www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 9 Sep 2015, Page No. 14319-14327

Optimized Reserving Room Approach For Reversible Data Hiding Algorithm Before Encryption On Encrypted Digital Images

C. Vara Lakshmi [1] B.Geetha Rani[2]

Department of ECE (DECS) [1] Assistant Professor M.TECH [2]

G. Pullaiah College Of Engineering And Technology, Kurnool, Andhra Pradesh, INDIA

Abstract

Reversible data hiding is the technique in which data in the cover image reversibly can retrieve after the extraction of hidden data in it. The technique provides the secrecy for a data, and also for its cover image. Ancestor methods of reversible data hiding were vacates room for data hiding after encryption, which leads to some errors at the time of data extraction and image recovery. Here describes a novel method of reversible data hiding in which, Reserving room before encryption in images, so that image extraction is subjected to free of errors. Here we are proposing an LSB plane method for the data hiding, which will result more space for embedded secret data. Moreover the usage of colour images as cover images will helps to store more data in different channels.

KEYWORDS: Data hiding, Reserving Room, RDH Algorithm, Payload

1. INTRODUCTION

In most cases of data hiding, the cover images will experience some distortion due to data hiding and cannot be inverted back to the original form. That is, some permanent distortion has occurred to the cover image even after the hidden data have been extracted out. In a wide range of applications like medical, military and law forensic fields, distortion of cover images does not allowed. So reversible data hiding is essential for these cases. In this technique the cover image can lossless recover after the extraction of hidden data. So many RDH techniques have introduced in recent years.

One of a general framework for RDH is first extracting compressible features of original cover and then compressing them lossless, more space can be saved for embedding auxiliary data. Another popular method is based on difference expansion (DE), in which the difference of each pixel group is expanded like multiplication by even numbers. Then the least significant bits (LSBs) of the difference are all-zero and can be used for embedding messages. Another considerable strategy for RDH is histogram shift (HS), in which space is surplus for data embedding by shifting the highest possible value of histogram of gray values.

Encryption is an effective and popular means of privacy protection. A content owner can encrypt his message before sending to another person as it converts the original and meaningful content to abstruse one. This process is performing with the help of spatial correlation in decrypted image. In another method, at the decoder side by further exploiting the spatial correlation using a different estimation equation and side match technique, which provides much lower error rate. These two methods mentioned above rely on spatial correlation of original

image to extract data. Implies that, decryption must be done in the encrypted before data extraction.

All the above methods try to vacate room from the encrypted images directly. Because of the entropy of encrypted images has been maximized, these techniques can only obtain small payloads or generate marked image with poor quality for large payload and all of them are subject to some error rates on data extraction and/or image restoration. Some methods can use error correcting codes, also pure payload can consume. In this paper, we proposes a novel method for RDH in encrypted color images, for which we do not ―vacate room after encryption‖, but ―reserve room before encryption‖ . We can first empty out spaces by embedding LSBs of some pixels into other pixels with a LSB plane method and then encrypt the image, so the place of these LSBs in the encrypted image can be used to hide data bits. This method separate data extraction from image decryption and data extraction and image recovery are free of any error.

2. RELATED WORKS

Reversible data hiding was first established as a technique of attaining the cover image after the extraction of hidden data. Here utilizes the zero or the minimum points of the histogram of an image and slightly modifies the pixel gray scale values to embed data into the image. The computational complexity for technique is low, but it is only applicable at gray scale images. Then the Reversible Data Hiding with Optimal Value Transfer emerges to find the optimal rule of value modification under a payload-distortion criterion.

An optimal value transfer matrix can be obtained by maximizing a target function using iterative algorithm, for a practical reversible data hiding scheme. The technique undergoes the prediction so Computation complexity will be higher. Encryption is an effective and popular means of privacy protection. Through the work Reversible Data Hiding in Encrypted Image proposes a reversible data hiding scheme for encrypted image. Then the additional data can be embedded into the image by modifying a small portion of encrypted data, after encrypting the entire data of a gray image. The received image provides original image through decryption using encryption key and data using data hiding key. For encrypted images, the compression efficiency can be improved as how the source dependency is exploited. So a work developed as Efficient Compression of Encrypted Gray scale Images.

