ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Robust Reversible Data Hiding in Image Encrypted Domain via Key Modulation

Kantheti John Babu
Department of Electronics & Communication Engineering,
DNR College of Engineering & Technology,
Balusumudi, Bhimavaram, West Godavari District,
Andhra Pradesh 534202, India.

Kopalli Venkanna Naidu
Department of Electronics & Communication Engineering,
DNR College of Engineering & Technology,
Balusumudi, Bhimavaram, West Godavari District,
Andhra Pradesh 534202, India.

ABSTRACT:

This project proposes a novel reversible image data hiding (RIDH) scheme over encrypted domain. The data embedding is achieved through a public key modulation mechanism, in which access to the secret encryption key is not needed. At the decoder side, a powerful two-class SVM classifier is designed to distinguish encrypted and non-encrypted image patches, allowing us to jointly decode the embedded message and the original image signal. Compared with the state-of-the-arts, the proposed approach provides higher embedding capacity, and is able to perfectly reconstruct the original image as well as the embedded message. Extensive experimental results are provided to validate the superior performance of our scheme.

Keywords: — Data hiding, Reversible data hiding, Image encryption, Image decryption.

1. INTRODUCTION

Reversible Image Data Hiding (RIDH) is a special category of data hiding technique, which ensures perfect reconstruction of the cover image upon the extraction of the embedded message. The reversibility makes such image data hiding approach particularly attractive in the critical scenarios, e.g., military and remote sensing, medical images sharing, law forensics and copyright authentication, where high fidelity of the reconstructed cover image is required. The majority of the existing RIDH algorithms [1] are designed over the plaintext domain, namely, the message bits are embedded into the original, un-encrypted images. The early works mainly utilized the lossless compression algorithm to compress certain image features, in order to vacate room for message embedding. However, the embedding capacity of this type of method is rather limited and the incurred distortion on the watermarked image is severe. Histogram shifting (HS)-based technique, initially designed by Ni et al. Is another class of approach achieving better embedding performance through shifting the histogram of some image features The latest difference expansion (DE)-based schemes and the improved prediction error expansion (PEE)-based strategies [2] were shown to be able to offer the state-of-the-art capacity distortion performance. Recently, the research on signal processing over encrypted domain has gained increasing attention, primarily driven by the needs from Cloud computing platforms and various privacy preserving applications [3]. This has triggered the investigation of embedding additional data in the encrypted images in a reversible fashion. In many practical scenarios, e.g., secure remote sensing and Cloud computing, the parties who process the image data are un-trusted. To protect the privacy and security, all images will be encrypted before being forwarded to an un-trusted third party for further processing. For instance, in secure remote sensing, the satellite images, upon being captured by on-board cameras, are encrypted and then sent to the base station(s), as illustrated in Fig. 1. After receiving the encrypted images, the base station

Cite this article as: Kantheti John Babu & Kopalli Venkanna Naidu, "A Robust Reversible Data Hiding in Image Encrypted Domain via Key Modulation", International Journal & Magazine of Engineering, Technology, Management and Research, Volume 5 Issue 6, 2018, Page 282-288.

embeds a confidential message, e.g., base station ID, location information, time of arrival (TOA), local temperature, wind speed, etc., into the encrypted images. Eventually, the encrypted image carrying the additional message is transmitted over a public network to a data center for further investigation and storage. For security reasons, any base station has no privilege of accessing the secret encryption key K pre-negotiated between the satellite and the data center. This implies that the message embedding operations have to be conducted entirely over the encrypted domain. In addition, similar to the case of Cloud computing, it is practically very costly to implement a reliable key management system (KMS) in such multi-party environment over insecure public networks [4], due to the differences in ownership and control of underlying infrastructures on which the KMS and the protected resources are located. It is therefore much desired if secure data hiding could be achieved without an additional secret data hiding keyshared between the base station and the data center. Also, we appreciate simple embedding algorithm as the base station usually is constrained by limited computing capabilities and/or power. Finally, the data center, which has abundant computing resources, extracts the embedded message and recovers the original image by using the encryption key K.

