ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Novel Reserving Room Approach for Reversible Data Hiding Algorithm before Encryption on Encrypted Digital Images

Krishnaveni Chatlawar
Associate Professor,
Konkan Gyanpeeth College of Engineering.

M.J.Lengare
Principal,
Konkan Gyanpeeth College of Engineering.

Abstract:

In this paper a novel framework for data hiding based on reversible data hiding is presented for lossless data recovery approach. Reversible data hiding on encrypted images is most successful approach for its excellent property of lossless data recovery. Data hiding in digital images is a challenging task from last few decades since maintaining the image contents confidentiality and security to hidden data is an area of concern. The conventional algorithms in the literature which are proposed are subjected to errors at data extraction or data restoration process because these algorithms embed data by reversibly vacating room from the encrypted images.

The proposed algorithm is more efficient than the conventional algorithm propose a novel method 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. The proposed method can achieve real reversibility, that is, data extraction and image recovery are free of any error. Finally the Experiments show that this novel method can embed more than 10 times as large payloads for the same image quality as the previous methods.

INTRODUCTION:

Reversible data hiding (RDH) approach in digital images is an innovative technique, where the information related to original cover recovered lossless approach, this lossless data extraction is done once the extraction of embedded message is successfully completed. The applications related to reversible data hiding are medical imagery, military imagery and law forensics, where no distortion of the original cover is allowed. In literature, Reversible data hiding is always a interesting area of research for many researchers because of its excellent of recovering the information without any loss. A researcher named Kalker et al [1] firstly introduced rate distortion model approach for reversible data hiding algorithm which attracts many researchers, since this rate distortion approach proposes a recursive code construction model which yields the data in lossless manner. After some years a researcher named Zhanget at el [2], [3] proposes a novel enhanced rate distortion framework seeing the tremendous importance of reversible data hiding.

This algorithm improves the performance of rate distortion model for binary covers and proves that this algorithm reconstruction approach achieves the rate distortion bound and this algorithm achieves the rate distortion as soon as the compression algorithm reaches the entropy and further equivalence approach is successfully established between the reversible data hiding for binary covers and the data compression respectively. The reversible data approach proves its mettle in security related applications and many research works has been reported for better application approach. In paper [16] Zhang reported a novel approach based on reversible data hiding. In this approach the respective encrypted image is divided into three blocks. The half portion of each block in flipped by three LSB's, room can be vacated for the embedded bit. The data extraction and image recovery proceed by finding which part has been flipped in one block.

This process can be realized with the help of spatial correlation in decrypted image. Hong et al.[17] ameliorated Zhang's method at the decoder side by further exploiting the spatial correlation using a different estimation equation and side match technique to achieve much lower error rate. These two methods mentioned above rely on spatial correlation of original image to extract data. That is, the encrypted image should be decrypted first before data extraction. To separate the data extraction from image decryption, Zhang [18] emptied out space for data embedding following the idea of compressing encrypted images [14], [15]. Compression of encrypted data can be formulated as source coding with side information at the decoder [14], in which the typical method is to generate the compressed data in lossless manner by exploiting the syndromes of parity-check matrix of channel codes.

The method in [18] compressed the encrypted LSBs to vacate room for additional data by finding syndromes of a parity-check matrix, and the side information used at the receiver side is also the spatial correlation of decrypted images. All the three methods try to vacate room from the encrypted images directly. However, since the entropy of encrypted images has been maximized, these techniques can only achieve small payloads [16], [17] or generate marked image with poor quality for large payload [18] and all of them are subject to some error rates on data extraction and/or image restoration. In the present paper, we propose a novel method for RDH in encrypted images, for which we do not "vacate room after encryption" as done in [16]–[18], but "reserve room before encryption". In the proposed method, we first empty out room by embedding LSBs of some pixels into other pixels with a traditional RDH method and then encrypt the image, so the positions of these LSBs in the encrypted image can be used to embed data.

Not only does the proposed method separate data extraction from image decryption but also achieves excellent performance in two different prospects:

- Real reversibility is realized, that is, data extraction and image recovery are free of any error.

- For given embedding rates, the PSNRs of decrypted image containing the embedded data are significantly improved; and for the acceptable PSNR, the range of embedding rates is greatly enlarged.

