Secured Reversible Data Hiding In Encrypted Images Using Hyper Chaos

T.M. Amarunnishad *amarnishad@rediffmail.com*
Professor
T K M College of Engineering
Kollam, Kerala, India

Aslam Nazeer *aslamnaz06@gmail.com*
Research Scholar
T K M College of Engineering
Kollam, Kerala, India

Abstract

Recently, more and more attention is paid to reversible data hiding (RDH) in encrypted images, since it maintains the excellent property that the original cover can be losslessly recovered after embedded data is extracted while protecting the image content's confidentiality. All previous methods embed data by reversibly vacating room from the encrypted images, which may be subject to some errors on data extraction and/or image restoration. Here a novel method is proposed by reserving room before encryption with a traditional RDH algorithm, and thus it is easy for the data hider to reversibly embed data in the encrypted image. Moreover data to be embedded is shuffled using a hyper chaotic function which is difficult to be extracted from the stego image without original key. A digital water mark is also embedded which ensures integrity of the data. The proposed method has been validated against three other available RDH schemes and it is observed that the proposed scheme outperforms these RDH schemes both in visual quality and payload. The proposed method can achieve real reversibility, that is, data extraction and image recovery are free of any error.

Keywords : Reversible Data Hiding, Encrypted Image, Self-reversible Embedding, Digital Watermarks, Hyper Chaotic System, Peak Signal To Noise Ratio.

1. INTRODUCTION

Data transfer through internet is common now a days. Since internet is a public network, confidential data have to be secured during transmission. Steganography [1] is used in such cases. It is the process of hiding secret data into a carrier in order to convey secret messages confidentially. The carrier may be audio, video or digital images. Due to availability and convenience digital images are widely used as carriers. The original image is known as cover image and the image up on which data embedded is known as stego image. Due to data embedding some distortion occurs in the stego image and these distortions are known as embedding distortions. A good embedding algorithm produces only less embedding distortions.

There are two types of data hiding, Reversible and Non-reversible. In reversible data hiding (RDH), the original image is losslessly recovered after extracting the embedded data where as in non-reversible data hiding once the image is distorted it cannot be reconstructed back. RDH is used where the image as well as the data is equally important. This technique is widely used in medical imaging, military purpose etc.

There are many RDH techniques available now a days based on lossless compression like histogram modification [2], difference expansion (DE) [3] etc. Among these, histogram based techniques are attracted much. Histogram based methods modify the histogram so that secret data can be embedded in to the modified histogram. The first histogram based method is proposed by Ni et al [4] in which data is embedded in to the image based on zero/peak pixel value. This method is simple and execution time is short. The stego image quality is also high but embedding capacity (EC) is low and the algorithm does not work if the image is having a flat histogram. Moreover it has overflow or underflow problem.

Tian et al [5] proposed another RDH method based on DE. In this method the neighbour pixel value differences are calculated and some differences are selected for DE. The payload is embedded in the difference number. It explores the redundancy in the image. Zhang et al [6] used histogram flipping method to embed data by which the encrypted image is divided in to several blocks. The three LSB's of half of pixels of each block are flipped to embed the extra bit. Zhang et al [7] proposed another method in which space to embed data are created before encryption and data is embedded into those specified areas using LSB replacement mechanism. Zhang [8] proposed a separable reversible data hiding scheme in encrypted images. In this scheme, at first the content owner encrypts the original uncompressed image using an encryption key. Then, a data-hider may compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate some additional data. Thodi et al [9] proposed another technique in which a combination of histogram shifting as well as DE technique is used. Hu et al [10] proposed another technique based on DE. Jung et al [11] proposed another technique which uses interpolation method.

The objective of the proposed method is to develop an RDH scheme with increased security. In order to increase the security a chaotic function is introduced in the proposed technique. A digital watermark is also introduced in order to identify transmission error or any explicit modification done by any third party.

The rest of this paper is organized as follows. The proposed data hiding scheme is explained in section 2, experimental and theoretical analysis is presented in section 3, and finally conclusions in section 4.

2. PROPOSED METHOD

In the existing RDH techniques "vacating room after encryption (VRAE)" [12] is used in which the original image is encrypted using a standard cipher and an encryption key. Then the data hider embeds data into encrypted image using some RDH technique and the encrypted image is send to the receiver. The receiver or third party having encryption key can extract the image from encrypted image and those who possess data hiding key can extract the embedded data in the image. It means those who have both keys can extract both image and data. This method is illustrated in figure 1(a).

