REVERSIBLE DATA HIDING IN ENCRYPTED IMAGE WITH PREDICTION ERROR CLUSTERING AND RANDOM PERMUTATION

Sruthi K V[1], Sreetha Sreedhar[2]

[1] *Student, Electronics and Communication Engineering, Malabar Institute of Technology, Kerala, India*
[2] *Assistant Professor, Electronics and Communication Engineering, Malabar Institute of Technology, Kerala, India*

ABSTRACT

A scheme is proposed to implement reversible data hiding (RDH) in encrypted images using prediction error clustering and random permutation. The original image is encrypted by the content owner using prediction error clustering and random permutation. We also demonstrate that an arithmetic coding-based approach can be exploited to efficiently compress the encrypted image. Then the data hider modifies the bits taken from the encrypted image to accommodate the secret data. On the receiver side, the secret data can be extracted if the receiver is available with the embedding key only. If the receiver has the encryption key only, original image can be recovered. If both the embedding and encryption keys are available at the receiver side, he/she can extract the secret data and recover the original image using logistic substitution decoding. The scheme is a good choice for secure image transmission.

Keyword: - *Index Terms—Image encryption, image recovery, image transmission, reversible data hiding.*

1. INTRODUCTION

With the rapid development of Internet technology, such media data as images, audios or videos are used more and more widely in human's daily life. This makes media data not only easy to be transmitted, but also easy to be copied and spread out. Thus, the legal issue rises that some media data should be protected against unauthorized users or operations.

In some cases, the content owner may not trust the service supplier, and needs to encrypt the data before uploading. Encryption is the process of encoding a message or information in such a way that only authorized parties can access it. Encryption does not of itself prevent interference, but denies the intelligible content to a would-be interceptor. In an encryption scheme, the intended information or message, referred to as plaintext, is encrypted using an encryption algorithm, generating cipher text that can only be read if decrypted. For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. It is in principle possible to decrypt the message without possessing the key, but, for a well-designed encryption scheme, considerable computational resources and skills are required. An authorized recipient can easily decrypt the message with the key provided by the originator to recipients but not to unauthorized users.

Some works have been done for data processing in encrypted domain, such as, compressing encrypted images [3],[5]adding a watermark into the encrypted images [4], and reversibly hiding data into the encrypted image [6-9]. The reversible data hiding in encrypted images allows the service supplier to embed additional messages such as image metadata, labels, notations or authentication information inside the encrypted images without accessing the original image. The original image along with the secret data is required to be recovered at the receiving side. Reversible data hiding is desirable. For example, to protect the patient's privacy, content of the medical image might be unavailable for the technician who embeds the information into the medical image.

In this paper, we propose a separable reversible data hiding method for encrypted images using prediction error clustering and random permutation. With the two different keys, the system is separable. The hidden data can

be completely extracted using the embedding key, and the original image can be approximately reconstructed using the encryption key. With both keys available, the hidden data can be completely extracted, and the original image perfectly recovered. The proposed method avoids the operations of room-reserving by the sender.

The rest of the paper is organized as follows. The proposed system is described in section II, which presents the procedure of image encryption, data embedding, data extraction and image recovery. Section III presents the experimental results. Section IV concludes the paper.

2. PROPOSED SYSTEM

Sketch of the proposed system is shown in Fig. 1. The system mainly consists of three phases: image encryption, data embedding and data retrieval/image recovery. In the first phase, the original image is encrypted to form the encrypted data using an encryption key. The encrypted data are sent to the data hider who embeds the secret data into the encrypted data using an embedding key in the second phase. There are three cases for the receiver to extract secret bits or recover the image. In the third phase, if the receiver has only the embedding key, he/she can extract the secret data independently. If he has only the encryption key, he can approximately recover the original image. If both the embedding key and the encryption keys are available for the receiver, the secret bits can be extracted and the original image can be perfectly recovered. Details of the procedure are as follows.

