Separable Reversible Data Hiding using Chaotic Encryption and Histogram Shifting

Arun K Mohan, Saranya M R, K Anusudha
Department of Electronics Engineering
School of Engineering & Technology
Pondicherry University
Puducherry, India.
arun.k.mohan@ieee.org, saranyamr.pu@gmail.com, anusudhak@yahoo.co.in

***Abstract* — In recent years, the chaotic system based cryptographic algorithms have suggested some new and efficient ways to develop secure image encryption techniques. Likewise, a new technology called Reversible Data Hiding (RDH) has been gaining popularity since its inception in the beginning of twenty first century. Here we propose a new approach by clubbing the two different techniques for achieving an algorithm for secure image transfer while concealing some data in the communication and discloses only upon the verification of the authenticity of the destination node. Beauty of the proposed system is that at the content owner side, sequential processing of the two algorithms is mandatory whereas at receiver the reverse process can be done in any succession. Content owner first encrypts the image by using user defined key derived chaotic sequence with the help of standard logistic map function. Then the data hider embeds the data by histogram modification method. By exploiting the advantage of histogram shifting, this method can provide minimum bound of Peak Signal to Ratio (PSNR) value as 48.13 dB, which is much higher than the existing methods. Zero error in the retrieved information, higher quality of recovered cover image and directly decrypted marked stego-image, better embedding capacity and etc. are achieved with the proposed method.**

Keywords— Reversible Data Hiding (RDH); Image Encryption; Key derivation; Chaos; Logistic map; Decryption; Histogram shifting.

I. INTRODUCTION

In the last decades, globe have witnessed an explosive growth in the frequent flow of digital images through the transmission Medias, now its security is very vital. Many applications like military images, confidential video conferencing, medical imaging system, cable TV, online personal photograph album, etc. require reliable, fast and robust security system to store and transmit digital images. To prevent all the privacy threats and fulfil all the security needs researchers in the cryptography branch of study have developed efficient set of encryption schemes. Cryptography basically does the scrambling of data for ensuring secrecy, as well as authenticity of the information. It enables us to transmit data across insecure networks so that it cannot be interpreted partially or completely by malicious end, or other than the authorized recipient. Two main branches of cryptography are cryptology and cryptanalysis, Cryptology is to keep plaintext secret from intruder while cryptanalysis deals with the defeating techniques from forging information in between. In cryptography, we usually employ encryption schemes to prevent any data from unauthorized access. In last few years, numerous encryption algorithms [1] have been proposed in the literature by utilizing different techniques. Among them, chaotic system based encryption techniques are practically used as these techniques provide a good amalgamation of high security, reasonable computational overheads, speed, computational power and complexity. Strong correlation among adjacent pixels, redundancy of data, being less sensitive as compared to the text data are the desirable characteristics of a digital image. Since traditional data encryption algorithms require large computational time and high computing power, IDEA, AES, DES, RSA etc. are not suitable for real time image encryption. Only those ciphers which take a lesser amount of time and at the same time without compromising security are only preferred for real time image encryption schemes. Systems with higher degree of security aspect and lower computational speed are obsolete in practical.

In the very beginning of twenty first century researchers are attracted towards a new scheme called reversible data hiding (RDH) in encrypted images. RDH is a technique to embed a secret message into some distortion unacceptable cover media like military images, medical images, and etc. in such a way that the cover image can be perfectly restored after extracting the secret message. There are a number of schemes which perform data hiding and encryption jointly. In some of them, a part of the cover is used to carry additional data and the rest of the cover is encrypted. For example, in [2], watermark is added to the amplitude of DCT coefficients, and motion vector difference, intra-prediction mode and signs of DCT coefficients are encrypted. A reversible data hiding technique in the encrypted image is described in [3], which hides data into completely encrypted image. But in this method, image decryption and data extraction are not separable. The method in [4] hides data into an encrypted image in a separable manner. A variety of schemes have been proposed since the inception of RDH to perform data hiding techniques in encrypted image, [5] has adopted difference expansion technique. In this method, one bit can be embedded into two consecutive pixels. Maximum embedding capacity is 0.5 bpp (bits per pixel). Later this method was generalized and the embedding capacity has been improved to (n-1)/n bpp. A different scheme of reversible data hiding, called reserving room before encryption is discussed in [6]-[7]. Another domain of RDH is histogram based method. [8]- [10] cover different methods under this technique. A new technique for RDH is implemented in [11] by estimating the errors. Methods of [5], [6], [7], [11] deal with RDH in separable Manner.

