International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

Flexible Histories Hiding in Encoded Imaginings

Chandana V[1], Prof Nithya E[2]

[1] *M. Tech Student , Department of Computer Science & Engineering, Dr. Ambedkar institute of Technology,Bangalore-560056*

[2] *Associate Professor, Department of Computer Science & Engineering, Dr. Ambedkar institute of Technology, Bangalore-560056,*

Abstract - *This paper proposes a method of reversible data hiding in encrypted images (RDH-EI) based on progressive recovery. Three parties are involved in the framework, including the content owner, the data-hider, and the recipient. The content owner encrypts the original image using a stream cipher algorithm and uploads ciphertext to the server. The data-hider on the server divides the encrypted image into three channels and respectively embeds different amount of additional bits into each one to generate a marked encrypted image. On the recipient side, additional message can be extracted from the marked encrypted image, and the original image can be recovered without any errors. While most of the traditional methods use one criterion to recover the whole image, we propose to do the recovery by a progressive mechanism. Rate-distortion of the proposed method outperforms state-of-the-art RDH-EI methods.*

Key Words: **Reversible data hiding, information hiding, encrypted image**

1. INTRODUCTION

Idea of reversible data hiding in encrypted images (RDH-EI) originates from reversible data hiding (RDH) in plaintext images [1][2]. It is feasible in the applications like cloud storage and medical systems. In cloud storage, a content owner can encrypt an image to preserve his/her privacy, and upload the encrypted data onto cloud [3][4]. On the cloud side, when managing huge amount of encrypted images, an administrator can embed additional messages (e.g., labels, time stamps, category information, etc.) into the ciphertext. This embedding not only saves the storage overhead, but also provides a convenient way of searching encrypted images. On the recipient side, when a user downloads the encrypted data containing additional messages from the server, he/she can losslessly recover the original images after decryption.

Some attempts on RDH-EI have been made. In [5], a content owner encrypts the original image using stream enciphering, and a data-hider embeds additional bits into ciphertext blocks by flipping three least significant bits (LSB) of half the pixels in each block. On recipient side, the ciphertext image is decrypted and two candidates for each block are generated by flipping again. This method was improved in [6] by exploiting spatial correlation between neighboring blocks to achieve a better embedding rate, which was further improved in [7] using a full embedding strategy to achieve larger embedding rate. Secure RDH-EI can be ensured by public key modulation [8]. RDH-EI can also be realized in encrypted JPEG bitstreams by slightly modifying the encrypted data [9]. One problem in [5]-[9] is that data extraction can only be done after image decryption. Separable RDH-EI was proposed to resolve this problem, allowing one to extract hidden data directly from the encrypted image. In [10], the data-hider permutes and divides the encrypted pixels into segments, and compresses some LSB layers of each segment to fewer bits using a predefined matrix. The recipient extracts additional message from the marked encrypted image. After decryption, the original LSBs are recovered by comparing the estimated bits with the compressed. If higher bitplanes are used [11-13], better embedding rate can be achieved. Some RDH-EI methods were also proposed to enlarge embedding rates by vacating embedding room before encryption, e.g., [14] and [15].

2. PROPOSED SYSTEM

The proposed system is illustrated in Fig. 1, including three parties: the content owner, the data-hider, and the recipient. The content owner encrypts the original image and uploads the encrypted image onto a remote server. The data-hider divides the encrypted image into three sets and embeds message into each set to generate a marked encrypted image. The recipient extracts message using an extraction key. Approximate image with good quality can be obtained by decryption if the receiver has decryption key. When both keys are available, the original image can be losslessly recovered by progressive recovery.

On the recipient side, additional messages can be extracted if the receiver has the key KEMB. The marked encrypted image is separated to the Square set, the Triangle set and the Circle set again. With the embedding key, the recipient permutes pixels in each set independently, and divides the permuted sets into segments, each of which contains Li (i=1,2,3) pixels. Collect the bits of three LSB-layers in each segment and reconstruct the groups B'i(ki)=[Ci(ki, 1), Ci(ki, 2), …, Ci(ki, 3Li–P), Ai(ki, 1), Ai(ki, 2), …, Ai(ki, P)]T (ki∈[1, Ri]). From

each group, the additional bits [Ai(ki, 2), ..., Ai(ki, P)]T are extracted.