The paper proposes a resolution progressive compression scheme which compresses the encrypted image progressively in resolution, such that the decoder can attain low resolution version of the image. The statistics can be used to analyze next resolution level. An Improved method of Reversible Data Hiding in Encrypted Images is Using Side Match. Here proposes an improved data extraction and image recovery method over Zhang's work. The Zhang's work partitions an encrypted image into blocks, and each block carries one bit by flipping three LSBs of a set of pre-defined pixels. Thus leads to estimation of new algorithm for better calculation of smoothness of image blocks.

According to the descending order of the absolute smoothness difference between two candidate blocks, the extraction and recovery of blocks are performed. Also side match technique reduces the error rate.

In the first phase, a content owner encrypts the original image using a key. Then, a data-hider may vacates some spare space by performing compression on least significant bits using a data-hiding key to accommodate some additional data. Using Separable Reversible Data Hiding in Encrypted Image, with an encrypted image containing additional data, even receiver does not know the image content he can extract the additional data using data hiding key. Also receiver can decrypt the received data to obtain an image similar to the original one using encryption key, but cannot extract the additional data. Receiver can recover the original content and extract additional data without any error using the encryption key and data-hiding key. Watermarking embeds information into a digital signal. After the hidden data is extracted, receiver can restore the original image without any distortion. Reversible Image Watermarking is a scheme using an interpolation technique,

which can embed a large amount of data into images with unknowable modification.

Here assigns the interpolation-error. Due to lesser modification of pixels, quality of image will be higher. Reversible Watermarking Algorithm Using Sorting and Prediction is another algorithm without using a location map is used for reversible watermarking. This algorithm employs prediction errors to hide data into an image. To record the prophesy errors based on magnitude of its local variances a sorting technique is used. Using sorted prophesies errors and a reduced size location map allows us to hide more data in the image with less distortion.

3. FOR HIDING DATA IN AN IMAGE

The method in segments the encrypted image into a number of non-overlapping blocks; each block is used to carry one additional bit. The method reduced the error rate of the method by fully exploiting the pixels in calculating the smoothness of each block and using side match. The method in compressed the encrypted LSBs to vacate room for additional data by finding syndromes of a parity-check matrix and to separate the data extraction from image decryption, emptied out space for data embedding following the idea of compressing encrypted images. Here, a novel method is proposed so as to encrypt images using RDH, for which "vacate room after encryption" is not done, but "reserve room before encryption"where, 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 which achieves excellent performance in two different prospects

4. WHAT IS RDH (REVERSIBLE DATA HIDING?)

Reversible data hiding is a procedure to embed extra message into some distortion-unacceptable cover media, such as military or medical images, with a reversible behavior so that the original cover content can be flawlessly restored. In Zhang divided the encrypted image into several blocks. By flipping 3 LSBs of the half of pixels in each block, 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. 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 emptied out space for data embedding following the idea of compressing encrypted images. Compression of encrypted data can be formulated as source coding with side information at the decoder, 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 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 or generate marked image with poor quality for large payload and all of them are subject to some error rates on data extraction and/or image restoration.

Although the methods can eliminate errors by error correcting codes, the pure payloads will be further consumed. 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, but "reserve room before encryption".

5. CONVENTIONAL WORK

In the proposed algorithm, a content owner encrypts the original image using a standard cipher with an encryption

DOI: 10.18535/ijecs/v4i9.44

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 gray-scale 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 S_1 and S_2 according to a data hiding key. If the additional bit to be embedded is 0, flip the 3 LSBs of each encrypted pixel in S_1, otherwise flip the 3 encrypted LSBs of pixels in S_2. For data extraction and image recovery, the receiver flips all the three LSBs of pixels in S_1 to form a new decrypted block, and flips all the three LSBs of pixels in S_2 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.

6. 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

.