Proposed system has deployed two different techniques to achieve better security and data hiding. One is image encryption which is adopted from cryptography and another one is steganography. In this system, the content owner encrypts the cover image (here onwards cover) using the chaotic system based encryption scheme. Security enhancement is provided by introducing large key space and sensitive key, using derivation mechanism. Encrypted image has unaltered statistical parameter histogram; hence data hider without any knowledge about the image content hides some data into the cover by utilizing this advantage of chaotic based encryption schemes. Hence, the confidentiality of image content has been maintained. At the receiver side, the decryption and data extraction can be performed in any succession according to the availability of the key. Beauty of the system is that at the content owner side, order of processing the algorithm are same but at the receiver side it has maintained separable processing order. That is, if the receiver has the data hiding key, then the hidden data can be extracted out but not the cover and vice versa.

II. PROPOSED SYSTEM

Here we propose a separable reversible data hiding technique in encrypted images using histogram shifting method. Overall block diagram of the proposed system is shown in the figure below.

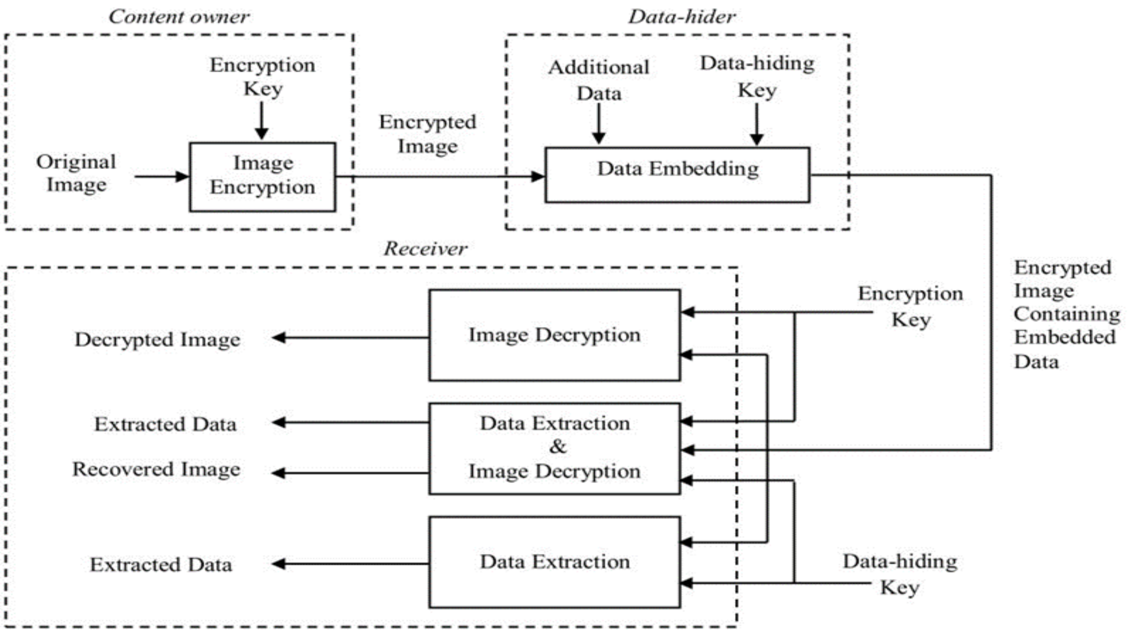


Fig. I. Complete Block Diagram of Proposed System [8]

In the upcoming sections, we will discuss each module deployed throughout the development. Basically, we have divided our system into three modules namely image encryption module, data hiding module and separable inversion module. Separable inversion module is categorized into three different modules namely image decryption module, data extraction module and data extraction & image decryption module. First two modules are deployed at the content owner side whereas the last one is deployed at the receiver side.

