Secure Data Transmission using Rabbit Stream Cipher

Sruthi L., Ansha Beevi S., Divya Sreenivasan, Sajiyya K
Department of Computer Science
College of Engineering Karunagapally
Kollam, India

***Abstract*-Reversible data hiding(RDH) is a method by which original cover can be perfectly recovered after the embedded data is extracted while protecting the image's confidentiality. This method uses the RRBE framework. All the existing cryptographic methods have certain demerits. In order to ensure security our proposed method uses Rabbit stream cipher to encrypt the image, which provides better security than any other existing methods.**

Keywords: Reversible data hiding, RRBE, Rabbit stream cipher, Image encryption

I. INTRODUCTION

Data hiding is a method in which secret data can be embedded into images, videos audios etc. This method is mainly used to hide confidential data and find application in fields such as military, medical imaging, law forensic etc. Reversible data hiding technique retrieve the embedded data and recover the cover image perfectly, which is an essential property needed in medical and military fields.

Many methods have been proposed for reversible data hiding. Earlier all the algorithms are based on embedding data into plain image. In all these methods data is embedded into LSBs of each pixel to avoid introducing much variation.Tian[1] established a reversible data embedding algorithm which uses difference expansion. Data is embedded into the expandable difference values of pixels. This method explores the redundancy in digital images to achieve very high embedding rates. Another promising method in RDH is histogram shift [2], which utilizes the zero and maximum points of histogram of image. The space for data embedding is achieved by shifting the bins of histogram towards the maximum or zero points. Al-Fahoum in [3] extends the previous method for improving the embedding capacity, which divides the image into blocks and perform contrast stretching in each block to embed data.

To improve the embedding rates and reversibility many other methods are also proposed. Tian's expansion method [1] has a best embedding capacity of 0.5b/pixel and the location map used in this method also needs 0.5b/pixel. Hence reducing the size of location map is a key issue. Sachnev [4] proposed a method which uses prediction errors to embedded data into images and also it doesn't need location map. Also this method uses a sorting method to arrange the prediction error based on the magnitude of its local variance, since smaller variance values are better for data hiding. Luo [5] utilizes interpolation error, the difference between interpolation value and corresponding pixel value is used to embed bits.

All these methods focus only on reversible data embedding and don't consider the confidentiality and security of image and embedded data. Later many methods are proposed to provide confidentiality for images; encryption is an effective method which converts the original content into incomprehensible one. Zhang [6] proposed the novel RDH scheme for encrypted images. The entire image is encrypted by a stream cipher; a small proportion of encrypted image is modified to embedded data. W.Hong [7] proposed an improved version of Zhang's method. Here the encrypted image is divided into blocks and each block is used to carry one bit by flipping the three LSBs of a set of pre-defined pixels. This method uses a side match scheme which avoids the incorrectness in data extraction of previous method. That is previous method didn't fully exploit the pixels in calculating the smoothness of blocks and didn't consider the correlation of pixels in the border of neighbouring blocks. All these methods rely on special correlation of images; encrypted image should be decrypted first before data extraction.

Zhang[8] proposed a method to separate data extraction from image decryption. Space is emptied out in compressed images. The encrypted LSBs are compressed to vacate room for additional data by finding syndromes of parity check matrix.

Reserving Room After Encryption (RRAE) is the method followed in all these methods. But this may result in certain difficulties. Since the entropy of encrypted image is maximised, these techniques can achieve only small payloads also the extracted images are subjected to some error rates. In [9] W.Zhang proposed a method in which the order of encryption and vacating room, Reserving Room Before Encryption (RRBE), hence achieve real reversibility and separate data extraction.

In the present paper we propose a novel method for ensuring the security of confidential image by encrypting the image using Rabbit stream cipher. Rabbit is a stream cipher with new type of design. It provides a strong non-

linear mixing of the inner state between two iterations, which is opposed to almost all other designs currently available using either linear feedback shift register or S-boxes. In the proposed method we first encrypt the image using Rabbit stream cipher and then data is embedded into the image.

II. PROPOSEDWORK

Our method uses RRBE frame work for reversible data hiding. This method first reserve room for embedding secret data by embedding the LSB's of some pixels into other by any traditional RDH methods. Then the image is encrypted by Rabbit stream cipher and data is hidden in this image. Major steps in proposed system are

- Generating encrypted image using Rabbit stream cipher
- Data hiding in encrypted image
- Data extraction and image recovery

The process is illustrated in figure 1.