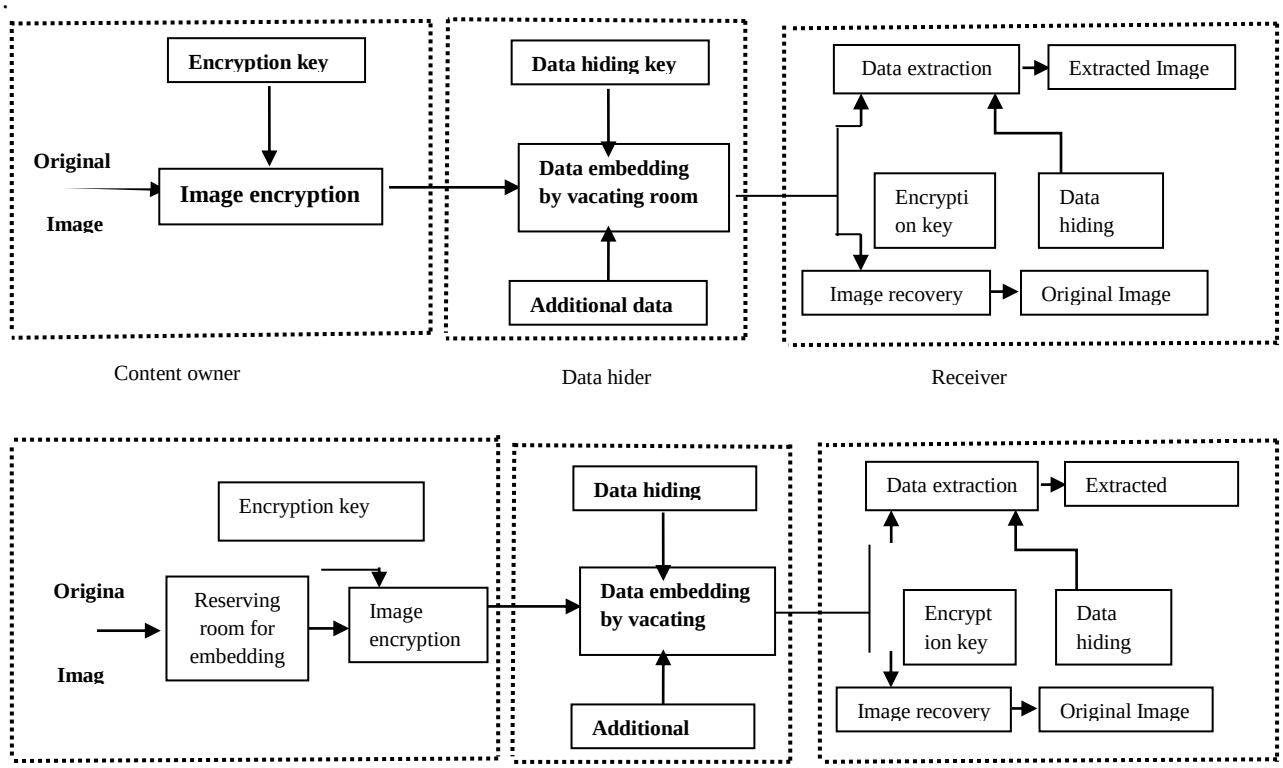

Figure 1: (a) Framework VRAE. (b) Framework RRBE

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.

6.1 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 $C_{i,j} \in [0,255]$, $1 \le i \le M$, $1 \le j \le N$. 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

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

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.

