Reversible Image Data Hiding over Encrypted Domain using Key Modulation

Dr. M Nivas
Head of the Department
Department of Computer Applications
KVM CE & IT Cherthala ,Kerala, India

Aswin Raju
Student
Department of Computer Applications
KVM CE & IT Cherthala ,Kerala, India

Abhishek Lal
Student
Department of Computer Applications
KVM CE & IT Cherthala ,Kerala, India

Abstract

This work 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. The encrypted data is embed into the image using Bitwise XOR operation. Then the image after embedding divides into block by Image block splitting method. 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.

Keywords: Reversible image data hiding (RIDH), signal processing over encrypted domain, feature extraction, SVM Classifier, Bitwise XOR operation, Image block splitting

I. 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. The majority of the existing RIDH algorithms are designed over the plaintext domain, namely, the message bits are embedded into the original unencrypted images.

The early works mainly utilized the lossless compression algorithm to compress certain image features, 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. In recent years, signal processing in the encrypted domain has attracted considerable research interest. As an effective and popular 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.

However, in some scenarios that a content owner does not trust the processing service provider, the ability to manipulate the encrypted data when keeping the plain content unrevealed is desired. For instance, when the secret data to be transmitted are encrypted, a channel provider without any knowledge of the cryptographic key may tend to compress the encrypted data due to the limited channel resource. While an encrypted binary image can be compressed with a lossless manner by finding the syndromes of low-density parity-check codes, a lossless compression method for encrypted gray image using progressive decomposition and rate-compatible punctured turbo codes is developed in. With the lossy compression method presented in, an encrypted gray image can be efficiently compressed by discarding the excessively rough and fine information of coefficients generated from orthogonal transform. When having the compressed data, a receiver may reconstruct the principal content of original image by retrieving the values of coefficients. The computation of transform in the encrypted domain has also been studied. Based on the homomorphic properties of the underlying cryptosystem, the discrete Fourier transform in the encrypted domain can be implemented. In another work a composite signal representation method packing together a number of signal samples and processing them as a unique sample is used to reduce the complexity of computation and the size of encrypted data. There are also a number of works on data hiding in the encrypted domain. In a buyer–seller watermarking protocol, the seller of digital multimedia product encrypts the original data using a public key, and then permutes and embeds an encrypted fingerprint provided by the buyer in the encrypted domain.

After decryption with a private key, the buyer can obtain a watermarked product. This protocol ensures that the seller cannot know the buyer's watermarked version while the buyer cannot know the original version. An anonymous fingerprinting scheme that improves the enciphering rate by exploiting the Okamoto-Uchiyama encryption method has been proposed in. By introducing the composite signal representation mechanism, both the computational overhead and the large communication bandwidth due to the homomorphic public-key encryption are also significantly reduced. In another type of joint data-hiding and encryption schemes, a part of cover data is used to carry the additional message and the rest of the data are encrypted, so that both the copyright and the privacy can be protected. For example the intraprediction mode, motion vector difference and signs of DCT coefficients are encrypted, while a watermark is embedded into the amplitudes of DCT coefficients. In this work, we propose an encrypted-domain RIDH scheme by specifically taking the above-mentioned design preferences into consideration. The proposed technique embeds message through a public key modulation mechanism, and performs data extraction by exploiting the statistical distinguishability 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.

II. Literature survey

Some recent attempts were made on embedding message bits into the encrypted images.
Puech et. al. used a simple substitution method to insert additional bits into AES encrypted images. Local standard deviation (SD) was then exploited at the decoder side to reconstruct the original image

Zhang designed a method to embed additional message bits into stream cipher encrypted images by flipping 3 LSBs of half of the pixels in a block. The data extraction can be performed by utilizing the local smoothness inherent to natural images. This method was later improved by Hong et. al through a side match technique . As local smoothness does not always hold for natural images, data extraction errors can be observed in the high-activity regions. Further, Zhang proposed a separable RIDH method such that the protection scopes of data hiding key and encryption key are gracefully separated.