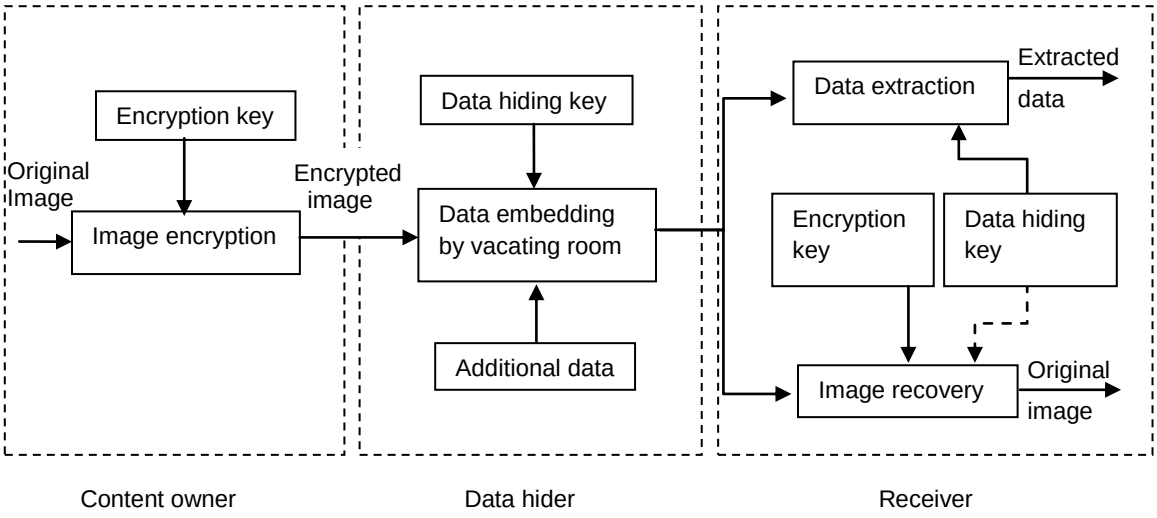
In the proposed method the order of creating space for data embedding and encryption is reversed. i.e space to embed data is reserved prior to image encryption and using some RDH technique data is embedded into these specified areas. This method is known as "reserving room before encryption (RRBE)" [7] scheme which is illustrated in figure 1(b).

In RRBE the content owner first reserves space in the original image and it is converted to encrypted form using an encryption key. The data hider needs only embedding of data on those reserved areas of the image. RRBE scheme primarily consists of four stages: generation of encrypted image, data hiding, image recovery and image extraction. The proposed RRBE technique is explained in the following subsections.

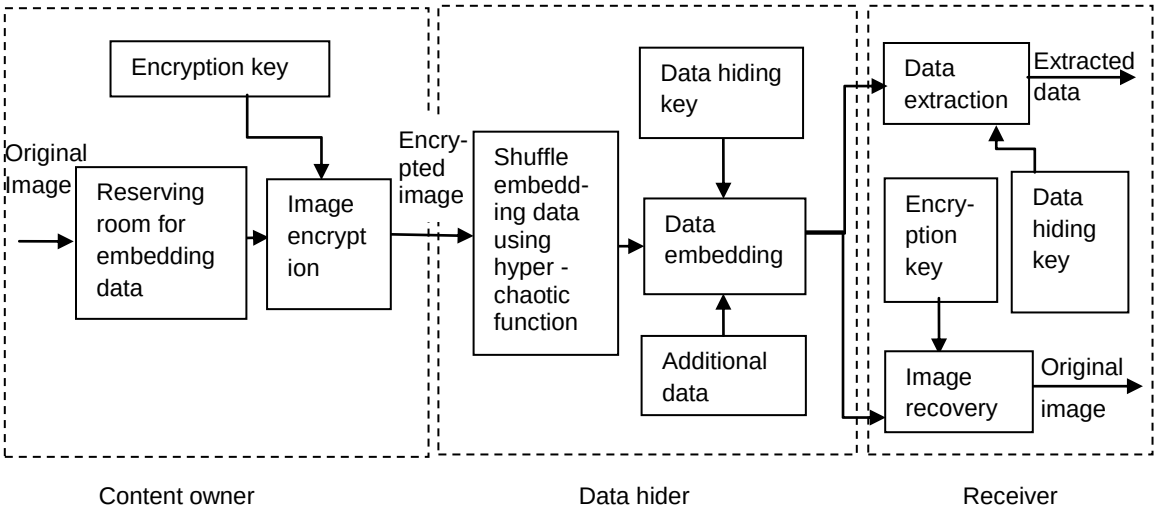
2.1 Generation of Encrypted Image

In order to produce an encrypted image using RRBE method, three steps are required: image partition, self reversible embedding and image encryption.