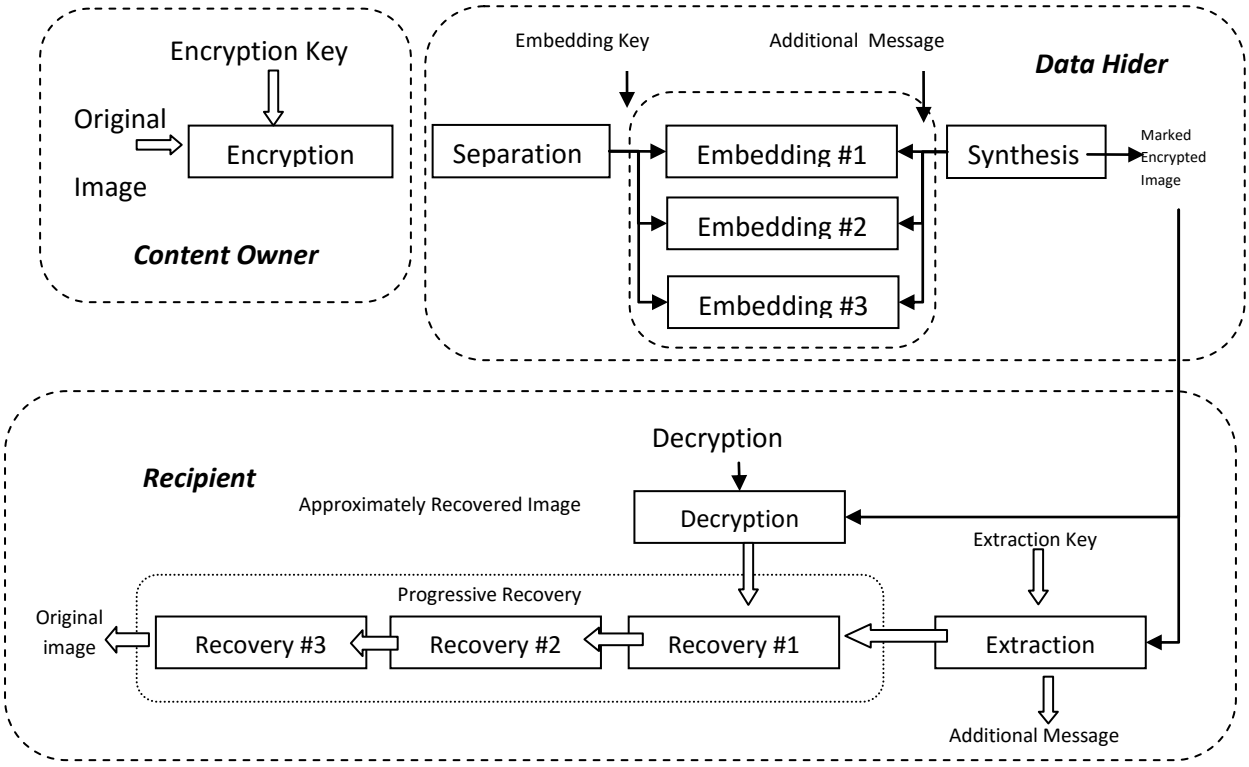


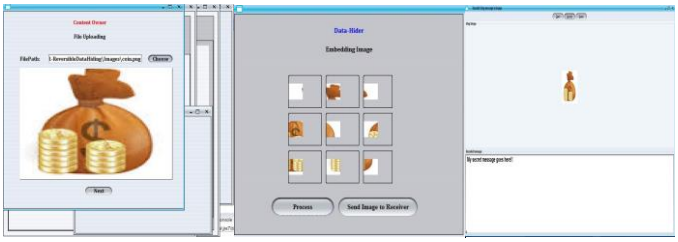
Fig. 1 Framework of the Proposed Method

If the recipient has only the key KENC, he/she decrypts the marked encrypted image using (2) to construct an approximate image. Since we limit the distortion to three LSB-layers, the directly decrypted image still preserves good quality.

In case both KENC and KEMB are available, the recipient can recover the original image. With KEMB, compressed bits Ci(ki)=[Ci(ki, 1), Ci(ki, 2), ..., Ci(ki, 3Li–P)]T are extracted from each group B'i(ki) (ki∈[1, Ri]). The recipient generates the matrices G1, G2 and G3 again, and accordingly constructs the parity-check matrices.