CONVENTIONAL WORK:

In the proposed algorithm, a content owner encrypts the original image using a standard cipher with an encryption key. After producing the encrypted image, the content owner hands over it to a data hider (e.g., a database manager) and the data hider can embed some auxiliary data into the encrypted image by lossless approach vacating some room according to a data hiding key. Then a receiver, maybe the content owner himself or an authorized third party can extract the embedded data with the data hiding key and further recover the original image from the encrypted version according to the encryption key.

In all methods of [16]–[18], the encrypted 8-bit grayscale images are generated by encrypting every bit-planes with a stream cipher. The method in [16] segments the encrypted image into a number of non overlapping blocks sized by a a; each block is used to carry one additional bit. To do this, pixels in each block are pseudo-randomly divided into two sets S1 and S2 according to a data hiding key. If the additional bit to be embedded is 0, flip the 3 LSBs of each encrypted pixel in S1, otherwise flip the 3 encrypted LSBs of pixels in S2. For data extraction and image recovery, the receiver flips all the three LSBs of pixels in S1 to form a new decrypted block, and flips all the three LSBs of pixels in S2 to form another new block; one of them will be decrypted to the original block. Due to spatial correlation in natural images, original block is presumed to be much smoother than interfered block and embedded bit can be extracted correspondingly. However, there is a risk of defeat of bit extraction and image recovery when divided block is relatively small (e.g., a=8) or has much fine-detailed textures.

PROPOSED WORK:

Since losslessly vacating room from the encrypted images is relatively difficult and sometimes inefficient, why are we still so obsessed to find novel RDH techniques working directly for encrypted images? If we reverse the order of encryption and vacating room, i.e., reserving room prior to image encryption at content owner side, the RDH tasks in encrypted images would be more natural and much easier which leads us to the novel framework, "reserving room before encryption (RRBE)".

As shown in Fig.1 (b), the content owner first reserves enough space on original image and then convert the image into its encrypted version with the encryption key.

Now, the data embedding process in encrypted images is inherently reversible for the data hider only needs to accommodate data into the spare space previous emptied out

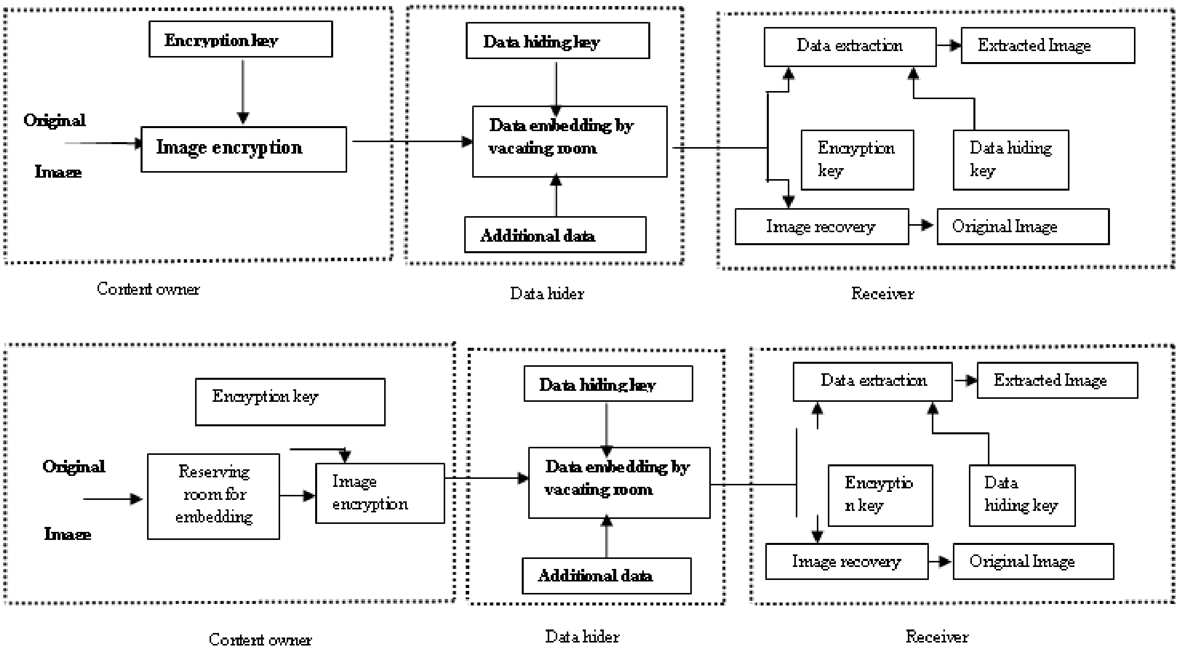

Figure 1
(a) Framework VRAE. (b) Framework RRBE.