A. Image Encryption

In this section, we discuss the step by step procedure of the proposed image encryption, consider a grayscale cover of size m×n, whose pixel values fall in the range [0,255]. Scan the cover into a 2-D matrix. We have used a logistic map function based chaotic system for the encryption of the cover. Least complexity of the equation which produces most complex chaotic behavior with the involvement of least computation makes the logistic map function best fit for the image encryption schemes. A sequence of pseudo random number of size m×n is generated using the logistic map function in equation (1).

$$a_{p+1} = \mu \times a_p \times (1 - a_p) \quad (1)$$

Where μ lies in the range [0, 4] but produces best chaotic behavior in the range [3.57, 4]. In our development, we have used the optimized μ value 3.9999[12]. a_p is the value of the chaotic system at p^{th} iteration. Initial value of the chaotic system is calculated from the 80 bit user defined key using key derivation algorithm. We deployed two different key calculation mechanisms available in the literature and combined mean have been chosen as the initial value of the logistic map function. In the first stream, we have divided the key into two blocks of 8 bit each, indicated as session keys.

$$K= (k1,\ k2,k3.....k20)\ (\text{in hexadecimal}) \tag{2}$$

K values are the alpha numeric characters in from (A-F) & (0-9). Each group of two alphanumeric characters represents a session key. Alternatively, this can be represented in ASCII mode as

$$K = (K1,\ K2,...K10)\ (\text{in ASCII}) \tag{3}$$

Here K represents one 8 bit block of secret key, *ie*, nothing else session key. Initial value calculated from the first branch is denoted by X_0, for the calculation we utilized the following procedure. Chosen the group of session keys in succession $K_4K_5K_6$ and convert them into binary strings as follows,

$$B_1=K_{41}\ K_{42}...K_{48}\ K_{51}\ K_{52}...\ K_{58}\ K_{61}\ K_{62}...K_{68} \tag{4}$$

Here K_{ij} is the binary bit of ith block of the session key. X_{01} is derived as follows,

$$X_{01}=(K_{41}\times2^0+K_{42}\times2^1...+K_{48}\times2^7+K_{51}\times2^9+K_{52}\times2^{10}+...K_{58}\times2^{15}+K_{61}\times2^{16}+K_{62}\times2^{17}...K_{68}\times2^{23})/2^{24} \tag{5}$$

Using equation (6) we developed another branch denoted by X_{02},

$$X_{02}=\sum(k_i/128) \tag{6}$$

Where k_i is the binary bit in the secret key. X_0 of the first branch is derived from the equation (7)

$$X_0= Mean(X_{01},\ X_{02}) \tag{7}$$

Now moving towards the calculation of initial value using second stream, for that we adopted the following method, keeping X_{01} as the initial value generate the first chaotic sequence. Using that we generated a sequence of 24 real numbers $f_1, f_2...f_{24}$ by iterating the first logistic map using the initial condition obtained by the above method. Keeping in mind that we have considered only those values, which fall in the interval [0.1, 0.9], the other values are discarded from the sequence. The real number sequence is converted into an integer sequence using the following formula,

$$P_k = int(23\times (f_k\text{-}1)/0.8)+1 \tag{8}$$

Where *k* lies in the range 1-24. As in the above steps, calculate initial condition Y_0 for the second stream of logistic map function. We choose three blocks of session keys i.e. $K_1\ K_2\ K_3$, and convert them into a binary string as:

$$B_2=K_{11}\ K_{12}...K_{18}\ K_{21}\ K_{22}...\ K_{28}\ K_{31}\ K_{32}...K_{38} \tag{9}$$

