Secure Reversible Data Hiding in Encrypted Images By Reserving Space In Advance

Besteena K J [1], Philumon Joseph[2]

Abstract:

Reversible data hiding with image encryption is a technique, in which a secret data is embedded inside a confidential image in such a way that, at the receiver side, both the image and the data can be extracted without any loss. All previous methods embed data by reversibly vacating room from the encrypted images, which may be subject to some errors on data extraction and/or image restoration. In this paper, we propose a novel method by reserving room before encryption with a traditional RDH algorithm, and thus it is easy for the data hider to reversibly embed data in the encrypted image. The proposed method can achieve real reversibility, that is, data extraction and image recovery are free of any error. It can be used in situations where both image and data have equal importance. The proposed method has its application in different areas such as medical imagery, military imagery and law forensics, where no distortion of the original cover is allowed.

Keywords:

Reversible data hiding, Image encoding, Image decoding,AES Encryption

[1] *Department of Compute Science,Government Engineering College, Idukki, Mahatma Gandhi University, Kottayam*

[2] *Department of Computer Science, Government Engineering College, Idukki, Mahatma Gandhi University, Kottayam*

Introduction

Communication is one of the most important needs of human beings. For communication purpose, most of the people are using different devices like mobile phones, laptops etc. Most of these devices use certain network to make the communication easier. Device level security can be ensured by using facilities like setting passwords, biometric authentication schemes etc. But while coming to network level security the most important challenge that world faces today is to ensure data security.

In the current trends of the world, the technologies have advanced so much that most of the individuals prefer using the internet as the primary medium to transfer data from one end to another across the world. There are many possible ways to transmit data using the internet: via e-mails, chats, etc. The data transition is made very simple, fast and accurate using the internet. However, one of the main problems with sending data over the internet is the security threat it poses i.e. the personal or confidential data can be stolen or hacked in many ways. Therefore it becomes very important to take data security into consideration, as it is one of the most essential factors that need attention during the process of data transferring.

Data security basically means protection of data from unauthorized users or hackers and providing high security to prevent data modification. This area of data security has gained more attention over the recent period of time due to the massive increase in data transfer rate over the internet. In order to improve the security features in data transfers over the internet, many techniques have been developed like cryptography, steganography and digital watermarking.

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

Literature survey

Resolution progressive compression scheme is proposed by Wei Liu et al. [1] which compress an encrypted image progressively in resolution. Decoder can observe a low resolution version of the image and study local statistics based on it. These statistics are used to decode the next resolution level. The encoder task is to send a down sampled version of the cipher text. At the decoder side, the corresponding low-resolution image is decoded and decrypted, from which a higher-resolution image is obtained by intra-frame prediction. The predicted image, together with the secret encryption key, is used as the side information (SI) in next resolution level to decode. This process is repeated until the whole image is decoded. So this multi-resolution approach makes it possible to have access to part of the spatial source data to generate more reliable spatial and temporal side information. But there is need to increase the efficiency of overall data compression to avoid the loss of any kind of data.

W. Puech et al. [2] proposed an analysis of the local standard deviation of the marked encrypted images in order to remove the embedded data during the decryption step for protection of multimedia based on Encryption and watermarking algorithms. These algorithms rely on the Kirchhoff's principle, details of the algorithm are known, and only the key for data encryption and data decryption should be secret. The first one is when there is homogeneous zones all blocks in these zones are encrypted in the same manner. The second problem is that block encryption methods are not robust to noise. Indeed, because of the large size of the blocks the encryption algorithms per block, symmetric or asymmetric cannot be robust to noise. The last problem we face is data integrity. The combination of data-hiding and encryption can solve these types of problems hence by using this approach a reversible data hiding method for encrypted images is able to embed data in encrypted images and then to decrypt the image and to rebuild the original image by removing the hidden data. But it is not possible to use when high capacity reversible data hiding method for encrypted images.