The data extraction and image recovery are identical to that of Framework VRAE. Obviously, standard RDH algorithms are the ideal operator for reserving room before encryption and can be easily applied to Framework RRBE to achieve better performance compared with techniques from Framework VRAE. This is because in this new framework, we follow the customary idea that first losslessly compresses the redundant image content (e.g., using excellent RDH techniques) and then encrypts it with respect to protecting privacy. Next, we elaborate a practical method based on the Framework "RRBE", which primarily consists of four stages: generation of encrypted image, data hiding in encrypted image, data extraction and image recovery. Note that the reserving operation we adopt in the proposed method is a traditional RDH approach.

(1) Generation of Encrypted Digital Images:

Actually, to construct the encrypted image, the first stage can be divided into three steps: image partition, self reversible embedding followed by image encryption. At the beginning, image partition step divides original image into two parts A and B; then, the LSBs of A are reversibly embedded into B with a standard RDH algorithm so that LSBs of A can be used for accommodating messages; at last, encrypt the rearranged image to generate its final version.

(a) Image Partition:

The operator here for reserving room before encryption is a standard RDH technique, so the goal of image partition is to construct a smoother area B, on which standard RDH algorithms such as [10], [11] can achieve better performance. To do that, without loss of generality, assume the original image C is an 8 bits gray-scale image with its size M N and pixels Ci, j [0,255], 1≤i≤M, 1≤j≤N. First, the content owner extracts from the original image C, along the rows, several overlapping blocks whose number is determined by the size of to-be-embedded messages, denoted by. In detail, every block consists of m rows, where m= [1/n], and the number of blocks can be computed through n=M-m+1. An important point here is that each block is overlapped by pervious and/or sub sequential blocks along the rows. For each block, define a function to measure its first-order smoothness.

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

Higher f relates to blocks which contain relatively more complex textures. The content owner, therefore, selects the particular block with the highest f to be A, and puts it to the front of the image concatenated by the rest part with fewer textured areas, as shown in Fig. 2.

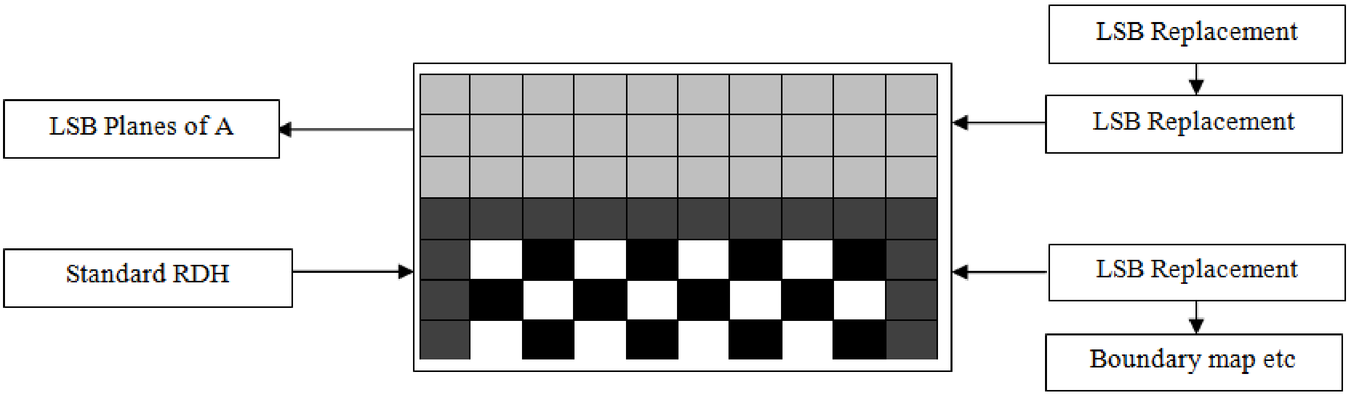

Figure 2 Image partition and embedding process.

