AN EFFICIENT DATA HIDING IN ENCRYPTED IMAGE

PL.Subramanian
B.Manikandan

ABSTRACT

Data transmission is boon to communication. In having secured and efficient data transfer within allotted bandwidth, the compression and encryption technology are of vital importance. The way in which data is compressed and encrypted also plays major role for optimization. Having compression after encrypting the source yields better result than with other case. Security is enhanced by having two separate keys for data and image.

With an encrypted image containing additional data, if a receiver has the data-hiding key, it can extract the additional data though it does not know the image content. If the receiver has the encryption key, it can decrypt the received data to obtain an image similar to the original one, but cannot extract the additional data. If the receiver has both the data-hiding key and the encryption key, it can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image when the amount of additional data is not too large.

Index Terms: Data Hiding, Encryption, Decryption, correlation.

***Student, PG-Final year, University College of Engineering, BIT Campus, Trichy**
**** Assistant Professor, University College of Engineering, BIT Campus, Trichy**

I .INTRODUCTION

Encryption is the process of transforming information (referred to as plaintext) using an algorithm (called a cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The result of the process is information (in cryptography, referred as cipher text). The reverse process, i.e., to make the encrypted information readable again, is referred as decryption (i.e., to make it unencrypted).

The objective of image compression is to reduce irrelevance and redundancy of the image data in order to be able to store or transmit data in an efficient form. Image compression may be lossy or lossless. Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences may be called visually lossless.

Encryption, by itself, can protect the confidentiality of messages, but other techniques are still needed to protect the integrity and authenticity of a message; for example, verification of a message authentication codes (MAC) or a digital signature. Standards and cryptographic software and hardware to perform encryption are widely available, but successfully using encryption to ensure security may be a challenging problem. A single slip-up in system design or execution can allow successful attacks. Sometimes an adversary can obtain unencrypted information without directly undoing the encryption. See, e.g., traffic analysis, TEMPEST, or Trojan horse.

When doing compression before the encryption there is not improvement in compression gain. Moreover the information security plays a vital role than any other parameters in network communication. Considering all the factors in mind a solution must be attained that could satisfy security transmission as well as data compaction, which facilitates freer bandwidth for additional data transfer in the same time interval.

The source is first compressed to its entropy rate using a standard source coder. Then, the compressed source is encrypted using one of the many widely available encryption technologies. At the receiver, decryption is performed first, followed by decompression. In this paper, we investigate the novelty of reversing the order of these steps, i.e., first encrypting and then compressing the encrypted source, as shown in Fig. 2. The compressor does not have access to the cryptographic key, so it must be able to compress the encrypted data (also called ciphertext) without any knowledge of the original source. At first glance, it appears that only a minimal compression gain, if any, can be achieved, since the output of an encryption will look very random.

However, at the receiver, there is a decoder in which both decompression and decryption are performed in a joint step. A significant compression ratio can be achieved if compression is performed after encryption. This is true for both lossless and lossy compression. In some cases, we can even achieve the same compression ratio as in the standard case of first compressing and then encrypting. The fact that we can still compress the encrypted source follows directly from distributed source-coding theory.

II.EXISTING METHODS

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.

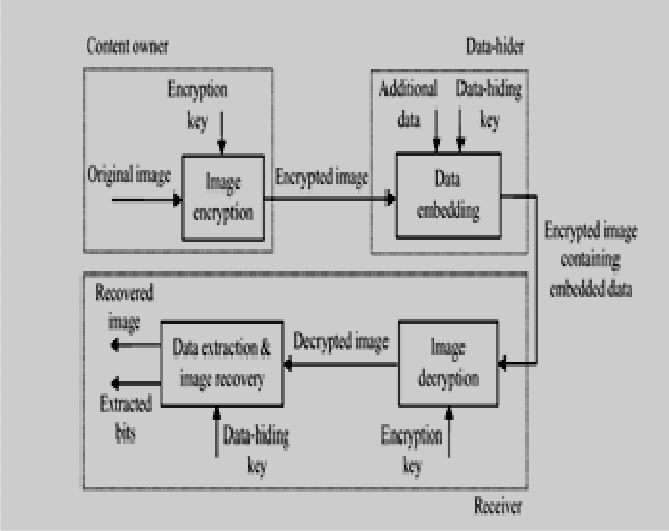

Fig.1 Existing encryption methodology

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 [3], a lossless compression method for encrypted gray image using progressive decomposition and rate-compatible punctured turbo codes is developed in [2]. With the lossy compression method 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.

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. 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.

III. PROPOSED METHODOLOGY

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. 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.

