ISSN 2395-1621

Data Hiding in Color Image for Secure Data Transmission with RDHbyRRBE

Komal Khandale [#1], Mrunal Patil [#2], Rohini Tale [#3], Hemavati Tanpure [#4]

[1]Komalkhandale@gmail.com

[2] mrunalpatil5@gmail.com

[3]talerohini@gmail.com

[4] madhuritanpure1993@gmail.com

[#1234] Department of Computer Engineering

NESGOI PUNE.

ABSTRACT

The data hiding plays vital role, the proposed method provides confidentiality and integrity for both the data as well as image. Reversible data hiding (RDH) in encrypted images provides property such that the original image which is used to hide the data can be recovered without loss quality of the image. Errors on the data and image extraction resulted due to previous methods. In this paper the reserving room before encryption with a RDH algorithm results in easiness for the data hider to reversibly embed data in the encrypted images. The real reversibility can be achieved by this RRBE method, that is data extraction and image recovery are without any error. Experiments show that this novel method can embed more than 10 times lager payloads (i.e.) efficiency for the same image quality as the previous methods. It is easier to modify and misuse the valuable information through hacking because of availability of such huge network. Cryptography and steganography are the two ways to sending data securely without modifications by unauthorized person. The proposed method deals with the image steganography also with the different security aspects and about the steganographic algorithms like Least Significant Bit (LSB) algorithm. It also checks those algorithms in means of speed, correctness, accuracy and security. The color image is used to hide data, all previous methods only support for gray scale image.

***Keywords*— Reversible data hiding (RDH), LSB Algorithm, Visual cryptography algorithm, encryption, privacy protection.**

ARTICLE INFO

Article History

Received : 26th January, 2015

Received in revised form :

2nd February, 2015

Accepted : 4th February, 2015

Published online :

11th February 2015

I. INTRODUCTION

Data security basically means protection of data from unauthorized users or hackers and providing high security to prevent data modification. The domain 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 skills have been developed like: Cryptography, Steganography. While cryptography is a method to conceal information by encrypting it to "cipher texts" and transmitting it to the intended receiver using an unknown key, Steganography provides further more security by hiding the cipher text into a seemingly invisible image or other formats.

Reversible data hiding (RDH) is a technique by which we can get the data hidden into image safely as well as the original image quality. And hence we can maintain the quality of original image. In the RDH framework as the name indicates, we are able to embed our data into image reversibly without knowing the various contents of referring image. In previous method, the data extraction process contains the large error-rate. But In this paper, to overcome the above problem an improved method is presented in which the error-rate is decreased and reserves the quality of encrypted image.

The well-known best steganographic method is LSB (Least Significant Bit), which removes the least significant bits of each pixels of selected image into another, that vacant bits are used to hide the information. Within Least Significant Bit Embedding's (LSB) there are various general steganographic techniques that will help to embed data into several digital evidences. The changes are made at only least significant bit of considering pixels so the resulting cover is contrast or similar to original one. And hence is too difficult to understand that the image is having the message also. The advantage of LSB embedding is its simplicity and many techniques use these methods. LSB replacing also allows high level transparency.

In this scheme we are separating the image encryption and data encryption. In the first phase, using an encryption key sender encrypts the original uncompressed image. Then content owner may compress the least significant bits of the encrypted image only. So there will be reserved space to accommodate some additional data.

II. PREVIOUS ARTS

In this framework, at sender side the original image encrypted using a standard cipher with an encryption key. The data hiding process comes in view after producing the encrypted image, the sender can easily embed data in encrypted image with data hiding key. Then at receiver side, the receiver itself maybe the data owner or an authorized third party who can extract the data which is embedded in encrypted image with the data hiding key, later recover the original image from the encrypted version with the help of the encryption key.