As in the above steps, here $K_{ij}s$ are the binary digits (0 or 1) of the ith block of the session key. The computed Y_{01} using the following method,

$$Y_{01}= (B_2)_{10}/2^{24} \tag{10}$$

Further we compute another real number Y_{02} using equation (11)

Chaotic Sequence is Generated with $X_{n+1} = X_n \times \mu \times (1 - X_n)$ Initial Seed Calculated $X_0 = 0.3601$

$\mu = 3.9999$ (Chosen for optimized chaotic behaviour)

$X_1 = X_0 \times \mu \times (1 - X_0)$ $X_1 = 0.3601 \times 3.9999 \times (1 - 0.3601)$ $X_1 = 0.9217$

$X_2 = X_1 \times \mu \times (1 - X_1)$ $X_2 = 0.9217 \times 3.9999 \times (1 - 0.9217)$ $X_1 = 0.2887$

$X_3 = X_2 \times \mu \times (1 - X_2)$ $X_1 = 0.2887 \times 3.9999 \times (1 - 0.2887)$ $X_1 = 0.8214$

.
.
.

$X_9 = X_8 \times \mu \times (1 - X_8)$ $X_9 = 0.9698 \times 3.9999 \times (1 - 0.9698)$ $X_1 = 0.1170$

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9
0.9217	0.2887	0.8214	0.5868	0.9699	0.1169	0.4132	0.9698	0.1170

Fig. II Illustrative Example for Chaotic Generation

$$Y_{02} = \left(\sum_{k=1}^{24} \underline{B_2[P_k]} \times 2^{k-1} \right) / 2^{24}, \qquad (11)$$

Initial value from the second stream of derivation has been evolved from equation (12)

$$Y_0 = Mean(Y_{01}, Y_{02}) \qquad (12)$$

Final initial seed to the actual chaotic system can be derived using equation (13)

$$X_{0\_Final} = Mean(X_0, Y_0) \qquad (13)$$

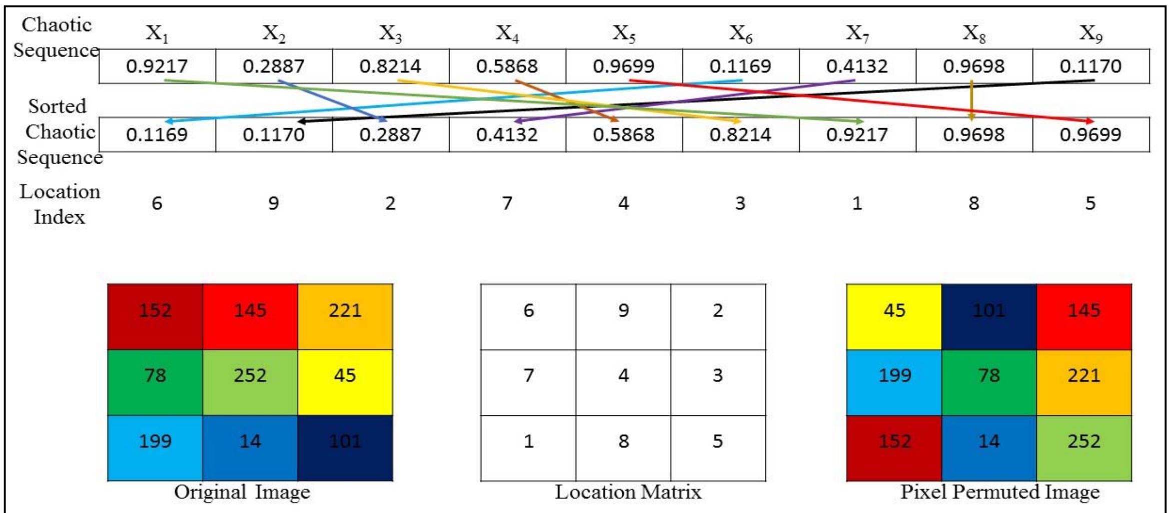
Now generate final chaotic system that can be used for the image encryption using equation (1) by feeding output evolved in equation (13). Generate N = m×n pseudorandom numbers using (1) and store them in a vector called Picture vector. Sort the Picture vector in the ascending order by noting the position changes during sorting to generate a location map LM. Arrange pixels of the image into a vector V. Shuffle V according to the location map LM. Rearrange vector V into a matrix sized m × n to get the encrypted image. Fig. II & III Illustrates image encryption steps.