1) Image Partition: The goal of this step is to divide the image into two parts A and B using a smoothness function so that a smoother area B is constructed on which standard RDH algorithm [13], [14] can achieve better performance. Consider an original image I of size M x N and pixels $B_{i,j} \in [0,255]$, $1 \leq i \leq M$, $1 \leq j \leq N$. At first the size of to be embedded message is calculated and denoted as l. The original image is divided into several overlapping blocks along the rows, whose number is determined by l. Each block having m blocks, where m=$\lceil l/N \rceil$ and the number of blocks can be calculated by n= M-m+1.



(a)



(b)

FIGURE 1: Framework of (a) Vacating Room After Encryption and (b) Reserving Room Before Encryption.

Here each block is overlapped by previous or sub sequential block. A function is defined to measure the smoothness of each block

$$f = \sum_{u=2}^{m} \sum_{v=2}^{N-1} \left| C_{u,v} - \frac{C_{u-1,v} + C_{u+1,v} + C_{u,v-1} + C_{u,v+1}}{4} \right| \quad (1)$$

As the value of f increases, it means that those blocks contain complex structures. So the block with higher f is considered as A and puts in front of the image which is concatenated by rest part which is considered as B with fewer textured area, shown in figure 2.

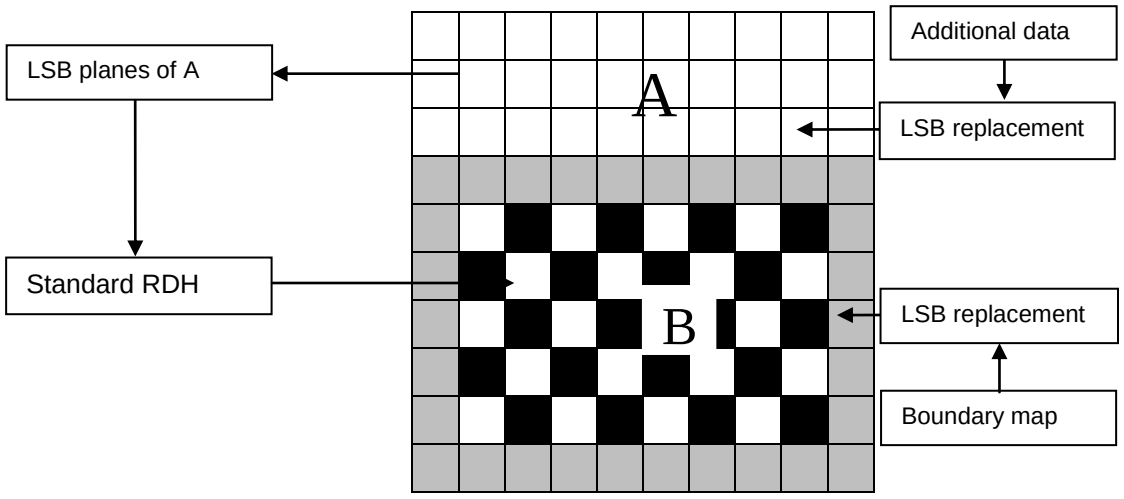


FIGURE 2: Illustration of image partition and embedding process.

2) Self-Reversible Embedding: The goal of this step is to embed the LSB planes of A into B using any RDH algorithm. Interpolation method is a commonly used RDH algorithm which is used here. Pixels in the area B are divided in to 2 sets: white and black. White pixels are those whose indices satisfy (i + j) mod 2 = 0 and black pixels are (i +j) mod 2 = 1. Then the value of each white pixel is estimated by interpolation value obtained with four black pixels surrounding it.

