International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Secure Text/Image Data Hiding in Images with Efficient Key Management

Anjaly Mohanachandran[1], Mary Linda P.A.[2]

[1, 2]Department of Computer Science and Engineering, KMCT College of Engineering and Technology, Calicut, Kerala, India

Abstract: ***The security plays an important role in transmission of confidential data over internet. So, as a part of improving security in data transmission, we will hide the data inside an encrypted image. Thereby, confidentiality of the image as well as the embedded data is maintained. The embedded data can be extracted without any error, and also the cover image restoration is also free from error. So, here we are implementing a system in which text data can be embedded inside an encrypted image for security and also cover image can be send separately to the destination by hiding it inside another image. It maintains the excellent property that the original image can be losslessly recovered after embedded data is extracted. The major application of this method is medical imagery, military where both data and cover image is confidential. The most improved technique of data hiding is that we can make use of separate keys in case of encryption and data hiding in sender side and receiver side. Thus the concept of efficient key management improves the security.***

Keywords: Data Hiding, Image encryption, image Decryption, Data Embedding, Data Extraction, LSB method.

1. Introduction

Steganography is a method of hiding a secret message inside other information so that the existence of the hidden message is concealed. Cryptography, in contrast, is a method of scrambling hidden information so that unauthorized persons will not be able to recover it. The main advantage steganography has over cryptography is that it hides the actual existence of secret information, making it an unlikely target of spying attacks. To achieve higher security, a combination of steganography with cryptography may be used.

Nowadays transmission of data by embedding it in digital images has widely increased. The security can be improved by sending data in this way. The situation in which both the transmitted data and the cover image is confidential, then we can make use of reversible data hiding technique in encrypted images. While transferring the data from the source to destination, there is a chance of occurring intruder attack and that steals the confidential information. So, this type of transmission is restricted by some applications such as military imagery, law forensic etc.

The encryption can be done using random secret key and decryption can be done using another random secret key at receiver side. So the security is maintained. Likewise the data hiding and extraction can also be done using random secret key.

A new algorithm is presented to hide information in the least significant bits (LSBs) of image pixels. The algorithm uses a variable number of hiding bits for each pixel, where the number of bits is chosen based on the amount of visible degradation they may cause to the pixel compared to its neighbors. The amount of visible degradation is expected to be higher for smooth areas, so the number of hiding bits is chosen to be proportional to the exclusive-or (XOR) of the pixel's neighbors. Analysis showed effectiveness of the algorithm in minimizing degradation while it was sensitive to the smoothness of cover images. By using this algorithm it is possible to hide the image separately inside another image and retrieve the hidden image without any error.

Lots of research has been done in the area of data hiding. In last few years various efficient methods have been proposed for data hiding and image cryptography. Some noticeable work in area of data hiding is as follows:

Xinpeng Zhang presented a scheme in which, a content owner encrypts the original image using an encryption key, and a data-hider embeds additional data into the encrypted image using a data-hiding key yet he does not know the original content. 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 activity of data extraction is not separable from the activity of content decryption. In other words, the additional data must be extracted from the decrypted image, so that the principal content of original image is opened before data extraction, and, if someone has the data-hiding key but not the encryption key, he is not able to extract any information from the encrypted image containing additional data [1].

Z. Ni, Y. Shi, N. Ansari, and S. Wei ,[2] have proposed a system that perform the Reversible Data hiding by using the histogram shift operation for RDH. In this system used the spare space for embedding the data by shifting the bins of gray scale values. The embedding capacity measured by the use of number of pixels in peak point. This system has some benefits such as it is simple and has constant PSNR ratio, capacity is high and distortion is very low. This system has some disadvantages such as more time consuming while searching the image number of times.

J.Tian [3] has proposed a system which uses difference expansion method for embedding data in reversible manner for digital images. Reversible data embedding means lossless embedding. Here quality degradation was very low after embedding the data. This paper describes how to measure the

performance of the system by using the concept of reversible data embedding. This can be measure through various factors such as the payload capacity limit, visual quality and complexity. This system uses the differences between two neighboring pixels. The LSB's of the differences are all zero and this embedded to the message. The benefits of the system are no loss of data while performing compression and decompression. This system is useful for audio and video data. The drawbacks of the system are achieving error because of division by 2 and due to bit replacement visual quality degrade.

Hiding in LSBs of each pixel is desired since their modification will cause less distortion compared to other bits. The number of bits used should be variable and related to the stego image to minimize distortion [12], [13]. However, some applications, such as lossy compression, involve image alteration where some LSBs are lost. In such cases, more significant bits are used by transformation algorithms that utilize the special features of these applications. These techniques generally append coding information to the image with minimal or no change to the original pixels [14], [15].