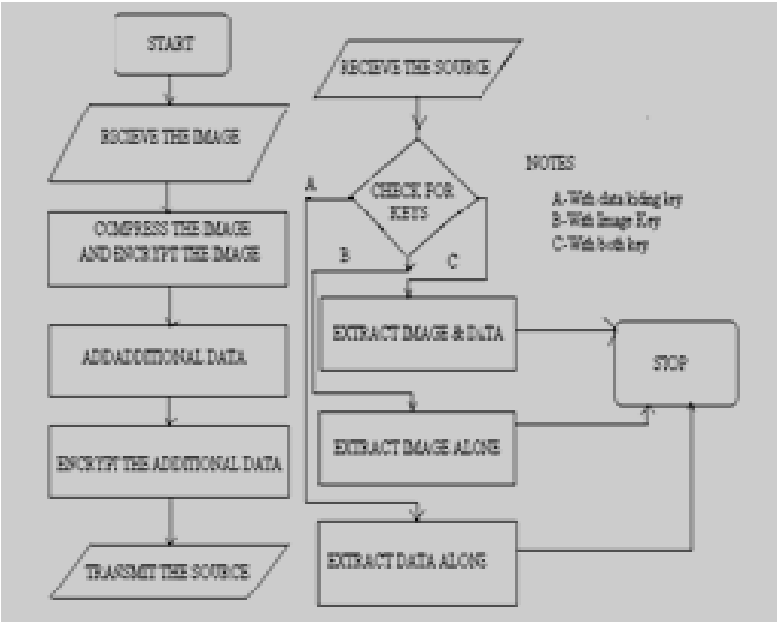

Fig.2 Flow diagram of proposed method

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. Fig. 2 shows the three cases at the receiver side.

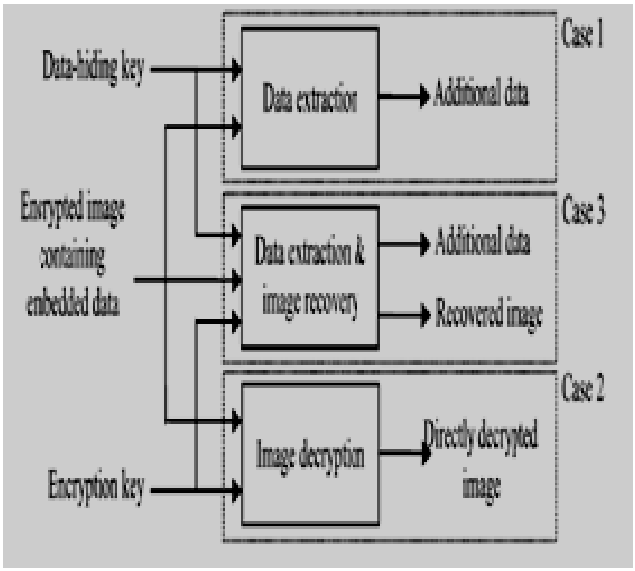

Fig.3 Proposed receiver architecture

We will consider the three cases as in fig 3 that a receiver has only the data-hiding key, only the encryption key, and both the data-hiding and encryption keys, respectively. With an encrypted image containing embedded data, if the receiver has only the data-hiding key, he may first obtain the values of the parameters M,L and S from the LSB of the N_p selected encrypted pixels. Then, the receiver permutes and divides the other N-N_p pixels into (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 N_p original LSB of selected encrypted pixels and (N-Np)*(S/L) - N_p additional bits. Note that because of the pseudo-random pixel selection and permutation, any attacker without the data-hiding 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 data, he cannot get any information about the original image content. Consider the case that the receiver has the encryption key but does not know the data-hiding key. Clearly, he cannot obtain the values of parameters and cannot extract the embedded data.

However, the original image content can be roughly recovered. Denoting the bits of pixels in the encrypted image containing embedded data as $B'_{i,j,0}$, $B'_{i,j,1}$, …….$B'_{i,j,}$($1<=i<=N_1$ and $1_{<=}j_{<=}N_2$), the receiver can decrypt the received data

$$b'_{ij,n}=B\oplus_{,n} \quad r_{ij,n}$$

where $r_{ij,n}$ are derived from the encryption key. The gray values of decrypted pixels are $P'_{ij},n=2^n(b'_{ijo+}b'_{ij1+}…….+b'_{ij})$.

Since the data-embedding operation does not alter any MSB of encrypted image, the decrypted MSB must be same as the original MSB. So, the content of decrypted image is similar to that of original image. The probability of this case is $(1/2^S)$, and, in this case, the

original(M*L-S) bits in the M LSB-planes can be correctly decrypted. Since S is significantly less than M*L, we ignore the distortion at other S decrypted bits. If there are nonzero bits among B(k,M*L-S+1), B(k,M*L-S+2),…… B(k,M*L), the encrypted data in the M LSB-planes have been changed by the data-embedding operation, so that the decrypted data in the M LSB-planes differ from the original data. The distortion in the N_p selected pixels is also ignored since their number is significantly less than the image size N. So, the value of PSNR in the directly decrypted image is

$$PSNR=10*(\log_{10}(A_E))$$

Where A_E is average energy of distortion. Table I gives the theoretical values of PSNR with respect to S and M.

