International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 4, April - 2014

Implementing Separable Reversible Data Hiding In Encrypted Image Using CLM RC4 Method

Pallavi S. Bangare
Dept. of Information Technology
Sinhgad Academy of Engineering
Pune, India

Mandar S. Raut
Dept. of Information Technology
Sinhgad Academy of Engineering
Pune, India

Amogh S. Bagade
Dept. of Information Technology
Sinhgad Academy of Engineering
Pune, India

Sahil Pandita
Dept. of Information Technology
Sinhgad Academy of Engineering
Pune, India

Sunil L. Bangare
Dept. of Information Technology
Sinhgad Academy of Engineering
Pune, India

Vishnu V. Menon
Dept. of Information Technology
Sinhgad Academy of Engineering
Pune, India

***Abstract*— In this paper an innovative method for Separable Reversible Data Hiding in encrypted image is discussed. A novel secure encryption mechanism is proposed which is a combination of chaotic logistic mapping and RC4 stream cipher. The encryption of plain image using an encryption key constitutes the first phase. The second phase is the data embedding phase where the data hider uses the data hiding key to compress the LSB and insert some additional data in the space created. The third and last phase is Data retrieval and Image recovery phase. In this phase, the receiver having encrypted image with additional data can retrieve hidden data or recover image or both, depending on the key possessed by him. 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 key 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—cryptography; chaotic logistic map; reversible data hiding;

I. INTRODUCTION

Due to the exponential growth of multimedia application, security becomes an important issue of communication and storage of images. Encryption is one of the ways to ensure high security. Images are used in many fields such as medical science, military; they are stored or transferred through network, security of such image data is important. Nowadays when more and more sensitive information is stored on computers and transmitted over the Internet, we need to ensure information security and safety. Images are also an important part of information. Therefore it is very important to protect images from unauthorized access. Security of multimedia information is used to protect the multimedia content from unauthorized access. The simplest and most widely used full-reference quality metric is the mean squared error (MSE), computed by averaging the squared intensity differences of distorted and reference image pixels, along with the related quantity of peak signal-to-noise ratio (PSNR) [9].

In an encryption scheme, the image is encrypted using an encryption algorithm, turning it into an indistinct or unclear data. This is usually done with the use of an encryption key. It is needed as it enhances the security and helps in maintaining privacy.

As an effective and popular means for privacy protection, encryption converts the ordinary signal into incomprehensible data, so that the general signal processing typically takes place before encryption or after decryption. However, in some situations that a content owner does not trust the service provider, the ability to manipulate the encrypted data when keeping the plain content secret is desired.

Reversible data embedding, which is also called lossless data embedding, embeds invisible data (which is called a payload) into a digital image in a reversible fashion [5]. As a basic requirement, the quality degradation on the image after data embedding should be low. An intriguing feature of reversible data embedding is the reversibility, that is, one can remove the embedded data to restore the original image [7].

There are also a number of works on data hiding in the encrypted domain. In a buyer–seller watermarking protocol [8] the seller of digital multimedia product encrypts the original data using a public key, and then permutes and embeds an encrypted fingerprint provided by the buyer in the

encrypted domain. After decryption with a private key, the buyer can obtain a watermarked product. This protocol ensures that the seller cannot know the buyer's watermarked version while the buyer cannot know the original version.

Fig. 1 shows the existing methodology.

Here,

1st Stage: The image is encrypted with the help of an encryption key.

2nd Stage: The additional data is added with the help of a data hiding key.

Now on the receiver side the image has to be decrypted first now then it is possible to remove the hidden data. The user who is interested in only the data must be in possession with both the keys as the image has to be decrypted with the help of the encryption key then the data can be obtained with the help of the data hiding key. The user has to go through these phases irrespective of what he wants only the image or the data 0or both.

