Reversible Encrypted Data Concealment in Encrypted Images by Reserving Room for Data Protection

Bhagyashri S. Jatte [1], Asst. Prof. A.S. Deshpande[2]
Electronics & comm. Department,
Savitribai Phule Pune University

***Abstract* - This paper proposes a novel scheme for separate reversible data hiding in encrypted images. In the first phase, a content owner encrypts using an encryption key the original image. Then, a data-hider may compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate some additional data. With an encrypted image containing additional data, if a receiver has the data-hiding key, he can extract the additional data though 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 one, but cannot extract the additional data. If the receiver has both the data-hiding and the encryption key, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image when the amount of additional data is not too large**

***Keywords* - Image Encryption, lifting wavelet transform, Data Hiding, Chaos Encryption Technique, LSB method, etc.**

I. INTRODUCTION

Data hiding [1] is referred to as a process to hide data in cover media. That is, the data hiding process links two sets of data, a set of the embedded data and another set of the cover media data. For instance, in covert communications, the hidden data may often be irrelevant to the cover media. In authentication, however, the embedded data are closely related to the cover media. In these two types of applications, invisibility of hidden data is an important requirement. We tend to review sample cases where watermarking has been deployed. Although cryptography achieves positive security effects, they produce the messages unreadable and meaningless. The knowledge concealment technique uses the accommodative LSB replacement rule for concealing secrete message bits into the encrypted image.

In most cases of data hiding, the cover media will experience some distortion due to data hiding and cannot be inverted back to the original media. That is, some permanent distortion has occurred to the cover media even after the hidden data have been extracted out. In some application, such as medical diagnosis, it is critical to reverse the marked media back to the original cover media after the hidden data are retrieved for the cover media can be losslessly recovered after the hidden data have been extracted out, thus providing an additional avenue of handling two different sets of data. It is the art of hiding the existence of data in another transmission medium to achieve secret communication.

The data hiding technique uses the adaptive LSB replacement algorithm for concealing the secret message bits into the encrypted image. In the data extraction module, the secret data will be extracted by using relevant key for choosing the encrypted pixels to extract the data. By using the decryption keys, the image and extracted text data will be extracted from encryption to get the original information. Since losslessly vacating room from the encrypted images is relatively difficult and sometimes in efficient, why are we still so obsessed to find novel RDH techniques working directly for encrypted images? If we 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)." Finally the performance of this proposal in encryption and data hiding will be analyzed based on image and data recovery. Data Protection system for secret data transmission based on, Reversible encrypted data concealment in encrypted images using chaos encryption, Asymmetric key encryption and adaptive least significant bit replacement technique. Methodologies used are Lifting Wavelet Transformer, Chaos based image encryption. Asymmetric key algorithm based text encryption, Adaptive LSB Replacement, Data Recovery by decryption, Parameter Analysis (MSE, PSNR, Correlation, Elapsed time. Obviously, most of the existing data hiding techniques are not reversible.

II. PAGE LAYOUT

We calculate the differences of neighboring pixel values, and select some difference values for the difference expansion (DE). The original content restoration information, a message authentication code, and additional data (which could be any data, such as date/time information, auxiliary data, etc.) will all be embedded into the difference values. In this paper we will consider grayscale images only. For color images, there are several options. One can decorrelate the dependence among different color components by a reversible color conversion transform [2], and then reversibly embed the data in the decorrelated components. Or one can reversibly embed each color component individually.

Obviously, most of the existing data hiding techniques are not reversible. For instance, the widely utilized spread-spectrum based data hiding methods (e.g., [2]–[5]) are not invertible owing to truncation (for the purpose to prevent over/underflow) error and round-off error. The well-known least significant bit plane (LSB) based schemes (e.g., [6] and [7]) are not lossless owing to bit replacement without "memory." Another category of data hiding techniques, quantization-index-modulation (QIM) based

schemes (e.g., [8] and [9]), are not distortion-free owing to quantization error.
Recently, some reversible marking techniques have been reported in the literature. The first method [10] is carried out in the spatial domain.