$$B'_{i,j} = w_1 B_{i-1,j} + w_2 B_{i+1,j} + w_3 B_{i,j-1} + w_4 B_{i,j+1} \quad (2)$$

where the weight w_i ,$1 \le i \le 4$, is determined by the same method proposed in [13]. The estimating error is calculated by $e_{i,j} = B_{i,j} - B'_{i,j}$ and then data are embedded into estimating error sequence using histogram shift. After calculating all values of white pixels the estimating error of black pixels are calculated by using modified white pixels and data are embedded into estimated error sequence of white pixels. Multilayer embedding is used if more data have to be embedded. For every single layered embedding two estimating error sequence are required for embedding messages. By using bidirectional histogram shift, the messages are embedded in the error sequence, ie. the estimated error histogram is divided into 2 parts, right and left. The highest point in each part is denoted as RM and LM and the zero point in each part is denoted as RN and LN. For ideal images RM = 0 and LN = -1. To embed messages into RM all values between RM+1 and RN-1 are shifted towards right by one step. The embedding process in the left part is similar except the shifting direction is left.

In this method the overflow/underflow problem is eliminated by embedding data only on those pixels whose values are between 1 and 254. Another problem arises when non-boundary pixels such as 1 is changed to 0 or 254 to 255.These newly created boundary pixels are known as pseudo boundary pixels. So a boundary map is maintained inorder to identify whether the boundary pixels are pseudo or natural. A binary sequence bit "0" is used to denote natural boundary pixel and "1" for pseudo boundary pixel. The marginal area is selected to embed boundary map. The parameters such as RN, LN, RM, LM, payload, start row, end row of A in original image are embedded into marginal area.

3) Image Encryption: After rearranging the self-embedded image it is encrypted by using any encryption algorithm. The encrypted image is denoted as X. After encryption a third party cannot access the image without encryption key, thus the privacy of the content is maintained.

2.2 Data Hiding in Encrypted Image

After encryption the encrypted image X is sent to the data hider. The data hider does not have any access to the real image. In the encrypted image, the region up on which data to be embedded are already identified and taken into front which is denoted as A_E .The data to embedded are shuffled randomly by using hyperchaotic function and embedded in A_E. The hyper chaotic function makes use of some keys which are known as data hiding key. One can't extract the data without data hiding key.

2.2.1 Chaotic functions

Chaos based encryption is first proposed in 1989 [15], [16] and after that many research works were appended in literature. The importance of chaotic function is "nearby" input does not generate "nearby" output. Recently, hyper chaos is widely used for encryption because it has more complex dynamical characteristics than chaos. In the proposed method hyper chaos is used to increase the security of the embedded data. The following hyper chaotic system is selected and it is used for generating random sequences

$$x_{n+1} = a_1 \times x_n + a_2 \times y_n \; ; \qquad y_{n+1} = b_1 + b_2 \times x^2{}_n + b_3 \times y_n \; ; \tag{3}$$

Here a_1 , x_n , a_2, y_n , b_1, b_2, b_3 can hold random values and which will be decided by user and that can be considered as encryption key and data hiding key. Here we set a_1=-0.95, a_2=-1.3, b_1=-0.45, b_2=2.4, b_3=1.05, x_n=0.0391 and y_n=0.019

2.2.2 Digital Watermarking

A digital watermark [17] is a kind of marker which is embedded in to the cover. It is typically used to identify ownership of the copyright. "Watermarking" is the process of hiding digital information in a carrier signal. Digital watermarks may be used to verify the authenticity or integrity of the carrier signal or to show the identity of its owners. The digital watermark should not alter carrier signal, it just marks data, but does not degrade it nor controls access to the data. A small digital watermarking is embedded into the marginal pixels of the cover image so that if any transmission error or any explicit modification done by a third party can be easily identified.

2.3 Data Extraction and Image Recovery

Data extraction is the reverse of data embedding and image decryption is the reverse of image encryption. If the receiver having both keys i.e. encryption key as well as data hiding key he can decrypt the image as well as extract the data. If he has only encryption key he can only decrypts the image, he can't extract data. If he having only data hiding key he can extract the data, but cannot decrypt the image.