Separable Reversible Data Hiding in Encrypted Image

This method proposes a novel scheme for separable reversible data hiding in encrypted images. As first step, a content owner encrypts the original uncompressed image using an encryption key. Then, a data-hider can compress the LSBs of the encrypted image using a data-hiding key to create a empty space to embed some additional data. If a receiver with an encrypted image containing additional data has the data-hiding key, he can extract the additional data even he does not know the image content. If the receiver has the encryption key, he can decrypt the received data to obtain an image similar to the original image, but he cannot extract the additional data from it. One who has both the data-hiding key and the encryption key can extract the additional data and recover the original content without any error. For this he need to exploit the spatial correlation in natural image when the amount of additional data is not too large.

Proposed System

Since losslessly vacating room from the encrypted images is relatively difficult and sometimes inefficient, why are all still so obsessed to find novel RDH techniques working directly for encrypted images? If it 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 Figure 3.1, the content owner first reserves enough space on original image and then convert 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.

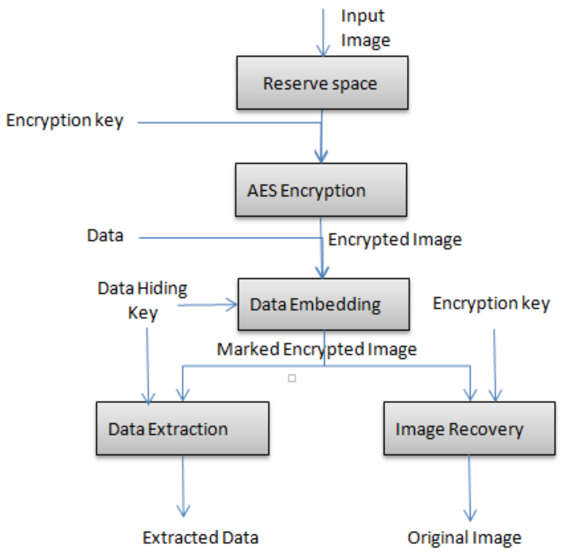


Figure 3.1: framework: reserving room before encryption (RRBE)

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 this new framework 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. The 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 adopt in this proposed method is a traditional RDH approach.

Generation of Encrypted Image

Actually, to construct the encrypted image, the first stage can be divided into three steps: image partition, self reversible embedding followed by image encryption. At the beginning, image partition step divides original image into two parts 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.

Image Partition

This step detects the area, where Reversible Data Hiding algorithms can yield better performance. Hence the main goal of image partition is to find a smoother area B. To do that, without loss of generality, assume the original image C has M*N pixels. First, the content owner extracts several overlapping blocks from the original image, along the rows. Number of overlapping blocks is determined by the size of the message to be embedded. For each block, find the first order smoothness by defining the following function.

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

Higher relates to blocks which contain relatively more complex textures. The content owner, therefore, selects the particular block with the highest to be A , and puts it to the front of the image concatenated by the rest part with fewer textured areas, as shown in Figure 3.2 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 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, need to 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.

Self reversible embedding

The objective of self reversible embedding is to find appropriate space to hide the secret data and it will embed the data in the image using LSB algorithm. The secret message will be embedded in the A part obtained in the previous step. For that the original LSB plane of A is embedded in the B part in the marginal area.

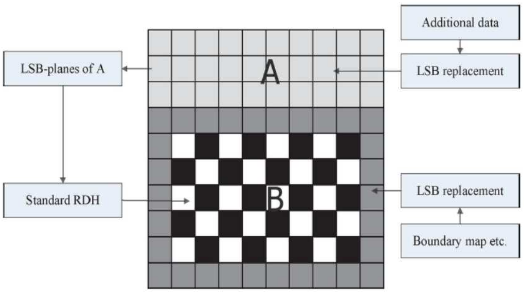


Figure 3.2: Illustration of image partition and embedding process

Pixels in the rest of image B are first categorized into two sets: white pixels with its indices i and j satisfying (I + j) mod 2 = 0 and black pixels whose indices meet (I + j) mod 2 = 1 , as shown in Figure 3.3 Then, each white pixel,Bi;j , is estimated by the interpolation value obtained with the four black pixels surrounding it as follows

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

where the weight wi; 1 ≤ i ≤ 4, is determined by the same method as proposed in [10]. The estimating error is calculated via ei;j = Bi;j – B'i;j and then some data can be embedded into the estimating error sequence with histogram shift, which will be described later. After that, 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 generated which can accommodate messages as well. Furthermore, it can also implement multilayer embedding 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.

