DATA HIDING USING RC4 ALGORITHM IN IMAGE FORM

Geetanjali N. Narhare[1], Savita R. Bhosale[2]

[1]PG Scholar, Computer Engineering, MGMCET Kamothe, Navi Mumbai (India)

[2]AssistantProfessor, Electronics and Telecommunication, MGMCET Kamothe, Navi Mumbai (India)

ABSTRACT

As we know that the improved technologies in communication, Security is very important in several fields. So we should make an arrangement of security of our valuable data. Data contain different types of data that include text, audio, video, graphics images. This paper enlighten the problem of transmitting redundant data over an insecure, bandwidth-constrained communications channel. In this proposed system, secret messages are encrypted before embedding it into the cover image which gets high security to secret data RC4 algorithm is used to encrypt secret messages and Least Significant Bits (LSB) based data embedding technique is used to high encrypted messages. To hide encrypted messages into BMP image file, pseudorandom sequences are used. A content owner encrypts the original uncompressed image using an encryption key. After that, a data-hider may compress the least significant bits (LSB) of the encrypted image using a data-hiding key to create a sparse space to accommodate some additional data. If the receiver has both the data-hiding key and the encryption key, the receiver can extract the additional data and the original image without any loss.

Keywords: Digital Image Processing, Image encryption, Image Recovery, RC4, Reversible Data Hiding, Security

I. INTRODUCTION

It is important that the dada transmitting in high security which was not affected by visible loss of data. To hide secret data in such manner that data can be reversed, Reversible Data Hiding (RDH) technique is used. Data can be restored to its original manner without any loss and also without using any other information. This can be termed as safe embedding. At the end of the receiver, hidden data is extracted and image is also restored in its original form. This technique is more useful in applications in which original image should remain intact even after data embedded is retrieved [1, 2]. Reversible data embedding can be viewed as an information carrier. It is impossible for human eyes to distinguish between embedded image and original image. Because of this, reversible data embedding can be thought as secret communication model. As an effective and standard means for privacy protection, encryption converts the ordinary signal into unintelligible data, so that the traditional signal processing usually takes place before encryption or after decryption [3, 4]. The traditional way of transmitting redundant data is to first compress the data to reduce the redundancy, and then to encrypt the compressed data to mask its meaning. The sender should encrypt the original data and the network provider tend to compress the encrypted data without any knowledge of the cryptographic key and the original data. At the side of receiver, a decoder integrating decompression and decryption functions for reconstructing the original data [5,6].

II. PROCEDURE OF ENCRYPTION AND DECRYPTION WITH FLOW DIAGRAM

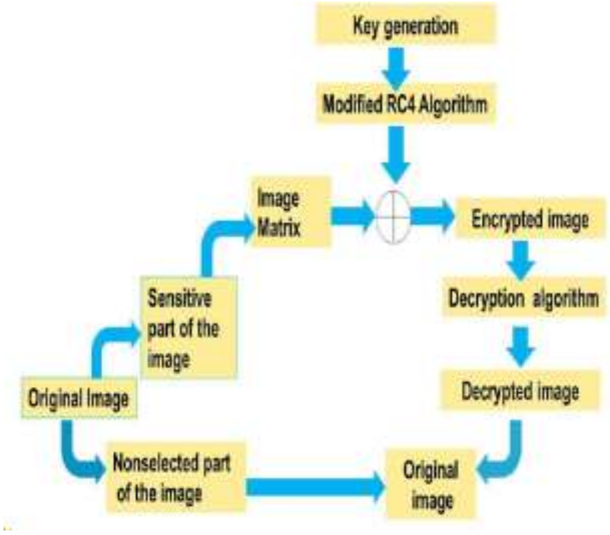


Fig. 1: Procedure of Encryption And Decryption

Normally cryptography starts with taking an image as an input after that applying the required algorithm encrypting the image that is called encrypted image. But for selected image encryption first of all we have to specify the regions we are going to encrypt. Then the encryption algorithm works. By using algorithm the selected parts of the image is being encrypted and the other parts remains as it is. We got the encrypted image after the end of this step. With the help of same algorithm we can decrypt the selected regions. We again got the original image back, after the end of this process. The overall procedure is shown in Figure 1.

