RAME PUBLISHERS
INTERNATIONAL JOURNAL OF COMPUTATIONAL AND ELECTRONICS ASPECTS IN ENGINEERING
ISSN: 2395-4116 (PRINT)
ISSN: 2395-4124 (ONLINE)
A Quarterly Publishing Journal

International Journal of Computational and Electronics Aspects in Engineering
Vol. 1, Issue 1, Dec 2014, Revised in March 2020, pp 5-10.
https://doi.org/10.26706/ijceae.1.1.20141202

Enhancing Reversible Data Hiding Technique in Encrypted Images

Shital B. Tiwaskar[1]
sheetaltiwaskar@gmail.com

Gajendra Singh Chandel[2]
gajendrasingh86@rediffmail.com
Assistant Professor

Department of Computer Science and Engineering,
Shri Satya Sai Institute Of Science & Technology, Sehore (M.P), India

Abstract- **Since several years, the protection of multimedia data is becoming very important. The protection of this multi- media data can be done with encryption or data hiding algorithms. Recently, more and more attention is paid to reversible data hiding (RDH) in encrypted images, since it maintains the excellent property that the original cover can be losslessly recovered after embedded data is extracted while protecting the image content's confidentiality. In this paper, we propose a novel method by reserving memory space before encryption with a traditional RDH technique, and thus it is easy for the data hider to reversibly embed data in the image. The proposed method can achieve real reversibility, that is, data extraction and image recovery are free of any error. A new idea is to apply reversible data hiding algorithms on encrypted images by wishing to remove the embedded data before the image decryption. Recent reversible data hiding methods have been proposed with high capacity, but these methods are not applicable on encrypted images. The image like differential expansion, histogram shift or the combination of both the techniques. This is useful in the way that this method recovers the image with its original quality with improved PSNR ratio.**

Index Terms — Reversible data hiding, Image encoding, Image, decoding, Image compression, Data Encryption, Image Encryption, Privacy Protection, Data Extraction, Histogram shifting

I. Introduction

The amount of digital images has increased rapidly on the Internet. Image security becomes increasingly important for many applications, e.g., confidential transmission, video surveillance, military and medical applications. Two main groups of technologies have been developed for this purpose. The first one is based on content protection through encryption. There are several methods to encrypt binary images or gray level images. In this group, proper decryption of data requires a key. The second group bases the protection on digital watermarking or data hiding, aimed at secretly embedding a message into the data. These two technologies can be used complementary and mutually commutative.

RDH has attracted considerable research interest. With regard to providing confidentiality for images, encryption is an effective and popular means as it converts the original and meaningful content to incomprehensible one. Although few RDH techniques in encrypted images have been published yet, there are some promising applications if RDH [9] can be applied to encrypted images. Then the process of data hiding is done using the separable reversible data hiding. A data-hider may compress the least significant bits of the encrypted image using a data-hiding

Research Paper
First Online on – 30 Dec 2014, Revised on – 30 March 2020

Cite this article – Shital B. Tiwaskar and Gajendra Singh Chandel, "Enhancing Reversible Data Hiding Technique in Encrypted Images", *International Journal of Computational and Electronics Aspects in Engineering*, RAME Publishers, vol. 1, issue 1, pp. 5-10, 2014, Revised in 2020.
https://doi.org/10.26706/ijceae.1.1.20141202

key to create a sparse space to accommodate some additional data. This additional data may include the data, some that at receiver side these contents are restored back in image to get image with original.

To separate the data extraction from image decryption, the idea of compressing encrypted images and the space for data embedding; Compression of encrypted data can be formulated as source coding with side information at the decoder, in which the typical method is to generate the compressed data in lossless manner by exploiting the syndromes of parity-check matrix of channel codes. The method in compressed the encrypted LSBs to memory space 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 quality. If the receiver has both the keys i.e. data-hiding key and the encryption key, he can extract the extra data and recover the original content.

II. Related Work

As when data is embedded into the image then the quality of image get disturbed. So it is expected that after the data extraction the image quality should be maintained just like the original image. But that image contains some distortions. The reversible data hiding in encrypted image is investigated. Most of the work on reversible data hiding focuses on the data embedding/extracting on the plain spatial domain [8]. the rate-distortion bounds of RDH for without memory covers and proposed a recursive code development which, however, does not move towards the bound [1].In this they used the encoding and decoding But, in some applications, 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.

And it is also hopeful that the original content should be recovered without any error after image decryption and message extraction at receiver side. Fig. 1 gives the sketch. A content owner encrypts the original image using an encryption key, and a data-hider can embed additional data into the encrypted image using a data-hiding key though he does not know the original content. Another promising strategy for RDH is histogram shift (HS), in which space is saved for data embedding by shifting the bins of histogram of gray values. In this process the data embedding process is done in three steps as, first the histogram is drawn then the peak point is taken into consideration then whole image is scanned row by row.