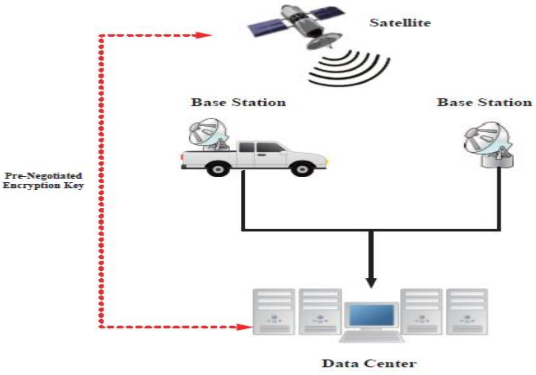


Fig. 1. Image data hiding in the scenario of secure remote sensing

In this thesis, we propose an encrypted-domain RIDH scheme by specifically taking the above-mentioned design preferences into consideration [5]. The proposed technique embeds message through a public key modulation mechanism, and performs data extraction by exploiting the statistical distinguish ability of encrypted and non-encrypted image blocks. Since the decoding of the message bits and the original image is tied together, our proposed technique belongs to the category of non-separable RIDH solutions. Compared with the state-of-the-arts, the proposed approach provides higher embedding capacity, and is able to achieve perfect reconstruction of the original image as well as the embedded message bits. Extensive experimental results on 100 test images validate the superior performance of our scheme [6]. The rest of this paper is organized as follows. Overviews the related work on RIDH over the encrypted domain. The proposed data hiding technique in encrypted images. The schematic diagram of the proposed message embedding algorithm over encrypted domain is depicted in following figure. In this work, we do not consider the case of embedding multiple watermarks for one single block, meaning that each block is processed once at most. For simplicity, we assume that the number of message bits to be embedded is n · A, where A ≤ B and B is the number of blocks within the image. The steps of performing the message embedding are summarized as follows:

Step 1: Initialize block index i = 1.

Step 2: Extract n bits of message to be embedded, denoted by Wi.

Step 3: Find the public key Q[Wi]d associated with Wi, where the index [Wi]d is the decimal representation of Wi. For instance, when n = 3 and Wi = 010, the corresponding public key is Q2. Step 4: Embed the length-n message bits Wi into the ith block via

$$[[f]]_i^w = [[f]]_i \oplus Q_{[W_i]_d}$$

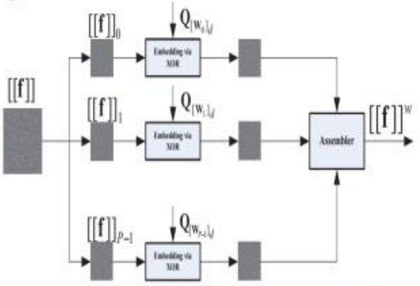


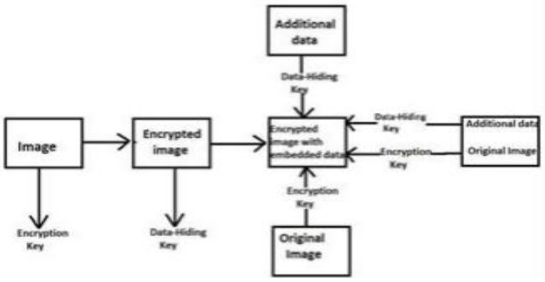
Fig. 2.schematic diagram of data hiding over encrypted domain