4. EXPERIMENTAL RESULTS

A group of experimental results are shown in Fig. 4, in which (a) is the original images sized 512×512. Fixed parameters P=5 and {L1=150, L2=125, L3=100} are used to hide 11350 bits (0.043 bpp) additional message into the encrypted image. Fig. 4(b) shows the marked encrypted images. Fig. 4(c) shows the approximate image by directly deciphering Fig. 4(b). The directly decrypted image preserves good quality, PSNR of which is equal to 38.1dB. From the marked encrypted image, additional bits can be extracted without any error. The original image can also be losslessly recovered to the same one as (a).



(a) (b) (c)

Fig. 2 Experimental Results for "Lena", (a) shows the original image, (b) the marked encrypted image, (c) the directly decrypted image, and (d) the losslessly recovered image.

The proposed method is compared with the separable RDH-EI method in [10]. Both methods embed message into three LSB-layers of the encrypted image. Maximal embedding rates Re of the different images are shown in Table I, where Re stands for the embedding rate (bit-per-pixel, bpp). We use a parameter P=5 in the proposed method, equaling to S=5 in [10]. The parameter L used in [10] means the segment length. Other parameters also are listed in Table I. The results show that the proposed method achieves a better embedding rate. We also compare the average of maximal embedding rates in the proposed method with other methods. Algorithms in

[5]~[7] and [10] are implemented in 50 natural images sized 512×512 with various features. All images can be downloaded from http://pan.baidu.com/s/1gdjPID1. When calculating maximal embedding rates, we use P=5 and ensure the lossless recovery. All methods use three LSB-layers of the encrypted image for data hiding. Average rates are shown in Table II, indicating that the proposed method outperforms previous ones.

Some methods may have higher embedding rates than the proposed. In [11], additional message is embedded into the sixth or higher bitplanes to achieve a high embedding rate, e.g., 0.0625 bpp for Lena. MSB is also used to carry additional messages [12], resulting in an embedding rate as high as 0.3 bpp. However, high rates in these methods are achieved at the expenses of serious distortions. When directly decrypting the marked encrypted image, the generated image has poor quality. Embedding rate in [15] is more than 1.0 bpp in most cases, higher than the other methods. However, the content owner has to reserve room before encryption, and then the data-hider may embed message into the specified room in the encrypted image, which is equivalent to RDH in plaintext images.

A. Data Flow Diagram

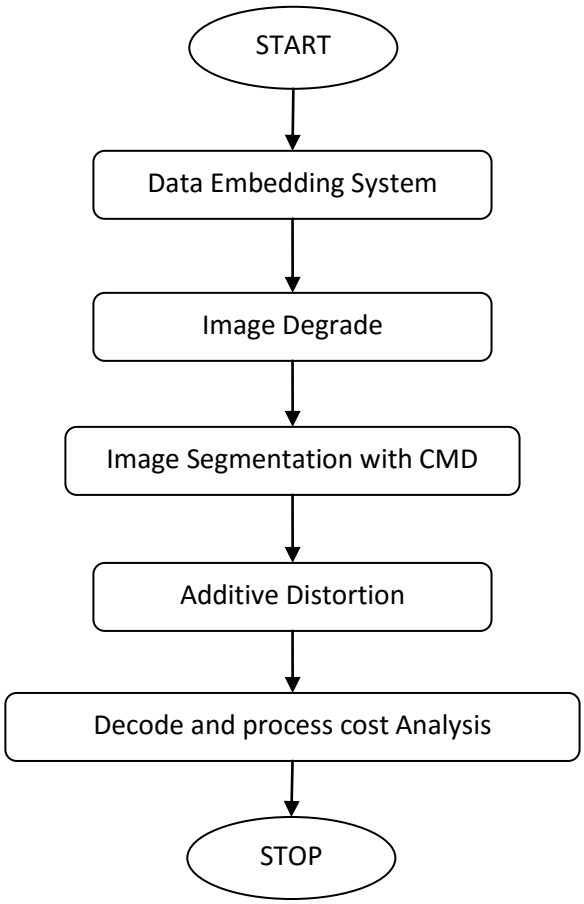


Fig 3 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

Data Embedding System: STEGANOGRAPHY aims to hide secret messages into innocuous digital media without drawing suspicion. It faces challenges posed by modern steganalysis. This intends to detect the traces of data hiding. We select steganography image and Then Read hidden Data from the Source. The Embedding System has activated after Start the Image and data read from source. In this module we have Embed the data's inside of image pixel. The data will be hidden, then the embedded image has stored in server.