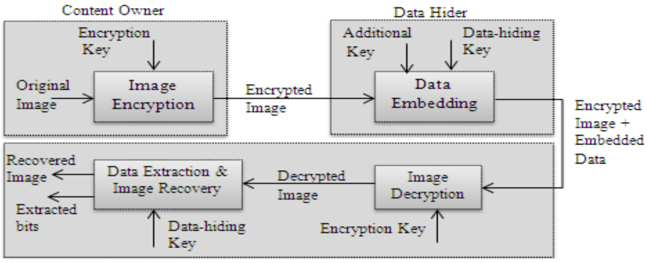

Fig.1 Non-Separable Reversible Data Hiding in Encrypted Image

With an encrypted image containing additional data, a receiver may first decrypt it according to the encryption key, and then extract the embedded data and recover the original image according to the data-hiding key. In the scheme, the data extraction is not separable from the content decryption. In other words, the additional data must be extracted from the decrypted image, so that the principal content of original image is revealed before data extraction, and, if someone has the data-hiding key but not the encryption key, he cannot extract any information from the encrypted image containing additional data. Some reversible data hiding methods uses the concept of differential expansion transform which is based on haar wavelet transform. Another concept used is the histogram shift. Some attempts on RDH in encrypted images have been made.

Zhang divided the encrypted image into numerous blocks. By spinning 3 LSBs of the half of pixels in every block, space can be created for the embedded bit. The data extraction and image recovery proceed by finding which part has been spin in one block. This process can be grasped with the help of spatial correlation in decrypted image.

As shown in Fig. 2(b), the content owner first reserves enough space on original image and then converts the image into its encrypted version with the encryption key.

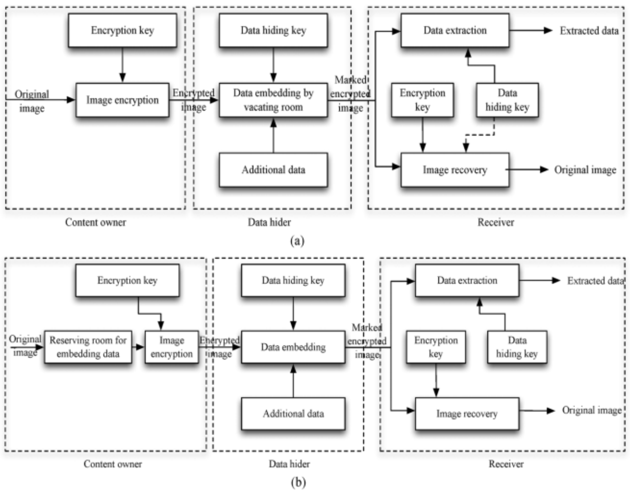

Fig. 2. Framework: "vacating room after encryption (VRAE)" versus framework: "reserving room before encryption (RRBE)." (Dashed line in (a) states that the need of data hiding key in image recovery varies in different practical methods). (a) Framework VRAE. (b) Framework RRBE.

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. 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 and then encrypts it with respect to protecting privacy.

A. Image Encryption

The user will browse the image from computer and encrypt the image and system will auto generate encryption key.

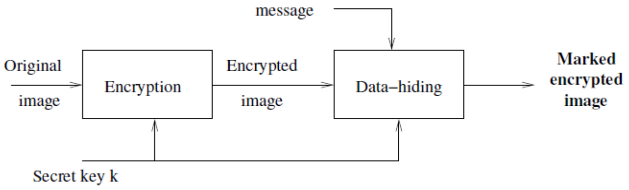

Figure 3. Overview of the encoding method

Encryption is an effective means of privacy protection. To share a secret image with strange person, a content owner may encode the image before broadcast. In some cases, a channel administrator needs to add some extra message, such as the source data, image information 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 extra message at receiver end. That means a reversible data hiding method for encrypted image is advantageous.

B. Data Encryption

The user will browse data that he want send and encrypt the original data and system will auto generate the data encryption key.

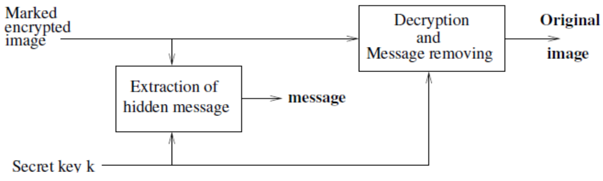

Figure 4. Overview of the decoding method.

The decoding algorithm is also composed of two steps which are the extraction of the message and the decryption removing. The overview of the decoding method is presented in the scheme from Fig. 4. The extraction of the message is very simple: it is just enough to read the bits of the pixels we have marked by using the secret key k and the same PRNG. But after the extraction, each marked cipher-text is still marked. The problem is then to decrypt the marked encrypted image. The decryption removing is done by analyzing the local standard deviation during the decryption of the marked encrypted images.