(b) Self Reversible Embedding:

The goal of self-reversible embedding is to embed the LSB-planes of A into B by employing traditional RDH algorithms. For illustration, we simplify the method in [10] to demonstrate the process of self-embedding. Note that this step does not rely on any specific RDH algorithm. Pixels in the rest of image B are first categorized into two sets: white pixels with its indices i and j satisfying (i+j) mod2=0 and black pixels whose indices meet (i+j) mod2=1, as shown in Fig. 2. Then, each white pixel Bij, is estimated by the interpolation value obtained with the four black pixels surrounding it as follows:

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

Where the weight w= 1≤i≤4, is determined by the same method as proposed in [10]. The estimating error is calculated via eij=Bij-B'ij and then some data can be embedded into the estimating error sequence with histogram shift, which will be described later. After that, we further calculate the estimating errors of black pixels with the help of surrounding white pixels that may have been modified. Then another estimating error sequence is generated which can accommodate messages as well.

Furthermore, we can also implement multilayer embedding scheme by considering the modified B as "original" one when needed. In summary, to exploit all pixels of B, two estimating error sequences are constructed for embedding messages in every single-layer embedding process. By bidirectional histogram shift, some messages can be embedded on each error sequence. That is, first divide the histogram of estimating errors into two parts, i.e., the left part and the right part, and search for the highest point in each part, denoted by LM and RM, respectively. For typical images, LM=-1 and RM=0. Furthermore, search for the zero point in each part, denoted by LN and RN. To embed messages into positions with an estimating error that is equal to RM, shift all error values between RM+1 and RM-1 with one step toward right, and then, we can represent the bit 0 with RM and the bit 1 with RM+1. The embedding process in the left part is similar except that the shifting direction is left, and the shift is realized by subtracting 1 from the corresponding pixel values.

(c) Image Encryption:

After rearranged self-embedded image, denoted by X, is generated, we can encrypt X to construct the encrypted image, denoted by E. With a stream cipher, the encryption version of X is easily obtained. For example, a gray value Xi,j ranging from 0 to 255 can be represented by 8bits, , such that Xi,j(0), Xi,j(1), Xi,j(2)……..Xi,j(7)

$$X_{i,j}(k) = \left\lfloor \frac{X_{i,j}}{2^k} \right\rfloor \bmod 2, \qquad k = 0, 1, \ldots, 7 \qquad (3)$$

The encrypted bits Ei,j(k) can be calculated through exclusive-or operation.

$$E_{i,j}(k) = X_{i,j}(k) \oplus r_{i,j}(k) \qquad (4)$$

Where rij(k) is generated via a standard stream cipher determined by the encryption key. Finally, we embed 10 bits information into LSBs of first 10 pixels in encrypted version of A to tell data hider the number of rows and the number of bit-planes he can embed information into. Note that after image encryption, the data hider or a third party can not access the content of original image without the encryption key, thus privacy of the content owner being protected.

(2) Data Hiding in Encrypted Image:

Once the data hider acquires the encrypted image E, he can embed some data into it, although he does not get access to the original image. The embedding process starts with locating the encrypted version of A, denoted by AE. Since AE has been rearranged to the top of E, it is effortless for the data hider to read 10 bits information in LSBs of first 10 encrypted pixels. After knowing how many bit-planes and rows of pixels he can modify, the data hider simply adopts LSB replacement to substitute the available bit-planes with additional data m. Finally, the data hider sets a label following m to point out the end position of embedding process and further encrypts m according to the data hiding key to formulate marked encrypted image denoted by E’. Anyone who does not possess the data hiding key could not extract the additional data.

(3) Data encryption and image recovery:

Since data extraction is completely independent from image decryption, the order of them implies two different practical applications. 1) Case 1: Extracting Data from Encrypted Images: To manage and update personal information of images which are encrypted for protecting clients’ privacy, an inferior database manager may only get access to the data hiding key and have to manipulate data in encrypted domain. The order of data extraction before image decryption guarantees the feasibility of our work in this case. When the database manager gets the data hiding key, he can decrypt the LSB-planes of and extract the additional data by directly reading the decrypted version. When requesting for updating information of encrypted images, the database manager, then, updates information through LSB replacement and encrypts updated information according to the data hiding key all over again. As the whole process is entirely operated on encrypted domain, it avoids the leakage of original content. 2) Case 2: Extracting Data from Decrypted Images: In Case 1, both embedding and extraction of the data are manipulated in encrypted domain. On the other hand, there is a different situation that the user wants to decrypt the image first and extracts the data from the decrypted image when it is needed. The following example is an application for such scenario.