Step 5: Increment i = i + 1 and repeat Steps 2-4 until all the message bits are inserted. The watermark length parameter A needs to be transmitted alone with the embedded message bits. There are many ways to solve this problem. For instance, we can reserve some blocksto embed A. Or, we can append an end-of-file symbol to the message to be embedded, such that the decoder can implicitly determine A. Both strategies can be readily implemented in practice with negligible affect to the actual embedding rate. For the sake of simpler presentation, we exclude the discussion of embedding A in the sequel. From the above steps, it can be seen that the message embedding is performed without the aid of a secret data hiding key. As will be proved, high level of embedding security can still be guaranteed, thanks to the protection offered by the encryption key K. In addition, the computations involved in message embedding are rather small (simple XOR operations), and all the block-by-block processing can be readily made parallel, achieving high-throughput. It is emphasized that the possibility of eliminating the data hiding key is not unique to our proposed method, but rather arguably applicable for all non-separable RIDH schemes over encrypted domain. For instance, the existing non-separable RIDH schemes, upon trivial modifications, can still ensure embedding security even if the data hiding key is eliminated. If we fix the way of partitioning a block into S0 and S1 (namely, do not use data hiding key to randomize the block partitioning), then an attacker still cannot compute the fluctuation function. So as to decode the embedded message [7]. This is because an attacker does not access to the secret encryption key K. In other words, the protection mechanism in the encrypted domain can be naturally extended to provide security for message embedding, eliminating the necessity of introducing an extra data hiding key. This could lead to significant reduction of the computational cost and potential risk of building up a secure KMS, which has been proved to be very challenging in the multi-party environment. Though the possibility of removing the data hiding key holds for all non-separable RIDH schemes over encrypted domain, it has never been pointed out in the existing work. It can be witnessed by the fact that all the existing RIDH schemes, including separable and non-separable ones, involve a data hiding key that has to be shared and managed between the data hider and the recipient. In addition to identifying this property, we, in the following Section VI, will exploit the message in distinguishability to prove that the removal of data hiding key will not hurt the embedding security. Before presenting the data extraction and image decryption methods, let us first investigate the features that can be used to discriminate encrypted and non-encrypted image blocks. The classifier designed according to these features will be shown to be crucial in the proposed joint data extraction and image decryption approach.

EXISTING TECHNIQUES

Reversible Data Hiding

Data hiding is the way of hiding information into a cover media. It requires two set of data that are embedded data and set of cover media data. In some case cover media distorted due to perform hiding operation but this type of changes are not acceptable by some applications such as medical imagery, military imagery and law-forensic etc. so that a novel method become more popular among the researches i.e. known as Reversible data hiding (RDH).Itis the technique that perform lossless embedding operation and recover the origin after the extraction. If cover medium distorted permanently when hidden message have been removed. Original Image encrypted into image encryption by using the encryption-key algorithm at the side of image owner. After that in the data hider module [8] we can embed some additional data with the use of data hiding key, finally gets the encrypted image that containing additional data and that image require to decryption at the receiver side. This concept describe by following figure.



Reversible data hiding techniques can be generally classified into two frameworks

1. Vacate room after encryption
2. Reserve room before encryption

In the first framework, vacate room after encryption (VRAE) [3-5], a content owner first encrypts the original image using a standard cipher with an encryption key. After producing the encrypted image, the content owner hands over it to a data hider (e.g., a database manager) and the data hider can embed some auxiliary data into the encrypted image by losslessly vacating some room according to a data hiding key. Then a receiver, maybe the content owner himself or an authorized third party can extract the embedded data with the data hiding key and further recover the original image from the encrypted version according to the encryption key.

In the second framework, reserve room before encryption (RRBE), the content owner first reserve enough space onoriginal image and then converts the image into its encrypted version with the encryption key. Now, the data embed ding process in encrypted images is inherently reversible for the data hider only needs to accommodate data into the spare space previous emptied out. The data extraction and image recovery are identical to that of Framework VRAE. Obviously, standard RDH algorithms are the ideal operator for reserving room before encryption and can be easily applied to Framework RRBE to achieve better performance customary idea [7] is followed in which the redundant image content is losslessly compressed and then encrypts it with respect to protecting privacy.

PROPOSED METHOD

Problem and some space created at the time of embedding. So this is also time consuming process. After extracting the data we cannot achieve the originality. Some distortion exists in the system. So our aim is to remove this type of distortion form the system. There are lots of problems in the existing system. So objective is to recover the problems in future, which are described below:

- The extracted data may contain error.
- Time-consuming process.
- Availability of memory space.
- The key contents are not store of original image.