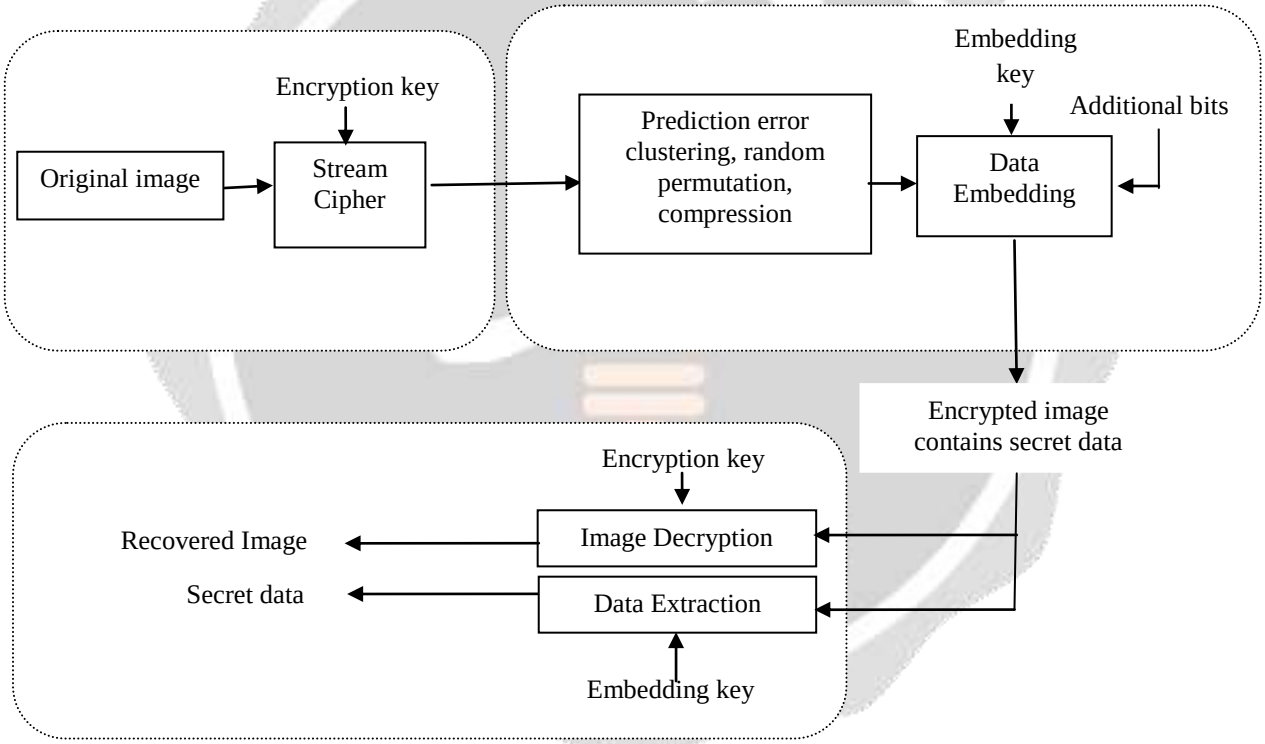

Fig-1: Proposed architecture

2.1 IMAGE ENCRYPTION

The password for encryption is chosen by the content owner as encryption key. The encryption algorithm should simultaneously consider the security and the ease of compressing the encrypted data. Here, the image encryption scheme is operated over the prediction error domain. For each pixel $I_{i,j}$ of the image I to be encrypted, a prediction $-I_{i,j}$ is first made by using an image predictor. In our work, the GAP is adopted. The prediction result $-I_{i,j}$ can be further refined to $\sim I_{i,j}$ through a context-adaptive, feedback mechanism. Consequently, the prediction error associated with $I_{i,j}$ can be computed by

$$e_{i,j} = I_{i,j} - (-I_{i,j}) \quad (1)$$

Each of which is sequentially mapped to a value between 0 to 255. The mapped prediction error is denoted by ~ $e_{i,j}$. The algorithmic procedure of performing the image encryption is then given as follows:

Step 1: Compute all the mapped prediction errors ~ $e_{i,j}$ of the whole image *I*.