1. Chaotic System Properties

Chaotic systems are very suitable for data message encryption because they have several good properties, for example, (a) chaotic motion is neither periodic nor convergent, and the domain is limited. With time passing, the points of the movement trace traverse all over domain, namely the ergodicity of the chaotic orbit; (b) the flexing and collapsing are carried continually through the limited domain. Therefore the outputs of chaotic systems are very irregular, similar to the random noise; (c) because chaotic systems are extremely sensitive to their initial conditions, the movement of any two closed points can be separated in an exponent rule. The long-term movement trace of systems cannot be forecasted. These dynamics characteristics cause chaotic sequences to be wideband, pseudo-random, and unmasked hardly. Different chaotic sequences can be produced with the different initial values of the systems. Therefore, the encrypting space is very wide. Because chaotic sequences are easy to control and easy to regenerate, the possibility for encryption and decryption has been provided [4]. A relationship between chaotic systems and cryptosystems is given by Fridrich [3].

2. Discrete Chaotic Encryption

The Discrete Chaotic Encryption [1] proposed is based on Chebyshev chaotic sequences. Chebyshev mapping is a simple mapping, and the n rank Chebyshev mapping can be represented as:
$T\,n(x) = \cos(n \arccos(x))$
It can be easily deduced
$T\,n+1(x) = 2xT\,n(x) - T\,n\text{-}1(x)$ Equation. (1)
For n∈ N, n ≥ 2, and x ∈ [-1, 1], every $T\,n(x)$ is chaotic [5,6].
The discrete sequences of the chaotic dynamical system are gained by the following equation
$x_{k+1} = T\,n(x_k)$ Equation. (2)
For n = 5, from Eq. (1) & Eq. (2), the following relationship is established
$x_{k+1} = T\,5(x_k) = 16x_k^5 - 20\,x_k^3 + 5x_k$ Equation. (3)
Where k = 0, 1, 2...
Choosing any initial value x0 in [-1, 1], a discrete Chebyshev chaotic sequence with any length {x1, x2,……, xM} can be generated using Equation. (3).

2.1 Encryption Process

If a digital image A of size M x N pixels needs to be encrypted, a corresponding discrete Chebyshev chaotic encrypting algorithm of digital images is expressed as follows [1]:

1. Arbitrarily select two values $x0$ and y0 in the interval [-1, 1].
2. Generate two sufficiently long Chebyshev chaotic sequence using Eq. (3) and $x0$ and y0 as the initial conditions. The lengths of the two sequences should be much larger than M and N respectively.
3. Arbitrarily intercept two sequences of length M and N from the above two sequences respectively i.e. {x1,x2,……, xM} & {y1,y2,……, yN}
4. Rearrange the two sequences obtained in step 3 either in ascending or descending order to get two new discrete chaotic sequences of length M and N respectively i.e. {x'1,x'2,……, x'M} & {y'1,y'2,……, y'N}
5. Decide the position of each $xi \in \{x1, x2, \ldots\ldots, xM\}$ in sequence $\{x'1, x'2, \ldots\ldots, x'M\}$ and generate replacement address set S1={$a1$, $a2$, … , aM }
6. Similarly decide the position of each $y \in$ {y1,y2,……, yN} in the sequence {y'1,y'2,……, y'N} to generate replacement address set S2= {$b1$, $b2$, … , bN }
7. The address set S1 is used in row scrambling of the pixels in the digital image. Similarly, the second sequence S2 is used to scramble the columns of the digital image.

The decrypting process of the image is the inverse process of the encryption.

3. Logistic Map Encryption

The basic Logistic-map is formulated as Equation. (4)
Where, x (0, 1). The parameter μ and the initial value $x0$ can be adopted as the system key (μ, $x0$). The research result shows that the system is in chaos on condition that 3.569 < μ < 4.0 [7].
The encryption scheme is composed of two chaotic systems. One creates a binary stream and the other creates a permutation matrix P. First, the pixel values of the plain image are modified randomly using the binary stream by the traditional stream ciphers technology, namely bit-wise XOR operation. Then the modified image is encrypted again by matrix P.

3.1 Encryption Process

Consider the plain image to be represented by A, and A (i, j) stands for an individual pixel in the image. The system key is denoted as k = (k1; k2), where k1 = (μ1, x01) and k2 = (μ2, x02) are the initial conditions of the two chaotic systems respectively. The encrypting process consists of following five steps [2]:

1. Generate a chaotic sequence using the sub-key k1 as the initial conditions of the first chaotic system.
2. Transform the chaotic sequence into a binary stream by a threshold function.

3. Modify pixel values of the plain image A (i, j) using the binary stream as a key stream and get the image A' (i, j). The operation is bit-wise XOR.
4. Construct a permutation matrix P using the sub-key k2 as the initial conditions of the second chaotic system.
5. Encrypt the image A' (i, j) by permutation matrix P and get the encrypted image A" (i, j).