In our proposed method we perform the encryption/decryption using the CLM RC4 algorithm and the data embedding Is carried out with the help of LSB Compression.[7]

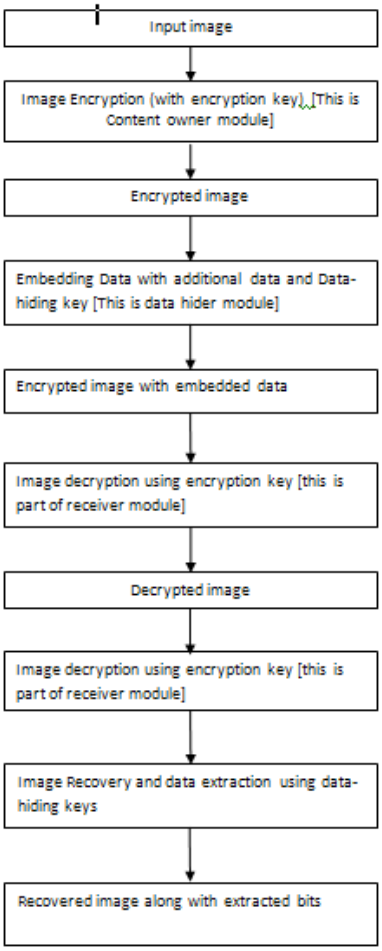


Fig. 1: Non-Separable Process

II. PROPOSED SCHEME

Our Proposed Scheme consists of Three phases:-

1. Image Encryption- At this stage Image is encrypted using RC4 stream cipher algorithm and chaotic logistic map.
2. Data Embedding-The data is embedded into the encrypted image using the LSB Compression Algorithm.
3. Data Extraction and Image Recovery-At this stage the data embedded as well as the image encrypted can be recovered.

A. Image Encryption-

For the purpose of image encryption we have used an innovative and secure algorithm, which based on RC4 stream cipher algorithm and chaotic logistic map.

Transmitting a digital image over the internet possesses many threats like illegal copying, loss of confidential data. The problem that many current data encryption methods such as DES, RSA, AES, and others is that they are only suitable for text data, but not for digital image.

To overcome these problems we have used digital image encryption using the combination of RC4 stream cipher and chaotic logistic map function. The reasons behind using this combination are (i) the simplicity of RC4 algorithm, (ii) RC4 requires only byte-length manipulations so it is suitable for embedded systems, (iii) even though RC4 has vulnerabilities [4], and we combined this with chaotic systems to make it almost impossible to break.

Chaotic Logistic Map Function

Ni G. A. P. Harry Saptarini et al have given following research contribution [3]. Chaos is phenomenon that exists in nonlinear systems, in which seemingly random events are actually predictable from simple deterministic equations [2]. One of the important properties of chaos is extreme sensitivity to initial conditions. Chaotic logistic map is one of the popular chaotic systems. Consider a CLM function as shown in Equation (1).

$$X_{n+1} = \lambda X_n (1 - X_n) \quad (1)$$

Where λ is a control parameter on the interval λ = [0,4] and X_n is real number on the interval

X_n = [0, 1].This system is said to be chaotic if λ has a value on the interval λ = [3.569955672, 4]. In this paper, we use $\lambda = 4$ so the complete formula is shown in Equation (2).

$$X_{n+1} = 4 X_n (1 - X_n) \quad (2)$$

Image Encryption Algorithm:

The structure of our encryption method (as shown in Figure 1) consists of three main units:

(i) Converter unit,
(ii) CLM function unit, and

(iii) RC4 stream cipher unit.

The Converter unit's task is to convert the encryption key to initial value [2]. It uses Equations (3), (4), (5), (6), (7) and (8) to convert key to initial value X0.The output of converter unit is an initial value X0 which will be used by CLM function unit to generate 256-bytes of array U[i]. U[i] is also known as key array.

In the last step contents array U[i] and array S[i] is swapped and its result is XORed with byte streams of input plain image to produce cipher image. When the cipher-image is given as input to this process it produces the original plain image. In this encryption method we do not manipulate with the header part of the image but only the gray value of pixel.