Generally, the related previous work did not focus on hiding images inside other images. In addition, related image steganography researches make use of same encryption and decryption key at the server side and client side, thereby reduces the security. The new algorithm of this paper handles hiding different images inside other images of various types. And also improves the security.

2. Hiding data in Encrypted Cover Image

A sketch of the proposed scheme is given in Fig. 1. A content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image, and then a data hider embeds additional data into the encrypted image using a data-hiding key though he does not know the original content. With an encrypted image containing additional data, a receiver may firstly decrypt it using the encryption key, and the decrypted version is similar to the original image. According to the data-hiding key, he can further extract the embedded data and recover the original image from the decrypted version.

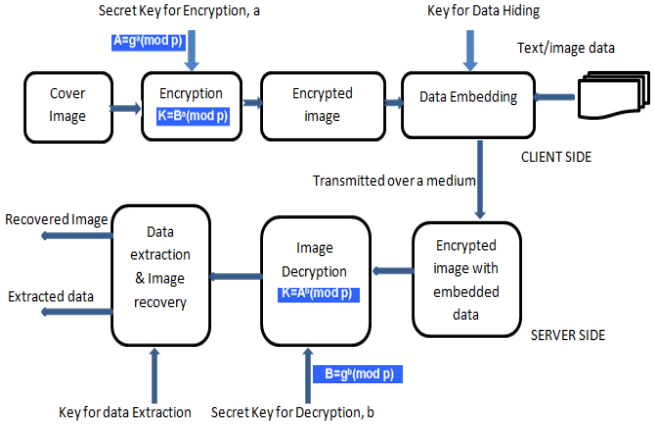

Figure 1: Sketch of proposed scheme

2.1 Image Encryption

Each pixel of uncompressed image with gray value falling into [0, 255] is represented by 8 bits. Denote the bits of a $b_{i,j,0}$, $b_{i,j,1}$..... $b_{i,j,7}$ pixel as where(i,j) indicates the pixel position, and the gray value as $P_{i,j}$. Thus

$$b_{i,j,k} = \left\lfloor \frac{p_{i,j}}{2^k} \right\rfloor \mod 2, \quad k = 0, 1, \ldots, 7$$

and

$$p_{i,j} = \sum_{u=0}^{7} b_{i,j,k} \cdot 2^k.$$

In encryption phase, the exclusive-or results of the original bits and pseudo-random bits are calculated

$$B_{i,j,k} = b_{i,j,k} \oplus r_{i,j,k}$$

Where $r_{i,j,k}$ are determined by an encryption key using a standard stream cipher.Then, $B_{i,j,k}$ are concatenated orderly as the encrypted data. A number of secure stream cipher methods can be used here to ensure that anyone without the encryption key, such as a potential attacker or the data hider, cannot obtain any information about original content from the encrypted data.

2.2 Text data Embedding

Firstly, segments the encrypted image into a number of blocks sized by s*s. Then, each block will be used to carry one additional bit. For each block, divide the pixels into two sets S0 and S1 according to a data-hiding key. If the additional bit to be embedded is 0, flip the 3 least significant bits (LSB) of each encrypted pixel in S0. If the additional bit is 1, flip the 3 encrypted LSB of pixels in S1. The other encrypted data are not changed.

2.3 Text data Extraction

The receiver will extract the embedded bits and recover the original content from the encrypted image. According to the data-hiding key, he may segment the decrypted image into blocks and divide the pixels in each block into two sets in a same way. For each decrypted block, the receiver flips all the three LSB of pixels in S0 to form a new block, and flips all the three LSB of pixels in S1 to form another new block. We denote the two new blocks as H1 and H0. There must be that either H1 or H0 is the original block, and another one is more seriously interfered due to the LSB flip operation. If H0 is the original content of the block then the extracted bit be 0. Otherwise, regard H1 as the original content of this block and extract a bit 1. Finally, concatenate the extracted bits to retrieve the additional message and collect the recovered blocks to form the original image.

2.4 Key Management

The key exchange, used here is somewhat related to Diffie-Hellman key exchange, is a method of digital encryption that uses numbers raised to specific powers to produce decryption keys on the basis of components that are never directly transmitted, making the task of a would-be code breaker

mathematically overwhelming. To implement proposed key exchange, the two end users Alice and Bob, while communicating over a channel they know to be private, mutually agree on positive whole numbers p and q, such that p is a prime number and q is a generator of p. Once Alice and Bob have agreed on p and q in private, they choose positive whole-number personal keys a and b, both less than the prime-number modulus p. Neither user divulges their personal key to anyone; ideally they memorize these numbers and do not write them down or store them anywhere. Next, Alice and Bob compute public keys a* and b*and set it as 1.
From these public keys, a number x can be generated by either user on the basis of their own personal keys. Alice computes x using the formula $x = (b^*)^a \bmod p$ .Bob computes x using the formula $x = (a^*)^a \bmod p$.