3. EXPERIMENTS AND RESULTS

In this section, the simulation results and testing of performance of the proposed scheme by the key space and key sensitivity analyses are presented. All the experiments have been performed on a personal computer with a 2.4 GHz Intel Core2 i3 processor, 2G memory and 250 GB hard disk with a Windows 7 operating system. The proposed method has been tested on standard

publically available images such as Lena, Airplane, Barbara, Peppers and Boat and each image is of size 512 x 512

3.1 Key space analysis

An algorithm's key space refers to the set of all possible keys that can be used to generate a key, and is one of the most important attributes that determines the strength of a cryptosystem. In encryption algorithms the most effective attack is the Brute Force attack, where the enemy performs a complete search through all possible keys of the key space to find the right one. To prevent an adversary from using a brute-force attack to find the key used to encrypt a message, the key space is usually designed to be large enough to make such a search infeasible. On average, half the key space must be searched to find the solution. Another desirable attribute is that the key must be selected truly randomly from all possible key permutations. If keys were not randomly selected the attacker is able to determine some factor that may influence how the key was selected, so that the search space can be significantly reduced. Humans do not select passwords randomly; therefore attackers frequently try a dictionary attack before a brute force attack, as this approach can often produce the correct answer in far less time than a systematic brute force search of all possible character combinations. If a key were eight bits (one byte) long, the key space would consist of 2^8 or 256 possible keys.

Key space analysis handles maximum number of keys that can be used to find out the original key. Key length should be large to avoid brute force attack. In our proposed system we use a hyper chaotic function to determine the data hiding key. The key can be represented as (A, X1). A= (a_2, a_4, b_1, b_2, b_3) X1=(x_0, y_0). These keys are used for shuffling the pixel positions of the image. If we are using 32 bit number for generation of A and X1, then the key space of data hiding key is equal to the $2^{32} \times 2^{32} \times 2^{32} \times 2^{32} \times 2^{32} \times 2^{32} \times 2^{32} = 2^{224}$. Because of this large key space brute force attack is difficult.

3.2 Key sensitivity analysis

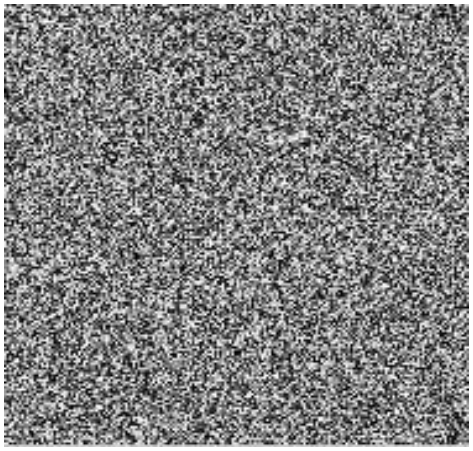
This subsection specifies sensitivity of data hiding key in proposed method. That means what changes will occur if a slight change is made in data hiding key. First, we hide some data into an image with key a_1= -0.95; a_2=-1.3; b1=-0.45; b2=2.4; b3=1.05; xn=0.0291; y_n=0.019; And then make a slight change with x_n=0.0391; y_n=0.019; So new key for data hiding is a_1= -0.95; a_2= -1.3; b_1= -0.45; b_2=2.4; b_3=1.05; x_n=0.0391; y_n=0.019. The recovery of data using incorrect key and correct key are shown in Table 1. When data is extracted using incorrect keys the extracted secret data is wrong. i. e only correct keys can extract original data. From this we can conclude that data hiding is sensitive to data hiding keys.

Figure 3 shows real time example of original image, encrypted image,decrypted image with data embedded in it and fully recovered image. The difference between original and extracted image is calculated. If the difference is zero it means that the image is losslessly recovered. Here the difference is zero which means the original image is losslessly recovered from stego image.

Data embedded using original key	Data extracted using original key	Data extracted using wrong key
"Hi…How are you"	"Hi…How are you"	"Ê¨ááGê¯cØi~"
"Who are you?"	"Who are you?"	"'ÉQwkJÛ/'"
"Hello world"	"Hello world"	"5AÇ • v/¾ • è"
"India is our country'	"India is our country"	"Ð²I¯È4Ã'Ì¹âQþ±"

TABLE 1: Key sensitivity analysis using original and wrong keys.

(a) Original Image



(b) Encrypted Image

(c) Decrypted Image

(d) Recovered image

FIGURE 3: Images after different steps of proposed RDH.

3.3 Implementation Issues

The peak signal-to-noise ratio (PSNR) is the objective criteria to find out the quality of the images after decryption. PSNR is the ratio between a signal's maximum power and the power of the signal's noise. Each picture element (pixel) may get changed when an image is modified. Logically, a higher value of PSNR is good because it means that the signal to noise ratio is higher. Signals can have a wide dynamic range, so PSNR is usually expressed in decibels, which is a logarithmic scale. PSNR values of the test images are recorded and evaluated inorder to check the quality of images and to check the efficiency of the proposed algorithm. To achieve high PSNR the following measures have to be taken.