III. PROPOSED SYSTEM DESIGN

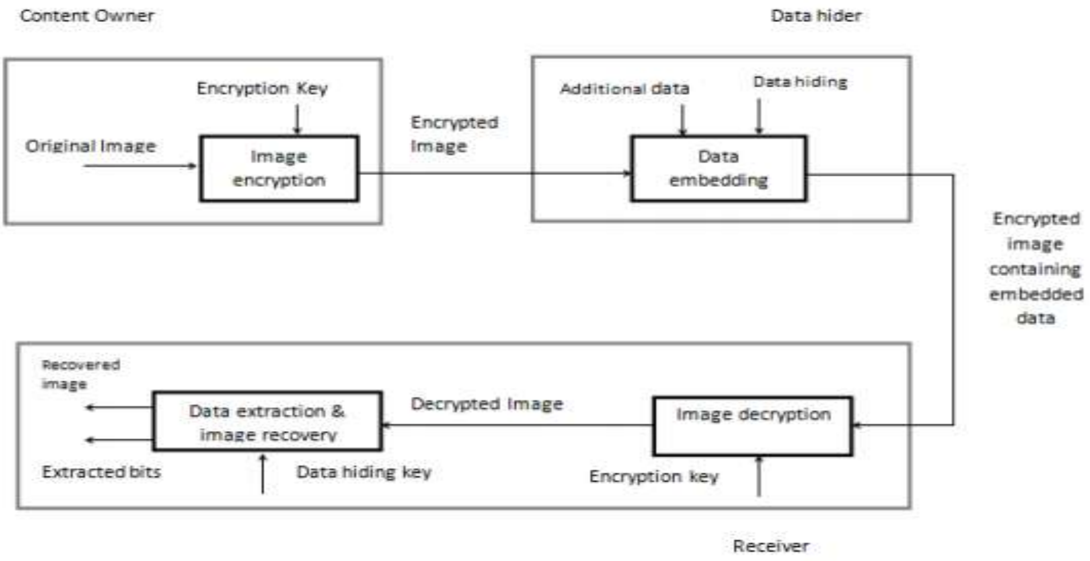


Fig 2.Proposed System Data Hiding İn Encrypted İmage

A owner of content encrypts the original image using an encryption key, then a data-hider can embed additional data into the encrypted image with the help of a data-hiding key though the receiver does not know the original data. With an encrypted image containing additional data, a receiver may first decrypt it according to the encryption key, and then recover the original image and extract the embedded data according to the data-hiding key.

3.1 Image Encryption Types

Image encryption involves two types, generation of encryption key and generation of pseudo-random sequence.

3.1.1 Generation of Encryption Key

Encryption key is 128 bit value. By using the random function, encryption key generated randomly. The random function generates the random key in an uniformly distributed function.

3.1.2 Generation of Pseudo-Random Sequence

It consists of random bits generated using the encryption key. In this paper, RC-4 algorithm is used to create the pseudo-random sequence using the 128-bit encryption key. It is represented as sequence of bytes or an array of bytes. The number of bytes generated should be equal to the number of pixels in the input image provided the pixels are represented as 8-bit values. If the pixels are represented as 16-bit values then the number bytes in pseudo-random sequence should be double the number of pixels.

IV. DATA EMBEDDING

In embedding, encrypted pixels are formed with the use of some parameters and LSB's of other encrypted pixels are compressed to create more space so that more data can be added. Data hider pseudo-randomly chooses Np encrypted pixels that will be used to carry the parameters for data hiding according to a data-hiding key,. Here, Np is a positive integer, for example, Np=20.The other (N-Np) encrypted pixels are pseudo-randomly permuted and divided into a small number of groups. Each group contain L pixels. The permutation way is also determined by the data-hiding key. For each pixel-group, collect the M least significant bits (LSB) of the L pixels, and indicate 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 produces a matrix G sized (M.L-S) × M.L, which is composed of two parts.

$$G= [\ I_{M.L-S}\ Q] \qquad (1)$$

Data hiding key is used to derive pseudo binary matrix i.e. Q sized (M.L-S) × S which is right part, and left part is (M.L-S) × (M.L-S) identity matrix. Where S is small positive integer. Then next procedure is embed all the values of parameters M, L and S into LSB of NP. So, the rate is:

$$R = ((N\text{-}NP).\ (S/L) – NP)\ /\ N = S\ /\ L \qquad (2)$$

where, R is encrypted data embedded rate, N is Number of pixels present in the encrypted image, NP is Number of pixels which carries the parameters, S is Small positive integer and L is Number of pixels in each pixels group.

