International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)

International Conference on Information Systems and Computing (ICISC-2013), INDIA.

EMBEDDING OF EXECUTABLE FILE IN ENCRYPTED IMAGE USING LSB MECHANISM

P.Sathish Kumar[1], Mr. C.BalaKrishnan[2]

[1]*M.E.(CSE), Dept. of CSE, S.A. Engineering College, Chennai, Tamil Nadu, India*
[2]*Research Scholar, MIT Campus, Anna University, Chennai, Tamil Nadu, India*

sathish23_83@yahoo.com, kknbalki_saec@yahoo.com

Abstract

Embedding of executable file in encrypted image encrypts the original uncompressed image using advanced encryption standard algorithm with an encryption key. Then, the executable file sender will compress the least significant bits of the encrypted image using an embedding key to create sparse space to accommodate the executable file. Using embedding key the executable file will be embedded into created space. With an encrypted image containing the embedded executable file, if the receiver has encryption key, he can decrypt the received data to obtain an image not similar to the original one and cannot extract the executable file. If the receiver has both the embedding key and encryption key, he can extract the executable file and recover the original content without any error by exploiting the spatial correlation in natural image.

***Keywords*-- Image encryption, Image recovery, Executable file hiding.**

I. INTRODUCTION

With the continued growth of multimedia applications, security is an important issue in communication and image storage, and encryption is the best way to ensure security. An encryption image technique attempts to convert the original image to another image which is difficult to identify as the original image. The purpose is to keep the image confidential among users and embedding of executable file into an encrypted image, in other words, it is essential that not just anyone can determine the contents of the image without a decryption key. In addition, the algorithm can find application where special storage and transmission security and reliability of digital images necessary such as military communication and information technology industries, etc. In fact, the use of a communication network to exchange data presents certain risks, which requires the existence of appropriate security measures. For example, the transmitted images can be saved and copied during their transmission without loss of image quality. Image and data can be hacked in time during an exchange of digital information storage and this is of course illegal. It is therefore necessary to develop a tool for effective protection of transferred data against arbitrary interference. Data encryption is very often the only effective way to meet these requirements. In this paper we are interested in the security of image and executable file data, which is considered as complex data, in particular, due its size and the information is two-dimensional and redundant in nature.

These features of the data make the algorithms developed in the literature unusable in their traditional form due to speed limitations and loss of information that can be caused by advanced encryption standard algorithm. It can be argued that there is no particular encryption algorithm which satisfies the requirements of all image types. In order to decrease the high correlation among pixels and increase the entropy value of an image, we propose a process based on shifting the rows and columns of the image using the following technique. The shifting process will be used to divide the original image into a number of blocks that are then shifted through the rows and the columns within the image based on a shifted table that is generated by another algorithm before the encryption process starts.

The generated image is then fed into the AES encryption algorithm. Fig. 1 gives the sketch. A content owner encrypts the original image using an encryption key, and a exe file sender can embed executable file into an encrypted image using a embedding key though he does not know the original content. With an encrypted image containing embedded executable file, a receiver may first decrypt it according to the encryption key, and then extract the embedded file. In the scheme, the executable file must be extracted from the decrypted image, so that the principal content of original image is revealed before file extraction, and if someone has the embedding key but not the encryption key, he cannot extract any information from the encrypted image containing executable file.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

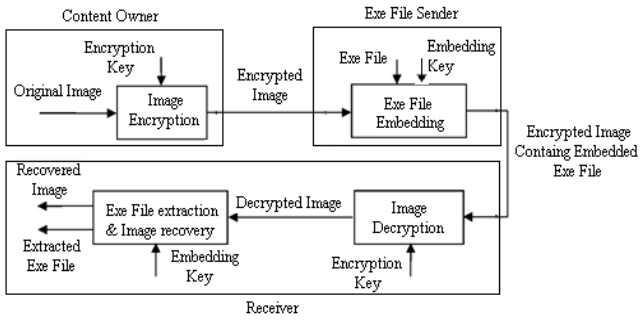

Fig.1. General Block Diagram

This paper proposes a novel scheme for embedding of executable file in encrypted image using LSB mechanism.In the proposed scheme, the original image is encrypted using an encryption key and the executable file are embedded into the encrypted image using an embedding key. With an encrypted image containing additional data, if the receiver has only the embedding key, he can extract the 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 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.