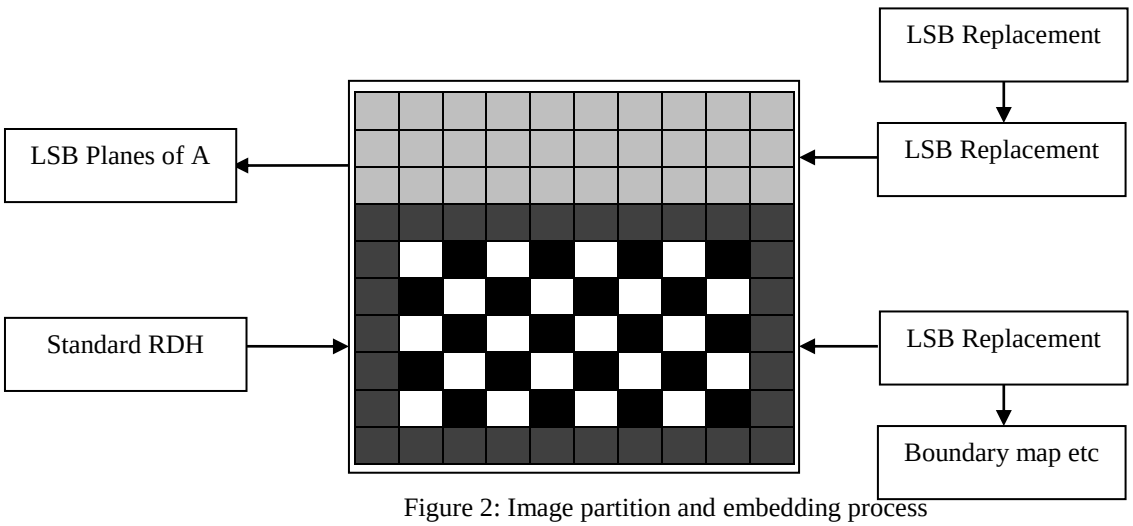

Figure 2: Image partition and embedding process

(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) mod*2=0 and black pixels whose indices meet *(i+j) mod*2=1, as shown in Fig. 2. Then, each white pixel B_{ij}, 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= $1 \le i \le 4$, is determined by the same method as proposed in [10]. The estimating error is calculated via $e_{ij}=B_{ij}-B'_{ij}$ 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 $X_{i,j}$ ranging from 0 to 255 can be represented by 8bits, , such that $X_{i,j}(0)$, $X_{i,j}(1)$, $X_{i,j}(2)$…….. $X_{i,j}(7)$

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

The encrypted bits $E_{i,j}(k)$ can be calculated through exclusive-or operation

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

Where $r_{ij}(k)$ 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.

6.2 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 A_E. Since A_E 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.

DOI: 10.18535/ijecs/v4i9.44

6.3 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.

6.4 Generating the marked Decrypted image

Step 1: With the encryption key, the content owner decrypts the image except the LSB-planes of A_E. 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.

7. SIMULATION RESULTS

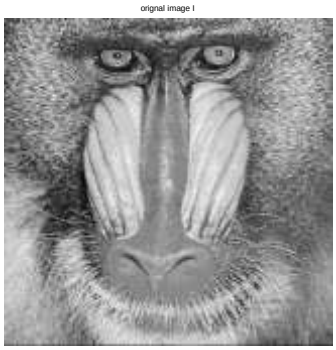

Figure 3: Original Image

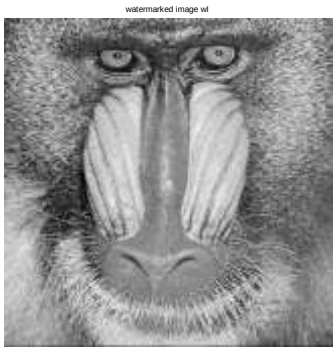

Figure 4 Watermarked image

The PSNR of WI is: 29.89 dB

The actual embedding rate is 0.50 bpp

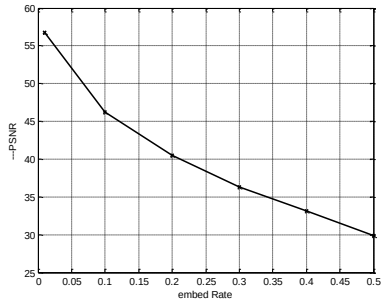

Figure 5: Embedding rate graph

8. 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.

9. EXTENSION

The proposed method has been written on the digital images, in this work images are used as media to hide the secret image by using the an approach where mosaic image generation has done by dividing the secret image into fragments and transforming their respective color characteristics into corresponding blocks of the target image. Usage of the LSB transformations helps to yield the lossless recovered image based on the untransformed color space values. So in extension work we did the same algorithm on the digital videos. The approaches towards videos are totally different from the images, so algorithm on videos is the contribution to the proposed work.

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.

[11] 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.

[12] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. Boca Raton, FL, USA: CRC, 1996.

[13] K. Hwang and D. Li, "Trusted cloud computing with secure resources and data coloring," IEEE Internet Comput., vol. 14, no. 5, pp. 14–22, Sep./Oct. 2010.

[14] 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.

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

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

[17] 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.

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

[19] Miscelaneous Gray Level Images [Online]. Available: http://decsai. ugr.es/cvg/dbimagenes/g512.php