B. Data Hiding

Since the shuffling based encryption does not alter the histogram of the original image, histogram shifting based method in [1] can be used to hide data in the encrypted image. The method follows:

1. Pseudo randomly permute the encrypted image pixels using data hiding key, for that use the same method deployed in encryption.
2. First few pixels are reserved for the embedding of header information which will be useful for the retrieval phase.
3. We included maximum grey level as the header information. Convert the grey level into eight bit binary stream and embed in the first eight pixels using single plain LSB substitution.
4. Histogram of remaining pixels is generated, and the grey level with the maximum pixel count is opted for embedding.
5. Introduce shift in the histogram, by providing a uniform unity accretion to the pixel grey levels in the range of grey levels of maximum pixel count and grey levels of minimum pixel count(zero is not considered for in this counting process). This process will leave out an empty space in histogram for the next successive gray level corresponding to the gray level having maximum number of pixels.
6. Convert the data to be embedded into the binary sequence.
7. Scan for the pixel grey levels of maximum pixel count & introduce a change in the grey level of the pixel according to the binary bit stream of data to be hidden. If the binary bit to be hidden is '1' then give a unity increment in the pixel grey level otherwise leave it as same.
8. Proceed the step for all the pixels satisfying the above prescribed criteria.
9. Inverse shuffle the image to retrieve the encrypted image.

Steps described above will embed the data reversibly in



	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9
Chaotic Sequence	0.9217	0.2887	0.8214	0.5868	0.9699	0.1169	0.4132	0.9698	0.1170
Sorted Chaotic Sequence	0.1169	0.1170	0.2887	0.4132	0.5868	0.8214	0.9217	0.9698	0.9699
Location Index	6	9	2	7	4	3	1	8	5

Original Image

152	145	221
78	252	45
199	14	101

Location Matrix

6	9	2
7	4	3
1	8	5

Pixel Permuted Image

45	101	145
199	78	221
152	14	252

Fig. III Illustrative Example for Image Encryption

Colors are used in illustrative example to provide better understanding, should not be confused with RGB image.

the encrypted cover. It is called marked stego-image, which is used in the transmission through the communication medium.

C. Data Extraction and Image Recovery

At the receiver side, according to the key supplied we have considered basically two processes and third one can be implemented by combining the basic two processes.

i. Data hiding Key alone supplied

ii. Encryption Key alone supplied

iii. Both data hiding & encryption key are supplied

i. Data Extraction and Image Recovery

When the receiver is supplied with the data hiding key and marked stego-image, the embedded data can be extracted out but the encrypted cover cannot decrypted. In order to extract the data, receiver has to follow the steps of data hiding module but in inverse. First receiver has to generate the chaotic sequence using the method described in image encryption module by utilizing supplied data hiding key. Using the chaotic sequence, perform pixel shuffling on marked stego-cover in order to obtain the header parameter and extract the embedded data. By reading the LSBs of the first eight bits obtain the grey level used for data hiding. Then sequentially scan for obtained grey level in the above process and unity accretion grey level of the obtained value, among the pixels other than the pixels used for header embedding. Whenever the grey level for maximum pixel count is encountered a bit ‘0’ is extracted and when unity accreted grey level corresponding maximum pixel count is encountered a bit ‘1’ is extracted. Those extracted bits are stored in the buffer and retrieved accordingly and generated meaningful embedded information. After the data extraction, provide a unity decrement to the pixels ranging from grey levels corresponding to maximum pixel count to grey levels corresponding to minimum pixel count, which will restore the shifted histogram. Then perform inverse shuffling to get the encrypted cover.

