Data Embedded Using *n*-ary Histogram Modification In Compressed Encrypted Image

Ms.SANDHRA B SIDHARTHAN
M.Tech Applied Electronics & Instrumentation
Department of ECE
Younus College of Engineering & Technology,
kollam, Kerala
sandhra1989@gmail.com

Mr.NISHIL A
Asst Professor
Department of ECE
Younus College of Engineering & Technology,
kollam ,Kerala
rafi.nishil@gmail.com

Abstract-This work is based on lossy image compression and data embedded using n-nary histogram modification in compressed encrypted image. In this, the original image is compressed by lossy compression method and encrypted using the encryption key. After the compressed image is encrypted by the sender, the data-hider embeds the secret message into the encrypted image by a histogram modification and n-nary data hiding scheme. If the receiver has encryption key then he can recover the image after decompression, if data hiding key he can extract the data, if both data hiding key and encryption key then he can extract the data and recover the original image after decryption and decompression This method does not require the sender to reserve embedding room by himself, and can perfectly recover the original image. Compared to the existing reversible data hiding methods for encrypted images, the proposed method largely improves the embedding capacity.

Keywords- Lossy Compression, Encrypting an image, Data hiding, Recovery of image/ Extraction of data.

I. INTRODUCTION

Image security becomes increasingly important for many applications such as, military, confidential transmission and medical applications. Confidential data used for medical and military purpose are transmitted over the internet. Nowadays, the transmission of images happens frequently and it is necessary to find an efficient way to transmit them over networks .While using secret images, security issues should be taken into consideration because hackers may utilize weak link over communication network to steal the necessary information. Data security basically means protection of data from unauthorized users or hackers and providing high security to prevent data modification. This area of data security has gained more attention over the recent period of time due to the massive increase in data transfer rate over the internet.In order to improve the security features in data transfers over the internet, many techniques have been developed like: Cryptography and Steganography. While Cryptography is a method to conceal information by encrypting it to cipher texts and transmitting it to the intended receiver using an unknown key, Steganography provides further security by hiding the cipher text into a seemingly invisible image or other formats.

Encryption and compression technologies are the important to the efficient solving of network bandwidth and security issues. Encrypting the data and compressing it, so that the compressor will not have the knowledge of the encryption key so that the secrecy is maintained [1]. The analysis of lossless image where the image data undergoes stream-cipher based encryption before compression is developed in [2].

Original image is encrypted by pseudorandom permutation and then compressed. Iterative reconstruction is used to retrieve the values of coefficients of original image. In this lossy compression is used [3]. The difference between original image and embedded image is almost imperceptible from human eyes, loseless data embedding could be thought as a convert communication channel. In this they calculate the neighboring pixel values and select some different values for the difference expansion [4]. A new loseless data embedding technique can embed large amount of data by using the algorithm.

This technique allows complete recovery of original host signal and introduces only a small amount of distortion. The n-nary histogram modification is used. In [7] an embedded pixel value is generalized according to the difference between predicted pixel value and its original pixel value. It has great data capacity and quality of them is increased. The original image is encrypted using the encryption key and data is hided into the image using the data hiding key. A receiver first decrypt the image using the encryption key and then extracts the data and recover the image using the data hiding key in [8]. Fig.1 gives the separable reversible data hiding in an encrypted image. In this original image is encrypted using the encryption key and the data are hided into the encrypted image using the data hiding key. If the receiver has encryption key he can recover the image, if data hiding key he can extract the data, if both encryption and data hiding key he can recover the original image and extract the data without any error in [9].

This paper proposes a scheme on lossy image compression and data embedded using n-nary histogram modification in compressed encrypted image. The original image is compressed using the 2D Haar wavelet compression method and then the image is encrypted using the encryption key.

After the compressed image is encrypted by the sender, the data-hider embeds the secret message into the encrypted

image by a histogram modification and n-nary data hiding scheme. If the receiver has encryption key then he can recover the image after decompression, if data hiding key he can extract the data, if both data hiding key and encryption key then he can extract the data and recover the original image after decryption and decompression This method does not require the sender to reserve embedding room by himself, and can perfectly recover the original image. Compared to the existing reversible data hiding methods for encrypted images, the proposed method largely improves the embedding capacity also Compression ratio will be higher and file size will be reduced.