Our proposed encryption method consists of three different steps. The entire encryption method is described as follows:

Step (1). Convert key to initial value.

The key has 16 ASCII characters in length where each character Ki of consists of 8-bit.

$$K = K_1, K_2 \ldots K_{16}(\text{ASCII code}) \quad (3)$$

For each Ki value, we convert them to bit stream B_0 and B_1.

$$B_0 = K_{11} \ldots K_{18} K_{21} \ldots K_{28} K_{31} \ldots K_{38} K_{41} \ldots K_{48} K_{51} \ldots K_{58} K_{61} \ldots K_{68} K_{71} \ldots K_{78} K_{81} \ldots K_{88} \quad (4)$$

$$B_1 = K_{91} .. K_{98} K_{101} .. K_{108} K_{111} .. K_{118} K_{121} .. K_{128} K_{131} .. K_{138} K_{141} .. K_{148} K_{151} .. K_{158} K_{161} .. K_{168} \quad (5)$$

where each Kij from Equation (4) and (5) has binary representation (0 or 1), where *i* refers to character position (*i=1,2,...,16*) and *j* refers to bit position of character (*j=1,2,...,8*). Using binary representation of Kij value, the real number X01and X02will be counted.

$$X_{01} = (K_{11} \times 2^0 + \cdots + K_{18} \times 2^7 + K_{21} \times 2^8 + \cdots + K_{28} \times 2^{15} + K_{31} \times 2^{16} + \cdots + K_{38} \times 2^{23} + K_{41} \times 2^{24} + \cdots + K_{48} \times 2^{31} + K_{51} \times 2^{32} + \cdots + K_{58} \times 2^{39} + K_{61} \times 2^{40} + \cdots + K_{68} \times 2^{47} + K_{71} \times 2^{48} + \cdots + K_{78} \times 2^{55} + K_{81} \times 2^{56} + \cdots + K_{88} \times 2^{63}) / 2^{64} \quad (6)$$

$$X_{02} = (K_{91} \times 2^0 + \cdots + K_{98} \times 2^7 + K_{101} \times 2^8 + \cdots + K_{108} \times 2^{15} + K_{111} \times 2^{16} + \cdots + K_{118} \times 2^{23} + K_{121} \times 2^{24} + \cdots + K_{128} \times 2^{31} + K_{131} \times 2^{32} + \cdots + K_{138} \times 2^{39} + K_{141} \times 2^{40} + \cdots + K_{148} \times 2^{47} + K_{151} \times 2^{48} + \cdots + K_{158} \times 2^{55} + K_{161} \times 2^{56} + \cdots + K_{168} \times 2^{63}) / 2^{64} \quad (7)$$

Next step, real number X01 in Equation (6) and X02 in Equation (7) is used to create initial value X0. The complete formula for creating initial value is shown in Equation (8).

$$X_0 = (X_{01} + X_{02}) \bmod 1 \quad (8)$$

Step (2). Generate a key array (pseudo random number sequence) using chaotic logistic map function.

The initial value X0 in Equation (8) will be used by CLM function to generate a key array of pseudorandom number sequence by using the formula in Equation (2). Generally, the chaotic process uses initial value X0 to get X1 value, then X1 value will be used to get X2 value, and so on. In order to strengthen CLM against any statistical attacks, we generate Xn value after a certain number of iterations. We determine the number of iterations by taking two digits after decimal point. For example, the initial value X0 is 0.937696878979928 then the number of iterations required to get the first value of chaos X1 is 93,thus after iterations, X1 value will be 0.8080204084200282. We can say that the value of Xn which is obtained at the end of iteration will act as a new "Xn" to calculate Xn+1 and so on. After each Xn value is obtained, it will be converted to integer form by taking eight points started after the decimal point of real numbers. For example, assuming that the value of Xn is 0.8080204084200282. After converting this value to integer form will yield 80802040, and then it will be modulo 256. The result value will be stored in array U[i] where *i = 0,1, ..., 255*. This process will be repeated until U[255] is filled.