3.3.1 Choice of LSB Plane Number

According to the proposed algorithm at first the image is divided into two parts A and B. The size of A is determined by the size of the message to be embedded and also by the number of LSB planes reversibly embedded in B. The choice of multiple LSB planes increases the size of B with an increase in embedding capacity. Table 2 shows the PSNR comparison between three different choices of LSB planes for five test images under various embedding rates measured by bits per pixel (bpp). From table 2 we can find that single LSB plane is better at a low embedding rate of

less than 0.2 bpp. For an embedding rate of 0.2 bpp and beyond, the choice of multiple LSB planes flips between 2 and 3 for a longer PSNR value.

PSNR results									
Embedding rate (bpp)		**0.005**	**0.01**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
Lena	1 LSB-plane	67.31	63.80	56.45	52.34	49.07	45.00	40.65	35.96
	2 LSB-plane	66.78	62.65	55.14	51.76	48.99	45.10	42.56	38.74
	3 LSB-plane	64.62	60.94	53.30	50.06	46.98	43.98	43.98	39.06
Airplane	1 LSB-plane	65.79	62.90	57.40	54.05	49.90	48.26	44.87	40.72
	2 LSB-plane	65.31	61.12	56.33	53.79	47.68	48.10	45.45	42.12
	3 LSB-plane	63.20	59.81	53.90	50.44	47.10	45.79	43.12	42.05
Barbara	1 LSB-plane	63.78	62.72	55.56	51.46	45.68	43.56	39.24	34.00
	2 LSB-plane	62.71	62.01	54.72	50.71	40.51	43.70	40.78	36.45
	3 LSB-plane	60.13	61.10	53.16	49.68	46.00	42.81	40.34	37.06
Peppers	1 LSB-plane	63.48	61.82	54.17	51.02	46.00	42.08	36.16	____
	2 LSB-plane	63.17	61.03	53.50	50.50	46.16	42.65	39.47	34.57
	3 LSB-plane	62.14	60.33	52.22	49.18	45.43	42.10	39.40	36.34
Boat	1 LSB-plane	66.55	64.42	56.75	52.71	49.10	45.21	41.44	36.60
	2 LSB-plane	65.91	63.34	55.75	51.02	48.40	44.98	42.26	40.87
	3 LSB-plane	64.17	61.81	53.73	50.45	46.71	43.81	41.75	39.44

Table 2: PSNR comparison for three different LSB-plane choices under various embedding rates.

5.3.2 Boundary Map

Boundary map is used to distinguish between natural and pseudo boundary pixels. Its size is a criteria to the applicability of the proposed approach. In most cases no boundary map is needed. Table 3 shows the boundary map size of the five standard images. The marginal area of the image must be large enough to record the boundary map.

From table 3 we can find that the images Lena, Airplane, Barbara and Boat are not using any boundary map for various embedding rates. But in the case of pepper image, up to an embedding rate of 0.4 bpp it can hold boundary map. Beyond that embedding rate for pepper, it does not have enough marginal pixels to hold boundary map. So embedding data in image pepper beyond 0.4 bpp is not possible in 1-LSB plane.

Boundary map size (bits)								
Embedding rate (bpp)	**0.005**	**0.01**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
Lena	0	0	0	0	0	0	0	0
Airplane	0	0	0	0	0	0	0	0
Barbara	0	0	0	0	0	0	0	0
Peppers	0	1	43	92	291	797	1741	____
Boat	0	0	0	0	0	0	0	0

Table 3: Length of boundary map under various embedding rates.

3.4 Performance Comparison with other RDH Schemes

For testing the performance of the proposed system, three other available RDH schemes from literature, Ni et. al [4], Tian [5] and Zhang [8] have also been implemented for the five test images: Lena, Boat, Barbara, Peppers and Airplane. The performance results for each test image are graphically shown in figure 4 with embedding rate (bpp) is taken on x-axis and PSNR values are taken on y-axis.

