International Association of Scientific Innovation and Research (IASIR)
(An Association Unifying the Sciences, Engineering, and Applied Research)

ISSN (Print): 2279-0063
ISSN (Online): 2279-0071

International Journal of Software and Web Sciences (IJSWS)
www.iasir.net

A HYBRID TECHNIQUES FOR EXECUTABLE FILE HIDING IN ENCRYPTED IMAGE

G.Archana (M.Tech Research Scholar)
Department of Computer Science and Engineering,
SRM University,Tamilnadu.
G.Niranjana
Department of computer science and Engineering,
SRM University,Tamilnadu.

Abstract
Data hiding is referred as a process of hiding a data into a cover medium. Also the process works with two links one is with a set of embedding of additional data and another one is with a set of the cover data. The relationship between these two sets of data is characterized for different applications. For instance, in some communications, the hidden data may often be irrelevant to the cover media. In authentication, however, the embedded data are closely related to cover media. The proposed approach uses a reversible data hiding mechanism for embedding the secrete information i.e. of an executable file to that of the encrypted image. When retrieving the executable files from the encrypted image the exe will be getting extracted without any distortions. Also the proposed approach uses digital image based data embedding technique to achieve a very high embedding capacity. The content owner encrypts original image into encrypted image using AES encryption algorithm. In the encrypted image the additional data will be embedded using the LSB based embedding approach. Thus the encrypted image will be embedded with the executable file and send to the receiver. The receiver will use the decryption key and embedding key to retrieve the original image and run the executable file. Thus the proposed method is used for all kinds of digital images.

Keywords: *Cover medium, data hiding, data embedding and authentication.*

I. Introduction

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. With the lossy compression method presented, 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 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. 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 the cover data in higher and lower bit-planes of transform domain are respectively encrypted and watermarked. In the

content owner encrypts the signs of host DCT coefficients and each content-user uses a different key to decrypt only a subset of the coefficients, so that a series of versions containing different fingerprints are generated for the users.
The reversible data hiding in encrypted image is investigated. Most of the work on reversible data hiding focuses on the data embedding/extracting on the plain spatial domain. But, in some applications, an inferior assistant or a channel administrator hopes to append some additional message, such as the origin information, image notation or authentication data, within the encrypted image though he does not know the original image content. And it is also hopeful that the original content should be recovered without any error after image decryption and message extraction at receiver side. Reference presents a practical scheme satisfying the above-mentioned requirements and Fig. 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.
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 AES 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 data-hiding key and the AES encryption key, then they can extract the additional data and recover the original image without any error.

II. Perceptive of Our Work

The use of computer networks for data transmissions has created the need of security. Many robust message encryption techniques have been developed to supply this demand. The encryption process can be symmetric, asymmetric or hybrid and can be applied to blocks or streams. Several asymmetric algorithms use long keys to ensure the confidentiality because a part of the key is known. These algorithms are not appropriate enough to be applied to images because they require a high computational complexity. In the case of block encryption methods applied to images, one can encounter three inconveniences. The first one is when we have homogeneous zones (regions with the same color), all blocks in these zones are encrypted in the same manner. The second problem is that block encryption methods are not robust to noise. Indeed, because of the large size of the blocks (which is at least of 128 bits) the encryption algorithms per block, symmetric or asymmetric, cannot be robust to noise. The third problem is data integrity. The combination of encryption and data-hiding can solve these types of problems.
The proposed approach is very simple, fast, accurate and which is been applied together as a double algorithm in order to provide better results under different complex circumstances like encryption ,data embedding and compressing of the images etc. Each of these algorithms are been discussed one by one below. Also the Advanced Encryption Standard algorithm is used to encrypt the original image. In addition to that a Data hiding mechanism is used to hide the executable file into the encrypted image.

A. Encryption Phase

The Advanced Encryption Standard (AES) algorithm consists of a set of processing steps repeated for a number of iterations called rounds. The number of rounds depends on the size of the key and the size of the data block. The number of rounds is 9 for example, if both the block and the key are 128 bits long. Given a sequence {X1,X2, ...,Xn} of bit plaintext blocks, each Xi is encrypted with the same secret key k producing the ciphertext blocks {Y1, Y2, ..., Yn}.