V. DATA-EXTRACTION AND IMAGE-RECOVERY

Our proposed scheme contains image encryption, data embedding and data extraction phase. At encryption side, encryption is done using an encryption key. Then, the data-hider compresses the LSB of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data. At the second side, embedded data can be easily retrieved from the encrypted image containing additional data according by using data-hiding key. It affects only LSB, a decryption with the encryption key, the result is similar to the original image. By using both encryption algorithms and data hiding keys we can effectively extract original image by using spatial correlation in natural image. Fig. 3 shows the three cases at the receiver side.

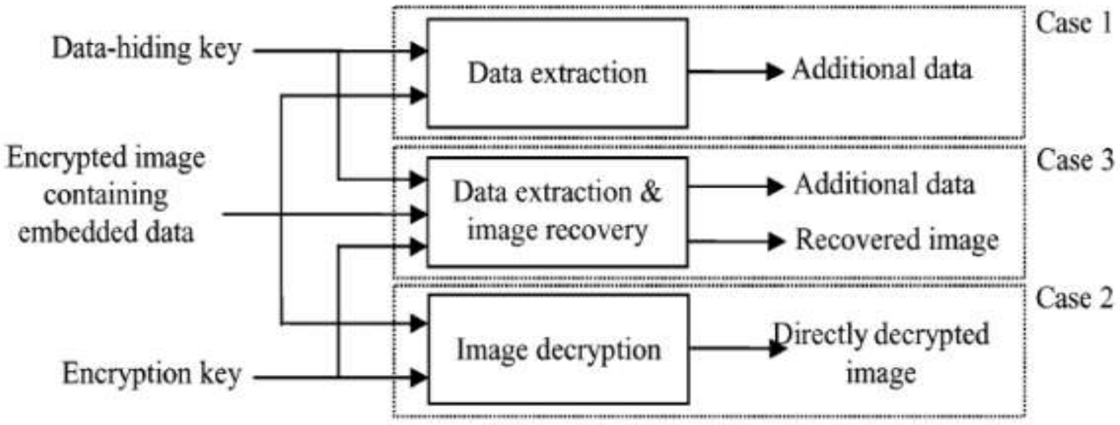


Fig.3. Three Cases At Receiver Side of The Proposed Separable

In this phase, there are three cases that, in first case a receiver has only the data-hiding key, in second case only the encryption key, and in third case both the data-hiding and encryption keys, respectively. If the receiver has only the data-hiding key by using an encrypted image containing embedded data, , he may first acquire the values of the parameters M, L and S from the LSB of the Np particular encrypted pixels. And then, the receiver permutes and divides the other. (N-NP) pixels into (N-NP)/L groups and extracts the S embedded bits from the M LSB-planes of each group. When consuming the total (N-NP).S/L extracted bits, then receiver can divide into NP new LSB of certain encrypted pixels and (N-NP).S/L-NP additional bits.

Important thing is that because of selection and permutation of the pseudo-random pixel, any attacker without helping the data-hiding key cannot obtain the parameter values and the pixel-groups, therefore the embedded data cannot extract. Moreover, the receiver can successfully extract the embedded data by using data-hiding key. Representing the bits of pixels in the encrypted image containing embedded data, the receiver can decrypt the received data from the encrypted data in the M LSB-planes. Assuming that the original distribution of the data in the M LSB planes is uniform, the alteration energy per each decrypted pixel is because the probability of this case.

De = $(2)^{\wedge-2M} = \sum_{\beta=0}^{2M-L}(\alpha - \beta)$ ^2 (3)

Here, the alteration in the NP selected pixels is also ignored since their number is significantly less than the image size N. So, the value of PSNR in the directly decrypted image

PSNR = 10. $\log_{10}$ (AE) (4)

Where AE is Average Energy. If the receiver has both the data-hiding and the encryption keys, receiver may goal to the embedded data extract and original image recovery. According to the data-hiding key, the values of M, L and S are the original LSB of the NP selected encrypted pixels, and the (N-NP).S/L-NP are additional bits can be extracted from the encrypted image enclosing embedded data. By placing the NP LSB into their original locations, the encrypted data of the NP selected pixels are recovered, and their original gray values can be properly decrypted using the encryption keys.

VI. RIVEST CIPHER 4 (RC4) ALGORITHM

This algorithm is developed by Ronald Rivest and thus, the name of the algorithm was put after Ronald's Rivest name. RC1, RC2, RC3, RC4, RC5 and RC6 is the series of RC algorithm.RC4 is symmetric key algorithm. It is one of the algorithm which is used for both encryption and decryption as the data stream is simply XORed with the generated key sequence. The key stream is completely independent of the plaintext used. It uses a flexible length key from 1 to 256 bit to modify a 256-bit state table. The state table is used for consequent generation of pseudo-random bits after that to generate a XORed pseudo-random stream with the plaintext to give the cipher text. The algorithm can be divided into two stages: First stage is initialization, and other operation. In the first stage the 256-bit state table, S is populated, using the key, K as a seed. Once the state table is setup, it continues to be modified in a regular pattern as data is encrypted. The initialization process can be summarized by the pseudo-code.