The maximum embedding rates of RDH scheme [4], [5] and [8] are 0.005, 0.4 and 0.025 respectively (figure 4). Since the implemented RDH schemes [4], [5] and [8] support data

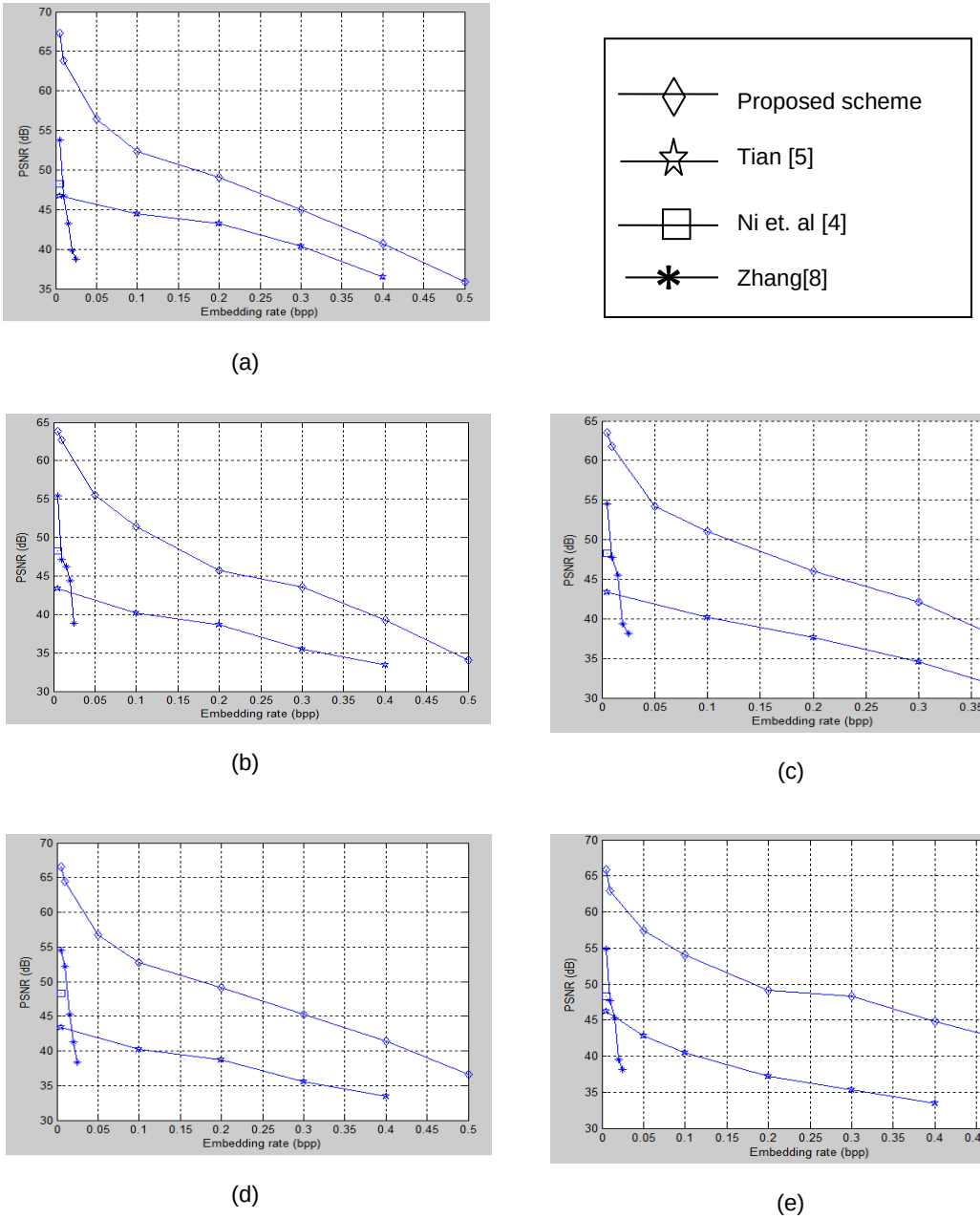


FIGURE 4: Performance of the proposed method against the three RDH schemes [4], [5], [8] for test images(a) Lena, (b) Barbara, (c) Peppers, (d) Boat, (e) Airplane.

embedding only in 1-LSB plane the proposed scheme with data embedding only in 1-LSB plane is taken for performance comparison. All the test images under proposed scheme can have a maximum embedding rate of 0.5 except peppers image. In 1-LSB plane peppers supports maximum embedding rate of 0.4 only.