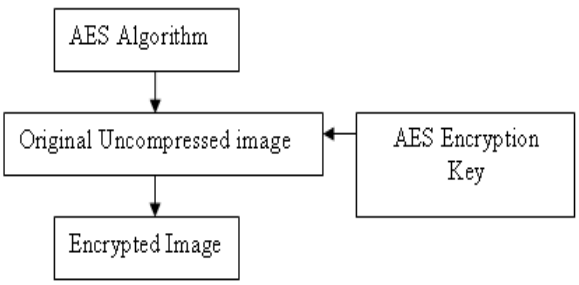

Figure 1. Encryption Module

AES operates on this 4×4 array of bytes termed the state. For encryption, each round of AES (except the last round) consists of four stages:

- Add Round Key — each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule.

- Sub Bytes — a non-linear substitution step where each byte is replaced with another according to a lookup table.
- Shift Rows — a transposition step where each row of the state is shifted cyclically a certain number of steps.
- Mix Columns — a mixing operation which operates on the columns of the state, combining the four bytes in each column using a linear transformation.
- The final round replaces the Mix Columns stage with another instance of Add Round Key. Finally it obtains an encrypted image.

B. Exe embedding in the Encrypted Image

The coding algorithm is composed of two steps which are the encryption and the data hiding step. The overview of the encoding method is shown in Figure 2. For each block Xi is composed of n pixels pj of an image of N pixels block. Ek() is the encryption function with the secret key k and Yi is the corresponding cipher-text to Xi.

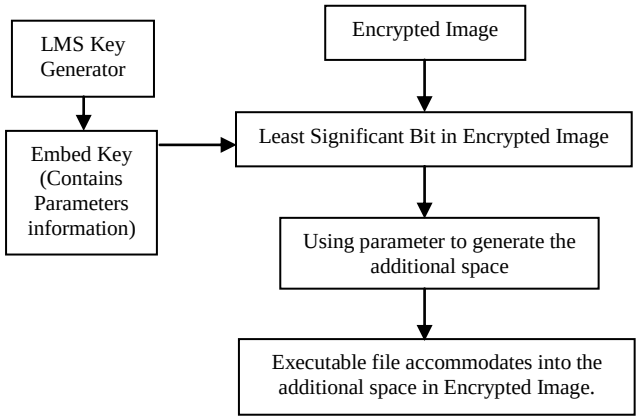

Figure 2. Executable File Embedding Process

In the data embedding, data hiding 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 executable file and the original data at the positions occupied by the parameters. The RSA key generator generates embed key. According to data-hiding key, the sender pseudo randomly selects encrypted pixels that will be used to carry the parameters for data hiding. Here, carried parameters pixels are a small positive integer. The other encrypted pixels are pseudo-randomly permuted and divided into a number of groups, each of which contains selected pixels. The permutation way is also determined by the data-hiding key. For each pixel-group, collect the number least significant bits of the selected pixels, and denote them as each indexed group pixels of bits. The sender also generates a matrix which is composed of two parts such as left part is an identity matrix, the right part is a pseudo-random binary matrix derived from the data hiding key. A total of bits made up of original Least Significant Bit of selected encrypted pixels and additional bits will be embedded into the pixel groups. The compressed bits space of pixels will generate by the inverse matrix of the indexed group pixel bits. Inverse indexed group LSB matrix of the original indexed group LSB of selected encrypted pixels and the additional data to be embedded. Then, replace the original indexed group LSB with the new Inverse matrix indexed group LSB, and put them into their original positions by an inverse permutation. At the same time, the most significant bits (MSB) of encrypted pixels are kept unchanged. Since selected compressed bits are embedded into each pixel-group with data file bits. The total data file bits can be accommodated in all groups.