The value of x turns out to be the same according to either of the above two formulas. However, the personal keys a and b, which are critical in the calculation of x, have not been transmitted over a public medium. Because it is a large and apparently random number, a potential hacker has almost no chance of correctly guessing x, even with the help of a powerful computer to conduct millions of trials. The two users can therefore, in theory, communicate privately over a public medium with an encryption method of their choice using the decryption key x.

2.5 The Image Hiding Algorithm

This algorithm uses a variable number of LSBs from each pixel of the cover image for hiding. A grayscale image consists of only one color matrix. A Red-Green-Blue (RGB) color image consists of three matrices representing the three colors. The number of bits chosen from each pixel color (red, green, and blue) is different. Images in other color formats may be converted to RGB matrices and converted back after the hiding process is done. The actual number of bits changes according to neighborhood information of each pixel color. When the resemblance between the neighbors of a pixel color entry is low, the pixel entry is located in a non-smooth area where change will not be detected easily. Therefore, the number of bits used for hiding is chosen to be proportional to the neighbors' XOR value for each pixel color entry.

The pixels used in hiding are those located in every line and every other column of the cover image, as in the white squares of a chess board. Pixels on the borders are not used for hiding. This means that approximately 50% of the pixels are used for hiding, while the rest of the pixels are used in determining hiding values and hiding capacity. For RGB images, each color is treated separately. The hiding process starts with the Red matrix, followed by the Green, and then the Blue. The XOR is computed for the value of each one of these pixels' four neighbors: left, right, above, and below. This comparison measures the smoothness of the pixel's neighborhood so that the number of hiding bits can be determined.

The algorithm for hiding in each color matrix is shown in Fig. 2, where *stegoC* is *stegoR*, *stegoG*, or *stegoB*, corresponding to the Red, Green, and Blue matrices of the original stego image, respectively. Each of these matrices has the same (*n***m*) dimensions as the original image. In grayscale images, *stegoC* is the single color matrix. This algorithm takes each color matrix individually, and it goes through every line of the matrix starting with the second line and stopping at the line before the last. It goes through the entries in every other column, taking odd and even numbered columns in odd and even numbered lines, respectively. Left and right border columns are not used for hiding. The XOR of the four neighbors of each examined entry is computed. If the XOR value is less than a given threshold, only one LSB is used for hiding. Otherwise, the number of LSBs (*numLSBs*) used will be the ceiling of one-half of the XOR value. In the implementation of this paper, was set to 9 and the maximum number of LSBs used for hiding in any pixel color was 4. To enhance avoidance of detection for RGB hidden images, avoid grouping all color information of a hidden pixel in a single location in the stego image.

The extraction process searches each of the three color matrices (Red, Green, and Blue), going through all lines and every other column as in the hiding procedure. The number of bits used for hiding in an entry, *stegoC*(*row*, *col*), is also determined by examining *x*; the XOR of the four neighbors as in the hiding process. All extracted hidden values are concatenated and grouped into bytes to form the original secret image.

```
row = 2
while (row ≤ n-2) and (the secret image is not finished)
  col = 2 + (row MOD 2)
  while col ≤ m-2
    x = stegoC(row-1,col) ⊕ stegoC(row+1,col) ⊕ stegoC(row,col-1)
      ⊕ stegoC(row,col+1))
    if x ≤ α
      numLSBs = 1
    else
      numLSBs = ⌈x/2⌉
    endif
    replace LSBs of stegoC(row,col) with the next numLSBs bits
      from the secret image
    col = col + 2
  endwhile
  row = row + 1
endwhile
```

Figure 2: Algorithm for hiding in one color matrix.