II. PROPOSED METHOD

2.1. Image Encryption

The Advanced Encryption Standard (AES) computer security standard is a symmetric block cipher that encrypts and decrypts 128-bit blocks of data. Standard key lengths of 128, 192, and 256 bits may be used. The algorithm consists of four stages that make up a round which is iterated 10 times for a 128-bit length key, 12 times for a 192-bit key, and 14 times for a 256-bit key.The first stage "SubBytes" transformation is a non-linear byte substitution for each byte of the block. The second stage "ShiftRows" transformation cyclically shifts (permutes) the bytes within the block.

The third stage "MixColumns" transformation groups 4-bytes together forming 4-term polynomials and multiplies the polynomials with a fixed polynomial mod (x^4+1). The fourth stage "AddRoundKey" transformation adds the round key with the block of data.In most ciphers, the iterated transform (or round). Typically in this structure, some of the bits of the intermediate state are transposed unchanged to another position (permutation). AES is composed of three distinct invertible transforms based on the Wide Trial Strategy design method.The Wide Trial Strategy design method provides resistance against linear and differential cryptanalysis. In the Wide Trail Strategy, every layer has its own function:

- The linear mixing layer: guarantees high diffusion over multiply rounds.
- The non-linear layer: parallel application of S-boxes that have the optimum worst-case non-linearity properties.
- The key addition layer: a simple XOR of the round key to the intermediate state.

Table 1.
AES Parameters

Algorithm	Key Length (Nk Words)	Block Size (Nb Words)	Number of rounds(Nr)
AES – 128	4	4	10
AES – 192	6	4	12
AES – 256	8	4	14

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.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

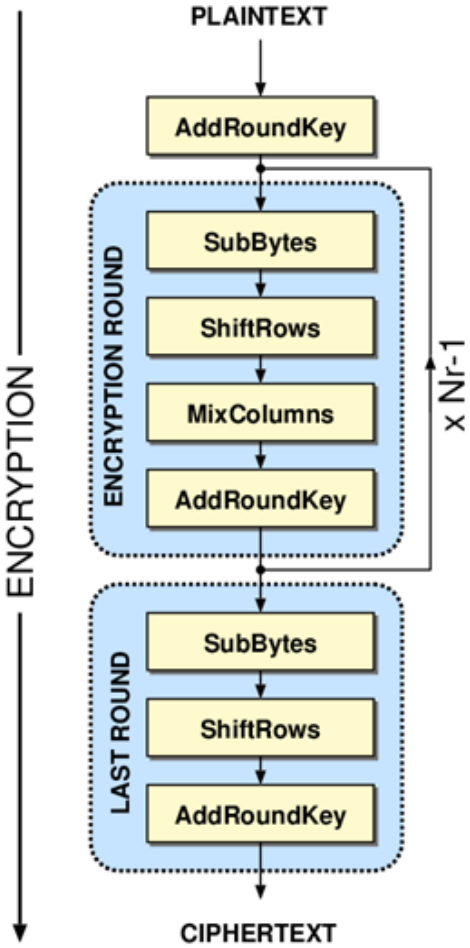

Fig.2. The Image Encryption Process

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 embedding of exe file can solve these types of problems.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}, as described in the scheme. To encipher a data block Xi in AES you first perform an AddRoundKey step by XORing a subkey with the block. The incoming data and the key are added together in the first AddRoundKey step. Afterwards, it follows the round operation. Each regular round operation involves four steps. In the SubBytes step, each byte of the block is replaced by its substitute in a substitution box (S-Box). In cryptography, an S-box is a basic component of symmetric key algorithms used to obscure the relationship between the plaintext and the ciphertext. The next one is the ShiftRows step where the rows are cyclically shifted over different offsets. The next step is the MixColumns, where each column is multiplied with a matrix over the Gallois Field, denoted as GF. The last step of the round operation is another AddRoundKey. It is a simple XOR with the actual data and the subkey for the current round. Before producing the final ciphered data Yi, the AES performs an extra final routine that is composed of (SubBytes, ShiftRows and AddRoundKey) steps, as shown in Fig. 2.