If the receiver has both the data-hiding and the encryption keys, he may aim to extract the embedded data and recover the original image. According to the data-hiding key, the values of M, L and S, the original LSB of the N_p selected encrypted pixels, and the (N- N_p).(S/L)-N_p) additional bits can be extracted from the encrypted image containing embedded data. By putting the N_p LSB into their original positions, the encrypted data of the N_p selected pixels are retrieved, and their original gray values can be correctly decrypted using the encryption keys. In the following, we will recover the original gray values of the other (N- N_p) pixels. For each vector, we attempt to put the elements in it to the original positions to get an encrypted pixel-group and then decrypt the pixel-group using the encryption key. Denoting the decrypted pixel-group as G_k and the gray values in it as $t_{i,j}$, calculate the total difference between the decrypted and estimated gray values in the group. The estimated gray values is generated from the neighbors in the directly decrypted image.

Clearly, the estimated gray values in (16) are only dependent on the MSB of neighbor pixels. Thus, we have 2^S different D corresponding to the 2^S decrypted pixel-group G_k. Among the 2^S decrypted pixel-group, there must be one that is just the original gray values and possesses a

low D because of the spatial correlation in natural image. So, we find the smallest D and regard the corresponding vector as the actual vector and the decrypted $t_{i,j}$ as the recovered content. As long as the number of pixels in a group is sufficiently large and there are not too many bits embedded into each group, the original content can be perfectly recovered by the spatial correlation criterion. Since the 2^Sdifferent D must be calculated in each group, the computation complexity of the content recovery is $O(N.2^S)$. On the other hand, if more neighboring pixels and a smarter prediction method are used to estimate the gray values, the performance of content recovery will be better, but the computation complexity is higher.