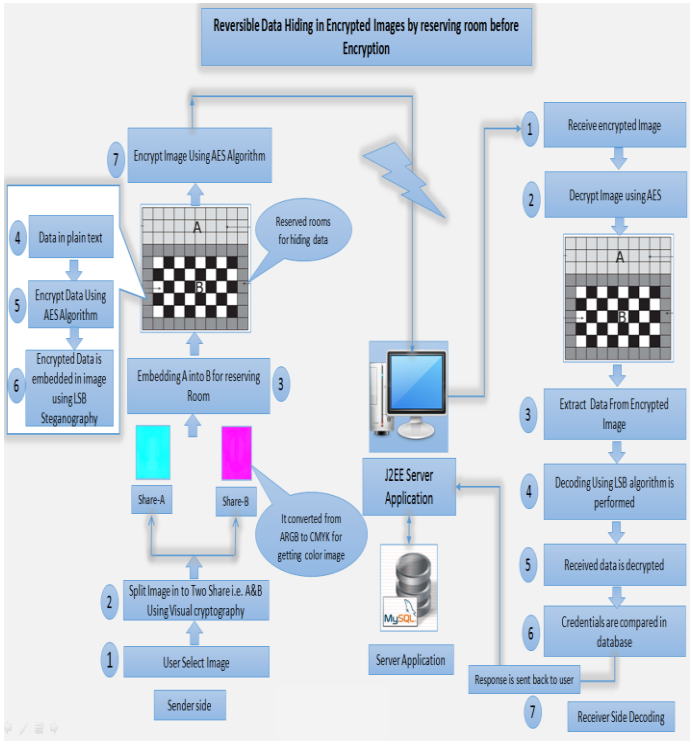
In previous system, image is encrypted using one of the encryption key and then for data hiding or embedding process the "emptying room after encryption" technique is used which vacate the rooms after encryption for data hiding or embedding. A receiver can retrieve the embedded data with the data hiding key and further recover the original image from the encrypted version according to the encryption key. All previous schemes are results in more error-rate on data extraction and image. The data hiding process become effort full because of Emptying Room After Encryption. Also in existing method only gray scale image is used for data hiding which is restriction on user and scope of system. The methods referred in can be described as the method, "Emptying Room After Encryption". In this technique the image is compressed and encrypted by using the encryption key and the data to hide is embedded in to the image by using the data hiding key. At the receiver side he first need to extract the image using the encryption key in order to extract the data and after that he'll use data hiding key to extract the embedded data. It is a sequential process and is not a separable process.

III. PROPOSED METHOD

In the present paper, we are using the "reserve room before encryption" with RDH rather than "emptying room after encryption". In the proposed method, first of all, the room will be emptied out by replacing LSBs of some pixels and adding into other pixels with a basic RDH algorithm and then the image will be encrypted, so the positions of the LSBs which are embedded in another the encrypted image can be used to embed data. The proposed method not only does separate data extraction from image decryption but also achieves excellent performance. Reserving room before image encryption at sender side, the RDH process in encrypted images would be more natural and much easier which leads us to the framework, "reserving room before encryption (RRBE)".

The emptying room after the encryption of image is hard and not efficient, so why we still use it. If the order of encryption technique and emptying room mechanism is reversed, that is, let reserving room done first then image encryption at sender side, then the process of RDH in encrypted images would be very natural and much easier which will results in new framework, "Reserving Room Before Encryption (RRBE)". The sender first reserves much space in original image and then image will be encrypted with the help of encryption key. The scheme involves Encryption of image, data embedding and data extraction/ image-recovery phases. The sender encrypts the original unreduced image using an encryption key to produce an encrypted image. Then, the data-hider reduces only the least significant bits (LSB) of the encrypted image using a data-hiding key and creates a small vacant space to absorb the valuable data over area. Using data hiding key the data stored in vacant space can be extracted from encrypted image. As the data stored only at the LSB which will affect the original image for small proportion, a decryption with the encryption key will be done which produces an image same as the original version.

IV. SYSTEM ARCHITECTURE



. Fig. System Architecture

First users will select one image from given set of images. Then image will be divided into two shares using visual cryptography algorithm to reserve room for data embedding. The data will be encrypted first then embedded. So the content owner is ready to hide their data in that reserved space in image. And hence we separated the process of data encryption and image encryption. It helps to maintain and preserve the original image quality. The original image quality assures the systems reliability. Now image encryption takes place by the AES algorithm and image is ready to transmit. The steps mentioned above will be reversely done at receiver side.

V. SYSTEM IMPLEMENTATION

The RRBE can be accomplished by three stages:

i) Generating encrypted image

ii) Hiding data in encrypted image

iii) Data extraction and image recovery.

i. Generating Encrypted Image

This stage describes the encryption of image to be transmitted. This can be done again by dividing this stage into following sub-steps: image partition, self reversible embedding and last image encryption.

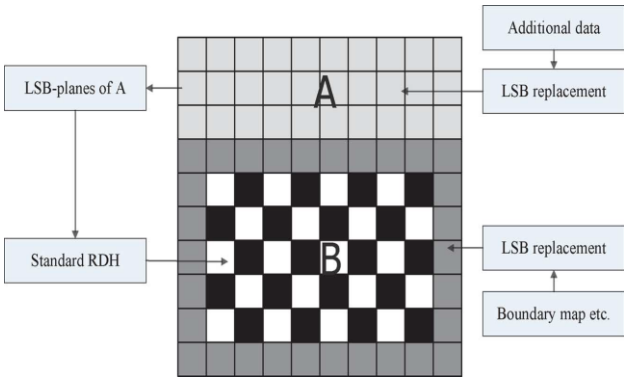


Fig. 2 Image partition and embedding process.