These entire problem recovered by using the concept of ―Reserving Room Before encryption (RRBE)‖. With the use of VRAE concept with cannot achieve original data after encryption. So that new concept used for achieve this property i.e. RRBE. The proposed system extracted data losslessly after encryption.

The proposed scheme is made up of image encryption, data embedding and dataextraction/image-recovery phases. The content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. Then, the data-hider compresses the least significant bits (LSB) [8] of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data. At the receiver side, the data embedded in the created space can be easily retrieved from the encrypted image containing additional data according to the datahiding key. Since the data embedding only affects the LSB, a decryption with the encryption key can result in an image similar to the original version. 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. Fig. shows the three cases at the receiver side.

Here propose a new reversible data embedding technique, which can embed a large amount of data (5–80 kb for a 512 x 512 x 8 Grayscale image) while keeping a very high visual quality for all natural images, specifically, the PSNR of the marked image versus the original image is guaranteed to be higher.

IMAGE ENCRYPTION

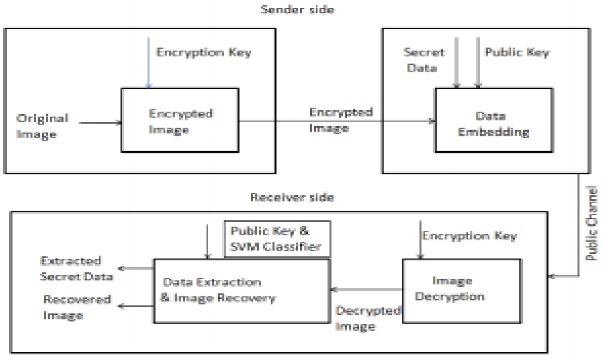
Assume the original image is in uncompressed format and each pixel with gray value falling into [0, 255] is represented by 8 bits.

$$B_{i,j,k} = b_{i,j,k} \oplus r_{i,j,k}$$

whereri,j,k are determined by an encryption key using a standard stream cipher. Then, B,i,j,k are concatenated orderly as the encrypted data. A number of secure stream cipher methods can be used here to ensure that anyone without the encryption key, such as a potential attacker or the data hider, cannot obtain any information about original content from the encrypted data.

DATA HIDING IN ENCRYPTED IMAGE

Once the data hider acquires the encrypted image, he can embed some data into it, although he does not get access to the original image. The embedding process starts with locating the encrypted version of A, denoted by . Since has been rearranged to the top of E, it is effortless for the datahider to read 10 bits information in LSBs of first 10 encrypted pixels. After knowing how many bit-planes and rows of pixels he can modify, the data hider simply adopts LSB replacement to substitute the available bit-planes with additional data. Finally, the data hider sets a label following to point out the end position of embedding process and further encrypts according to the data hiding key to formulate marked encrypted image denoted by E'. Anyone who does not possess the data hiding key could not extract the additional data.



In encryption phase, the exclusive-or results of the original bits and pseudo-random bits are calculated.

selectsNp encrypted pixels that will be used to carry the parameters for data hiding. Here, Np is a small positive integer, for example, Np=20. The other (N-Np) encrypted pixels are pseudo-randomly permuted and divided into a number of groups, each of which contains L pixels. The permutation way is also determined by the data-hiding key. For each pixel-group, collect the M least significant bits of the L pixels, and denote 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 generates a matrix G sized (M*L – S) * M*L, which is composed of two parts. The left part is the identity matrix and the right part is pseudo-random binary matrix derived from the data-hiding key.

For each group, which is product with the G matrix to form a matrix of size (M * L-S). Which has a sparse bits of size S, in which the data is embedded and arrange the pixels into the original form and permutated to form a original image.