Fig1. Shows the block diagram of seperable reversible data hiding in encrypted image.

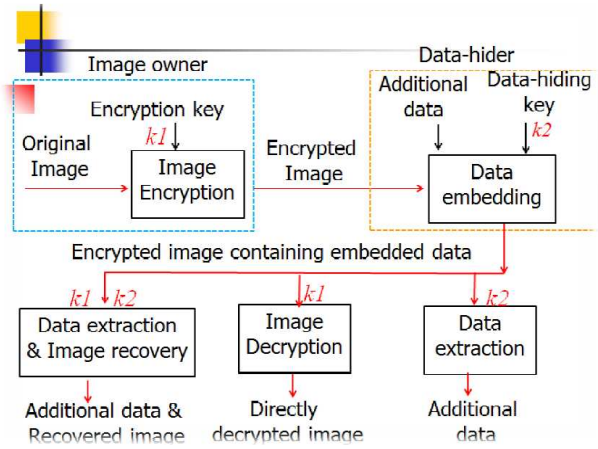

Fig1. Seperable reversible data hiding in encrypted image.

II. PROPOSED SCHEME

The proposed scheme consists of Haar wavelet compression, encrypting an image, Data hiding, Recovery of image and Extraction of data. The original uncompressed image is compressed using the Haar wavelet compression technique and the compressed image is encrypted using the encryption key. The data is hidden by histogram modification using a data hiding key. At the receiver side data hidden into the compressed image can be extracted using the data hiding key. The recovery of image is done after decryption. By using both data hiding and encryption key, data can be extracted from the histogram modification and the image is recovered. Fig.2 shows the sketch of lossy image compression and data embedded in compressed encrypted image.

A. 2D Haar Wavelet Compression:

To calculate Haar transform of array of n samples,

1. Find the average of each pair of samples. (n/2 averages)
2. Find the difference between each average and samples it was calculated from. (n/2 differences)
3. Fill the first half of the array with averages.
4. Fill the second half of the array with differences.
5. Repeat the process again to get 2D Haar wavelet transform.

The compression of image was done using the 2DHaar wavelet transform. By this the compression ratio is higher.

Advantages of Haar wavelet transform:

1. Best performance in terms of computation time.
2. Computation speed is high.
3. Simplicity.
4. HWT is efficient compression method

B. Image encryption

Assume the gray images with values belonging to [0, 255] for each pixel are used in the scheme. Denote the original image as **I** that sized $M \times N$. With a pre-defined key K_1, we first pseudo-randomly realign all the pixels into a disordered image **I'** which is semantically meaningless. However, histogram of the realigned image retains the same as the original. To avoid the information leakage of the histogram, we further define a mapping function associate with a new key K_h for pseudo-randomly reordering the indices of the histogram,

$$y=ENC(x, K_h), x,y \in [0, 255] \quad (1)$$

where the $ENC()$ is the mapping function to turn the intensity x into another intensity y by the key K_h. Thus, an encrypted image **E** is finally generated by the map-ping, $\mathbf{E}(a, b)=ENC(\mathbf{I}'(a, b), K_h)$, where $a=1,\ldots,M$ and $b=1,\ldots,N$.

After the encryption, the information of the contents and the histogram of the image are concealed. There are totally $(MN)!\cdot 256!$ possible permutations for the original image, it would be difficult for the data-hider or the adversaries to break the encryption. Record the keys K_p and K_h as the encryption key $\mathbf{K}_{enc}=\{k_1, K_h\}$.