C. Data Embedding

In the data embedding phase, some parameters are embedded into a small number of encrypted pixels, and the LSB of the other encrypted pixels are compressed to create a space for accommodating the additional data and the original data at the positions occupied by the parameters. Reversible data embedding hides some information in a digital image in such a way that an approved party could decode the hidden information and also restore the image to its original state. The presentation of a reversibledata-embedding algorithm can be measured using following,

- Data embedding capacity limit
- Visual quality
- Complexity

The data without any distortion embedding is the attractive feature 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 pleasing, particularly in military data and medical data. In such a circumstances, every small part of information is important.

The detailed procedure is as follows According to a data hiding key, the data-hider randomly selects Np encrypted pixels that will be used to carry the parameters for data hiding. Here, Np is a small positive integer, for example, Np=20. The other (N-Np) encrypted pixels are permuted and divided into a number of groups, each of which contains L pixels. The permutation way is also determined by the data-hiding key. For each pixel-group, collect the M least significant bits of the L pixels, and denote them as B (k,1) , B (k,2),…., .B(k,M*L) where k is a group index within [1,(N-Np)/L] and M is a positive integer less than 5. The data-hider also generates a matrix G, which is composed of two parts. The left part is the identity matrix and the right part is pseudo-random binary matrix derived from the data-hiding key. For each group, which is product with the G matrix to form a matrix of size (M * L-S). Which has sparse bits of size S, in which the data is embedded and arrange the pixels into the original form and permutated to form a original image.

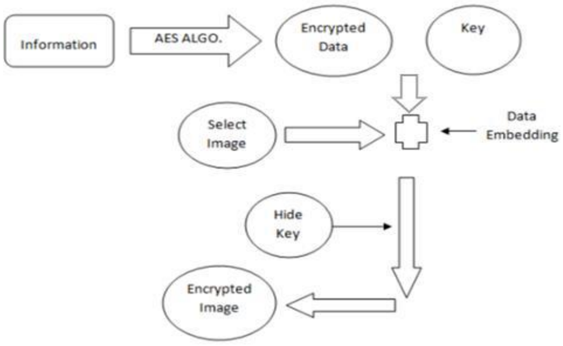

Fig. 5. Data Embedding Process

D. Image compression Techniques

When the data is embedded into the image then the required memory is created into the covering media. But if some additional data is required, it is embedded into image then the process of image compression is done.

When it is desired to transmit repeated data over bandwidth-constrained channel, it is important to first compress the data and then encode it. Mark Johnson investigated the innovation of reversing the order of these steps, i.e., first encoding and then compressing. He showed that in certain scenarios his scheme requires no more arbitrariness in the encryption key than the conservative system where compression precedes encryption.

- *Image Decryption*

After receiving file from user, if receiver have Data hiding and Image Encryption keys then he will only decrypt the image and he will not able to get original data.

- *Data Decryption*

If receiver has data hiding and data encryption keys then he will able to decrypt data. After decryption of data he will get the original data.

- *Data Extraction & Image Recovery*

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

III. AES ALGORITHM

The Advanced Encryption Standard (AES) algorithm consists of a set of processing steps repeated for a number of iterations called rounds. The number of rounds depends on the size of the key and the size of the data block. In our system we are using 128 bit key and in AES this is represented by Nb = 4, which reflects the number of 32-bit words (number of columns) in the State. The number of rounds is 9 for example, if both the block and the key are 128 bits long. Given a sequence {X1,X2, ...,Xn} of bit plaintext blocks, each Xi is encrypted with the same secret key k producing the cipher text blocks {Y1, Y2, ..., Yn}, as described in the scheme from Fig. 6.

To encipher a data block Xi in AES you first perform an AddRoundKey step by XORing a subkey with the block. The incoming data and the key are added together in the first AddRoundKey step. Afterwards, it follows the round operation. Each regular round operation involves four steps. In the SubBytes step, each byte of the block is replaced by its substitute in a substitution box (S-Box). In cryptography, an S-box is a basic component of symmetric key algorithms used to obscure the relationship between the plaintext and the ciphertext. The next one is the ShiftRows step where the rows are cyclically shifted over different offsets. The next step is the MixColumns, where each column is multiplied with a matrix over the Gallois Field, denoted as GF(28). The last step of the round operation is another AddRoundKey. It is a simple XOR with the actual data and the subkey for the current round. Before producing the final ciphered data Yi, the AES performs an extra final routine that is composed of (SubBytes, ShiftRows and AddRoundKey) steps, as shown in Fig. 6.