The encryption process is shown in Figure 1. The decrypting process is the reverse process of encrypting.

III. PROPOSED SCHEME

The proposed scheme combines the benefits provided by both systems mentioned above. As will be clear in the results, the first system based on Chebyshev chaotic sequence, is relatively simple and hence the time taken for the encryption process is very less. The proposed system as in the second system based on Logistic Map has two encryption stages. The first encryption stage uses a chaotic system that is based on the Logistic Map, but unlike the second system, the second encryption stage in the proposed system uses a discrete chaotic sequence based on Equation. (3).

A. ENCRYPTION ALGORITHM

The key is denoted as k = (k1; k2), where k1 = (μ, x01) and k2 = (x02, y02). The parameter μ is selected such that $3.569 < \mu < 4.0$ and $x01 \in (0, 1)$. The initial conditions x02 and y02 for the chaotic system of the second stage lie in [-1, 1]. Consider the plain image to be represented by A of size M x N, and A (i, j) stands for an individual pixel in the image. The encrypting process consists of following steps:

1. Generate a chaotic sequence using the sub-key k1 as the initial conditions of the first chaotic system.
2. Transform the chaotic sequence into a binary stream by a threshold function.
3. Modify pixel values of the plain image A (i, j) using the binary stream as a key stream and get the image A' (i, j). The operation is bit-wise XOR.
4. Generate two sufficiently long Chebyshev chaotic sequence using Equation. (3) and *x*02 and y02 as the initial conditions. The lengths of the two sequences should be much larger than M and N respectively.
5. Arbitrarily intercept two sequences of length M and N from the above two sequences respectively.
6. Rearrange the two sequences obtained in step 5 either in ascending or descending order to get two new discrete chaotic sequences of length M and N respectively.
7. Decide the position of each $xi \in \{x1, x2, \dots\dots, xM\}$ in sequence $\{x'1, x'2, \dots\dots, x'M\}$ and generate replacement address set S1 = {*a*1, *a*2, … , *aM* }
8. Similarly decide the position of each *yi* ∈ {y1,y2,……, yN} in the sequence {y'1,y'2,……, y'N} to generate replacement address set S2= {*b*1, *b*2, … , *bN* }
9. The address set S1 is used in row scrambling of the pixels in the image encrypted in stage I. Similarly, the second sequence S2 is used to scramble the columns of the image encrypted in stage I.

4.1 Generation of Binary Stream

The process for generating the binary stream mentioned in step 2 of section 5.1 above is as follows:

1. Generate a chaotic sequence using the sub-key k1 as the initial conditions of the first chaotic system. i.e.
2. Generate a binary stream from the above chaotic system *xi* by using a threshold function *F*. The threshold function *F* is as given below:

The proposed scheme combines the benefits provided by both systems mentioned above. As will be clear in the results, the first system based on Chebyshev chaotic sequence, is relatively simple and hence the time taken for the encryption process is very less. The proposed system as in the second system based on Logistic Map has two encryption stages. The first encryption stage uses a chaotic system that is based on the Logistic Map, but unlike the second system, the second encryption stage in the proposed system uses a discrete chaotic sequence based on Equation. (3).

Encryption Algorithms

The key is denoted as k = (k1; k2), where k1 = (μ, x01) and k2 = (x02, y02). The parameter μ is selected such that $3.569 < \mu < 4.0$ and $x01 \in (0, 1)$. The initial conditions x02 and y02 for the chaotic system of the second stage lie in [-1, 1]. Consider the plain image to be represented by A of size M x N, and A (i, j) stands for an individual pixel in the image. The encrypting process consists of following steps:

1. Generate a chaotic sequence using the sub-key k1 as the initial conditions of the first chaotic system.
2. Transform the chaotic sequence into a binary stream by a threshold function.
3. Modify pixel values of the plain image A (i, j) using the binary stream as a key stream and get the image A' (i, j). The operation is bit-wise XOR.
4. Generate two sufficiently long Chebyshev chaotic sequence using Equation. (3) and *x*02 and y02 as the initial conditions. The lengths of the two sequences should be much larger than M and N respectively.
5. Arbitrarily intercept two sequences of length M and N from the above two sequences respectively.
6. Rearrange the two sequences obtained in step 5 either in ascending or descending order to get two new discrete chaotic sequences of length M and N respectively.
7. Decide the position of each $xi \in \{x1, x2, \dots\dots, xM\}$ in sequence $\{x'1, x'2, \dots\dots, x'M\}$ and generate replacement address set S1 = {*a*1, *a*2, … , *aM* }
8. Similarly decide the position of each *yi* ∈ {y1,y2,……, yN} in the sequence {y'1,y'2,……, y'N} to generate replacement address set S2= {*b*1, *b*2, … , *bN* }