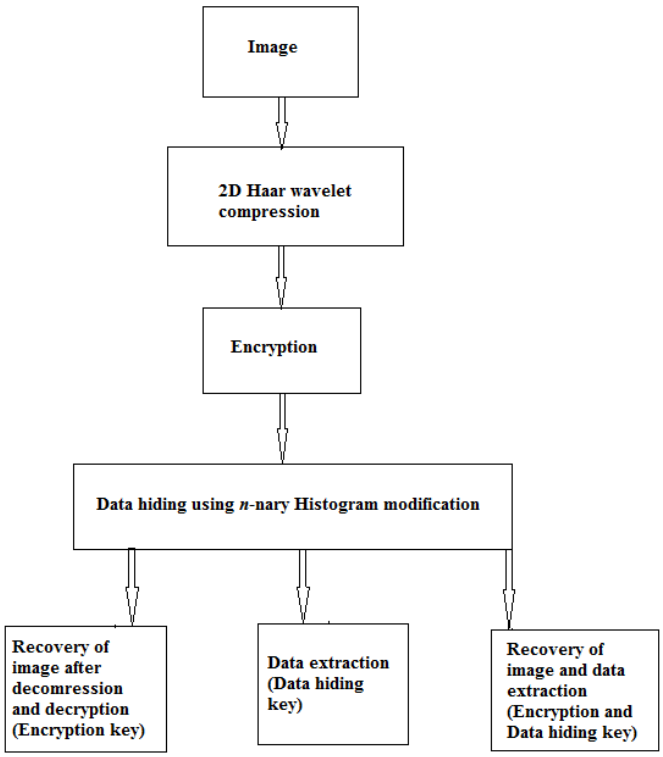

Fig2. Lossy image compression and data embedded using histogram modification in compressed encrypted image

C. Data hiding

After receiving the encrypted data **E**, the data-hider embeds the secret bits by modifying a small portion of the encrypted data. Firstly, the data-hider constructs the histogram of the encrypted image. Denote the histogram values for the encrypted image as h_i, i=0,1,…255. Find the intensity set that has maximum quantity in the histogram,

$$\mathbf{P} = \{\arg\max_i(h_i),\ i \in [0,255]\} \tag{2}$$

If **P** contains more than one member, discard the intensities except the first one, and generate a new set **P**={p_0} with only one element. Then the data-hider calculates

$$\mathbf{Q} = \{\arg\min_i(h_i),\ i \in [0,255]\} \tag{3}$$

where **Q**={q_0, q_1, …, q_{r-1}} is the set of intensities with minimum quantities. By merging the two sets, the data-hider generates a new set **T**=**P**∪**Q**. Intensities in the new set **T** are available for data hiding.

Accordingly, the data-hider uses an n-nary data hiding algorithm for information embedding. The value n is determined by the cardinality of the set **T**, that is n=| **T** |. Define the intensities in **T** as {t_0, t_1, …, t_{n-1}}, where t_0=p_0, t_i=q_{i-1}, i=1,2,…,n–1.

Generally, $h(q_i)$, the number of pixels that equals q_i, is equal zero or a small positive integer. If $h(q_i)$ is not equal to zero, the data-hider records the positions of the intensities that are equal to $h(q_i)$, and changes the corresponding values to p0 in the encrypted data. Denote these positions and the corresponding intensities as a sequence **W**.

After pseudo-randomly realign the secret bits by a new key K_s, the hider obtains the encrypted secret data that contains U bits **S**=b_0 b_1 … b_{U-1} where . Turn these bits into an n-nary array **S**$_{(n)}$=s_0s_1…s_{V-1} by

$$s_k = \left\lfloor \sum_{i=0}^{U-1} b_i \cdot 2^i / n^k \right\rfloor \bmod n,\quad k = 0,1,...,V \tag{4}$$

where V=h(p_0) and s_k∈[0, n-1].

Subsequently, modify all the intensities that are equal to p_0 in the encrypted im-age **E** to generate the stego **F** using equation (5). For example, to embed the n-nary value that equals k into the encrypted data, we find the intensity that equals p_0 and modify it to t_k.

$$\mathbf{F}(i,j) = \begin{cases} t_k & \text{if } \mathbf{E}(i,j) = p_0 \ \&\ s_l = k \\ \mathbf{E}(i,j) & \text{if } \mathbf{E}(i,j) \neq p_0 \end{cases} \tag{5}$$

Concatenating K_s, **T** and **W**, and compressing these data by a lossless compression algorithm, the embedding key K_{emb}={K_s, **T**, **W**} is generated.