1) Image Partition:

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 A of can be used for storing useful or valuable data; then, encrypt the re-arranged image to generate its final version. Here the visual cryptography algorithm for partition of image is used. For example if 5 users are there means we create five shares. For each share the user can reveal the image but only after five shares he can view the full image. This algorithm not uses the encryption key because if the key is obtained by some unauthorized person then he will reveal the image very easily.

a. In 1st step we convert RGB value of pixel of selected image into CMYK form. To achieve this we form three share images.

b. Share1 will contain pixels with only C value as 1 and other values 0. Share2 will contain pixels with only M value as 1 and other values 0. Share3 will contain pixels with only K value as 1 and other value

Step 2:

a. In this step for each share image we will take 1 cover image.

b. Then share1 is added to cover1 by dividing pixel value by 5(e.g. divide C by 5)

c. Same process is repeated for all share images.

Mathematical Formulae to calculate CMYK value

A. At Sender Side

1. Get color pixel (x, y)
2. Each pixel value is of 32 bit, so separate ARGB values from color

```
A = (color>>24) && 0xff;
R = (color>>16) && 0xff;
G= (color>>8) && 0xff;
B = (color>>0) && 0xff;
```

3. Convert ARGB values into CMYK values

```
R' = R/255;
G' = G/255;
B' = B/255;

K= 1- max(R',G',B');
C = (1-R'-K) / (1-K);
M = (1-G'-K) / (1-K);
Y = (1-B'-K) / (1-K);

K= K*255t;
C= C*255t;
M= M*255t;
Y= Y*255t;

K = K/2t;
C = C/2t;
M = M/2t;
Y = Y/2t;
```

4. Create two splits by CMYK values, Split-1 =>A and split-2=>B

```
A=put pixel (x, y) => (A, R, G, B) => (C, M, 0, 0)
B= put pixel (x, y) => (A, R, G, B) => (Y, K, 0, 0)
```

B. At Receiver Side

1. Get color pixel from image of (x, y) position

```
Int pixA=A.getRGB(i, j);
Int pixB=B.getRGB(I, j);
```

2. Calculate (C,M,Y,K) values from split A and B

```
c= (pixA>>16) && 0xff;
m= (pixA>>16) && 0xff;
```



```
y= (pixB>>16) && 0xff;
k= (pixB>>16) && 0xff;
```

3. Convert CMYK values into ARGB

```
c=2*c;
m=2*m;
y=2*y;
k=2*k/255t;

c=2*c/255;
m=2*m/255;
y=2*y/255;
k=2*k/255;

r=255*(1-c)*(1-k);
g=255*(1-m)*(1-k);

b=255*(1-y)*(1-k);
```

4. Create original image from RGB values

Original image=put pixel (x, y) => (A, r, g, b) => (A, R, G, B)

2) Self-Reversible Embedding:

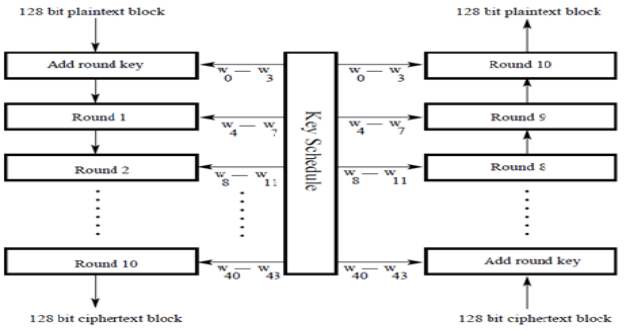
The goal of self-reversible embedding is to embed the LSB-planes of A into B by employing RDH algorithms. Each pixel is divided into two sets S1 and S2 with additional bit. If additional bit is 0 then 3 LSB's of S1 flipped otherwise, 3 LSB's of S2 will flip. To reduce ambiguities of value 0 and 1 the boundary map comes in view. Boundary map is used for checking natural and pseudo boundary pixels.

3) Image Encryption:

After self-reversible embedding, the image encrypted with the help of Advanced Encryption Algorithm (AES).

AES Encryption Algorithm:

AES is Symmetric block cipher. Before any round-based processing for encryption should begin, the input states an array is XORed with the first four words of the key. The similar working is there during decryption except that now we XOR the cipher text state array with the last four words of the key schedule.



For encryption, each round consists of the following four steps:

1) Substitute bytes,

2) Shift rows,

3) Mix columns, and

4) Add round key.

The last step consists of XORing the output of the previous three steps with four words from the key schedule. The key size used for an AES cipher specifies the number of repetitions of transformation rounds that modifies the input, called the plaintext, into the unreadable form, called the cipher text. The numbers of cycles of repetition are as follows:

10 cycles of repetition when 128-bit size key is used.

