Efficient technique of data hiding in encrypted images

K.Sravani[#1],M.O.V.Pavan Kumar[#2]
#1 PG scholar (VLSI)
#2 Assistant Professor
Department of electronics and communication engineering
Gokaraju Rangaraju Institute of engineering & technology,JNTUH

***Abstract*— Recently, more and more attention is paid to reversible data hiding (RDH) in encrypted images, since it maintains the excellent property that the original cover can be lossless recovered after embedded data is extracted while protecting the image content's confidentiality. All previous methods embed data by reversibly memory space from the encrypted images, which may be subject to some errors on data extraction and/or image restoration. In this paper, we propose a novel method by reserving memory space before encryption with a traditional RDH technique, and thus it is easy for the data hider to reversibly embed data in the image. The proposed method can achieve real reversibility, that is, data extraction and image recovery are free of any error.**

***Keywords*— Image encryption, Data embedding, Image extraction, Reversible data hiding.**

Introduction

Reversible data hiding (RDH) in images is a technique, by which the original cover can be lossless recovered after the embedded message is extracted. This important technique is widely used in medical imagery, military imagery and law forensics, where no distortion of the original cover is allowed. Since first introduced, RDH [9] has attracted considerable research interest. With regard to providing confidentiality for images, encryption is an effective and popular means as it converts the original and meaningful content to incomprehensible one. Although few RDH techniques in encrypted images have been published yet, there are some promising applications if RDH [9] can be applied to encrypted images. To separate the data extraction from image decryption, the idea of compressing encrypted images and the space for data embedding; Compression of encrypted data [1] can be formulated as source coding with side information at the decoder, in which the typical method is to generate the compressed data in lossless manner by exploiting the syndromes of parity-check matrix of channel codes. The method in compressed the encrypted LSBs to memory space for additional data by finding syndromes of a parity-check Matrix [7] and the side information used at the receiver side is also the spatial correlation of decrypted images.

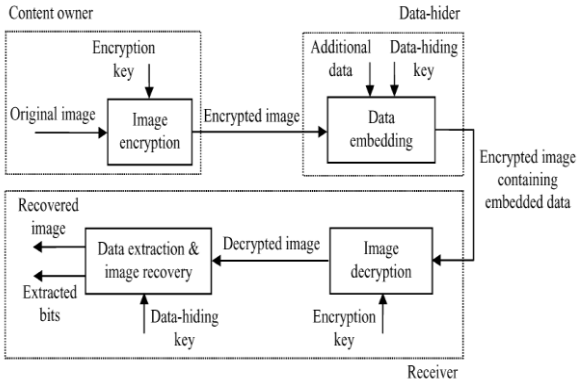


Fig.1 Non-Separable Reversible Data Hiding in Encrypted Image

And it is also hopeful that the original content should be recovered without any error after image decryption and message extraction at receiver side. Fig. 1 gives the sketch. A content owner encrypts the original image using an encryption key, and a data-hider can embed additional data into the encrypted image using a data-hiding key though he does not know the original content. With an encrypted image

Containing additional data, a receiver may first decrypt it according to the encryption key, and then extract the embedded data and recover the original image according to the data-hiding key. In the scheme, the data extraction is not separable from the content decryption. In other words, the additional data must be extracted from the decrypted image, so that the principal content of original image is revealed before data extraction, and, if someone has the data-hiding key but not the encryption key, he cannot extract any information from the encrypted image containing additional data. encryption key, and a data-hider can embed additional data into the encrypted image using a data-hiding key though he does not know the original content.

With an encrypted image containing additional data, a receiver may first decrypt it according to the encryption key, and then extract the embedded data and recover the original image according to the data-hiding key. In the scheme, the data extraction is not separable [4] from the content decryption. In other words, the additional data must be extracted from the decrypted image, so that the principal content of original image is revealed before data extraction,

ISSN: 2278 – 7798

and, if someone has the data-hiding key but not the encryption key, he cannot extract any information from the encrypted image containing additional data.

I. PROPOSED SYSTEM

The proposed scheme is made up of image encryption, data embedding and data-extraction/ 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) of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data [8].

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 data-hiding 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.

a) *Image Encryption*

The user will browse the image from computer and encrypt the image and system will auto generate encryption key.

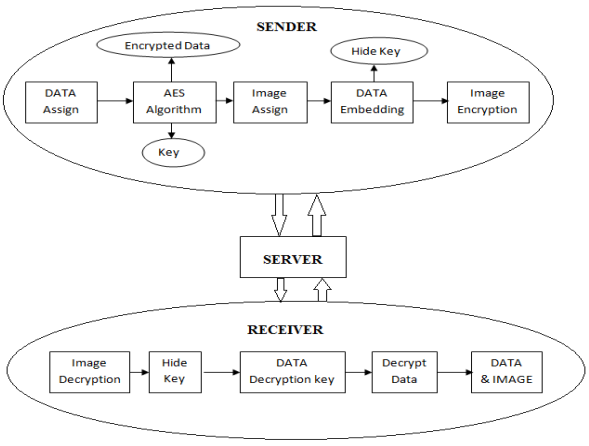
b) *Data Encryption*

The user will browse data that he want send and encrypt the original data and system will auto generate the data encryption key.

c) *Data Embedding*

User will hide the encrypted data in encrypted image and system will auto generate data hiding key and system will generate file extension as per user defined. In the data embedding phase, some parameters are embedded into a small number of encrypted pixels, and the LSB of the other encrypted pixels are compressed to create a space for accommodating the additional data and the original data at the positions occupied by the parameters. The detailed procedure is as follows According to a data-hiding key, the data-hider randomly selects Np encrypted pixels that will be used to carry the parameters for data hiding [8]. Here, Np is a small positive integer, for example, Np=20.