From figure 4 we can also find that the proposed method shows larger PSNR values at different embedding rates. It indicates that the proposed method outperforms the other RDH schemes [4], [5] and [6] both in visual quality and payload

4. CONCLUSIONS

Reversible data hiding (RDH) is drawing attention now a days because of its ability to recover the cover without any distortion. Encryption is also used along with RDH for privacy protection. In the currently available methods RDH is implemented in encrypted images by vacating space after encryption. Moreover data hiding is done by simple LSB method in which the security of data is very less. In the proposed method space to embed data is allocated before encryption. A hyper chaotic function is also used which ensures the security of the hidden data. The hyper chaotic function shuffles the secret data based on a set of keys and these keys are difficult to track by a third person or an attacker. Hence it is prevented from brute force attack and the security of the system is increased. So all traditional RDH algorithms could achieve better performance by using the proposed system without losing secrecy. More specifically, the system is well secured. A digital water mark is also embedded into the cover image so that transmission errors and explicit modifications done by any third party can be easily identified. The proposed method has been validated by comparing the performance against three other RDH schemes [4], [5] and [8] for five test images and it is observed that the proposed method outperforms these RDH schemes both in visual quality and payload. The proposed method can achieve real reversibility, separate data extraction and great improvement on the quality of decrypted images.

In the present systems data is embedded as plain text. To increase the security some symmetric key algorithms can be used for encrypting the data to be embedded so that the encrypted data can be embedded in the image. During the first extraction at the receiver side encrypted data is retrieved as output. When the symmetric key algorithm is used the original data can be retrieved.

5. REFERENCES

[1] Johnson, Z. Duric, S. Jajodia Information Hiding. Steganography and Watermarking - Attacks and Countermeasures: Kluwer Academic Press. Norwall, 2000.

[2] J. Hwang , Kim and J. U Choi , “A reversible watermarking based on histogram shifting,” Int. Workshop on Digital Watermarking, Lecture Notes in Computer Science, Jeju Island, Korea, 2006, vol. 4283,pp. 348–361, Springer-Verlag.

[3] A. M. Alattar, “Reversible watermark using the difference expansion of a generalized integer transform,” IEEE Transactions on Image Processing, Vol. 13, 2004, pp. 1147-1156.

[4] Z. Ni, Y. Shi, N. Ansari, and S. Wei, “Reversible data hiding,” IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar.2006.

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

[6] X. Zhang, “Reversible data hiding in encrypted images,” IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255–258, Apr. 2011.

[7] W. Zhang, K. Ma, X. Zhao, N. Yu, F. Li “Reversible data hiding in encrypted images by reserving room before encryption, ”IEEE trans. Info. forensics and security ,vol. 8, no. 3, pp. 553–558, March 2013.

[8] X. Zhang, “Separable reversible data hiding in encrypted image,” IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 826–832, Apr. 2012.

[9] D. M Thodi and J. Rodriguez, “Expansion embedding techniques for reversible watermarking,” IEEE Trans. Image Process., vol. 16, no. 3,pp. 721–730, Mar. 2007.

[10] Y. Hu, H.K Lee, Chen and J. Li, “Difference Expansion based reversible data hiding using two embedding direction,” IEEE trans. multimedia, vol. 10, no. 8, pp. 1500–1512, 2008

[11] K. H Jung and K. Y Yoo, “Data hiding method using image interpolation,” Journ. of Compute standard and interfaces, vol. 31, pp. 465–470, 1996

[12] W. Hong, T. Chen, and H.Wu, “An improved reversible data hiding in encrypted images using side match,” IEEE Signal Process. Lett., vol.19, no. 4, pp. 199–202, Apr. 2012.

[13] L. Luo et al., “Reversible image watermarking using interpolation technique, ”IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187–193,Mar. 2010.

[14] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.-Q. Shi, “Reversible watermarking algorithm using sorting and prediction,” IEEE Trans.Circuits Syst. Video Technol., vol. 19, no. 7, pp. 989–999, Jul.2009.

[15] Zhu, C. Zhao, X. Zhang “A novel image encryption-compression scheme using hyper-chaos and Chinese remainder theorem”, Elsevier,Image communication 670-680, 2013.

[16] Y. Wang, K. W Wong, X. F Liao, T. Xiang, G. R Chen , “A chaos based image encryption algorithm with variable control parameters “, Chaos ,solutions and Fractals 41(4) 1773-1783, 2009.

[17] I. Cox,M. Miller M, J. Bloom, J. Fridrich, T. Kalker. Digital Watermarking and Steganography Second Edition. Elsevier, 2008