12 cycles of repetition when 192-bit size key is used.

14 cycles of repetition when 256-bit size key is used.

Each round consists of several processing steps listed above, each containing four similar but different steps, available for encryption key itself. A set of inverse of that rounds are applied to transform cipher text back into the original plaintext using the same encryption key.

ii. Data Hiding in Encrypted Image

Once the sender gets the encrypted image, he can embed useful or valuable data into it, but still he does not get access to the original image. He can modify after knowing how many bit-planes and rows of pixels are there, the sender simply adopts LSB replacement to replace the available bit-planes with additional data.

Least Significant Bit (LSB) based steganography

The simplest and most common type of steganography is LSB (least significant bit). The one of the bit of a byte is used to embed the information. Suppose we want to encode the letter A (ASCII 65 or binary 01000001) in the following 8 bytes of a carrier file.

01011101 11010000 00011100 10101100

11100111 10000111 01101011 11100011

Becomes

0101110**0** 1101000**1** 0001110**0** 1010110**0**

1110011**0** 1000011**0** 0110101**0** 1110001**1**

iii. Data Extraction and Image Recovery

The data extraction is completely independent of image decryption; there are two cases for data extraction and image recovery.

1) Case 1: Retrieving Data From Encrypted Images:

In this case the data stored in image extracted from encrypted image. It depends on situation to consider this case. That is firstly data retrieved then image decrypted. To manage and update personal information of images which are encrypted for protecting clients privacy, authorized third party may only get access to the key for data hiding and have to manipulate data in encrypted domain. Also both embedding and extraction of the data are manipulated in encrypted domain.

2) Case 2: Retrieving Data From Decrypted Images:

In this case the data stored hidden in image retrieved from original image. That is firstly image decrypted then data extracted. On the other hand, 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..

VI. ADVANTAGES AND APPLICATIONS

The system is advantageous in perfect secrecy without leakage of data. This method helps for secure data transfer with confidential data hiding. In this scheme recovery and quality of image as well as data is achieved. Since reserving room before encryption is used the data hiding process becomes easier than previous one. Its purpose is to ensure privacy by keeping the information hidden from anyone for whom it is not intended. The proposed system used in military imagery, medical imagery, banking sectors, government sector. This system is applicable for all applications where there is a requirement of higher level of security.

VII. CONCLUSION

In recent there is more trend toward use of web services due to their functionality and scalability. But the major problem is requirement of preserving the privacy. The proposed system fulfils this requirement of security. The data hiding process and encryption becomes easy and convenient as the space is reserved in advance before the encryption. Steganography is mechanism which helps for private communications. Encryption is the conversion of data into some unreadable form or not understandable by unauthorized person. The scheme can gives benefit of basic RDH framework for all colour images and achieve benchmark with perfect secrecy. Further, this system can provide separate data extraction and greatly preserves the quality of decrypted images. When the receiver has both the data hiding key and encryption key, he is able to extract the hidden data and recover the original information without any error and retrieve selected image with its quality.

VIII. FUTURE SCOPE

The lossless compression method is used for the encrypted image containing hidden data, the useful data can be still extracted and the original content can be also recovered since the lossless compression does not change the content of the encrypted image containing embedded data. However, the lossy compression method compatible with encrypted images generated by pixel permutation is not suitable here since the encryption performed by bit-XOR operation. In the future, a comprehensive combination of image encryption and data hiding compatible with loss compression deserves further research. The implemented a Reversible approach can be enhanced in future by using the following provisions

• And MLSB technique can also be applied after embedding when there is lot of change in the pixel to retain nearest to the original value.

• Applicable in networking domain and the keys are sent and received securely.

REFRENCES

[1] Reversible data hiding in Encrypted Images by reserving room before Encryption, by Kede Ma, Weiming Zhang, Xianfeng Zhao, *Member, IEEE, Nenghai Yu, and Fenghua Li*, IEEE transactions on Information Forensics and Security,vol. 8,no. 3,March 2013.

[2] 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.

[3] 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.

[4] T. Margaret PG Student [Embedded System], Dept of ECE, Sathyabama University, Chennai, Tamil Nadu, India "Reversible data hiding in encrypted images b XOR Ciphering techniques" IJAREEIE vol. 3, issue 2, February 2014.

[5] Sozan Abdulla "New Visual Cryptography Algorithm For Colored Image" *JOURNAL OF COMPUTING*, vol. 2, ISSUE 4, APRIL 2010, ISSN 2151-9617.[6] X. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 826–832, Apr. 2012.

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

[8] 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.

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

[10] L. Luo *et al.*, "Reversible image watermarking using interpolation technique," *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.*

[12] 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.