Fig 2. shows the histogram of the pixel difference using laplacian distribution and the maximum value is very close to zero.

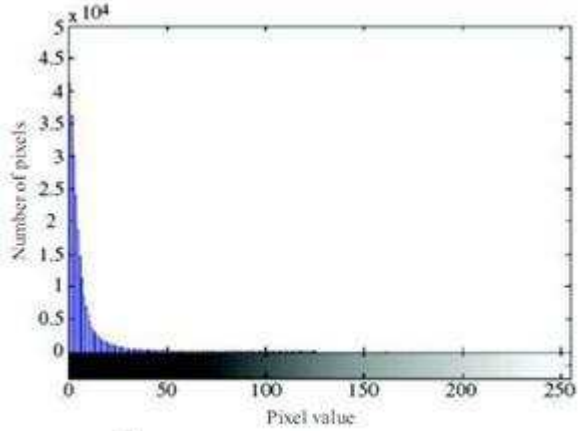

fig 3. Distribution of pixel differences.

D. Data extraction and image recovery

On the receiver side, the secret data can be extracted from the stego **F** with the embedding key K_{emb}. After decompressing the embedding key, data of K_s, **T** and **W** are recovered.

According to **T**, the n-nary data can be extracted by

$$s_i' = k,\ \text{if } \mathbf{F}(i,j) = t_k \tag{6}$$

where i=0,1,…V and k∈[0, n-1]. Then calculate the secret bits,

$$b_j' = \left\lfloor \sum_{i=0}^{V-1} s_i' \cdot n^i / 2^j \right\rfloor \bmod 2,\quad j = 0,1,...,U \tag{7}$$

By the key K_s, the receiver reorders these bits of b_k' to recover the original message.

Considering the receiver has both of the embedding key and the encryption key, he can further decrypt the received data and perfectly recovers the original image. The modified data of the encrypted image **E**' can be first recovered by

$$\mathbf{E}'(i,j) = p_0,\ \text{if } F(i,j) \in \mathbf{T} \tag{8}$$

Then, replace all the intensities on the positions recorded by **W** with the original values. This way, the encrypted data is totally restored. Next, recover the original image according to the encryption key **K**enc={Kp, Kh}. With the key Kh, the

mapping relationships are constructed again to change each data into the original intensity,

$$\mathbf{R}'(a, b)=ENC^{-1}(\mathbf{E}'(a, b), K_h), a=1,\ldots,M; b=1,\ldots,N \quad (9)$$

where *ENC*-1() is the reverse of the mapping function *ENC*(). Pseudo-randomly reorder **R'** using the key *Kp* to finally recover the original image **R**.

Additionally, if the receiver has only the encryption-key, content of the original image can also be approximately recovered after decompression. The receiver can use the key **K***enc*={*Kp*, *Kh*} to turn the stego **F** directly into an spatial image which is close to the original image. Although some "pepper & salt" noises will appear in the approximately decrypted image, main contents of the original image are well preserved.