III. Decryption and Data Extraction

The decoding algorithm is composed of two steps which are extraction of executable file and another one is of decrypting of encrypted image. The overview of the decoding method is presented in the scheme of Figure 3. The extraction of exe is very simple and is just enough to read the bits of the pixels that are marked by using the secret key k. In the extraction, each marked cipher-text is marked to decrypt the marked encrypted image. The decryption removing is done by analyzing the local standard deviation during the decryption of the marked encrypted images. For each marked cipher-text we apply the decryption function Dk() for the two possible values of the hidden bit (0 or 1) and we analyze the local standard deviation of the two decrypted blocks X0i and X1i. In the encrypted image, the entropy value must be maximal and greater than the original one. Moreover, the local standard deviation of the encrypted image is higher than for the original image. From this assumption the comparison is done for each block as the local standard deviation for X0i with X1i and select a bit value. Also an encrypted embedded image which contains embedded exe data and it can be only viewed if the receiver has the Embed key. The parameters values are taken from least significant bit and which has a small positive integer with a selected pixel value of the selected encrypted Image. By applying the AES encryption key to the encrypted image and reverse process is been done and thus finally we recovered the original image and executable file from the embedded image.

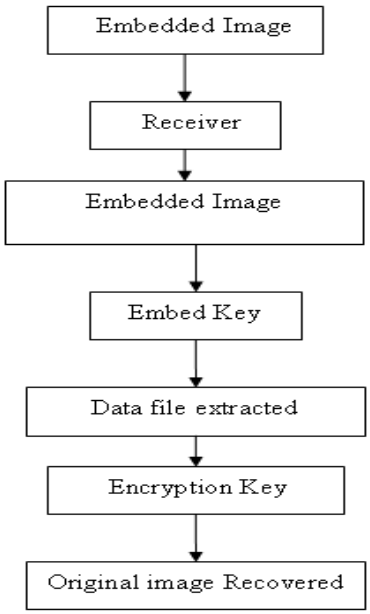

Figure 3: Decryption & Data Extraction

IV. RESULTS AND DISCUSSION

We have applied our method on various color images and we show the results of the proposed method which is applied on with different image pixels (1024 × 1024 pixels) illustrated in Figure. 4. To deliver an effective system we take test on various color images which includes many colors with numerous sizes. The test image Lena sized 512 X512 is shown in Fig. 4: (a) was used as an original image for processing in the result. After image encryption, the eight encrypted bits of each pixel are converted into a gray value to generate an encrypted image shown in Fig. 4(b). The encrypted image which contains the embedded executable file is shown in Fig. 4(c), and the embedding rate is higher than the existing system. The encrypted image containing embedded data (Executable file) is embedded using the data-hiding key. The AES Decryption method directly decrypts the encrypted image and retrieves the original information. The decrypted image is given and shown in figure 4 (d). By using both the data-hiding and the AES encryption keys, the embedded data could be successfully extracted and the original image could be perfectly recovered from the encrypted image that containing the embedded data. Thus these proposed techniques works well under any robust conditions like complex background and also with different face positions. These algorithms give different rates of accuracy under different conditions as experimentally observed. A test shown in figure 5a was taken on the training on different technique with that of the accuracy in percentage value were highlighted. The experiment was also conducted again and again and which provides a 100% result. Also the experiment was proved in Figure 5 b that the proposed algorithm has a tolerable Error rates compared with the other existing algorithm.

Figure 4: (a) Original Image, (b) Encrypted Image, (c) Encrypted image containing embedded data (EXE) with embedding rate, and (d) Decrypted Image