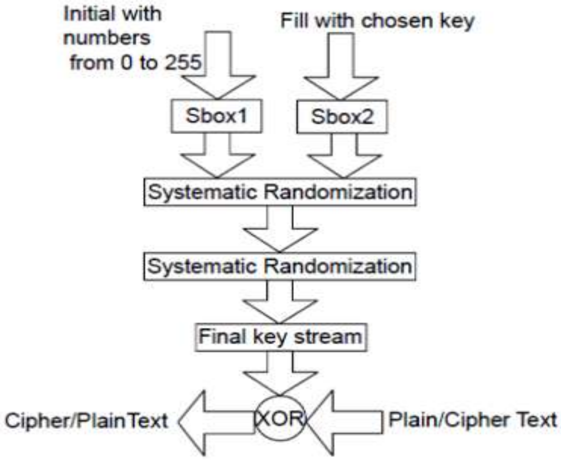
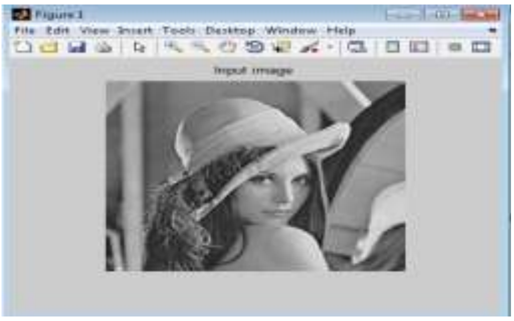


Fig 4. Rc4 Encryption Algorithm

This algorithm generates a pseudo-random stream values. XORed is the input value with these values, bit by bit. The process of encryption and decryption is the same as the data stream is just XORed with the generated key sequence. If it is fed in an encrypted message, it will produce the output of decrypted message, and if it is fed in plain text message, the encrypted version will produce. The RC4encryption algorithm is shown in Fig.4.

VII. RESULTS AND DISCUSSIONS

The test image Lena sized 512 ×512 is used as the original image in the experiment. After image encryption, the eight encrypted bits of each pixel are converted into a gray value to generate an encrypted image.



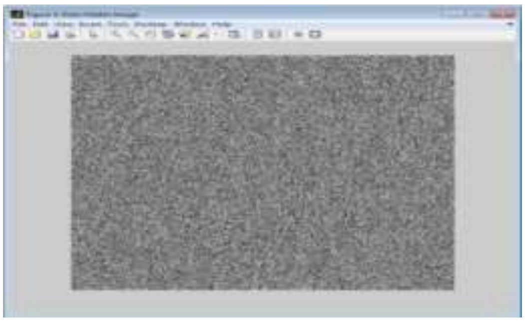


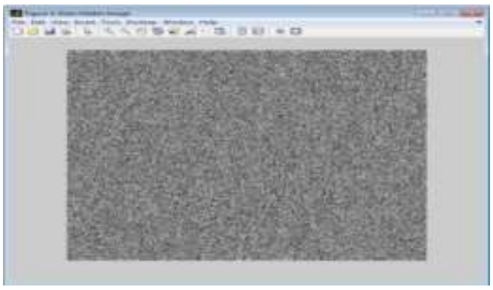
Fig 5. Original Image **Fig 6. Encrypted Image**

Let M=2, L=76 and S=1 to additional bits into the encrypted image. The encrypted image containing the embedded data embedding rate R is 0.013158 bit per pixel (bpp). With an encrypted image containing embedded data, we could extract the additional data using the data-hiding key. Then, embed the values of the parameters M, L and S into the LSB of NP selected encrypted pixels.