Zhang et. al. extended the lossless compression based RIDH approach to the encrypted domain, namely, losslessly compress half of the 4th LSBs of the encrypted image via LDPC code to create space for data hiding. As the source coding with side information at the decoder requires a feedback channel, this scheme would face severe challenges in many practical scenarios, e.g., secure remote sensing, where the feedback channel could be very costly. Ma et. al suggested a new embedding method by reserving room before encryption with a traditional reversible image watermarking algorithm. Significant improvements on embedding performance can be achieved by shifting partial embedding operations to the encryption phase. More recently, Qian et. al. proposed a RIDH framework that is capable of hiding data into an encrypted JPEG bit stream.

It should be noted that, for all the existing RIDH schemes including both non-separable as well as separable solutions, an extra data hiding key is introduced to ensure embedding security. Certainly, the data hiding key needs to be shared and managed between the date hider and the recipient. As mentioned earlier, the key management functions, e.g., the key generation, activation, de-activation, suspension, expiration, destruction, archival, and revocation, are difficult to be reliably implemented within such distributed infrastructure. A natural question arising now is whether we can design an encrypted-domain RIDH scheme, which does not require a secret data hiding key, while still ensuring that only the party with the secret encryption key K can disclose the embedded message. This could be very valuable in practice, as the cost and the potential risk of building up the KMS can be significantly reduced.

III. Proposed system

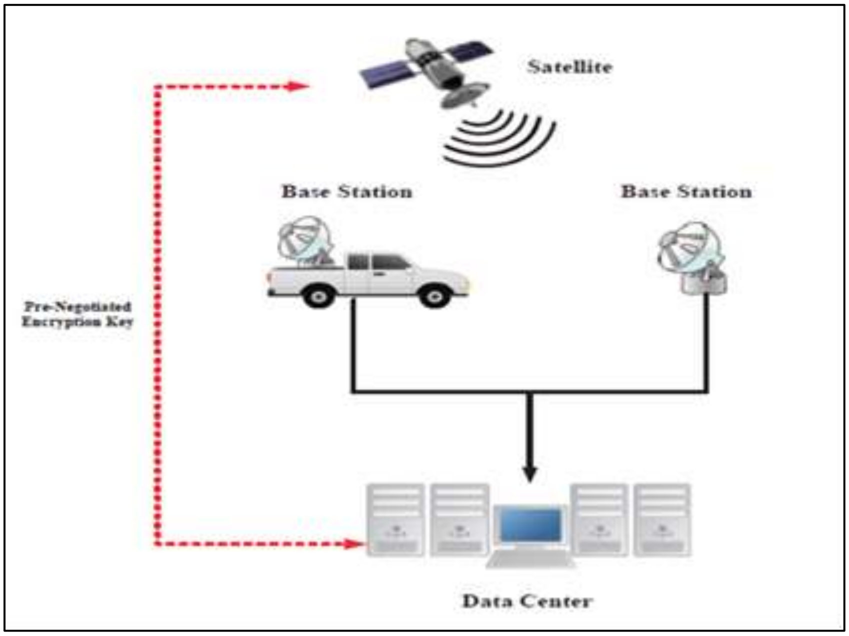

Fig. 1: Image data hiding in the scenario of secure remote sensing.

The proposed scheme is made up of image encryption, data embedding and data extraction/image-recovery phases. nstead of considering dedicated encryption algorithms tailored to the scenario of encrypted-domain data hiding, we here stick to the conventional stream cipher applied in the standard format. That is, the ciphertext is generated by bitwise XORing the plaintext with the key stream. If not otherwise specified, the widely used stream cipher AES in the CTR mode (AESCTR) is assumed.
When stream cipher is employed, the encrypted image is generated by

$$[[f]] = Enc(f,K) = f \oplus K \ (1)$$

where f and [[f]] denote the original and the encrypted images, respectively. Here, K denotes the key stream generated by using the secret encryption key K. In this work, without loss of generality, all the images are assumed to be 8-bit. Throughout the paper, we use [[x]] to represent the encrypted version of x. Clearly, the original image can be obtained by performing the following decryption function

$$f = Dec([[f]],K) = [[f]] \oplus K$$

In the proposed system the data is encrypted by using Advanced AES mechanism. The data after encryption is embedded with the image directly. Here the data embedding (Steganography) is done by Bitwise XORing with the image .During this method the data is attached to the free space in image.
After the XORing process the data bearing image is divided into blocks/frames(Block size M×N, where i is the block index. Each block is designed to carry n bits of message) depend on its size and length. block index. Each block is designed to carry n bits of message. Letting the number of blocks within the image be B, the embedding capacity of our proposed scheme becomes n·B bits. To enable efficient embedding, we propose to use S = 2n binary public keys Q0,Q1,··· ,QS−1, each of which is of length L = M × N × 8 bits. These image block is again encrypted and ready to send. Fig: 2 shows the proposed system architecture