Assume Alice out sourced her images to a cloud server, and the images are encrypted to protect their contents. Into the encrypted images, the cloud server marks the images by embedding some notation, including the identity of the images’ owner, the identity of the cloud server and time stamps, to manage the encrypted images. Note that the cloud server has no right to do any permanent damage to the images. Now an authorized user, Bob who has been shared the encryption key and the data hiding key, downloaded and decrypted the images. Bob hoped to get marked decrypted images, i.e., decrypted images still including the notation, which can be used to trace the source and history of the data. The order of image decryption before/without data extraction is perfectly suitable for this case. Next, we describe how to generate a marked decrypted image.

(a) Generating the marked Decrypted image:

Step 1. With the encryption key, the content owner decrypts the image except the LSB-planes of AE. The decrypted version of E’ containing the embedded data can be calculated by

$$X''_{i,j}(k) = E'_{i,j}(k) \oplus r_{i,j}(k) \qquad (5)$$

$$X''_{i,j} = \sum_{k=0}^{7} X''_{i,j}(k) \times 2^k \qquad (6)$$

Step 2. Extract the SR and ER in marginal area of B’’. he plain image containing embedded data is obtained.

SIMULATION RESULTS:

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

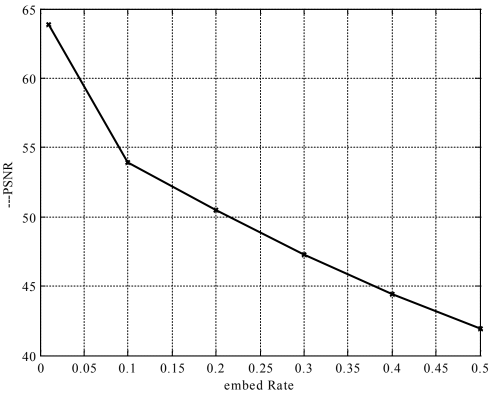

Figure 1: For Car image

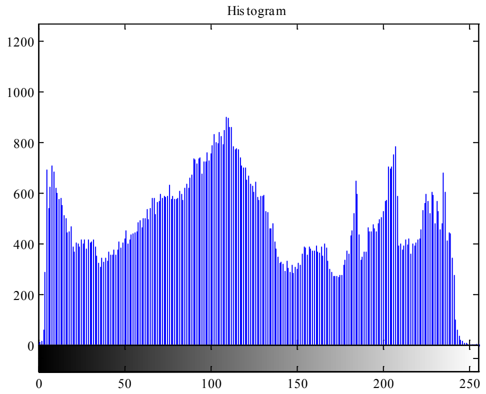

Figure 2: Histogram for Car Image

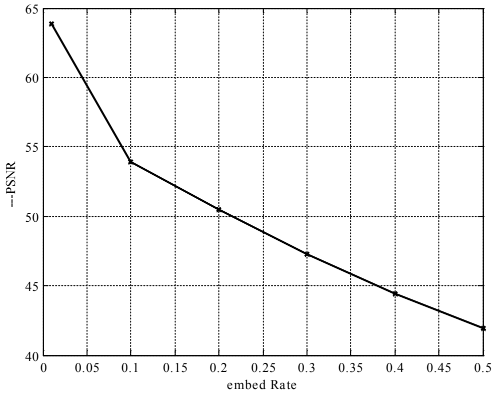

Figure 3: For Cameraman image

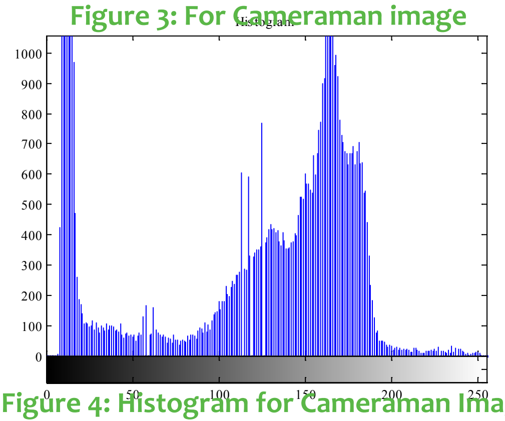

Figure 4: Histogram for Cameraman Image

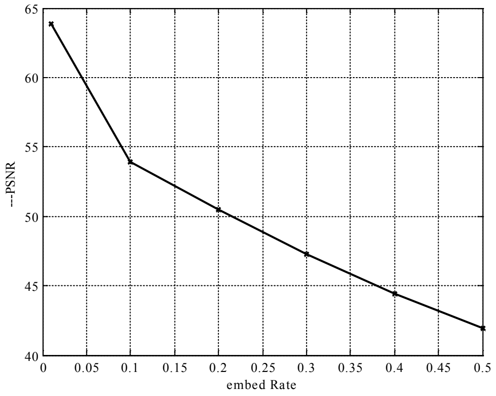

Figure 5: For Einstein image

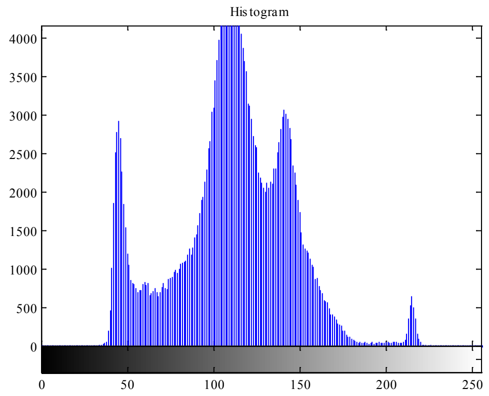

Figure 6: Histogram for Einstein Image

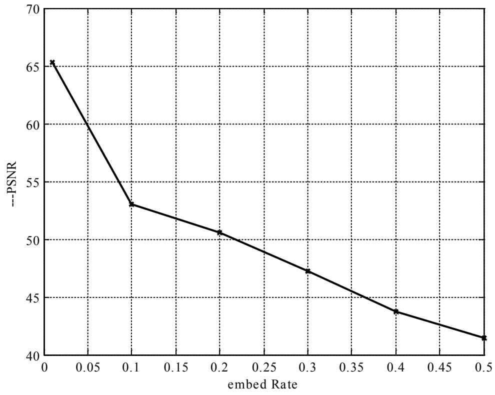

Figure 7: For Pig image

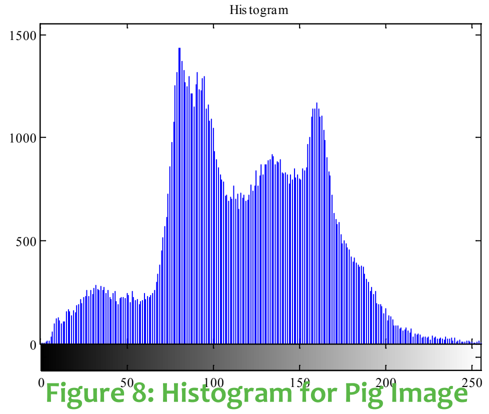

Figure 8: Histogram for Pig Image

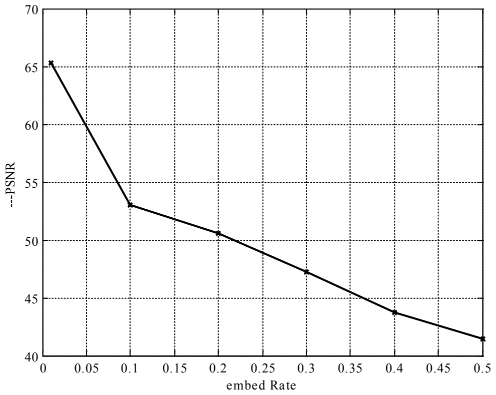

Figure 9: For Lena image