ii. Encryption Key alone supplied

When the receiver is supplied with encryption key and marked stego-cover, then the image similar to the original cover can be decrypted, but should not be able to read the embedded data. For this, key calculation mechanism & encryption strategy deployed in encryption module have been adopted. Generate the chaotic sequence using (1) with the help of already described method. Rearrange the pixels using the location map generated while sorting the chaotic sequence. This process can bring back all the pixels to their original positions. Only distortion in the decrypted image is a difference of 1 in grayscale value for those pixels used for data hiding. The lower bound of Peak Signal to Noise Ratio (PSNR) of this decrypted image can be proved to be larger than 48.13 dB as follows. It is observed, in the worst case, the value of every pixel differ by a value of 1 from their original value, hence Mean Square Error (MSE) of worst case is 1 and the lower bound of PSNR of the decrypted image containing hidden data is given by,

$$\begin{aligned} \text{PSNR} &= 10 \log 10\,(255^2/\text{MSE}) \\ &= 10 \log 10\,(255^2) \\ &= 48.13 \text{ dB} \end{aligned}$$

This resultant lower bound of PSNR is much higher than that of reversible data hiding in encrypted image techniques, [10] and [11], reported in the literature.

iii. Both data hiding & encryption key are supplied

When the receiver is supplied with both keys, both data extraction and cover recovery can be done in any order. *ie,* receiver can decide whether he want to decrypt the image first or extract the data first. According to that, receiver perform the above said operations.

III. EXPERIMENTAL RESULTS

We tested the proposed separable method with 4 commonly used test images namely Lena, Cameraman, Boat, and Lake using MATLAB 2012Rb. We implemented data embedding by utilizing only one peak point in the histogram, according to the requirement of embedding rates one can configure the number of peak points to be considered for embedding. When the receiver is having the encryption key only, he can decrypt the image and the PSNR of all directly decrypted images were observed to be above 48.13dB and verifies the theoretical result. When the receiver is having data hiding key only, he can extract the hidden data without any error. When the receiver is having both encryption key and data hiding key, he can extract the hidden data exactly and recover the image. The recovered image is exactly same as the original image as in [1]. Table I shows the images during different phases of the algorithm. Second column in the table projects that algorithm have ensured the privacy to the image content by encrypting the content of image before supplying to an external authority or channel administrator or a database manager for the data hiding work. One can easily note that directly decrypted image and original image have high degree of similarity or in other words algorithm maintains supreme quality of the cover image by introducing least distortion while embedding the data to the confused cover.

Table II shows the PSNR value of the directly decrypted images by utilizing single peak point embedding and multiple peak point embedding. For the better understanding and convenience only three level embedding values are shown in the table. Algorithm can support multiple peak point embedding by ensuring proper overflow suppression systems. From the table, one can arrive at the conclusion regarding the trade-off between the embedding capacity and PSNR value of the directly recovered cover image. Graphical representation of the above said relation is not presented in this work because it depends purely on the cover image.

Decrypted image after data extraction is exactly the same copy of the cover image and the result proves complete reversible nature of the proposed algorithm, which was obtained by considering irreversible header embedding by reversible embedding with the effective utilization of minimum gray level pixels.

TABLE I. RESULTS OBTAINED AT DIFFERENT STAGES OF ALGORITHM

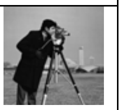
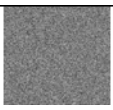
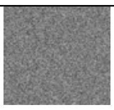
Original Image	Encrypted Image	Encrypted Image with Hidden Data	Directly Decrypted Cover	Retrieved Cover
				
				

TABLE II. MEASUREMENT OF EMBEDDING RATE AND PSNR BY CONSIDERING DIFFERENT PEAKS IN HISTOGRAM

		Boat	Cameraman	Lake	Lena
First Peak	Embedding Rates	0.0221	0.0257	0.0142	0.0088
	PSNR(dB)	58.4537	62.9540	63.0538	62.7082
Second Peak	Embedding Rates	0.0437	0.0490	0.0278	0.0175
	PSNR(dB)	54.8153	59.9023	59.8879	59.2499
Third Peak	Embedding Rate	0.0650	0.0716	0.0414	0.0258
	PSNR(dB)	52.5572	57.8779	57.7272	57.4110