Step (3). RC4 streams cipher and encryption/decryption process for RGB channel.

The output of Step (2) is an array U[i] which is also called "key streams" and consists of 256 pseudorandom numbers. Array S is created (as shown in Figure 2a.) where the content of array *S* are set equal to the values from 0 through 255 in ascending order; which is S[0] = 0, S[1] = 1, …. , S[245] = 254,S[255] = 255. Next step, array *U* is used to produce the initial permutation of array *S* (Figure 2b.). For each *S[i]*, swap *S[i]* with another byte in *S* according to the content of *U[i]* and this will cause the content of *S* still contains all the numbers from 0 through 255. In Figure 2a., streams generation is done by swapping *S[i]* with another byte in *S* according to a scheme dictated by the current configuration of *S*. The encryption/decryption process for each RGB channel is done by XORed each pixel's of RGB component of plain-image with the bytes of array S (or XORed the bytes of cipher-image with the bytes of array S when do a decryption process). The decryption algorithm is identical to the encryption algorithm discussed above except that the order of the basic operations is reversed.

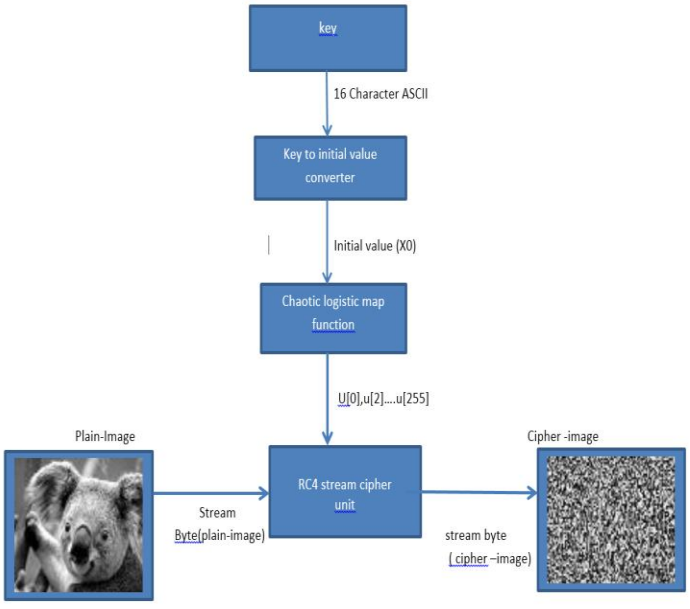


Fig. 2: Strucutre of Encryption Method

The algorithm of RC4 streams cipher [3] referred for this implementation is .

(a) Initialization of array S,

(b)Initial permutation of array S,

(c) Encryption/Decryption process for each RGB channel,

(d) Permutation process.

B. Data Embedding-

A unique reversible (lossless) data hiding (embedding) technique, which enables the exact recovery of the original host signal with the extraction of the embedded information. And this exact recovery with lossless data is nothing but the reversible data hiding [6]. Now by providing the data hiding key he can embed the required data in the image. And the image is sent on the extraction side. And by providing the necessary key u can get either the image or the data or both [1].

Some parameters are embedded into a small number of encrypted pixels. The LSB of the other encrypted pixels are compressed to create a space for accommodating the additional data.

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.

The other (N-Np) encrypted pixels are randomly permuted and divided into a number of groups, each of which contains L pixels.

The data hider also generates a matrix G sized (M . L – S) * M . L, which is composed of two parts, where S are the number of bits embedded into each pixel group.