DATA EXTRACTION AND IMAGE RECOVERY

When having an encrypted image containing embedded data, a receiver firstly generates ri,j,k according to the encryption key, and calculates the exclusive-or of the received data and ri,j,k to decrypt the image. We denote the decrypted bits as b1i,j,k . Clearly, the original five most significant bits (MSB) [7] are retrieved correctly. For a certain pixel, if the embedded bit in the block including the pixel is zero and the pixel belongs to S1, or the embedded bit is 1 and the pixel belongs to S0 , the data-hiding does not affect any encrypted bits of the pixel. So, the three decrypted LSB must be same as the original LSB, implying that the decrypted gray value of the pixel is correct. On the other hand, if the embedded bit in the pixel's block is 0 and the pixel belongs to S0, or the embedded bit is 1 and the pixel belongs to S1 , the decrypted LSB.

$$\begin{aligned} b'_{i,j,k} &= r_{i,j,k} \oplus B'_{i,j,k} \\ &= r_{i,j,k} \oplus \overline{B_{i,j,k}} \\ &= r_{i,j,k} \oplus \overline{b_{i,j,k} \oplus r_{i,j,k}} \\ &= \overline{b_{i,j,k}}, \qquad k = 0,\ 1,\ 2. \end{aligned}$$

That means the three decrypted LSB must be different from the original LSB. In this case:

$$b'_{i,j,k} + b_{i,j,k} = 1,$$

Extracting Data from Decrypted Images: In Case 1, both embedding and extraction of the data are manipulated in encrypted domain. On the other hand, there is a different situation that the user wants to decrypt the image first and extracts the data from the decrypted image when it is needed. The following example is anapplication for such scenario. Assume Alice outsourced her images to a cloud server, and the images are encrypted to protect their contents. Into the encrypted images, the cloud server marks the images by embedding some notation, including the identity of the images' owner, the identity of the cloud server and time stamps, to manage the encrypted images. Note that the cloud server has no right to do any permanent damage to the images. Now an authorized user, Bob who has been shared the encryption key and the data hiding key, downloaded and decrypted the images. Bob hoped to get marked decrypted images, i.e., decrypted images still including the notation, which can be used to trace the source and history of the data. The order of image decryption before/without data extraction is perfectly suitable for this case.

If the receiver has both the data-hiding, he may aim to extract the embedded data According to the data-hiding key, the values of M,L and S, the original LSB of the Np selected encrypted pixels, and the (N-Np) * S/L - Np additional bits can be extracted from the encrypted image containing embedded data. By putting the Np LSB into their original positions, the encrypted data of the Np selected pixels are retrieved, and their original gray values can be correctly decrypted using the encryption keys. In the following, we will recover the original gray values of the other (N-Np) pixels.

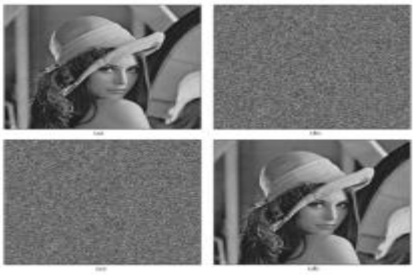


Fig. 3. (a) Original Lena, (b) its encrypted version, (c) encrypted image containing embedded data with embedding rate 0.017 bpp, and (d) directly decrypted version with PSNR 39.0 dB.

This project proposes a novel scheme for separable reversible data hiding in encrypted image. In the proposed scheme, the original image is encrypted using an encryption key and the additional data are embedded into the encrypted image using a data-hiding key. With an encrypted image containing additional data, if the receiver has only the data-hiding key, he can extract the additional data though he does not know the image content. If he has only the encryption key, he can decrypt the received data to obtain an image similar to the original one, but cannot extract the embedded additional data. If the receiver has both the datahiding key and the encryption key, he can extract the additional data and recover the original image without any error when the amount of additional data is not too large.

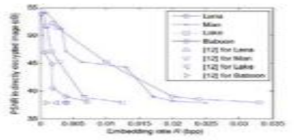


Fig. 4. Rate-PSNR comparison between the proposed scheme and the method in [12].