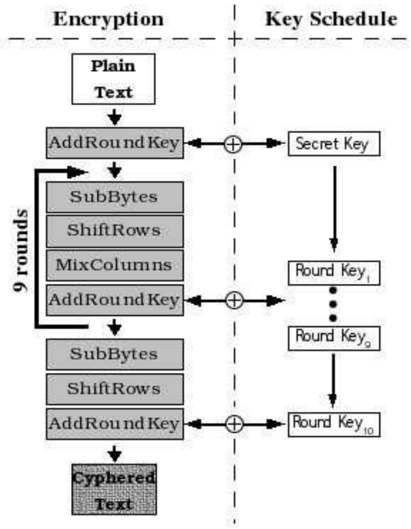

Figure 6. The scheme of the AES algorithm containing 9 rounds of processing steps.

The AES algorithm uses a round function that is composed of four different byte-oriented transformations:

1) Substitution using a substitution table (S-box).

2) Shifting rows of the State array by different offsets

3) Mixing the data within each column of the State array

4) Adding a Round Key to the State.

IV. Experimental Results

We have applied the same process to the image of Leena, Fig. 7.a, by using the AES algorithm in ECB mode to get the encrypted image illustrated in Fig. 7.b. The size of the blocks is also 16 pixels (128 bits). From this encrypted image we have then embedded 16384 bits to get the marked and encrypted image illustrated in Fig. 7.c. The image difference between the Fig. 7.b and c is illustrated in the Fig. 7.d. For the decoding process, after the extraction, if we apply only the decryption on the image of Fig. 7.b, we get the image histogram illustrated in Fig. 8.a. By analyzing the local standard deviation for each block during the decryption step we are able to find the original value of each bit and thus to remove the hidden data and to get the decrypted image.

The SNR Signal to Noise Ratio 5.4760 db PSNR Peak Signal to Noise Ratio 10.7242 db

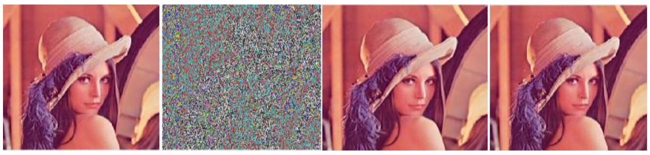

Figure 7. a) Original medical image of 1024 × 1024 pixels, b) Encrypted image with AES in ECB mode, c) Encrypted and marked image with 65536 hidden bits, d) Difference between b) and c).

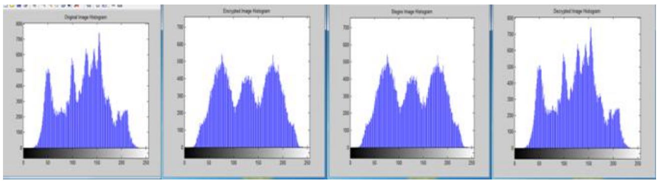

Figure 8. Histogram of: a) The original image histogram, b) The encrypted image histogram c) The stegno image histogram d) The decrypted image histogram

V. Objectives

The lot of work in this research area is done but there are number of problems in existing systems. So the objectives to be recovered in the future may be,

- The extracted data contains the errors as there may be data loss.
- The problem of availability of memory space can occur.
- It is time consuming process.

- The key contents of original image are not restored back, so image quality may hamper.

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 extracted data may contain errors because if there is no availability of sufficient space then some data may lost & that's why there is data missing at the receiver side which may called as data with error. Again the un-availability of memory space is the big problem, as some space is created at the time of data embedding which is the time consuming process. The quality of marked decrypted images is compared in the term of PSNR. Decrypted images under given embedding rates. Out of fairness, we modify the methods in with error-correcting codes to eliminate errors. The proposed method can take advantage of all traditional RDH techniques for plain images and achieve excellent performance without loss of perfect secrecy.

After data extraction the image recovered does not contain the qualities as was the original cover. Some distortions are there into that image.

VI. Conclusion

In conclusion, with our proposed reversible data hiding method for encrypted images we are able to embed data in encrypted images and then to decrypt the image and to rebuild the original image by removing the hidden data. Previous methods implement RDH in encrypted images by memory space after encryption, as opposed to which we proposed by reserving memory space before encryption. Our study helps constructing secure transmission of secrete file preventing any third party access and security level of data is increased by encrypting data. In this paper, we detailed all the steps of the proposed method and we illustrated the method with schemes.

We presented and analyzed various results by showing the plots of the local standard deviations. In the proposed method, the embedding factor is 1 bit for 16 pixels. This small value of the embedding factor is only is to have to choose between two values for each block during the decryption. For the future, we are thinking to improve this method by increasing the payload but also the complexity. Furthermore, this novel method can achieve real reversibility, separate 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 Processing (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 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.

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

[9] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encrypted rayscale images," IEEE Trans. Image Process., vol. 19, no. 4, pp. 1097–1102, Apr. 2010.

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

[11] 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.

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

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