For each group, Calculate:

$$\begin{bmatrix} B'(k,1) \\ B'(k,2) \\ \vdots \\ B'(k,ML-S) \end{bmatrix} = \mathbf{G} \cdot \begin{bmatrix} B(k,1) \\ B(k,2) \\ \vdots \\ B(k,ML) \end{bmatrix}$$

The embedding rate, i.e., a ratio between the data amount of net payload and the total number of cover pixels is:

$$R = \frac{((N - N_P) \cdot S/L - N_P)}{N} \approx \frac{S}{L}.$$

The diagram below [1] represents implementation scheme with Separable Processes:-

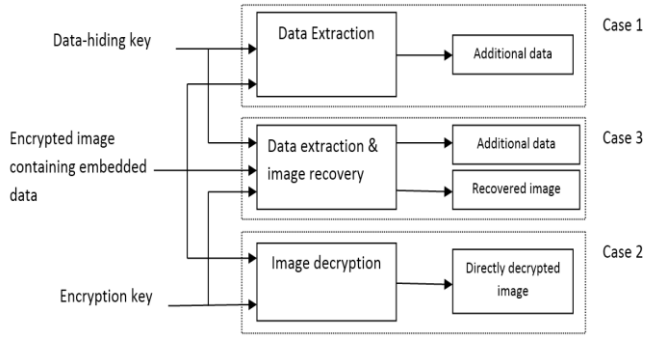


Fig. 3: implementation scheme with Separable Processes

3. Data Extraction and Recovery :-

In this phase, we will consider the three cases that a receiver has onlythe data-hiding key, only the encryption key, and both the data-hidingand encryption keys, respectively.

Case 1: Receiver has the data-hiding key

With an encrypted image containing embedded data, if the receiverhas only the data-hiding key, he may first obtain the values of the parametersM, L and S from the LSB of the Np selected encryptedpixels. Then, the receiver permutes and divides the other(N-Np)pixels into (N-Np)/L groups and extracts the S embedded bits fromthe M LSB-planes of each group. When having the total (N - Np).S/L extracted bits, the receiver can divide them into Np original LSBof selected encrypted pixels and (N - Np).S/L-Np additional bits.

Note that because of the pseudo-random pixel selection and permutation,any attacker without the data-hiding key cannot obtain the parametervalues and the pixel-groups, therefore cannot extract the embeddeddata. Furthermore, although the receiver having the data-hiding key cansuccessfully extract the embedded data, he cannot get any informationabout the original image content.

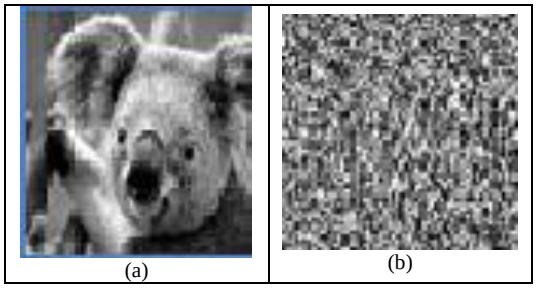
Case 2: Receiver has encryption key

Consider the case that the receiver has the encryption key butdoes not know the data-hiding key. Clearly, he cannot obtain thevalues of parameters and cannot extract the embedded data. However,the original image content can be recovered. The decryption algorithm is identical to the encryption algorithm discussed above except that the order of the basic operations is reversed.

Since the data-embedding operation does not alter any MSB of encrypted image, the decrypted MSB must be same as the original MSB.So, the content of decrypted image is similar to that of original image.if

Case 3: Receiver has both keys

In the event of receiver possessing both keys, he can decrypt the image to obtain original image using the encryption key.He can also use data-hiding key to retrieve the embedded data. The process of decryption and data retrieval is same as described above.



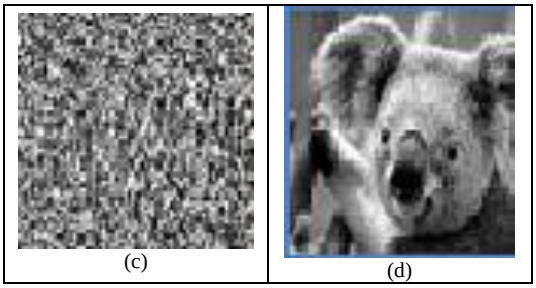


Fig. 4: Experimental Results after implementation: (a) Original image (b) Encrypted Image (c) Encrypted image with embedded data (d) Original image recovered at the receivers end.