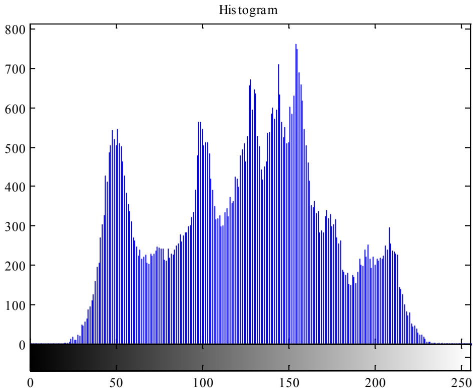

Figure 10: Histogram for Lena Image

IMAGE NAME	PSNR	STD	MEAN	VAR
CAMERAMAN	41.4569	62.3431	118.6952	554.44234
CAR	38.9082	67.5223	119.976	2124.3565
EINSTEIN	48.297	33.4767	110.6855	113.43944
PIG	39.7144	44.4065	115.3782	1584.4634
LENA	36.617	58.9147	128.4005	2348.2676

Tabular form with four different parameters values

CONCLUSION:

Reversible data hiding in encrypted images is a new topic drawing attention because of the privacy-preserving requirements from cloud data management. Previous methods implement RDH in encrypted images by vacating room after encryption, as opposed to which we proposed by reserving room before encryption. Thus the data hider can benefit from the extra space emptied out in previous stage to make data hiding process effortless. The proposed method can take advantage of all traditional RDH techniques for plain images and achieve excellent performance without loss of perfect secrecy. Furthermore, this novel method can achieve real reversibility, separate data extraction and greatly improvement on the quality of marked decrypted images.

REFERENCES:

[1] T. Kalker and F. M. Willems, “Capacity bounds and code constructions for reversible data-hiding,” inProc. 14th Int. Conf. Digital Signal Processing (DSP2002), 2002, pp. 71–76.

[2] W. Zhang, B. Chen, and N. Yu, “Capacity-approaching codes for reversible data hiding,” in Proc 13th Information Hiding (IH’2011), LNCS 6958, 2011, pp. 255–269, Springer-Verlag.

[3] W. Zhang, B. Chen, and N. Yu, “Improving various reversible data hiding schemes via optimal codes for binary covers,”IEEE Trans. Image Process., vol. 21, no. 6, pp. 2991–3003, Jun. 2012.

[4] J. Fridrich and M. Goljan, “Lossless data embedding for all image formats,” inProc. SPIE Proc. Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents, San Jose, CA, USA, Jan. 2002, vol. 4675, pp. 572–583.

[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] 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.