2.2. Embedding of Exe File

In the embedding of exe file 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 pseudo-randomly selects Np 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 Mleast significant bits of the L pixels, and denote them B(k,1),B(k,2)…..,B(k,M.L) where kis a group index within [1,(N-Np)/L] and Mis a positive integer less than 5. The data-hider also generates a matrix Gsized (M.L-S)xM.L, which is composed of two parts.

$$G=[IM.L\text{-}S\ Q] \qquad (1)$$

While the left part is an (M.L-S)x(M.L-S)identity matrix, the right part Q sized(M.L-S)xSis a pseudo-random binary matrix derived from the embedding key.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

Here, Sis a small positive integer. Then, embed the values of the parameters M, Land Sinto the LSB of Npselected encrypted pixels. For the example Np=20, the exe file sender may represent the values of M, L and S as 2, 14 and 4 bits, respectively, and replace the LSB of selected encrypted pixels with the 20 bits.
In the following a total of (N-Np).S/L bits made up of Np original LSB of selected encrypted pixels and (N-Np).S/L-Np additional bits will be embedded into the pixel groups. For each group,calculate

$$\begin{pmatrix} B'(k,1) \\ B'(k,2) \\ \vdots \\ \vdots \\ B'(k,M.L-S) \end{pmatrix} = G. \begin{pmatrix} B(k,1) \\ B(k,2) \\ \vdots \\ \vdots \\ B(k,M.L-S) \end{pmatrix} \quad (2)$$

Where the arithmetic is modulo-2 by (2), [B(k,1),B(k,2),…….,B(k, M . L)] are compressed as (M .L – S) bits, and a sparse space is therefore available for data accommodation. Let [B'(k, M . L – s + 1), B'(k, M . L – S +2), ……,B'(k, M . L)] of each group be the original LSB of selected encrypted pixels and the additional data to be embedded. Then, replace the [B(k,1),B(k,2),…….,B(k, M . L)] with the [B'(k,1),B'(k,2),…….,B'(k, M . L)] and put them into their original positions by an inverse permutation. At the same time, the (8 – M) most significant bits(MSB) of encrypted pixels are kept unchanged. Since S bits are embedded into each pixel – group, the total (N-Np) . S/L bits can be accommodated in all groups.