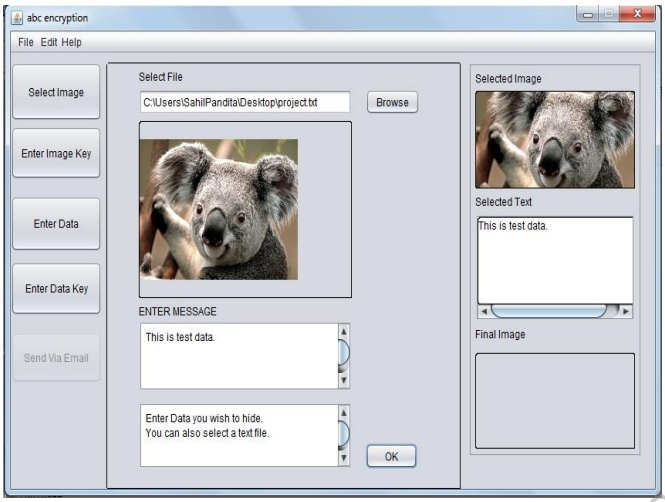


Fig. 5: Screenshot of the Experiment

CONCLUSION

In this paper, a novel scheme for separable reversible data hiding in encrypted image is implemented, which consists of image encryption, data embedding and data-extraction/image Recovery phases. In the first phase, the content owner encrypts the original uncompressed image using RC4 algorithm using CLM. Although a data-hider does not know the original content, he can compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data. With an encrypted image containing additional data, the receiver may extract the additional data using only the data-hiding key, or obtain an image similar to the original one using only the encryption key. When the receiver has both of the keys, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image if the amount of additional data is not too large. If the lossless compression method is used for the encrypted image containing embedded data, the additional 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.

REFERENCES

[1] Xinpeng Zhang, "Separable Reversible Data Hiding in Encrypted Image", IEEE Transactions on information forensics and security, Vol. 7, No. 2, APRIL 2012

[2] N.K. Pareek, "Image Encryption using Chaotic Logistic Map" , in *Image and Vision Computing,* Vol.24,pp. 926–934, 2006.

[3] Ni G. A. P. Harry Saptarini,Yosua Alberth Sir , "Digital Color Image Encryption Using RC4 Stream Cipher andChaotic Logistic Map", Information Systems International Conference (ISICO), Vol .2, Issue 1 January,February 2013.

[4] E. Tews ,M. Beck, "Practical attacks against WEP and WPA," in Proceedings the second ACM conferenceon Wireless network security, Zurich, pp. 79-86, 2009.

[5] Zhou Wang,Alan Conrad Bovik,Hamid Rahim Sheikh,Eero P. Simoncelli "Image Quality Assessment: From Error Visibility to Structural Similarity", IEEE Transactions on Image Processing,Vol. 13, No. 4, April 2004.

[6] Mehmet Utku Celik, Gaurav Sharma, Ahmet Murat Tekalp ,Eli Saber 'Lossless Generalized-LSB Data Embedding', IEEE Transactions on Image processing, Vol. 14, No. 2, February 2005.

[7] Jun Tian, "Reversible Data Embedding Using a Difference Expansion" IEEE trans Circuits and Systems for Video Technology, Vol. 13, No. 8, August 2003.

[8] Jessica Fridrich,Miroslav Goljan,Rui Du "Lossless Data Embedding—New Paradigm in DigitalWatermarking" EURASIP Journal on Applied Signal Processing 2002:2, Vol. 185–196, 2002.

[9] Zhou Wang ,Alan Conrad Bovik, Hamid Rahim Sheikh, Eero P. Simoncelli "Image Quality Assessment: From Error Visibility to Structural Similarity" IEEE Transactions on Image Processing, Vol. 13, No. 4, April 2004.