[7] D. M. Thodi and J. J. Rodriguez, “Expansion embedding techniques for reversible watermarking,”IEEE

Trans. Image Process.,vol.16,no.3, pp. 721–730, Mar. 2007.

[8] X. L. Li, B. Yang, and T. Y. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," IEEE Trans. Image Process., vol. 20, no. 12, pp. 3524–3533, Dec. 2011.

[9] P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting,"Signal Process., vol. 89, pp. 1129–1143, 2009.

[10] L. Luoet al., "Reversible image watermarking using interpolation technique,"IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187–193, Mar. 2010.

About Author's:

Krishnaveni Chatlawar

received the B.E degree in electronics and telecommunication from BabasahebAmbedkar University ,Maharashtra, India and M.E in electronics from Government college of engineering, BabasahebAmbedkar University Aurangabad, Maharashtra ,India in 1997 and 2008 respectively .since 2000 to 2007 worked as a lecturer in P.E.S.College of Engineering Aurangabad,India . She is currently working as a ''Associate Professor" in KonkanGyanpeeth College of Engineering, Karjat, Mumbai university, Maharashtra, India and developed laboratory of optical fiber communication. She published 2 papers at NCAC 2008, Nadiad,Gujrat. Her interest lies in digital signal processing and image processing and her post-graduation project was based on speech processing.

M.J.Lengare-

received B.E degree in instrumentation engg. from M arathwadauniversity,Maharashtra, India and received M.E and P.H.D degree from Swami RamanandTirth-Marathwada University, Maharashtra,India in 2000 and 2012 respectively. He is currently working as "Principal" at KonkanGyanpeeth College of Engineering, Mumbai University, Maharashtra, India and he is research guide at, Udaypur,Rajsthan. He is having 20 years of teaching experience and more than 10 years of experience at administrative level. His research interest include control system,geneticalalgorithm,system identification and signal processing .In those areas he has more than 15 Journal and conference publications and he is member of ISA and ISTE