Fig.3a. Original Image

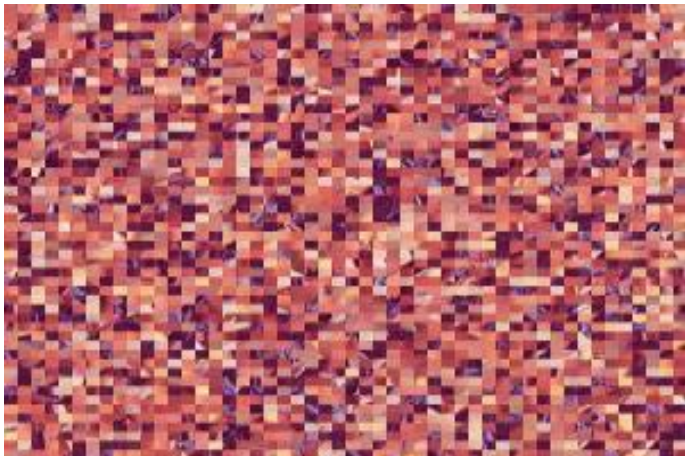

Fig.3b. Encrypted version

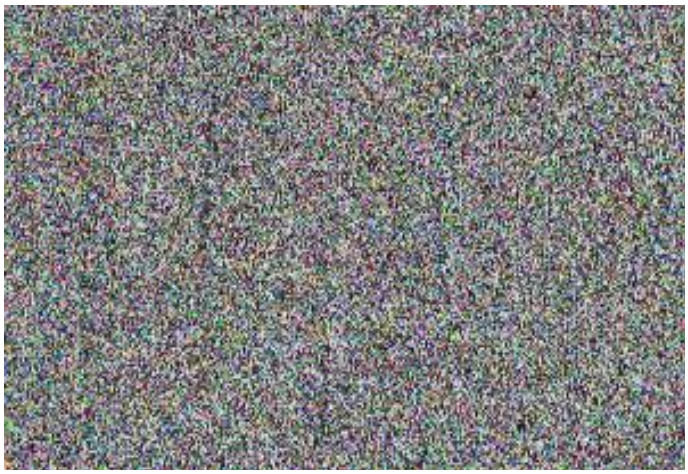

Fig.3c. Encrypted image containing embedded Exe File

Fig.3d. Decrypted image.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

Clearly, the embedding rate, a ratio between the data amount of net payload and the total number of cover pixels, is

$$R = \frac{N}{} = \frac{((N - Np) \cdot S/L - Np)}{L} \quad S \quad (3)$$

2.3. Data Extraction and Image Recovery

In this phase, we will consider the three cases that a receiver has onlythe embedding key, only the encryption key, and both the embedding key and encryption keys, respectively. With an encrypted image containing embedded data, if the receiver has only the embedding key, he may first obtain the values of the parameters M,L and S from the LSB of the Npselected encrypted pixels. Then, the receiver permutes and divides the other (N-Np). pixelsinto(N-Np)/L groups and extracts the S embedded bits from the M LSB-planes of each group. When having the total (N-Np). S/L extracted bits, the receiver can divide them into Nporiginal LSB of selected encrypted pixels and (N-Np).S/L-Np additional bits. Note that because of the pseudo-random pixel selection and permutation, any attacker without the embedding key cannot obtain the parameter values and the pixel-groups, therefore cannot extract the embedded data. Furthermore, although the receiver having the data hiding key can successfully extract the embedded executable file, he cannot get any information about the original image content.

III. Experimental Results

The test image Lena sized 512 x512 shown in Fig. 3(a) was used as the original image in the experiment. After image encryption, the eight encrypted bits of each pixel are converted into a gray value to generate an encrypted image shown in Fig. 3(b). Then, we M=3,L=128 and S=2to embed 4.4 $x10^3$additional bits into the encrypted image. The encrypted image containing the embedded data is shown in Fig. 3(c). With an encrypted image containing embedded executable file, we could extract the executable file using the embedding key. If we directly decrypted the encrypted image containing embedded executable file using the encryption key, the value of PSNR in the decrypted image was 39.0 dB, which verifies the theoretical value 39.1 dB calculated. The directly decrypted image is given as Fig. 3(d). By using both the executable filehiding and the encryption keys, the embedded data could be successfully extracted and the original image could be perfectly recovered from the encrypted image containing embedded data.

