National Conference on Innovations in Computing and Networks (NCICN) - 2019
© 2019 IJSRSET | Volume 5 | Issue 9 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099

Reversible Secure Image Data Hiding over Encrypted Domain using Key Modulation

Salini S Nair*[1], Dr T Mahalakshmi[2], Chithira R[3]
*[1]Assistant Professor, Sree Narayana Institute of Technology, Kollam, Kerala, India
[2]Principal, Sree Narayana Institute of Technology, Kollam, Kerala, India
[3]Student, Sree Narayana Institute of Technology, Vadakkevila, Kollam, Kerala, India

ABSTRACT

This work proposes a secure reversible image data hiding (RIDH) scheme over encrypted domain. The embedding of data is achieved through a public key mechanism, in which secret encryption key is not needed. At the decoder side, a powerful two-class SVM classifier is designed to differentiate 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: Reversible Image Data Hiding(RIDH), SVM

I. INTRODUCTION

Secure Reversible image data hiding (RIDH) is a special category of data hiding technique, which guarantee perfect reconstruction of the image upon the extraction of the embedded message. The reversibility makes such image data hiding approach particularly attractive in the special scenarios, e.g., military and remote sensing, medical images sharing, law forensics and copyright authentication, where high fidelity of the reconstructed image is required. The majority of the existing RIDH systems are designed over the plaintext domain, usually, the message bits are embedded into the original, un-encrypted images. The early works mainly used the lossless compression algorithm to compress certain image features, in order to arrange 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, is another class of approach achieving better embedding performance through shifting the histogram of some image feature. The Difference Expansion (DE)-based schemes and the improved Prediction Error Expansion (PEE)-based strategies 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. 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 a 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). After receiving the encrypted images, the base station 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, 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 key shared 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.

II. METHODS AND MATERIAL

Reversible data hiding Scheme based on histogram modification exploit a binary tree structure to solve the problem of communicating pairs of peak points. Distribution of pixel differences is used to achieve large hiding capacity while keeping the distortion low. We also adopt a histogram shifting technique to prevent overflow and underflow. Performance comparisons with other existing schemes are provided to demonstrate the superiority of the proposed scheme designed to solve the problem of lossless embedding of large messages in digital images using distribution of pixel differences.

Reversible data embedding has drawn lots of interest recently. Being reversible, the original digital content can be completely restored. In this paper, we present a novel reversible data embedding method for digital images. We explore the redundancy in digital images to achieve very high embedding capacity, and keep the distortion low. RDE hid information in a digital image and authorized party can decode hidden data and restore image to its original state.RDE has a self -authenticating scheme for restore image and it is used as information carrier.

Reversible data hiding scheme for encrypted image**,** after encrypting the entire data of an uncompressed image by a stream cipher, the additional data can be embedded into the image by modifying a small proportion of encrypted data with an encrypted image containing additional data, one may firstly decrypt it using the encryption key, and the decrypted version is similar to the original image. According to the data-hiding key, with the aid of spatial correlation in natural image, the embedded data can be successfully extracted and the original image can be perfectly recovered the additional message is embedded by modifying a part of encrypted data. On the receiver side the embedded data are successfully extracted while the original image is perfectly reconstructed.

Modification to the system is a Blocker mechanism which blocks the user who attempts to use executable files.

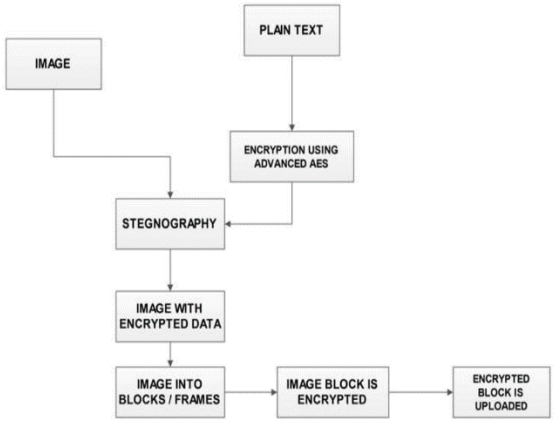

Figure 1:Architecture of the Proposed System

Following are the methods used for encryption, data embedding, data extraction and decryption.

A. Advanced AES Encryption

AES stands for Advanced Encryption Standard. Length of the input output block and the State is 128 for AES algorithm. This is represented by Nb = 4, which reflects the number of 32-bit words (number of columns) in the State. For the AES algorithm 128, 192, or 256 bits is the length of the Cipher Key, K. The key length of the block is denoted by Nk and its value is 4, 6, or 8. This value

reflects the number of 32-bit words (number of columns) in the Cipher Key. For the AES algorithm, during the execution of algorithm the numbers of rounds to be performed are dependent on the key size.

Nr is used to represent the number of round. AES algorithm uses a round function for both its Cipher and Inverse Cipher. This function is composed of four different byte-oriented transformations: 1. Using a substitution table (S-box) byte substitution, 2. By different offsets shifting rows of the State array, 3. Mixing the data within each column of the State array, and 4. adding a Round Key to the State. No. of round keys generated by key-expansion algorithm is always one more than the actual no. of round present in the algorithm.

In Advanced AES, the actual AES algorithm is executed twice to generate 256bit key by using two additional policies.

B. Steganography

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video. The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny. Plainly visible encrypted messages, no matter how unbreakable they are, arouse interest and may in themselves be incriminating in countries in which encryption is illegal. Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size.

Here Steganography is used for mainly image data hiding.

C. SVM Classifier

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples.

SVM Classifier here it used as to obtain the original image while extracting the data. Here use a two class SVM to compare the encrypted and non-encrypted image patches.