By bidirectional histogram shift, some messages can be embedded on each error sequence. That is, 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 RM - 1 with one step toward right, and then, it 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.

Suppose it should implement the embedding scheme x times to accommodate additional data. In the previous x - 1 single-layer embedding rounds, peak points of two error sequences are selected and utilized to embed messages as above mentioned. When it comes to the xth single-layer embedding, only a small portion of messages is left to be embedded, so it is unadvisable to accommodate such little data at the expense of shifting all error values between peak points and their corresponding zero points. To deal with this issue, it can either exploit only part of error sequences which has enough peak points to embed the remaining messages while leaving the rest error sequences unchanged, or find two proper points, denoted by LP and RP, whose sum is larger, however closest to, the size of remaining messages. By shifting error values between LP and RP with their corresponding zero points, messages can be embedded into LP and RP instead of peak points.

Image Encryption using AES

After rearranged self-embedded image, denoted by X, is generated, we can encrypt X to construct the encrypted image, denoted by E. The image X is encrypted by using AES algorithm to enhance the security of the secret image while transmitting it to the receiver side. Advanced Encryption Standards (AES) is an example for symmetric encryption. It takes a block of size 128 bits as input and produces the output block of same size. AES supports different key sizes like 128,192 and 256 bit keys. Each encryption key size will change the number of bits and also the complexity of cipher text. The 128 bit data block is divided into 16 bytes. These bytes are mapped into a 4*4 arrays called state array. AES is an iterative algorithm and each iteration is called a round. For 128 bit keys there will be 10 rounds. T here will be 12 rounds if the key size is 192 and for key size 256, there will be 14 rounds. All the rounds are same except the last round. Each round has substitute bytes, shift rows, mix columns, add round key. In the final round there are no mix columns. The important steps in AES algorithm is illustrated in Figure.3.3

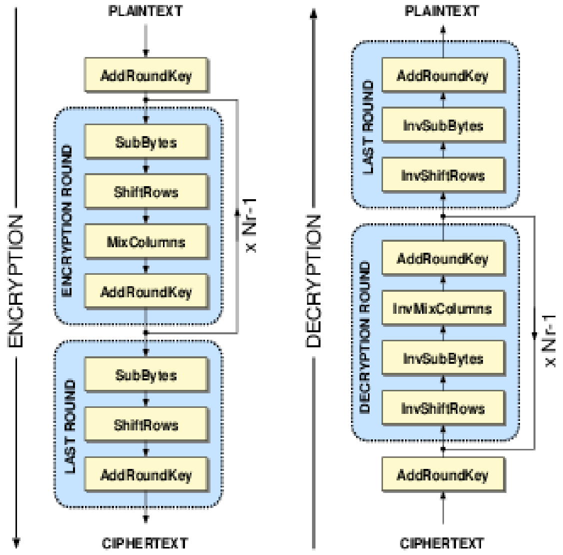


Figure.3.3 High level description of the algorithm:

Data Hiding in Encrypted Image

The data hiding process start with embedding data to encrypted image. This can be done by locating the encrypted version of the A .We denote this version as AE. AE rearranged to the top of E, so data hider can easily read 10 bits information in LSBs of first 10 encrypted pixels. After understanding number of bit-planes and rows of pixels can be modified, data hider use LSB substitution method for hiding additional data 'm' in to available bit planes. As last step, data hider sets a label following 'm' to point out the end position of embedding process. Then encrypts m according to the data hiding key to formulate marked encrypted image denoted by E'.Those who possess the data hiding key could only extract the additional data.

Data Extraction and Image Recovery

Data extraction is independent from image decryption. so we can use them for different practical applications.