Decrypted Images and Pulse to Signal Noise Ratio in recovered images when different M,Land Swere used for images Lena and Man. The embedding rate is dependent on S and L, and the larger Sand the smaller correspond to a higher embedding rate. On the other hand, the smaller the values of Mand S, the quality of directly decrypted image is better since more data in encrypted image are not changed by data embedding. Here, the large M, L and the small S are helpful to the perfect content recovery since more cover data and less possible solutions are involved in the recovery procedure. If Mand Lare too small or Sis too large, the recovery of original content may be unsuccessful, and the values of PSNR in recovered images are also given in Tables II and III. It show the rate-distortion curves of the four images, Man, Lake and Baboon. Here, three quality metrics were used to measure the distortion in directly decrypted image: PSNR, the Watson metric and a universal quality index Q. While PSNR simply indicates the energy of distortion caused by data hiding, the Watson metric is designed by using characteristics of the human visual system and measures the total perceptual error, which is takes into account three factors: contrast sensitivity, luminance masking and contrast masking. Additionally, the quality index Qworks in spatial domain, as a combination of correlation loss, luminance distortion and contrast distortion. Higher PSNR, lower Watson metric or higher Qmeans better quality. In these figures, while the abscissa represents the embedding rate, the ordinate is the values of PSNR, Watson metric or quality index Q. Figs.4-6 show the rate-distortion curves of the four images Lena, Man, Lake and Baboon. Here, three quality metrics were used to measure the distortion in directly decrypted image: PSNR, the Watson metric and a universal quality index Q. While PSNR simply indicates the energy of distortion caused by data hiding, the Watson metric is designed by using characteristics of the human visual system and measures the total perceptual error, which is takes into account three factors: contrast sensitivity, luminance masking and contrast masking. Additionally, the quality index Q works in distortion and contrast distortion. Higher PSNR, lowerWatson metric or higher Q means better quality. In these figures, while the abscissa represents the embedding rate, the ordinate is the values of PSNR,watson metric or quality index Q. The curves are derived from different M, L and S under a condition that the original content can be perfectly recovered using the exe file hiding and encryption keys. Since the spatial correlation is exploited for the content recovery, the rate-distortion performance in a smoother image is better.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

The performance of the non separable method is also given in Figs. 4–6. It can be seen that the performance of the proposed separable scheme is significantly better than that existing system. We also compared the proposed scheme with the non separable method over 100 images sized 2520 x 3776, which were captured with a digital camera and contain landscape and people. When meeting the perfect recovery condition, the proposed scheme has an average 203% gain of embedded data amount with same PSNR value in directly decrypted image, or an average gain of 8.7 dB of PSNR value in directly decrypted image with same embedded data amount.

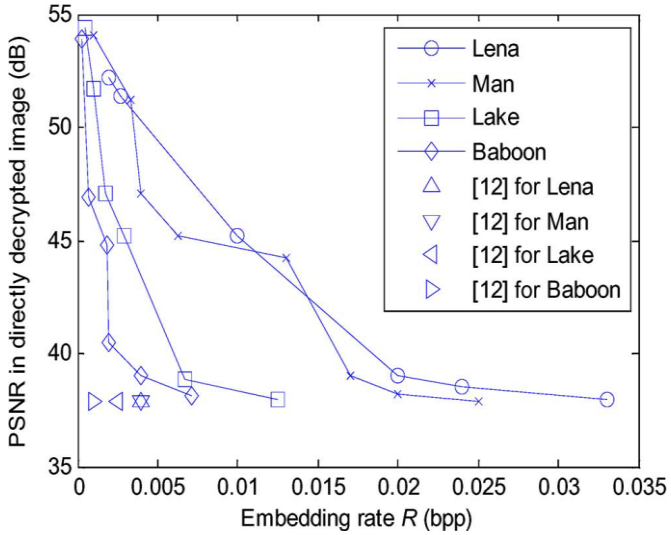

Fig.4 Rate-PSNR proposed scheme.

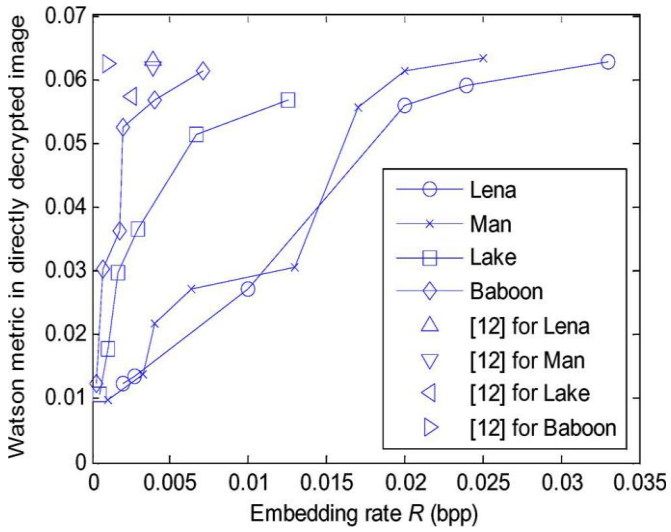

Fig.5 Rate-Watson metric proposed scheme.

