International Journal of Research in Advent Technology, Vol.6, No.11, November 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

Improved Reversible Data Hiding Using Sparse Representation Technique

Sonali Pawar[1], Vikas Marathe[2]
[1] *N.B. Navale Sinhgad College of Engineering; India; pawarsonali9292@gmal.com*
[2] *N.B. Navale Sinhgad College of Engineering; India; vksmarathe@gmail.com*

Abstract- Reversible data hiding is a type of data hiding techniques in which the cover image is first compressed and then processed for hiding data at transmitter and the hidden data is recovered exactly without losing its visuality and quality at the receiver side. As it is a lossless data hiding technique it is most suitable for medical and military applications. Another technique used in this paper is sparse representation technique. It is one of the most important techniques which are widely used for signal compression and representation in image processing. It has received more attention over the years because of its high efficiency and privacy. The aim of this paper is to present a technique of sparse representation applied to reversible data hiding. Here when applied to cover image, a sparse representation technique works at patch level rather than pixel level to increase the efficiency and privacy of the algorithm.

Index Terms-RDH; Image encryption; Image decryption; K-SVD;

1. INTRODUCTION

Data hiding is a process where amount of data is hidden (embedded) in a cover object like image audio or video. The type of data hiding which exactly recover both the cover image and the embedded secret information is known as reversible data hiding. Many fields like medical and military has been attracted to this method because of its high privacy and accuracy since in such fields a slight distortion in signal is also not tolerable. In early decades Many RDH algorithms have already been developed, such as image compression-based RDH, difference expansion based, histogram shift (HS)-based, image pixel pair based, and dual/multi-image hiding methods. Recently, due to the requirement of privacy protection, the cover owner usually encrypts the original content before transferring it to the data manager. Further for authentication or steganography purpose, the data manager may want to embed additional messages into the encrypted image even though the content of the original image is unknown to him. To meet such requirement hiding the data in an encrypted domain of image is required. For this purpose, some digital watermarking based schemes are proposed. The methods mentioned above provides promising performance in encrypted domains but they are not well suitable for more sensitive scenarios like military and medical fields. In such applications its necessary to losslessly recover the image contents after data extraction along with keeping it secret strictly. So its required to develop a technique embedding data in encrypted images ie. reversible data hiding in encrypted images(RDHEI) is desirable.

In the decades flow, many RDHEI schemes have been proposed in progress in which One common techniques is based on manipulating the least-significant-bit (LSB) planes. In LSB manipulating methods, they directly replace the three LSBs of the cover-image with the secret message bits by compressing the LSBs of the each pixel. So it's a kind of the pixel-level compressive method because in such methods, the encrypted image is devided into a number of non-overlapped blocks and each block is divided into two sets. By flipping three LSBs of a set for predefined pixels, one bit of secret message is embedded in each block. While doing so, they consider the pixel correlations in the border of neighboring blocks which results in decrease of the error rate of extracted-bits. This is very simple and easy to manipulate method but its not easy to vacate room by only choosing three LSBs of the encrypted images. To overcome this problem, researchers preferred to select a half of fourth LSB also as the space to carry the secret data. The smooth blocks in the encrypted image are selected to further improve the compression ratio and the additional data is embedded into the blocks. The blocks are sorted for embedding with respect to block smoothness by considering local HS. The image is divided into patches or groups to increase the hiding capacity but the LSB modification or compression is required to get the preserved spaces. Also the entropy of encrypted images is increased and its at maximum level. Hence using the above methods its really hard task to losslessly create the vacated room available for data hiding. To overcome this drawback, the methods of reserving room before encryption are put forward. Here for better results, only three LSBs are used for data hiding purpose.

Actually, for various computer applications, the image can be analyzed at the patch level rather than at the individual pixel level. Patches contain relevant information and have advantages in terms of computation and generalization. Specifically, because the pixels in certain ranges (like patches or regions) are of strong similarity, the information in any image is correlated in a way such that they can be compressed at greater compression rate which finally results in a large hiding room. Considering the two aspects, to better explore the correlations of neighbor pixels, here a method for high capacity separable reversible data hiding in encrypted image is proposed.

In this paper we follow the framework of RRBE ie. Reserving room before encryption and the image patch is described by sparse linear combinations of overcomplete dictionary atoms. Due to this, the number of signals which require space to record are decreased since only a small number of coefficients and the corresponding residual error $\tilde{e}$ caused by sparse representation require space to record. Thus, a higher capacity room is made available.