IV.SIMULATION RESULTS

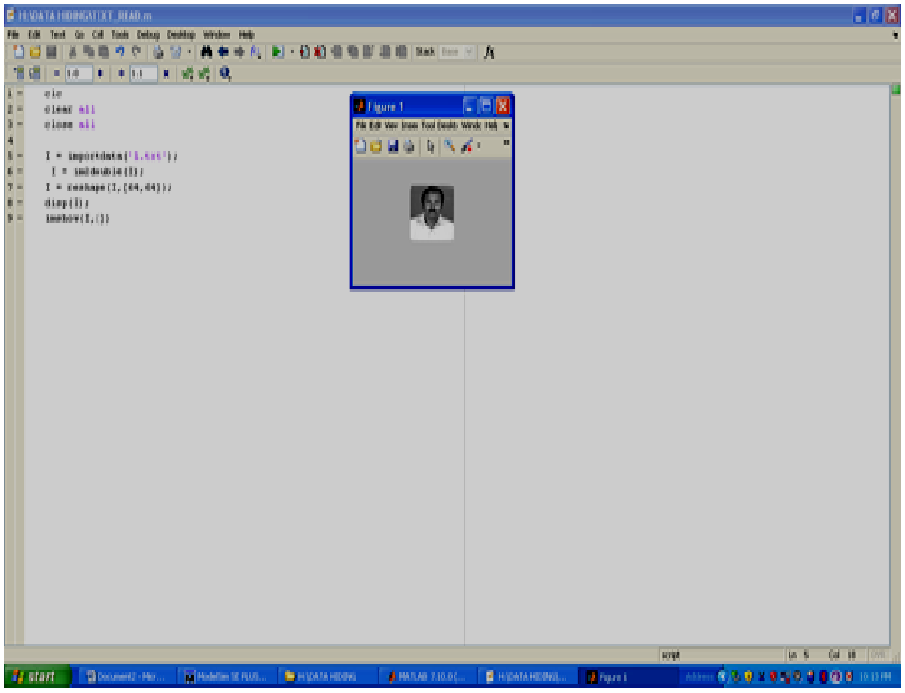

Fig.4 image before encryption and data hiding

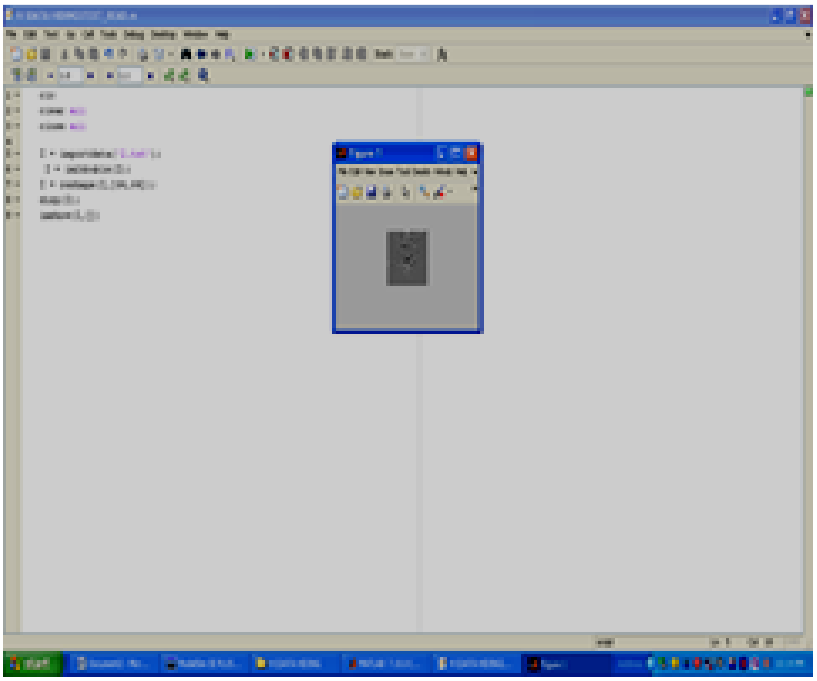

Fig.5 View of encrypted and data hided image

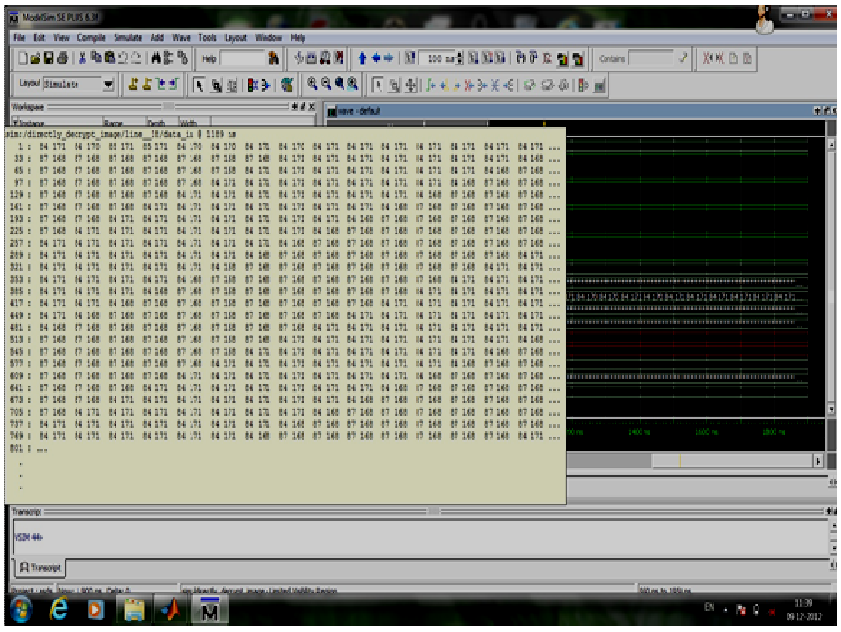

Fig.6 Image and data recovery in bit format

V. CONCLUSION

A novel scheme for separable reversible data hiding in encrypted image is proposed, which consists of image encryption, data embedding and data-extraction/image-recovery phases. In the first phase, the content owner encrypts the original uncompressed image using an encryption key. Although a data-hider does not know the original content, he can 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 receiver may extract the additional data using only the data-hiding key, or obtain an image similar to the original one using only the encryption key. When the receiver has both of the keys, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image if the amount of additional data is not too large. If the lossless compression method in [1] or [2] is used for the encrypted image containing embedded data, the additional data can be still extracted and the original content can be also recovered since the lossless compression does not change the content of the encrypted image containing embedded data. However, the lossy compression method in [3] compatible with encrypted images generated by pixel permutation is not suitable here since the encryption is performed by bit-XOR operation. In the future, a comprehensive combination of image encryption and data hiding compatible with lossy compression deserves further investigation.

VI. REFERENCES

[1] Xinpeng Zhang, “Separable Reversible Data Hiding in Encrypted Image” IEEE Transactions on information Forensics and Security,vol .7 No. 2, April 2012.

[2] 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.

[3] 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.

[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] 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. 180–187, 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 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.

[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.

[13] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[14] 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.

[15] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized- LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[16] W. Hong, T.-S. Chen, Y.-P. Chang, and C.-W. Shiu, "A high capacity reversible data hiding scheme using orthogonal projection and prediction error modification," *Signal Process.*, vol. 90, pp. 2911–2922, 2010.

[17] 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.