2.6 The Image Decryption

When having an encrypted image containing embedded data, a receiver firstly generates $r_{i,j,k}$ according to the encryption key, and calculates the exclusive-or of the received data and $r_{i,j,k}$ to decrypt the image. We denote the decrypted bits as $b'_{i,j,k}$. Clearly, the original five most significant bits (MSB) are retrieved correctly. For a certain pixel, if the embedded bit in the block including the pixel is zero and the pixel belongs to S1, or the embedded bit is 1 and the pixel belongs to S0, the data-hiding does not affect any encrypted bits of the pixel. So, the three decrypted LSB must be same as the original LSB, implying that the decrypted gray value of the pixel is correct. On the other hand, if the embedded bit in the pixel's block is 0 and the pixel belongs to S0, or the embedded bit is 1 and the pixel belongs to S1, the decrypted

LSB

$$\begin{aligned} b'_{i,j,k} &= r_{i,j,k} \oplus B'_{i,j,k} \\ &= r_{i,j,k} \oplus \overline{B_{i,j,k}} \\ &= r_{i,j,k} \oplus \overline{b_{i,j,k} \oplus r_{i,j,k}} \\ &= \overline{b_{i,j,k}}, \qquad k = 0,\ 1,\ 2. \end{aligned}$$

3. Implementation and Analysis

The test image Lena sized 512 512 shown in Fig. 3(a) was used as the original cover in the experiment. After image encryption, the 8 encrypted bits of each pixel are converted into a gray value to generate an encrypted image shown in Fig. 3(b).Then; we embedded 256 bits into the encrypted image by using the side length of each block. The decrypted image is given as Fig. 3(c), and the values of PSNR caused by data embedding are 37.9 dB, which is imperceptible and verifies the theoretical analysis. At last, the embedded data were successfully extracted and the original image was perfectly recovered from the decrypted image.

(a) (b) (c)

Figure 3: (a) Original Lena, (b) its encrypted version, and (c) a decrypted version containing embedded data.

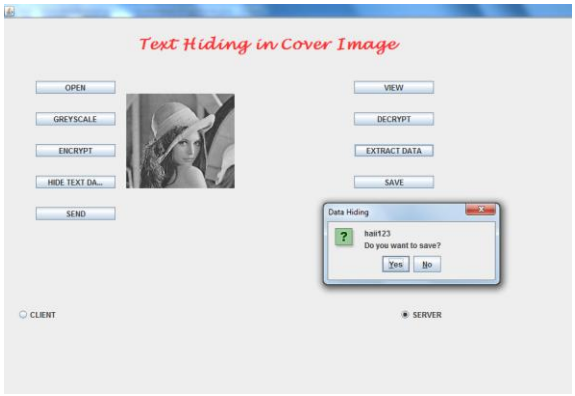

Figure 4: Extracted data

The algorithm was applied using 35 different images of different types and sizes for hiding. The sizes of these secret images ranged from 55*110 to 175*148 pixels. Fig. 5 shows one sample secret image, which is 148*175 pixels, and the cover images. Fig shows the stego images is hiding a copy of the secret image. As seen in the figures, the difference between the original images and the stego images is not visible to the human eye.

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

Figure 5: (a) Secret image, (b) Original Lena, (c)Marked Encrypted image (d) Extracted Secret image.(c) Recovered cover image

The average results for all 15 test images are shown in TABLE I. The average PSNR value was taken for the absolute values of correlation for all images, where the

Table 1: Margin specifications

Cover Image	*PSNR of Image Hiding*	*PSNR of Data Hiding*
Lena	43.13	38.23
Baboon	43.09	38.45
Cameraman	44.55	38.80
Boat	41.57	35.56

original cover image was compared to each of its stego images to obtain the individual correlation values.

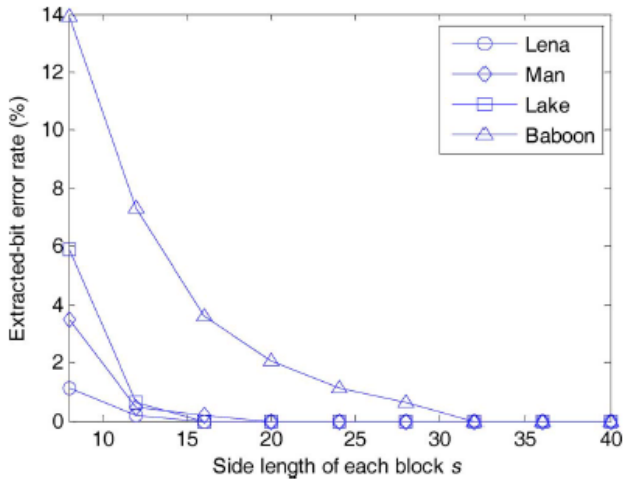

Figure 6: Extracted-bit error rate with respect to block sizes.

Fig. 6 shows the extracted-bit error rate with respect to block sizes when four test images Lena, Man, Lake and Baboon sized 512 512were used as the original covers. These covers are standard test images and freely available in many image databases. Here, the extracted- bit error rate is equivalent to the rate of unsuccessful block recovery. It can be seen that the smoother the cover image, the better is the performance of data extraction and image recovery. When the side length of block is not less than 32, for most cover images, all the embedded bits can be correctly extracted and the original image can be successfully recovered.