Case1: Extracting Data from Encrypted Images

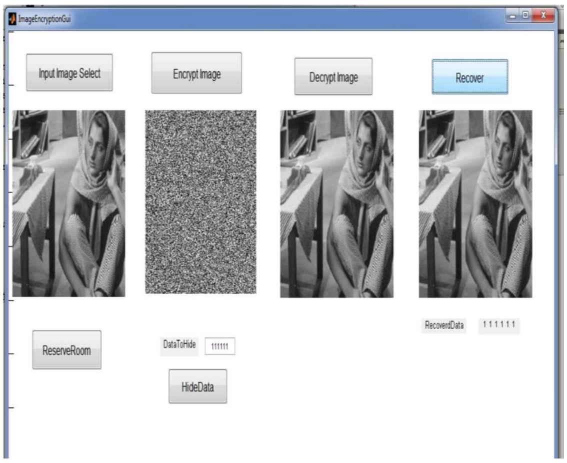
In some case such us updating personal information of images which are encrypted for ensuring privacy, the database manager get access to data hiding key only. He have to update data in encrypted domain. Since proposed system extract data independent of image decryption order, database manager can update information thought LSB substitution by decryption the LSB-planes of A_E and extract the data m by reading the decrypted version. After updating information, he can encrypts updated information according to the data hiding key. So there will not be any leakage of original content, since whole process is entirely carried out in encrypted domain.

Case2: Extracting Data from Decrypted Images

The previous case we explained was both embedding data and extracting data is carried out in encrypted domain. There is a second situation user need to decrypt the image first and extract data from the resulting image. One such example is customers outsourced their encrypted images to one cloud server and cloud server insert some identification mark for each image. Now an authorized user who has been shared the encryption key and the data hiding key, downloaded and decrypted the images. So authorized user hopes to get decrypted images which including the identification mark and 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.

Resuls and Discussion

We take standard image Barbara, shown in Figure. 4.1, to demonstrate the feasibility of proposed method. In Fig. 4.1 second one is the encrypted image containing embedded messages then next one is decrypted version with messages. Last one depicts the recovery version which is identical to original image.



DataToHide 111111

RecoverdData 1 1 1 1 1 1

Figure.4.1 . Result of implementation.

Future Enhancement

In this method of reserving room before encryption in RDH, it is implemented in available gray scale image for analysis purpose. But for military or medical application we need to implement it in real time with color images as cover image. So using RGB channel we can embed large data such as image or file to cover image. In-order to add more security to cover image and data we introduce dividing in to shares method. Also before transmitting the image containing the embedded data, to the receiver, the image is split into shares and each share will be encrypted to enhance the data security. So only authorized persons, who will be having the secret key can decrypt the shares at the receiver side. Also the original image can be recovered only if all the shares are combined together. The authorized receiver can also extract the embedded data from the image and it is then decrypted to get the original version of the secret data. In this way both image and the data can be extracted without any error.

Conclusion

Secure reversible data hiding in encrypted images by reserving room before encryption can enhance the data security while transmitting the secret data and secret image through the networks. Reversible data hiding in encrypted images is a new topic drawing attention because of the privacy-preserving requirements from cloud data management. Since AES algorithm with large key size is using for image encryption, no brute force attack will be there. Moreover it is extremely fast, simple and can be implemented with minimum amount of memory. Application of secure reversible data hiding with image encryption involves banking, medical images, organizations etc where both image and data are confidential.

REFERENCES

- [1] Wei Liu, Wenjun Zeng, Lina Dong, and Qiuming Yao, Efficient “Compression of Encrypted Grayscale Images”, Image Processing, IEEE Transactions Vol: 19, April 2010.

- [2] W. Puech, M. Chaumont and O. Strauss – “A Reversible Data Hiding Method for Encrypted Images”, SPIE Electronic Imaging,Security, Forensics, Steganography and Watermarking of Multimedia Con- tents,San Jose,CA, USA.

- [3] X. Zhang, "Reversible data hiding in encrypted images," IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255258, Apr. 2011.

- [4] 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. 199202, Apr. 2012.

- [5]X. Zhang, "Separable reversible data hiding in encrypted image," IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 826832, Apr. 2012.

- Singh, Akhand Pratap. (2014). Fortified Multimedia Application.*International Journal of Research*. 1 (5), p283-291.