III. IMPLEMENTATION

Reversible Secure Image Data Hiding is a system in which we use Advanced AES for encryption. It ensures high security for image encryption. Bitwise XOR enables reversible process for data extraction and image decryption. Here a SVM is used as a two class classifier used to obtain original image after decryption. Here SVM compares encrypted and non-encrypted image patches.

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 and 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)

Step 1**:** Initialize block index i= 1.

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

Step 3**:** Find the public key Q[Xi]d associated with Xi, where the index [Xi]d is the decimal representation of Xi. For instance, when n = 3 and Xi = 010, the corresponding public key is Q2.

Step 4**:** Embed the length-n message bits Xi into the ith block via

$$[[f\,]]xi = [[f\,]]i \oplus Q[Xi]d$$

Step 5**:** Increment i= i+ 1 and repeat **Steps 2-4** until all the message bits are inserted.

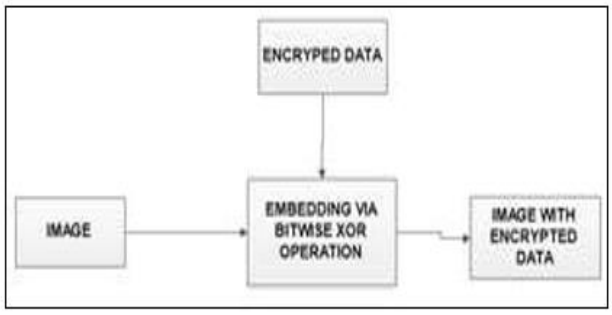

Figure 2: Bitwise XOR operation

C. Join Data Extraction and Image Decryption

The join data extraction and image decryption are signal separation problems. Here applies a two class SVM classifier, which differentiate encrypted and non-encrypted image patches, allowing us to jointly decode the embedded message and the original image signal

IV. RESULTS AND DISCUSSION

This work implement Advanced AES and steganography for encryption and message embedding at encoder side. A two class SVM classifier is used at decoder side for extracting the message and to obtain the original image. In this system the original image retrieval also have some particular importance. The results show that it is very secure for hiding confidential data. Here there is no need for private key modulation or no need to kept the key secret. This work also denies attempts to upload executable file. Because it may cause several virus attacks and it leads to the entire destruction of our system. So we here introduce a blocker mechanism to stop such attempts. When a user attempts to upload executable files the system blocks the user for a particular time. Thus it is secure Image data hiding system.
Following are the few screenshots of the data embedding, extraction and image decryption.

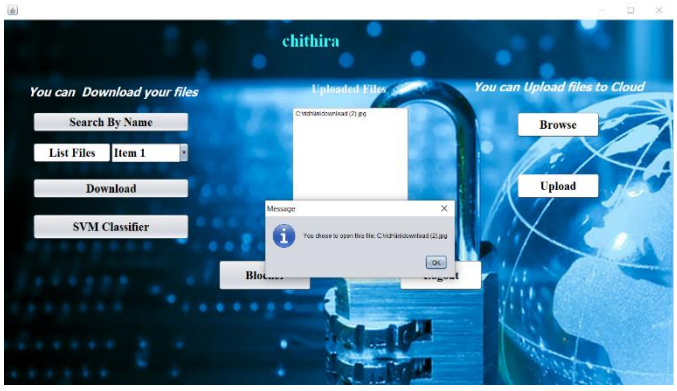

Figure 3: User browse image to embed the data

Figure 4: User enters the public key for encryption

Figure 5: User enters the message to be embedded

Figure 6: User downloading the encrypted data embedded image

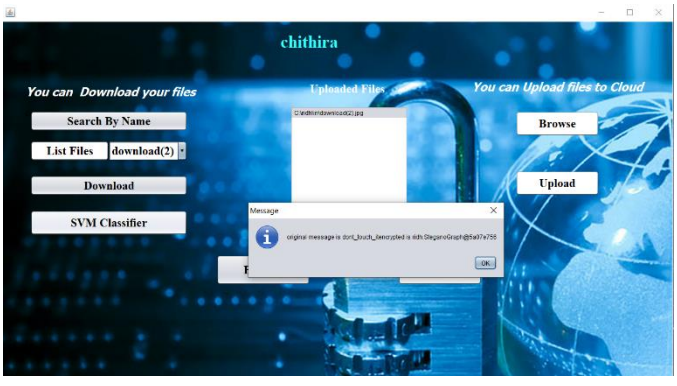

Figure 7: User applies SVM classifier and extracts message and decrypts original image

V. CONCLUSION

In this work, Secure reversible image data hiding (RIDH) scheme operated over a encrypted domain via a public key modulation. The public key modulation mechanism, which allows us to embed the data via simple XOR operations, without the need of accessing the secret encryption key such as private key. At the decoder side, there is a powerful two-class SVM classifier to differentiate encrypted and non-encrypted image patches, enabling us to jointly decode the embedded message and the original image signal perfectly.

In addition to this it improves security by implementing a blocker when an executable file is uploaded. Executable files may cause various kinds of virus attacks and it may destruct the entire system. So we have to take care of this. The blocker blocks such user attempts for a particular period of times. Thus it is a Secure Reversible Image Data Hiding system.

VI. REFERENCES

[1]. Jiantao Zhou, Weiwei Sun, Y.-H. Huang, Li Dong, Xianming Liu“ Secure Reversible image data Hiding over Encrypted Domain Via Key Modulation” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 7, pp. 1051-8215, 2015.

[2]. C. Qin, C.-C. Chang, Y.-H. Huang, and L.-T. Liao “An in painting-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 error correcting 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.