4. Conclusion

A novel data hiding scheme for encrypted image with a low computation complexity implemented, which consists of image encryption, data embedding and data- extraction/image recovery phases. The data can be either text or image. The original cover image is encrypted by a stream cipher. Although a data-hider does not know the original content, he can embed additional data into the encrypted image by modifying a part of encrypted data. With an encrypted image containing embedded data, a receiver may firstly decrypt it using the key generated using Algorithm, and the decrypted version is similar to the original image. According to the data-hiding key, the embedded data can be correctly extracted while the original image can be perfectly recovered. Although someone with the knowledge of secure encryption key can obtain a decrypted image and detect the presence of hidden data using LSB-steganalytic methods, if he does not know the data-hiding key, it is still impossible to extract the additional data and recover the original image. Thus the security of this system is improved.

5. Future Scope

This system 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. Inorder 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.

References

[1] X. Zhang, "Reversible data hiding in encrypted images," IEEE Signal Process. Lett.,vol. 18, no. 4, pp. 255258, Apr. 2011.
[2] Z. Ni, Y. Shi, N. Ansari, and S. Wei, Reversible data hiding, IEEE Trans.Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354362, Mar.2006.
[3] J. Tian, Reversible data embedding using a di erence expansion Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890 2003.
[4] D.M. Thodi and J. J. Rodriguez, Expansion embedding techniques for reversible watermarking, IEEE Trans.Image Process., vol.16,no. 3, pp. 721730, Mar. 2007.
[5] W. Zhang, B. Chen, and N. Yu, Improving various reversible data hiding schemes via optimal codes for binary covers ,IEEE Transactions,vol. 21, no. 6, pp. 29913003, June.2012.
[6] Wei Liu, Wenjun Zeng, Lina Dong, and Qiuming Yao Ecient Compression of Encrypted Grayscale Images, Image Processing, IEEE Transactions Vol: 19, April 2010,pp. 1097 1102.
[7] L. Luo et al., Reversible image watermarking using interpolation, IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187G. Xuan,J. Chen, J.Zhu, Y.Q. Shi, Z. Ni, and W. Su, Lossless Data hiding.
[8] X. L. Li, B. Yang, and T. Y. Zeng, on adaptive prediction-error expansion and pixel selection Image Process., vol. 20, no. 12, pp. 3524.44 Secure Data Hiding Main Project Report 2015
[9] 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.
[10] M. Johnson, P. Ishwar, V.M. Prabhakaran, D. Schonbergand K. Ramchandran, On compressing encrypted data, IEEE Trans. Signal Process., vol. 52, no. 10, pp. 2992-3006, Oct. 2004.
[11] Kede Ma, Wei. Zhang, Xianfeng Zhao, Reversible data Hiding in Encrypted Images by reserving Room before encryption,IEEE trans. On information forensics and security, vol,8 No.3 , march 2013.
[12] S. Janakiraman, R. Amirtharajan, K. Thenmozhi and J. Rayappan, "Pixel forefinger for gray in color: A layer by layer stego," Inf. Technol. J., vol. 11, no. 1, pp. 9-19, 2012.
[13] A. Pradhan, D. Sharma and G. Swain, "Variable rate steganography in digital images using two, three and four neighbor pixels," Indian J. Comput. Sci. & Eng., pp. 457-463, 2012.

[14] M. Al-Husainy, "A new image steganography based on decimal-digits representation," Comput. & Inf. Sci., vol. 4, no. 6, pp. 38-47, 2011.

[15] O. Zanganeh and S. Ibrahim, "Adaptive image steganography based on optimal embedding and robust against Chi-square attack," Inf. Technol. J., vol. 10, no. 7, pp. 1285-1294, 2011.

Author Profile

Anjaly Mohanachandran is P.G. student, Department of Computer Science and Engineering, KMCT College of Engineering, Calicut University. She obtained her B.Tech degree in Computer Science & Engineering from Indira Gandhi Institute of Engineering & Technology in 2013. She is currently pursuing the M.Tech .degree in Computer Engineering from Calicut University.

Mary Linda P.A. is Assistant Professor, Department of Computer Science and Engineering, KMCT College of Engineering, Calicut University. Her research focuses on Image processing, Internet security. She obtained her B.Tech degree in Computer Science & Engineering from KMCT College of Engineering in 2007. And she completed her M.Tech degree in Image processing from Model Engineering college, CUSAT in 2012.