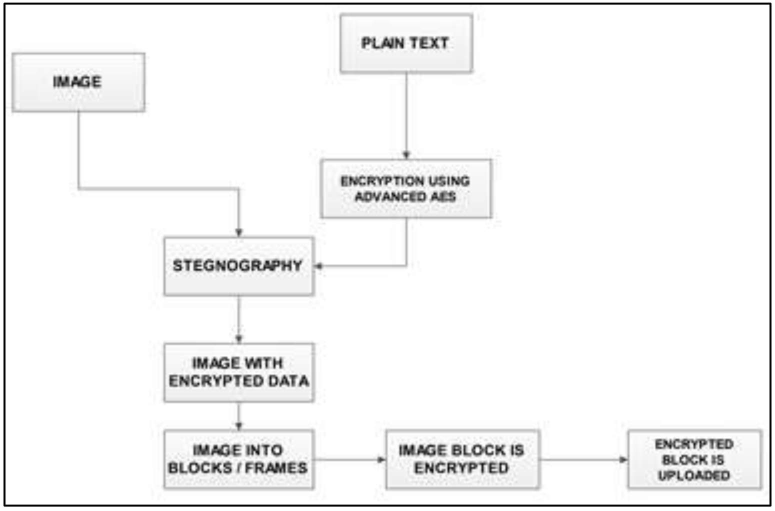
PLAIN TEXT
IMAGE
ENCRYPTION USING ADVANCED AES
STEGNOGRAPHY
IMAGE WITH ENCRYPTED DATA
IMAGE INTO BLOCKS / FRAMES
IMAGE BLOCK IS ENCRYPTED
ENCRYPTED BLOCK IS UPLOADED

Fig. 2: shows proposed system architecture

A. *Advanced AES Encryption*

It works same as normal AES but the difference is that we will introduce two libraries in JDK. They are
- US export policy
- Local policy

Which increases the actual key size in AES into its double value so that high security in encryption is provided.

B. *Bitwise XOR Operation (Steganography)*

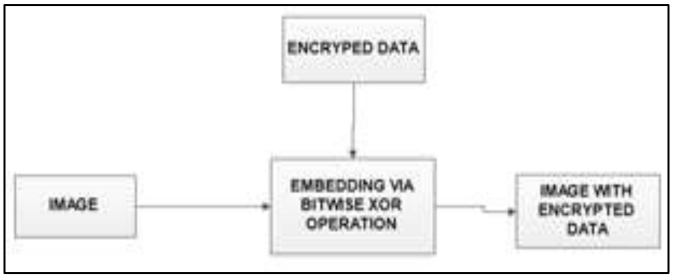
ENCRYPED DATA
IMAGE
EMBEDDING VIA BITWISE XOR OPERATION
IMAGE WITH ENCRYPTED DATA

Fig. 3: Shows Bitwise XOR Operation

1) Step 1 – Take the image to be encrypted
2) Step2 - Extract n bits of message to be embedded, denoted by Wi.
3) Step3 –Enter the key embed the data into image taking the inital data value as i=0

4) Step4 - Increment i = i + 1 and repeat Steps 2-3until all the message bits are inserted

C. *Joint Data Extraction and Image Decryption*

The decoder in the data center has the decryption key K, and attempts to recover both the embedded message and the original image simultaneously from [[f]]w, which is assumed to be perfectly received without any distortions. Note that this assumption is made in almost all the existing RIDH methods. Due to the interchangeable property of XOR operations, the decoder first XORs [[f]]w with the encryption key stream K and obtains

$$fw = [[f]]w \oplus K$$

The resulting fw is then partitioned into a series of nonoverlapping blocks fw i 's of size M×N, similar to the operation conducted at the embedding stage. From , we have

$$fw\ i = fi \oplus Q[Wi]d\ (13)$$

The joint data extraction and image decryption now becomes a blind signal separation problem as both Wi and fi are unknowns. Our strategy of solving this problem is based on the following observation: fi, as the original image block, very likely exhibits certain image structure, conveying sematic information. Note that Q[Wi]d must match one of the elements in Q = {Q0,Q1,··· ,QS−1}. Then if we XOR fw i with all Qj's, one of the results must be fi, which would demonstrate structural information. As will become clear shortly, the other results correspond to randomized blocks, which can be distinguished from the original, structured figure.