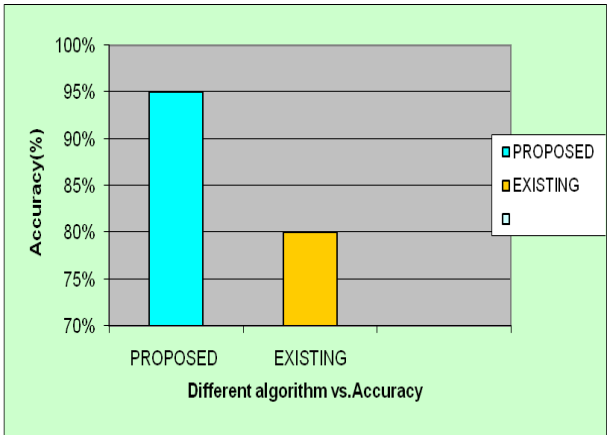

Figure. 5 a Different algorithm vs. accuracy

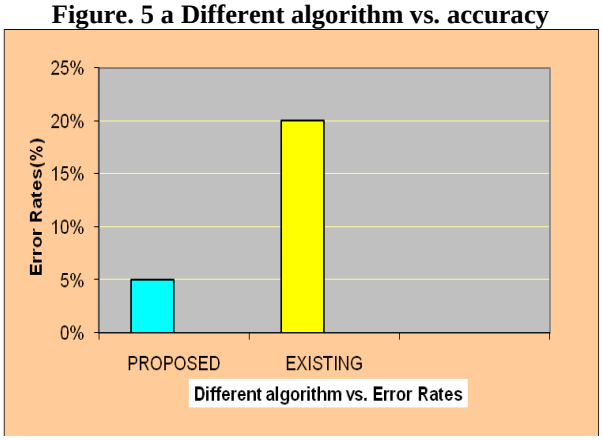

Figure.5 b Different algorithm vs. Error Rates

V. Conclusions

Separable reversible data hiding technique is been applied for the encrypted image and is been proposed. Also the approach works with an image encryption, exe embedding and exe file-extraction/image-recovery phases. In the first phase, the content owner encrypts the original uncompressed image using an encryption key. Also a data-hider does not know about the original content and then compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data. With an encrypted image containing additional data, the sender can extract the additional data using the data-hiding key, to obtain an image similar to the original one using the encryption key. When the receiver has both the keys, then they can extract the data and recover the original content without any error. However, the proposed method is more compatible with encrypted images and embedding of the executable files in them makes the system performance efficient.

References

1. Reversible Data Embedding Using a Difference Expansion Jun Tian.
2. Reversible Data Hiding Zhicheng Ni, Yun-Qing Shi, Nirwan Ansari, and Wei Su.
3. New Image Encryption and Compression Method Based on Independent Component Analysis Masanori Ito and Noboru Ohnishi Graduate School of Information Science Nagoya University Furo-cho, Chikusa-ku, Nagoya, 464–8603 Japan Email: ito-m@nagoya-u.jp Ayman Alfalou ISEN-Brest 20 rue de cuirass´e Bretagne 29228 Brest Cedex 2, France Ali Mansour ENSIETA2 Rue Franc¸ois Verny 29806 Brest, France.
4. On Compressing Encrypted Data Mark Johnson, Student Member, IEEE, Prakash Ishwar, Vinod Prabhakaran, Student Member, IEEE, Daniel Schonberg, Student Member, IEEE, and Kannan Ramchandran, Senior Member, IEEE.
5. 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, pp. 180–18, Feb. 2010.
6. N. Memon and P. W. Wong, "A buyer-seller watermarking protocol," IEEE Trans. Image Process., vol. 10, no. 4, pp. 643–649, Apr. 2001.
7.M. Kuribayashi and H. Tanaka, "Fingerprinting protocol for images based on additive homomorphic property," IEEE Trans. Image Process., vol. 14, no. 12, pp. 2129–2139, Dec. 2005.
8. M. Deng, T. Bianchi, A. Piva, and B. Preneel, "An efficient buyer-seller watermarking protocol based on composite signal representation," in Proc. 11th ACM Workshop Multimedia and Security, 2009, pp. 9–18.
9. S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative encryption and watermarking in video compression," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 6, pp. 774–778, Jun. 2007.
10. M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. B. Natale, and A. Neri, "A commutative digital image watermarking and encryption method in the tree structured Haar transform domain," Signal Processing:Image Commun., vol. 26, no. 1, pp. 1–12, 2011.