Fig. IV – VI can provide a basic perception regarding the histogram domain. Fig. IV shows the original histogram of the cover image or it can be called as the histogram of the encrypted image since the encryption algorithm does not employ diffusion methods to conceal the image content. Histogram shifting process introduces an empty slot to bins of gray level MAX+1, which is clearly visible in Fig. V. The empty bin slots are filled once the embedding algorithm has executed. Allocation of the gray levels to the emptied one completely depends on the data to be embedded to the cover image. Fig. VI projects the above for random data embedding.

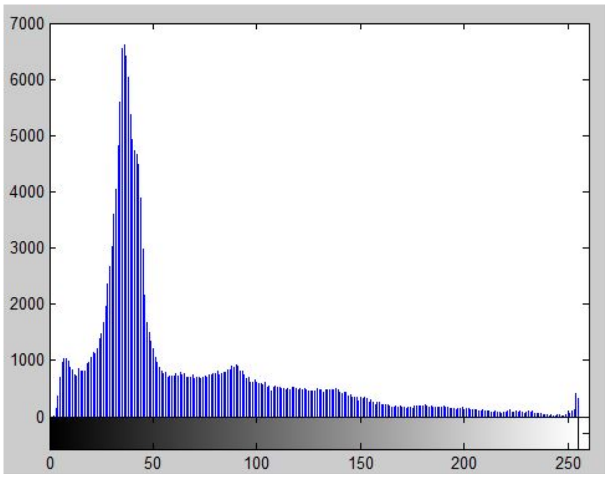


Fig. IV Original Image Histogram

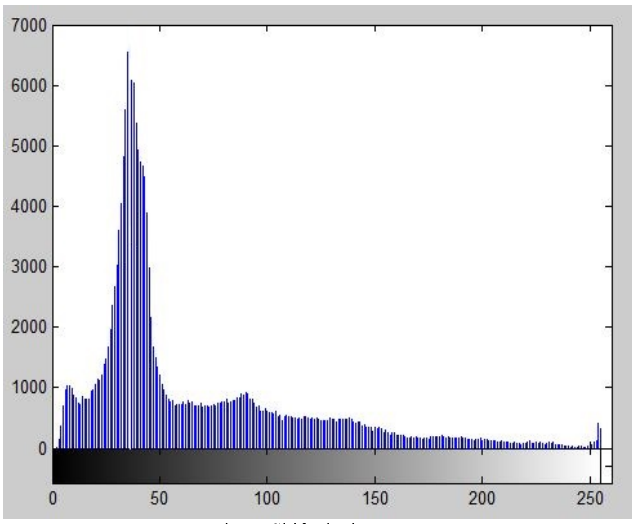


Fig. V Shifted Histogram

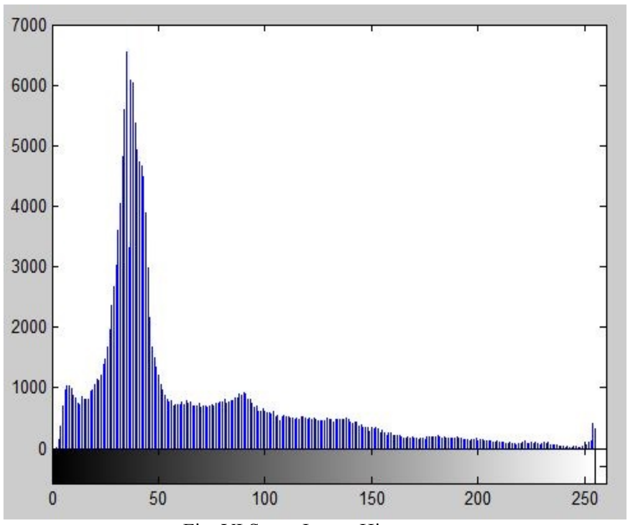


Fig. VI Stego-Image Histogram

IV. CONCLUSION

We proposed a novel separable reversible data hiding in encrypted image with improved performance. The method consists of image encryption, data hiding, and data extraction and image recovery phases. In the first phase, owner of the image performs encryption by chaotic permutation using encryption key. The data hider without knowing the original content can hide data into the encrypted image using data hiding key. For this, histogram modification based method is used. Embedding capacity of the proposed method is much higher than that of the existing RDH techniques in encrypted images. At the receiver side, data extraction and image recovery are performed in a separable manner. The receiver with data hiding key only can only extract the hidden data, but cannot decrypt the image. The receiver with encryption key can only generate an image similar to the original image by decryption, but cannot read the hidden data. The lower bound of PSNR for the decrypted image is 48.13 dB, which is much higher than that of the existing reversible data hiding techniques in encrypted image. If the receiver has both the keys, he can extract the data and recover the original image completely.

REFERENCES

[1] Nayak, Chinmaya Kumar, Anuja Kumar Acharya, and Satyabrata Das. "Image Encryption Using an Enhanced Block Based Transformation Algorithm." International Journal of Research & Reviews in Computer Science 2.2 (2011).

[2] Lian, Shiguo, et al. "Commutative encryption and watermarking in video compression." Circuits and Systems for Video Technology, IEEE Transactions on 17.6 (2007): 774-778.