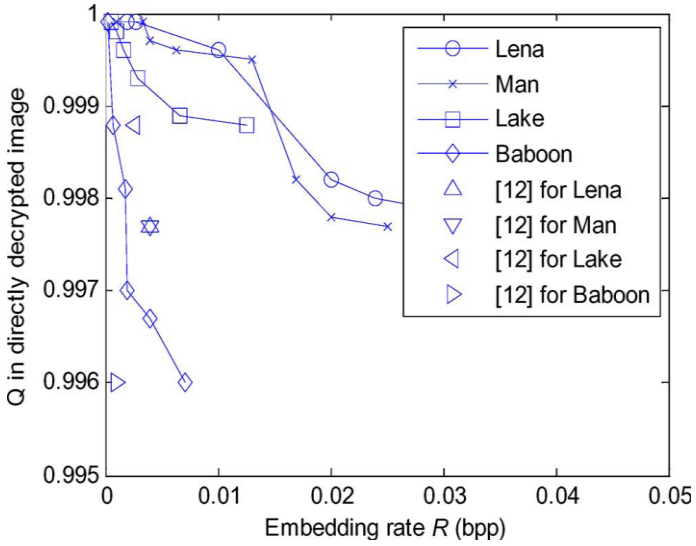

Fig. 6 Comparison between the proposed scheme and executable file hiding method.

IV. CONCLUSION

In this paper, a novel scheme for embedding of exe file in encrypted image is proposed, which consists of image encryption, Exe file embedding and exe fileextraction and imagerecovery phases. In the first phase, the content owner encrypts the original uncompressed image using an encryption key.

Acknowledgement

I must thank, first and foremost my Internal Guide Mr.C.Balakrishnan M.E.,(Ph.D) Assistant Professor, Department of Computer Science and Engineering and Project Coordinator Mr. Muthukumaraswamy M.E., without whose guidance and patience this dissertation would not be possible. I wish to record my thanks to Mrs. Umarani Srikanth M.E.,(Ph.D) Head of the Department, Department of Computer Science and Engineering, Project Panel Members, Professors of the Department of Computer Science and Engineering for their consistentence encouragement and ideas.

REFERENCES

[1] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K.Ramchandran, "On compressing encrypted data," IEEE Trans. SignalProcess., vol. 52, no. 10, pp. 2992–3006, Oct. 2004.

[2] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encryptedgrayscale images," IEEE Trans. Image Process., vol. 19, no. 4,pp. 1097–1102, Apr. 2010.

[3] X. Zhang, "Lossy compression and iterative reconstruction for encryptedimage," IEEE Trans. Inform. Forensics Security, vol. 6, no. 1,pp. 53–58, Feb. 2011.

[4] T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discreteFourier transform in the encrypted domain," IEEE Trans. Inform.Forensics Security, vol. 4, no. 1, pp. 86–97, Feb. 2009.

[5] T. Bianchi, A. Piva, and M. Barni, "Composite signal representation forfast and storage-efficient processing of encrypted signals," IEEE Trans.Inform. Forensics Security, vol. 5, no. 1, pp. 180–187, 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 imagesbased on additive homomorphic property," IEEE Trans. ImageProcess., vol. 14, no. 12, pp. 2129–2139, Dec. 2005.

[8] M. Deng, T. Bianchi, A. Piva, and B. Preneel, "An efficient buyer-sellerwatermarking protocol based on composite signal representation," inProc. 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.

[11] D. Kundur and K. Karthik, "Video fingerprinting and encryption principles for digital rights management," Proceedings IEEE, vol. 92, no. 6, pp. 918–932, Jun. 2004.

[12] X. Zhang, "Reversible data hiding in encrypted image," IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255–258, Apr. 2011.