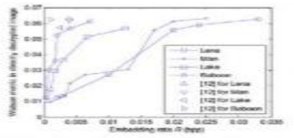


Fig. 5. Rate-Watson metric comparison between the proposed scheme and the method in [12].

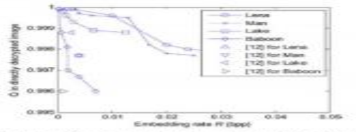


Fig. 6. Rate-Q comparison between the proposed scheme and the method in [12].

Table-1: Comparative analysis between proposed system and existing systems.

	Proposed system		RDH in encrypted images		RDH using side match	
Block Size	Capacity	Accuracy	Capacity	Accuracy	Capacity	Accuracy
8*8	8192 bits	100%	4096 bits	89.44%	4096 bits	92.04%
7*7	10,658bits	99.8%	5329 bits	87.20%	5329 bits	90.65%
6*7	12,410bits	99.5%	6205 bits	84.19%	6205 bits	88.88%

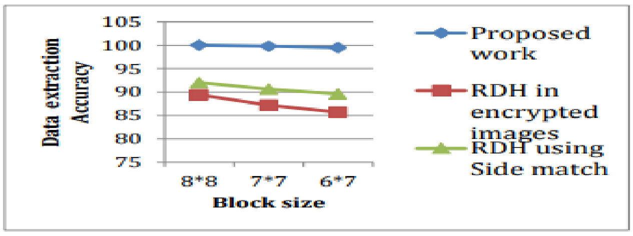


Chart-1: Comparative analysis with existing systems

CONCLUSION

In this project, a secure Reversible image data-hiding scheme is designed over an encrypted domain. This scheme can be used in various scenarios like military, medical data sharing, authentication and many more. A powerful stream cipher algorithm AES-CTR is used to encrypt and decrypt the images, which forms the encrypted domain. To overcome the disadvantages of the existing methods a public key modulation mechanism is used to embed the data without accessing the secret encryption key. It also eliminates the need of using an extra data-hiding key. Data embedding is carried out via simple XOR operations at the sender end. At the receiverend, a powerful two-class SVM classifier is used to discriminate randomized and original image patches. This enables simultaneous decoding of the embedded message and the cover image to perfection. Experimentation was performed to validate the embedding performance and embedding capacity of the proposed RIDH method over encrypted domain to obtain good results.

FUTURE ENHANCEMENT

Performed extensive experiments to validate the superior embedding performance of our proposed RIDH method over encrypted domain. And also would like to point out that the complexity of performing the joint decryption and data extraction may not be crucial in many applications, e.g. secure remote sensing, where the recipient has abundant computing resources.

REFERENCES

[1]. Alistair McMonnies, “Digital Image Processing 3rd Edition”. Pearson Education, and ISBN: 81-297-0649-0, First Indian Reprint 2004.

[2]. PoonamYadav, “Digital Image Processing” Edition 2002, Tata McGraw-Hill, Publishing Company Limited, New Delhi.

[3]. Roger S. Pressman “Software Engineering” Tata McGraw-Hill, Publishing Company Limited (1987).

[4]. Munesh Chandra Trivedi, “Digital Image Processing”, Second Edition, Pearson Education Asia, ISBN: 981-4035-20-3 (2000).

[5]. Tamal Bose, “Digital Signal and Image Processing (WSE series)”, Microsoft Press.

[6]. B. Chanda&D.DuttaMajumder, “Digital Image Processing And Analysis”, Prentice Hall.

[7] Zhaoxia Yin, Andrew Abel, Xinpeng Zhang and Bin Luo, "Reversible data hiding in encrypted image based on block histogram shifting," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, 2016, pp. 2129- 2133.

[8] M. Chandramouli, R. Iorga, and S. Chokhani, “Publication citation: Cryptographic key management issues & challenges in cloud services,” US Dept. Commerce, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 7956, 2013, pp. 1–31.