III. RESULTS

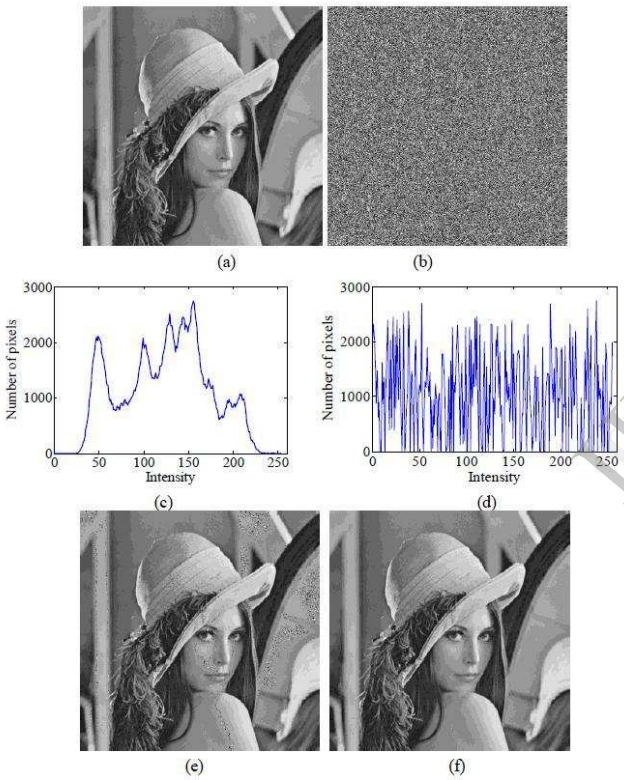

Fig.4. Loseless embedding in encrypted image "Lenna", (a) the original image, (b) the histogram of (a), (c) compressed and encrypted image where the data is embedded, (d) the histogram of (c), (e) the approximately recovered image, (f) the perfectly recovered image.

Experiments are conducted to verify the proposed method. The test images are standard gray images size 512×512. Fig. 4(a) is the original image "Lena", and Fig. 4(c) the corresponding histogram. After encrypting the compressed image with the encryption key, both the image and histogram are randomly distributed, which are shown in Fig. 4(b) and Fig. 4(d). Then, 14284 secret bits are embedded into the encrypted image by 42-nary data hiding using an embedding key. On the receiver side, the secret bits can be extracted without any errors if the embedding key is available. When the receiver has only the encryption-key, the original im-age can be approximately recovered, as shown in Fig. 4(e), with some salt & paper noises. If the receiver has both the embedding key and the encryption key, the original image can be perfectly recovered after decompression without any errors, which is shown in Fig. 4(f).

The image of Lena sized 512x512 is used as an original image. The image is compressed and encrypted using the encryption key and data is embedded into the compressed encrypted image. The coding is done using VHDL and Matlab. The image is directly decrypted by using the encryption key and the PSNR is 34.07. Table 1 explains the comparison of PSNR, compression ratio.

Table 1. Comparison of PSNR , Compression Ratio of existing and proposed method

Method	Parameter	Compression Ratio	PSNR of directly decrypted image	PSNR of recovered image
Proposed	M=2, L=15, S=2	0.02	34.0731	62.5
Existing	M=2, L=15, S=2	0.05	19.569	39.09

Table 2.Embedding capacity comparison Proposed method

Images	Method [9] (bits)	Method [10] (bits)	Method [11] (bits)	Proposed (bits)	Proposed *n*-nary
Lena	1024	1024	8650	14284	42
Man	655	1024	6554	46557	57
Lake	655	1024	3408	14828	19
Baboon	256	334	1966	13056	31

Embedding capacities of some images are listed in Table 2. We compare the proposed method to the methods in [9]~[11]. Results show that the proposed method largely improves the embedding capacity

IV. CONCLUSION

In this paper the proposed method is compressing the image and then the image is encrypted using the encryption key. The data is hided by histogram modification into the compressed encrypted image using the data hiding key. If the receiver has encryption key then the image can be directly decrypted after decompression of an image, if data hiding key then receiver can extract the data from the compressed encrypted image, if both data hiding and encryption key the data can be extracted and the image is recovered after the decompression and decryption of an image. 2D Haar wavelet compression is used here so the compression ratio will be higher and file size will be reduced while transmitting and also improves embedding capacity.

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

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

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

[5] 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.

[6] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber,

[7] "Lossless generalized-LSB data embedding," IEEE Trans. Image Process., vol. 14, no.2, pp. 253–266, Feb. 2005.

[8] 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.

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

[10] Hong W, Chen T, Wu H (2012) An improved reversible data hiding in encrypted images using side match, IEEE Signal Process. Lett., 19(4):199–202

[11] Zhang X (2012) Separable reversible data hiding in Encrypted image, IEEE Trans. Inf. Forensics Security,