Step 2: Divide all the prediction errors into *L* clusters **C***k* , for $0 \le k \le L-1$, where *k* is determined by (2), and each is formed by concatenating the mapped prediction errors in a raster-scan order.

$$k = \{k \mid q_k \le \Delta_{i,j} < q_{k+1}\} \quad (2)$$

Step 3: Reshape the prediction errors in each **C***k* into a 2-D block having four columns and [|**C***k* |/4] rows, where |**C***k*| denotes the number of prediction errors in **C***k* .

Step 4: Perform two key-driven cyclical shift operations to each resulting prediction error block, and read out the data in raster-scan order to obtain the permuted cluster **˜C***k*. **CS***k* and **RS***k* be the secret key vectors controlling the column and the row shift offsets for **C***k* . Here, **CS***k* and **RS***k* are obtained from the key stream generated by a stream cipher, which implies that the employed key vectors could be different, even for the same image encrypted at different sessions.

Step 5: The assembler concatenates all the permuted clusters **˜C***k*, for $0 \le k \le L-1$, and generates the final encrypted image.

$$I_e = \tilde{C}_0 \tilde{C}_1 ..... \tilde{C}_{L-1} \quad (3)$$

Step 6: Pass *Ie* to Charlie, together with the length of each cluster |**˜C***k* |, for $0 \le k \le L-2$. The values of |**˜C***k*| enables to divide *Ie* into *L* clusters correctly. In comparison with the file size of the encrypted data, the overhead induced by sending the length |**˜C***k* | is negligible.

2.2 DATA EMBEDDING AND LOSSLESS COMPRESSION

The compression of the encrypted file *Ie* needs to be performed in the encrypted domain. Assisted by the side information |**˜C***k* |, for $0 \le k \le L-2$, a de-assembler can be utilized to parse *Ie* into *L* segments **˜C**0, **˜C**1, . . . , **˜C***L*−1 in the exactly same way as that done at the encryption stage. An adaptive AC is then employed to losslessly encode each prediction error sequence **˜C***k* into a binary bit stream **B***k* .

After creating the encrypted form of original image, the content owner sends the encrypted image to the data-hider. An embedding key is chosen by the data hider for the secrecy of the data. The size of the data is calculated and it should be less than the encrypted image. The data is now embedded in the encrypted image by resizing the encrypted image. An assembler concatenates all **B***k* to produce the final compressed and encrypted bit stream **B**, namely,

$$B = B0B1 \ldots BL-1 \quad (4)$$

2.3 DECOMPRESSION, DATA EXTRACTION AND IMAGE RECOVERY

Upon receiving the compressed and encrypted bit stream **B**, aims to recover the original image *I* . According to the side information |**B***k* |, Bob divides **B** into *L* segments **B***k* , for $0 \le k \le L-1$, each of which is associated with a cluster of prediction errors. For each **B***k* , an adaptive arithmetic decoding can be applied to obtain the corresponding permuted prediction error sequence **˜C***k* . The corresponding de-permutation operation can be employed to get back the original **C***k* .

$$Ii, j = \tilde{} Ii, j + ei, j \quad (5)$$

As the predicted value ˜ *Ii, j* and the error energy estimator are both based on the causal surroundings, the decoder can get the exactly same prediction ˜ *Ii, j* . In addition, in the case of lossless compression, no distortion occurs on the prediction error *ei, j* , which implies ˆ *Ii, j* = *Ii, j* , i.e., error-free decoding is achieved. On the receiving end, the encrypted image contains data is available. If the encryption key is available, image recovery is done by the logistic substitution decoding. This is the reversal process of Section A. If the embedding key is available at the end, data extraction is done by subtracting the values from the processed image, which is the reverse of Section B.

3. EXPERIMENTAL RESULTS

Our proposed method is verified using standard gray images and color images, all sized (256×256). Fig. 2 illustrates a group of experimental results with Flower image. The original image in Fig. 2(a) is encrypted and shown in Fig. 2(b). Fig. 2(c) and 2(d) shows the resulting original image contains secret bits and encrypted image containing secret bits respectively. The hidden text (secret data) extracted is shown if Fig. 3.Similar to the encryption, decryption also performed in two stages. The output of decryption operation is illustrated in Fig. 4. Along with the hidden text, length of the hidden text, number of bits available for data hiding, number of ASCII characters hidden also calculated. Because no operation is performed before image encryption, the proposed method is a kind of VRAE. With the encryption key only, an approximate image can be reconstructed with high quality. The two aspects of security is considered here. Security of the image content and the security of the additional message. The content owner does not allow the service supplier to access the original image.