Image Degrade by data hider: While complete the embedding process, next start the image Degrade Process. Here, we retrieve the embedded image from server then apply decomposing function to the selective image.

Image segmentation With CMD: An image is decomposed into several sub-lattices, where pixels within the same sub-lattice are separated by a distance larger than the support width of the potential function and apply the CMD function Cost assignment and data embedding are performed in each sub-lattice sequentially.

Additive Distortion function: The distortion function quantifies the effect of modifying an input cover object to the corresponding output stego object. A distortion function is considered additive when it is expressed as a sum of embedding costs for individual pixels which element-wisely evaluate the effect of respective embedding modification.

Decode and Process Cost analysis by recipient: An end of the steganography Process, we decode the message using steganography image. After embedding for a sub-lattice, the costs of pixels in the remaining sub-lattices are updated. And find the performance analysis in cost and secure level.

4. CONCLUSIONS

Based on our previous work, a new RDH-EI protocol for three parties is proposed in this paper. Main improvement is extending the traditional recovery to the progressive based recovery. The progressive recovery based RDH-EI provides a better prediction way for estimating the LSB-layers of the original image using three rounds, which outperforms state-of-the-art RDH-EI methods. Since RDH-EI is equivalent to a rate-distortion problem, capability of the method should be evaluated by both the distortion and the embedding rate. For a fair comparison, this paper limits the distortion to three LSB-layers, and accordingly improves the embedding rate.

REFERENCES

[1] X. Hu, W. Zhang, X. Li, and N. Yu, Minimum Rate Prediction and Optimized Histograms Modification for Reversible Data Hiding, IEEE Transactions on Information Forensics and Security, 10(3): 653-664, 2015

[2] X. Li, W. Zhang, B. Ou, and B. Yang. A brief review on reversible data hiding: current techniques and future prospects, IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP), 426-430, 2014
[3] H. Wang, W. Zhang, and N. Yu, Protecting Patient Confidential Information based on ECG Reversible data hiding, Multimedia Tools and Applications, doi:10.1007/s11042-015-2706-2,2015
[4] Z. Fu, X. Sun, Q. Liu, et al. Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing, IEICE Transactions on Communications, 98(1): 190-200, 2015
[5] X. Zhang, Reversible data hiding in encrypted images, IEEE Signal Processing Letters, 18(4): 255–258, 2011
[6] W. Hong, T. Chen, and H. Wu, An improved reversible data hiding in encrypted images using side match, IEEE Signal Processing Letters, 19(4): 199–202, 2012
[7] M. Li, D. Xiao, A. Kulsoom, and Y. Zhang, Improved reversible data hiding for encrypted images using full embedding strategy, Electronic Letters, 51(9): 690-691, 2015
[8] J. Zhou, W. Sun, L. Dong, et al. Secure reversible image data hiding over encrypted domain via key modulation, IEEE Transactions on Circuits and Systems for Video Technology, 26(3): 441-452, 2016
[9] Z. Qian, X. Zhang, and S. Wang, Reversible data hiding in encrypted JPEG bitstream, IEEE Transactions on Multimedia, 16(5): 1486-1491, 2014
[10] X. Zhang, Separable reversible data hiding in encrypted image, IEEE Transactions Information Forensics and Security, 7(2): 826–832, 2012
[11] Wu X, Sun W. High-capacity reversible data hiding in encrypted images by prediction error, Signal processing, 104: 387-400, 2014
[12] Z. Qian, and X. Zhang, Reversible data hiding in encrypted image by distributed encoding, IEEE Transactions on Circuits and Systems for Video Technology, 26(4): 636-646, 2016
[13] X. Zhang, Z. Qian, G. Feng and Y. Ren, Efficient reversible data hiding in encrypted images, Journal of Visual Communication and Image Representation, 25(2): 322-328, 2014
[14] K. Ma, W. Zhang, et al. Reversible data hiding in encrypted images by reserving room before encryption, IEEE Transactions Information Forensics and Security, 8(3): 553-562, 2013
[15] X. Cao, L. Du, X. Wei, et al. High capacity reversible data hiding in encrypted images by patch-level sparse representation, IEEE Transactions on Cybernetics, 46(5): 1132-1143, 2016.