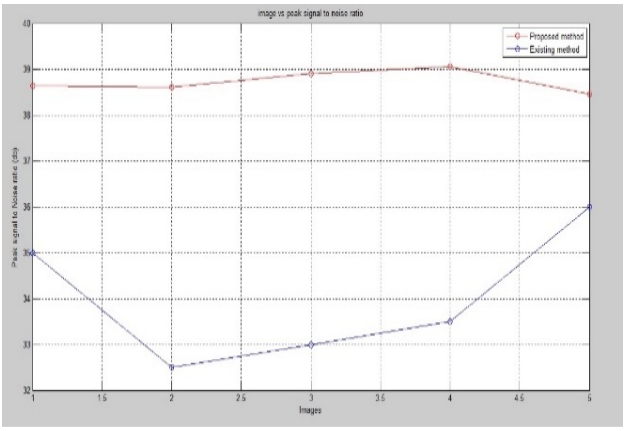
4.1 Data Extraction And Image Restoration:

After generating the marked decrypted image, the content owner can further extract the data and recover original image. The receiver must decrypt the image using the same encryption and extraction keys within the time limit fixed by sender. The process is essentially similar to that of traditional RDH methods

1. Record & decrypt the LSB-planes of A. According to the data hiding key; extract the data until the end label is reached.
2. Calculate estimating errors of the white pixels, and extract embedded bits and recover white pixels in the same manner. In marginal areas, if extracted bits are LSBs of pixels, then restore them quickly.
3. Replace marked LSB-planes of part 1 with its original bits extracted from part 2 to get original cover image.

5. Testing Result

In the earlier methods some error occurs during data extraction and/or image restoration. While in the proposed method is free of any error for all kinds of images. Another advantage of our approach is that the embedding rate is much wider for acceptable PSNRs. And the proposed method can embed more than 10 times as large payloads for the same acceptable PSNR which yields a very good potential for various practical applications.



6. CONCLUSION

The graph value of proposed system output shows that it has less or minimal disturbances meaning that the viewer gets the best and enhanced quality of image and data. An algorithm on Reversible Data Hiding on images and data, not only enhances the data transmission but also data security. Experimental results demonstrate that the data transfer by the proposed algorithm is visually pleasing, secured and natural looking.

REFERENCES

[1] Kede Ma, Weiming Zhang, Xianfeng Zhao, Member, IEEE, Nenghai Yu, and Fenghua Li, "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption", IEEE Transactions on Information Forensics and Security, Vol. 8, No. 3, March 2013.

[2] K. Arun Kumar & S.M. Riyazoddin, "Analysis of Data Hiding Techniques in Encrypted Images", Global Journal of Computer Science and Technology Graphics & Vision, Volume 13 Issue 4 Version 1.0 Year 2013.

[3] Xinpeng Zhang, "Separable Reversible Data Hiding in Encrypted Image**",** IEEE Transactions on Information Forensics and Security, Vol. 7, No. 2, April 2012.

[4] Xinpeng Zhang, "Reversible Data Hiding in Encrypted Image"**,** IEEE Signal Processing Letters, Vol. 18, No. 4, April 2011.

[5] Zhenfei Zhao, Hao Luo, Zhe-Ming Lu, Jeng-Shyang Pan, "Reversible data hiding based on multilevel histogram modification and sequential recovery", Int. J. Electron. Commun. (AEÜ) 65 (2011) 814– 826.

[6] Lixin Luo, Zhenyong Chen, Ming Chen, Xiao Zeng, and Zhang Xiong, "Reversible Image Watermarking Using Interpolation Technique**",** IEEE Transactions on Information Forensics and Security, Vol. 5, No. 1, March 2010

[7] 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, March 2006.

[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.