Np = 20, N = 262144, M = 2, L = 76 and S = 1

Enter a maximum of 214 characters: Geetanjali

Data Embedded Rate = 0.013158 bpp



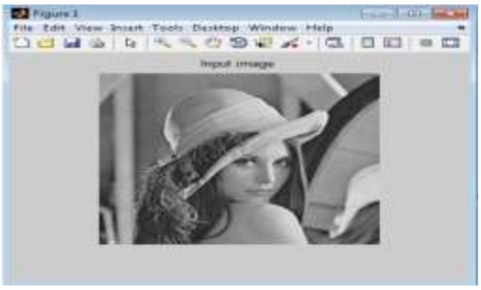
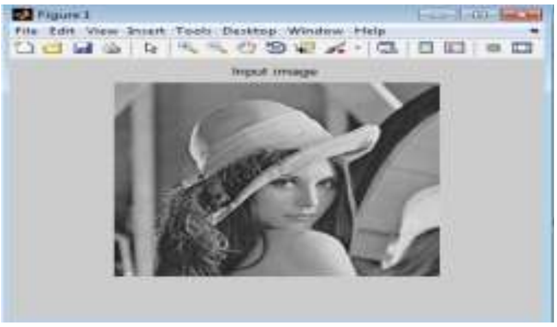


Fig.7. Data Hiding Image **Fig 8. Decrypted Image**

When putting the value Np, N, M, L and S with the maximum characteristics value the correspond encrypted image, data hiding image and decrypted images fallen one by one. If we directly decrypted the encrypted image containing embedded data using the encryption key, the value of PSNR in the decrypted image was 40.0 dB, which verifies the theoretical value 40.0 dB is calculated. PSNR of the decrypted image = 40.930937 dB

The recovered image and directly decrypted image are shown in Fig.9 and Fig.10 respectively. The image recovered using RC4 algorithm is same as the original image.



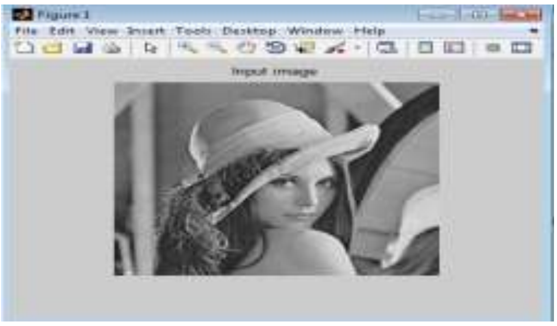


Fig 9. Image Recovery **Fig 10.Directlydecrypted Image**

VIII. CONCLUSION

With the encryption key Pseudo random structure consists of random bits generated. In this paper to create sequence of pseudo-random in the 128-bit encryption key by using RC-4 algorithm. With the parameters we can additional data inserted in to an encrypted image. Additional data which is encrypted in image. With an encrypted image containing additional data, with data-hiding key receiver can extract the additional data, or using only the encryption key can obtain an image similar to the original one. When using both of the encryption and data-hiding keys, the embedded additional data can be successfully extracted and the original image can be perfectly recovered by exploiting the spatial correlation in natural image. Compared with the other algorithms, the proposed system demonstrated successful accuracy in recovering the original images. In the future, a comprehensive combination of image encryption and data hiding compatible with lossy compression deserves further investigation.

IX. ACKNOWLEDGEMENT

I am Geetanjali Narhare, would like to thank Prof. S.R.Bhosale of Electronics and Telecommunication Department, MGMCET Kamothe, Navi Mumbai, India, for her contribution to this work.

REFERENCES

[1]. X. Zhang, “Lossy compression and iterative reconstruction for encrypted image,” IEEE Trans. Inform. Forensics Security, vol. 6, no. 1, pp. 53–58, Feb. 2011.

[2]. 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.

[3]. T. Bianchi, A. Piva, and M. Barni, “Composite signal representation for fast and storage-efficient processing of encrypted signals,” IEEE Trans. Inform. Forensics Security, vol. 5, no. 1, pp. 180–187,Feb.2010.

[4]. S. Lian, Z. Liu, Z. Ren, and H. Wang, “Commutative encryption and watermarking in video compression,” IEEE Trans. Circuits Syst. VideoTechnol., vol. 17, no. 6, pp. 774–778, Jun. 2007.

[5]. Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354– 362, Mar.2006.

[6]. M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K. Ramchandran, “On compressing encrypted data,” IEEE Trans. Signal Process., vol. 52, no. 10, pp. 2992–3006, Oct. 2004.

[7]. Glover, P. and M. Grant, Digital Communications, 2nd edition, Person Education, 2004.

[8]. M Joset Pieprzyk, et. al., Fundamentals of Computer Security, Springer, 2003.

[9]. M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. B. Natale, and A Neri, “A commutative digital image watermarking and encryption method in the tree structured Haar transform domain,” Signal Processing:

[10]. Image Commun., vol. 26, no. 1, pp. 1–12, 2011.Z. Wang and A. C. Bovik, “A universal image quality index,” IEEE Signal Process. Lett., vol. 9, no. 1, pp. 81–84, Jan. 2002.