II. BLOCK DIAGRAM:



III. MODULES:

1) Extracting data from encrypted images:

To manage and update personal information of images which are encrypted for protecting clients' privacy, an inferior database manager may only get access to the data hiding key and have to manipulate data in encrypted domain. The order of data extraction before image decryption guarantees the feasibility of our work in this case. When the database manager gets the data hiding key, he can decrypt the LSB-planes of encrypted version of A denoted by A extract the additional data m by directly reading the decrypted version. When requesting for updating information of encrypted images, the database manager, then, updates information through LSB replacement and encrypts updated information according to the data hiding key all over again. As the whole process is entirely operated on encrypted domain, it avoids the leakage of original content.

2) Extracting data from decrypted images:

In Module1, 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 an application 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. Next, we describe how to generate a marked decrypted image.

Fig 2: (a) Original image, (b) its encrypted image, (c) encrypted image containing embedded data and (d) directly decrypted image

ISSN: 2278 – 7798

2) AES Algorithm

In our system we are using 128 bit key and in AES this is represented by $N_b = 4$, which reflects the number of 32-bit words (number of columns) in the State. The length of the Cipher Key, K, is 128. The key length is represented by $N_k = 4$, 6, or 8, which reflects the number of 32-bit words (number of columns) in the Cipher Key. The number of rounds to be performed during the execution of the algorithm is dependent on the key size. The number of rounds is represented by N_r where $N_r = 10$ when $N_k = 4$, $N_r = 12$ when $N_k = 6$, and Nr = 14 when $N_k = 8$.

For both its Cipher and Inverse Cipher, the AES algorithm uses a round function that is composed of four different byte-oriented transformations:

1. Substitution using a substitution table (S-box).
2. Shifting rows of the State array by different offsets
3. Mixing the data within each column of the State array
4. Adding a Round Key to the State.

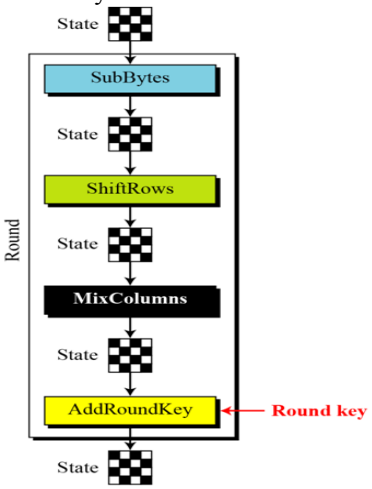


Fig 3: AES transformations

IV. CONCLUSION

Reversible data hiding scheme for encrypted image with a low computation complexity is proposed, which consists of image encryption, data embedding and data extraction/ image recovery phases. The data of original image are entirely encrypted by a stream cipher. Although a data hider does not know the original content, he can embed additional data into the encrypted image by modifying a part of encrypted data. With an encrypted image containing embedded data, a receiver 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 correctly extracted while the original image can be perfectly recovered.

Although someone with the knowledge of encryption key can obtain a decrypted image and detect the presence of hidden data using LSB steganalytic methods, if he does not know the data hiding key, it is still impossible to extract the additional data and recover the original image. For ensuring the correct data extraction and the perfect image recovery, It may let the block side length be a big value or introduce error correction mechanism before data hiding to protect the additional data with a cost of payload reduction.

V. FUTURE WORK

The implemented a reversible method can be enhanced in future by using the following provisions and MLSB technique can also be applied after embedding when there is lot of change in the pixel to retain nearest to the original value. It can be applied in networking and the keys are sent and received securely. The image produced by the reversible data hiding using two key has distortion. In order to remove distortion and to produce the image in a high quality using 3 key.

V. REFERENCES

1. 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.
2. Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354-362, Mar.2006.
3. C.-C. Chang, C.-C.Lin, and Y.-H.Chen, "Reversible data-embedding scheme using differences between original and predicted pixel values, "IET Inform. Security, vol. 2, no. 2, pp.35-46, 2008.
4. T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discrete Fourier transform in the encrypted domain," IEEE Trans. Inform. Forensics Security, vol. 4, no. 1, pp. 86-97, Feb. 2009.
5. W. Liu.W. Zeng, L.Dong, and Q.Yao, "Efficient compression of encrypted gray scale images," IEEE Trans. Image Process., vol. 19, no.4, pp. 1097-1102, Apr. 2010.
6. T. Bianchi, A. Piva, and M. Barni, "Composite signal representation for fast and storage- efficient processing of encrypted signals," IEEE Trans. Inform. Forensics Security, vol. 5, no. 1, p. 180-187, Feb. 2010.
7. X.Zhang, "Lossy compression and iterative reconstruction for encrypted image," IEEE Trans. Inform. Forensics Security, vol. 6, no.1, pp. 53-58, Feb. 2011.
8. Xinpeng Zhang "Separable Reversible Data Hiding in Encrypted Image" IEEE Trans. VOL. 7, no. 2, Apr 2012.
9. Kede Ma, Weiming Zhang, "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption" IEEE Trans. VOL. 8, no. 3, Mar 2013.

ISSN: 2278 – 7798