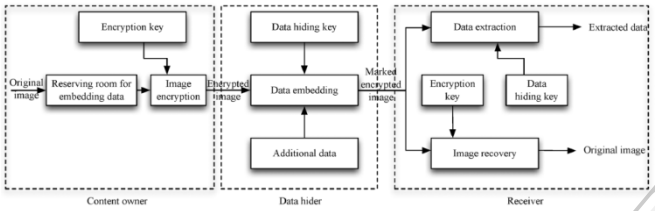

Fig 1: Proposed method

A. *Generating encrypted image using Rabbit stream cipher*

Encrypted image can be constructed in three steps: partitioning the image, self-reversible embedding and image encryption.

Partitioning the image

Since our method uses RRBE for RDH, we need to find a smoother area in the image, B on which apply standard RDH algorithm for better performance. Let the original image, I be a 8-bit grayscale image with size M×N. To partition the image content owner first divide the image into several overlapping blocks along the row, whose number is determined by the size of the message to be embedded, *l*. The block consist of *m* rows and *n* columns, where $m = \lceil l/N \rceil$ and *n*= *M*-*m*+1. For each block calculate the smoothness using the following function:

$$f = \sum_{u=2}^{m} \sum_{v=2}^{N-1} \left| I_{u,v} - \frac{I_{u-1,v}+I_{u+1,v}+I_{u,v-1}+I_{u,v+1}}{4} \right| . (1)$$

The block with higher *f* contain relatively complex textures, A; hence those blocks are put in front of the image and concatenate with Bthe fewer textured areas.

Self-reversible embedding

Here self-reversible embedding is used to embed the LSB-planes of A into B using traditional RDH algorithms. We can use the RDH scheme mentioned in [5] for better results.

Pixels in B part are first divided into two sets: white and black; white pixels with indices *i* and *j* satisfying the condition *(i + j)* mod2=0 and black pixels as*(i + j)* mod2=1. Then each white pixel in $B_{i,j}$, is estimated by the interpolation value obtained with the surrounding black pixels:

$$B'_{i,j} = w_1 B_{i-1,j} + w_2 B_{i+1,j} + w_3 B_{i,j+1} + w_4 B_{i,j-1} (2)$$

where$w_i$is the weight 1≤ i ≤ 4 and the estimating error $e_{i,j} = B_{i,j} - B'_{i,j}$; secret data can be embedded into the estimating error with histogram shift.

Message can be embedded into error sequence by bidirectional histogram shift. Divide the histogram of estimating error into two parts: left part and right part, and find the highest point in each part denoted by LM and RM also the zero point in each part LN and RN. For typical images LM=-1 and RM=0. Embedding messages into position with estimating error equal to RM is done by shifting all the error values from RM+1 to RN-1 towards right, so that we can represent the bit 0 with RM and 1 with RM+1. Embedding in left side is similar to this except the shifting is towards left.

There may be some overflow or underflow conditions occur when the natural boundary pixel changes from 255 to 256 and 0 to -1 also non boundary pixels are changed from 1 to 0 and 254 to 255. To avoid this, a boundary map is used. LSB of marginal area of the cover image is embedded with: LSB planes of *A*, parameters such as LN, LM, RN, RM and the boundary map. These play an important role in the recovery process.

Image Encryption

Previous step results in a rearranged self-embedded image X and this step the image X is converted into an encrypted image, E.

Rabbit is a synchronous stream cipher, with no crypto graphical weakness have been revealed until now. Takes 128 bit key and a 64 bit IV (Initial Vector) as input and generates for each iteration an output block of 128 pseudo-random bits, s_i from a combination of the internal state bits.