[3] Zhang, Xinpeng. "Reversible data hiding in encrypted image." Signal Processing Letters, IEEE 18.4 (2011): 255-258.

[4] Chen, Ruiliang, et al. "Toward secure distributed spectrum sensing in cognitive radio networks." Communications Magazine, IEEE 46.4 (2008): 50-55.

[5] Zhang, Xinpeng. "Separable reversible data hiding in encrypted image." Information Forensics and Security, IEEE Transactions on 7.2 (2012): 826-832..

[6] Jun Tian, “Reversible data embedding using a difference expansion”, IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, No. 8, Aug. 2003.

[7] K. Ma, W. Zhang, X. Zhao, N. Yu, F. Li, Reversible data hiding in encrypted images by reserving room before encryption, IEEE Trans. Inf Forensic Secur. 8(3) (2013) 553-562.

[8] W. Zhang, K. Ma, N. Yu, Reversibility improved data hiding in encrypted images, Signal Process. 94 (2014) 118–127.

[9] Mohan, Arun K., M. R. Saranya, and K. Anusudha. "An algorithm for enhanced image security with reversible data hiding." Contemporary Computing and Informatics (IC3I), 2014 International Conference on. IEEE, 2014.

[10] Mohammad Ali Bani Younes, Aman Jantan, “An image encryption approach using a combination of permutation technique followed by encryption”, International Journal of Computer Science and Network Security, vol.8 No.4, April 2008.

[11] Saranya, M. R., Arun K. Mohan, and K. Anusudha. "A composite image cipher using DNA sequence and genetic algorithm." Contemporary Computing and Informatics (IC3I), 2014 International Conference on. IEEE, 2014.

[12] Xiaotian Wu, Wei Sun, “High capacity RDH in Encrypted Images by prediction error”, IEEE Signal Processing Letters, vol. 18, No. 4, Apr. 2011.

[13] N K Pareek, Vinod Patidar, K K Sud,“Image Encryption using Chaotic map,” Image & Vision computing, vol. 24, pp. 926–934 Feb.2006.

[14] Chang, I-Cheng, et al. "High capacity reversible data hiding scheme based on residual histogram shifting for block truncation coding." Signal Processing 108: 376-388, 2015.

[15] Marin, John, and Frank Y. Shih. "Reversible data hiding techniques using multiple scanning difference value histogram modification." Journal of Information Hiding and Multimedia Signal Processing 5.3: 461-474, 2014.

[16] Mathew, Nice, and A. Grace Selvarani. "A Novel Reversible Data Hiding Technique based on Histogram Shifting and Efficient use of Location Map." International Journal of Computer Applications 89.1 (2014): 25-29.