More specifically, we first create S decoding candidates by XORing fw i with all the S possible public keys Q0,Q1,··· ,QS−1 f(0) i = fw i ⊕ Q0 = fi ⊕ Q[Wi]d ⊕ Q0 f(1) i = fw i ⊕ Q1 = fi ⊕ Q[Wi]d ⊕ Q1 . . . f(S−1) i = fw i ⊕ QS−1 = fi ⊕ Q[Wi]d ⊕ QS−1 As mentioned earlier, one of the above S candidates must be fi, while the others can be written in the form

$$f(t)\ i = fi \oplus Q[Wi]d \oplus Qt\ (15)\ \text{where}\ t\ 6= [Wi]d.$$

The result f(t) i = Enc(fi,Q[Wi]d ⊕ Qt) corresponds to an encrypted version of fi with equivalent key stream being Q[Wi]d ⊕ Qt. Notice that all the public keys Qj's, for 0 ≤ j ≤ S − 1, are designed to have maximized minimum Hamming distance, and the upper bound is given in Hence, f(t) i tends to loose the image structural information, making it appear random.

To identify which candidate corresponds to fi, we apply the designed two-class SVM classifier to these S candidates. Let r = (r0,r1,··· ,rS−1)′ be the vector recording the classification results, where rj = 0 and rj = 1 correspond to the original (structured) and randomized blocks, respectively. If there exists a unique j such that rj = 0, then we decode the embedded message bits as
Wi = [j]2 where [j]2 denotes the length-n binary representation of j and n = log2 S. For example, if n = 3 and j = 7, then [j]2 = 111. Upon determining Wi, the original image block can be easily recovered by

$$fi = fw\ i \oplus Q[Wi]d.$$

IV. CONCLUSIONS

In this paper, we design a secure reversible image data hiding (RIDH) scheme operated over the encrypted domain. We suggest a public key modulation mechanism, which allows us to embed the data via simple XOR operations, without the need of accessing the secret encryption key. At the decoder side, we propose to use a powerful two-class SVM classifier to discriminate encrypted and non-encrypted image patches, enabling us to jointly decode the embedded message and the original image signal perfectly. We also have performed extensive experiments to validate the superior embedding performance of our proposed RIDH method over encrypted domain.

REFERENCE

[1] M. U. Celik, G. Sharma, and A. M. Tekalp,"Lossless watermarking for image authentication: a new framework and an implementation," IEEE Trans. Image Process., vol. 15, no. 4, pp. 1042-1049, 2006.
[2] C. Qin, C.-C. Chang, Y.-H. Huang, and L.-T. Liao,"An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism," IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 7, pp. 1109-1118, 2013.
[3] Z. Ni, Y. Shi, N. Ansari, and W. Su,"Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354-362, 2006.
[4] W. L. Tai, C. M. Yeh, and C. C. Chang,"Reversible data hiding based on histogram modification of pixel differences," IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 6, pp. 906-910, 2009.
[5] X. Li, W. Zhang, X. Gui, and B. Yang,"A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," IEEE Trans. Inf. Forensics Secur, vol. 8, no. 7, pp. 1091-1100, 2013.
[6] J. Tian,"Reversible data embedding using a difference expansion," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890-896, 2003.
[7] J. MacDonald,"Design methods for maximum minimum-distance errorcorrecting codes," IBM J., pp. 43-57, 1960.
[8] C.-C. Chang and C.-J. Lin,"Libsvm: A library for support vector machines," ACM Trans. Intelligent Syst. and Technol., vol. 2, no. 3, pp. 27-53, 2011.
[9] R. Hamming,"Error detecting and error correcting codes," Bell Sys. Tech. J., vol. 29, pp. 147-160, 1950.
[10] A. Buades, B. Coll, and J. Morel,"A non-local algorithm for image denoising," in Proc. of CVPR, 2005, pp. 60-65.