Fig-2(a): Original image

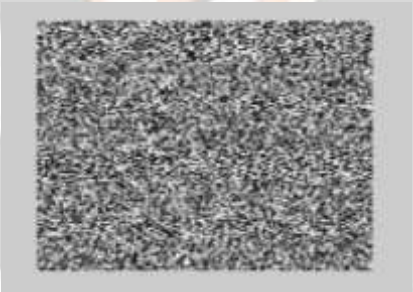

Fig-2 (b): Output of encryption operation

The data hider does not allow adversaries to crack the system for embedded message. The original image is encrypted with a stream cipher using an encryption key. For the data hider, the additional bits are also protected with the embedding key. Data extraction and image reconstruction is separable in this method. The three different cases at the receiving end is hence solved here; with the encryption key only, with the embedding key only and with the two keys together.

Fig-2(c): Original image contains secret data

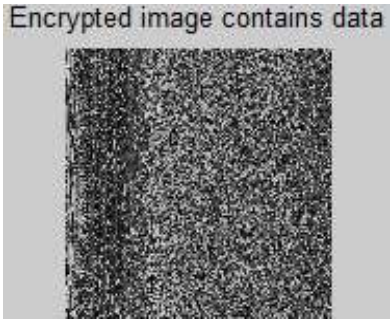

Fig-2(d): Encrypted image contains secret data
Fig-2: Image encryption

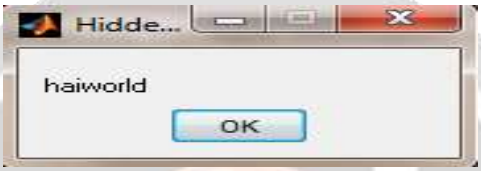

Fig-3: Extracted secret data

Fig-4: Output of decryption

4. CONCLUSIONS

This paper proposes a system of reversible data hiding in encrypted images using prediction error clustering and random permutation. An arithmetic coding-based approach can be exploited to efficiently compress the encrypted image. After encrypting the original image, encrypted image is modified for the additional secret data. On the receiver side, after decompression, all hidden data can be extracted with the embedding key only and the original image approximately recovered with high quality using the encryption key only. When both the embedding and encryption keys are available to the receiver, the hidden data can be extracted completely and the original image recovered perfectly. Because embedding operations are performed to the encrypted data, the data-hider cannot access the contents of the original image. That ensures security of the contents in data hiding.

5. ACKNOWLEDGEMENT

The authors would like to thank Dr.C.Sorna Chandra Devadass (Principal, MIT) and Asst. Prof Jacob Zachariah (Head of the Department, Electronics and Communication Engineering, MIT) for their immense encouragement and support. This work was done at Malabar Institute Of Technology, Kannur.

6. REFERENCES

[1]. Zhenxing Qian, Xinpeng Zhang, "Reversible data hiding in encrypted image with distributed source encoding,"

IEEE Trans. on Circuits and Systems for Video Technology, vol. 26, pp. 636 - 646, April 2016.
[2]. Jiantao Zhou, Xiaming Liu, Oscar C. Au, Yuan Yan Tang, "Designing an Efficient Image Encryption-Then-Compression System via Prediction Error Clustering and Random Permutation," IEEE Trans. On Information Forensics and Security, vol. 9, pp. 39-50, January 2014.
[3]. 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.
[4]. S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative encryption and watermarking in video compression," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 6, pp. 774–778, Jun. 2007.
[5]. 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.
[6]. W. Puech, M. Chaumont and O. Strauss, "A reversible data hiding method for encrypted images," Proc. SPIE 6819, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, 68191E, Feb. 26, 2008, doi:10.1117/12.766754.
[7]. X. Zhang, "Reversible data hiding in encrypted images," IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255–258, Apr. 2011.
[8]. 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.
[9]. K. Ma, W. Zhang, et al. "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption," IEEE Trans. Inf. Forensics Security, vol. 8, no. 3, 553-562, 2013.