For example gray value $X_{i,j}$ ranging from 0 to 255 can be represented by 8 bits, $X_{i,j}(0), X_{i,j}(1), \ldots, X_{i,j}(7)$ such that

$$X_{i,j}(k) = \left\lfloor \frac{X_{i,j}}{2^k} \right\rfloor mod\ 2, \quad k = 0, 1, \ldots, 7. \quad (3)$$

The encrypted bits $E_{i,j}(k)$ can be calculated by XORing the bit values with the output obtained from each iteration of rabbit cipher, s_i

$$E_{i,j}(k) = X_{i,j}(k) \oplus s_i \quad (4)$$

To inform the data hider the number of rows and number of bit planes he can embedded information, we embed 10 bits of information into the LSBs of first 10 pixels in the encrypted version of A. Rabbit was designed to be faster than commonly used ciphers and to justify a key size of 128 bits for encrypting up to 2^{64}blocks of plain text. This means that for an attacker who does not know the key, it should not be possible to distinguish up to2^{64} blocks of the cipher output from the output of a truly random generator, using less steps than would be required for an exhaustive key search over 2^{128}keys.

B. Data Hiding in Encrypted Images

Data hider acquires the encrypted image E, and then embedded secret data into the image. The embedding process is carried out on the encrypted version of A, denoted by A_E. Since A_E is arranged on the top of E, it is effortless to read first 10 bit information in LSBs of first 10 encrypted pixels. Thus the data hider came to know about how many bit planes and how many pixels he can embed additional data, data hider substitute the available bit planes with additional data, m also set a label following m to indicate end position of embedding. Then encrypt m according to the data hiding key. The marked encrypted image is denoted by E'.

C. Extraction and Recovery

Extraction and decryption are purely independent process; the order of them implies two different practical applications.

1) Extracting data from encrypted images

Consider the case of a hospital database, in order to protect the clients privacy the database manager can only get into access to the data hiding key and manipulate data in encrypted domain. This situation demands data extraction before decryption.

With the data hiding key, an authorised one can simply extract and modify the data, since the data is embedded in the LSB planes of marked encrypted image also the data hider indicates the end of embedding.

2) Extracting data from decrypted images

This method founds application in cloud client server system. One send images into cloud, image is encrypted to protect the content. The server embed some additional information regarding the identity of the image owner and timestamp to manage the encrypted images. Now, an authorised user who has access to both the data hiding key and encryption key, can download the image and decrypt it. He obtain decrypted image still including the notation, which can be used to trace the source.

Marked decrypted image can be obtained by XORing it with the encryption key and image can be restored by extracting the additional data and apply the reverse process of embedding. With the help of boundary map and the parameters LM, LN, RM and RN the cover image can be perfectly recovered.

III. EXPERIMENTAL RESULTS

The proposed method is tested over standard images, which includes “Lena”, “Baboon”, “Peppers” etc. The figure 2 below shows the results of proposed system. Fig 2(a) shows the original image and (b) shows the encrypted version of the image using rabbit cipher, (c) shows the marked decrypted image and (d) the original recovered image.

Fig 2: (a) Original image (b) Encrypted image

(c) Marked decrypted image (d) Recovered image

IV. CONCLUSION

Reversible data hiding is a method which draws attention in now days because it founds application in many fields such as medical imaging, military etc. The propose scheme uses RRBE framework. Here the encryption process uses Rabbit stream cipher which provides better security than any other existing algorithms, also no crypto graphical weakness have been revealed until now with this cipher. That is this method achieves excellent secrecy of embedded data, real reversibility and separate data extraction.

REFERENCES

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

[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] A.S. Al-Fahoum, "Reversible data hiding using contrast enhancement approach", International Journal of Image Processing(IJIP),Vol (7):Issue (3):2013

[4] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.-Q. Shi, “Reversible watermarking algorithm using sorting and prediction,” IEEE Trans.Circuits Syst.

Video Technol., vol. 19, no. 7, pp. 989–999, Jul. 2009.

[5] L. Luo et al., "Reversible image watermarking using interpolation technique," IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187–193,Mar. 2010.

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

[7] 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. 199–202, Apr. 2012.

[8] X. Zhang, "Separable reversible data hiding in encrypted image," IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 826–832, Apr. 2012.

[9] Kede Ma, Weiming Zhang, Xianfeng Zhao, "Reversible Data Hiding In Encrypted Images By Reserving Room Before Encryption" IEEETransactions On Information Forensics And Security, Vol. 8, No. 3, March 2013

[10] Martin Boesgaard, MetteVesterager, Thomas Christensen, Erik Zenner, "The Stream Cipher Rabbit", CRYPTICO A/S, Fruebjergvej 3, 2100 Copenhagen, Denmark, info@cryptico.com