9. The address set S1 is used in row scrambling of the pixels in the image encrypted in stage I. Similarly, the second sequence S2 is used to scramble the columns of the image encrypted in stage I.

Generation of Binary Stream

The process for generating the binary stream mentioned in step 2 of section 5.1 above is as follows:

1. Generate a chaotic sequence using the sub-key k1 as the initial conditions of the first chaotic system. i.e.
2. Generate a binary stream from the above chaotic system *xi* by using a threshold function *F*. The threshold function *F* is as given below:

IV. CONCLUSIONS

This paper presents the recent research work in the field of steganography deployed in spatial, transform, and compression domains of digital images. In this paper, we have presented a simple and efficient reversible date embedding method for digital images. We explored the redundancy in the digital content to achieve reversibility. Both the payload capacity limit and the visual quality of embedded images are among the best in the literature. Here, the results of different kinds of images with different typical histogram distribution are presented. In all experiments, two pairs of maximum and minimum points are used in data embedding and extraction. Transform domain techniques make changes in the frequency coefficients instead of manipulating the image pixels directly, thus distortion is kept at minimum level and that's why they are preferred over spatial domain techniques. But when it comes to embedding capacity, spatial domain techniques give better results. According to the data hiding key, with the aid of spatial correlation in natural image, the embedded data can be correctly extracted while the original image can be perfectly recovered. Although someone with the knowledge of encryption key can obtained encrypted 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. For ensuring the correct data-extraction and the perfect image recovery, we may let the block side length be a big value, such as 32, or introduce error correction mechanism before data hiding to protect the additional data with a cost of payload reduction. However, there exists a trade-off between the image quality and the embedding capacity. Hiding more data will result directly into more distortion of the image. In recent years, some researchers have concentrated on embedding secret data into the compression codes of images. Such need arises keeping in mind the bandwidth requirements. For instance terrorists may use this technique for their secret secure communication or antivirus systems can be fooled if viruses are transmitted in this way. However, it is proved that steganography has numerous useful applications and will remain the point of attraction for researchers.

V. 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 Trans. on information forensics and security*, vol. 8, no. 3, march 2013 553.

[2]Zhang Dinghui, GU Qiujie, Pan Yonghua and Zhang Xinghua.2008 "*Discrete Chaotic Encryption and Decryption of Digital Images*." 2008 International Conference on Computer Science and Software Engineering.

[3] J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[4] A. Z. Tirkel, C. F. Osborne, and R. G. Van Schyndel, "Image watermarking-a spread spectrum application," in *Proc. IEEE 4th Int. Symp. Spread Spectrum Techn. Applicat.* vol. 2, Sep. 1996, pp. 785–789.

[5] J. Huang and Y. Q. Shi, "An adaptive image watermarking scheme based on visual masking," *Electron. Lett.*, vol. 34, no. 8, pp. 748–750, 1998.

[6] J. Huang, Y. Q. Shi, and Y. Shi, "Embedding image watermarks in DC component," *IEEE Trans. Circuits Syst.: Video Technol.*, vol. 10, no. 6, pp. 974–979, Sep. 2000.

[7] J. Irvine and D. Harle, *Data Communications and Networks: An Engineering Approach*. New York: Wiley, 2002.

[8] M. M. Yeung and F. C. Mintzer, "Invisible watermarking for image verification," *Electron. Image*, vol. 7, no. 3, pp. 578–591, Jul. 1998.

[9] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[10] F. Perez-Gonzlez and F. Balado, "Quantized projection data hiding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Sep. 2002, pp. 889–892.

[11] C. W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel, "Lossless Recovery of an Original Image Containing Embedded Data," U.S. Patent 6 278 791 B1, Aug. 21, 2001.

[12] Jiri Fridrich.1997. "*Image Encryption Based on Chaotic Maps"*. Proceedings of IEEE Conference on Systems, Man and Cybemetics. 1105-1110, 1997.

[13] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.