2. RELATED WORK

In RDH, user hides data in a cover object by using different types of protocols and techniques. This work is carried out since many years. In recent decades, a series of Reversible Data Hiding in Encrypted Images (RDHEI) schemes have been developed in progress. There are two types of RDHEI scheme. First is Vacating Room After Encryption (VRAE) which first encrypts the image and then make space for additional data embedding. Second is Reserving Room Before Encryption(RRBE) which first makes space available for additional data embedding and then encrypt that image. [4]–[8] are impersonating different approaches for VRAE technique. For RRBE, there are two mechanisms as [2] and [3]. In [4], the sender first encrypts the original cover image and then embeds secret data by modifying a small portion of the encrypted image. The receiver first performs decryption of the encrypted image and then extraction of the embedded data and recovers the original cover image. This method is efficient but the disadvantage is the increased error rate in extracted data. In [5] W. Hong proposed a scheme for reversible data hiding using side match. This method uses the side-match mechanism to decrease the error rate of extracted bits. The error rate obtained by [5] is much lower than [4]. However, separability property is not taken into account in [4], [5]. That is, the data extraction and content decryption operations are not separately done. By taking this point into account, Zhang [6] come up with a novel scheme for separable RDHEIs which include advancement of reversible data hiding with cryptography. In this, first the user performs encryption of original uncompressed image by using encryption key to vacate the room and then the data hider compresses the LSBs of the encrypted image using a data hiding key. Depending on the keys provided by use, there are 3 cases at the receiver side. Further Zhang et al. put forward an advanced scheme of RDHEIs in [7]. He losslessly compressed the encrypted data using LDPC code. Here half of the fourth LSB in the cipher-text image is compressed by the data hider and then inserted the compressed data and the additional data into the half of the fourth LSB using efficient reversible data hiding method. Moreover, Yin et al. came forward with a RDHEI scheme in [8] which inheriting the error-free data extraction in addition of previous method. In this proposed scheme the cover image is partitioned into number of overlapping blocks and then encryption mechanism is applied to those blocks. Depending on smoothness of blocks the data hider selects the several smoother blocks for data embedding process.

It is well known that encryption process increases the entropy of the image. So the room vacating is more difficult work in the entire process of RDH. The phenomenons in [4]–[8] can achieve a small amount of payloads embedding capacity even they have advancement of compressing encrypted images. The maximum embedding rate (MER) in [4]–[8] are all less than 0.2 bits per pixel. This implies that losslessly vacating room from encrypted images is somewhat difficult and problematic task and sometimes it is totally inefficient. To overcome this disadvantage, the RRBE methods [2], [3] are kept forward. Zhang et al. in [2] approximated a few pixels before encrypting the image and then the additional data are embedded in the estimating errors, instead of embedding data in encrypted images directly. Ma et al. [3] designed an effective scheme in improvement of this phenomenon by RRBE. In his designed framework, instead of encrypting the image directly, he first empty out room by embedding LSBs of some pixels in one region into another region via a traditional RDH method. Since the LSBs are used to hide the data, the embedding rate is increased in comparison of previous methods. Also this method can separately extract hidden data and decrypt the original cover image. But since the spare space emptied out is limited to at most three LSB-planes per pixel, the MER is only about 0.5 bits per pixel of method proposed in [2]. To overcome this drawback, the advance method is put forward by Xiaochun Cao in [1]. His proposed method take over the merits of RRBE technique based sparse representation at patch level. The proposed method separates the data extraction from image decryption along with the excellent achievement in terms of embedding capacity and visual quality of image. Moreover, different from the above methods mainly considering pixel-level compressive property, our scheme takes the patch as a

whole, and represents them using sparse signal representation coding. As a result, a high capacity is achieved.

3. SYSTEM MODEL

In this section, we give a detailed introduction about our scheme in the following three aspects:
1)encrypted image generation; 2) data hiding in the encrypted image; and 3) data extraction and image recovery. Figure.1 shows the block diagram of proposed work- Analysis and implementation of high efficient steganography using patch level sparse representation.

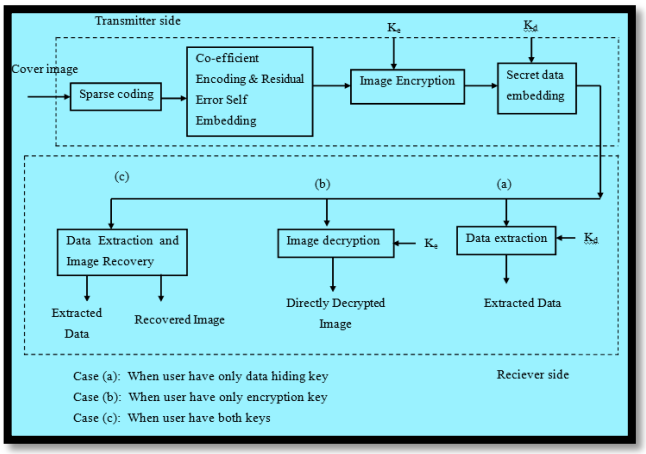

Fig.1 Overview of proposed system.

The proposed framework aims to perform 3 main operations.
1. Encrypted image generation
2. Data embedding in encrypted image
3. Data extraction and image recovery
First two operations will be performed at transmitter side and third will be at receiver side.

Transmitter: Transmitter consist of 4 blocks as-

1. Sparse coding

2. Coefficient encoding and residual error self embedding

3. Image encryption

4. Secret data embedding

1. Sparse coding: Designing the dictionary for signal representation using sparse coding is the first part of this proposed algorithm. Given a cover image of size 512*512, we first divide this cover image into patches of size 8*8. These divided patches are then represented according to an over complete dictionary D via sparse coding technique. After the sparse representation of signal, the smoother patches with lower residual errors are selected from those patches and they are subjected for room reserving process. These selected patches are represented by the sparse coefficients.

Here we train the dictionary based on K-means singular value decomposition (K-SVD) algorithm, which is widely used for designing over-complete dictionaries that lead to sparse signal representation. The K-SVD training is an offline procedure and hence the corresponding dictionary produced by K-SVD training is then considered fixed throughout the whole procedure.

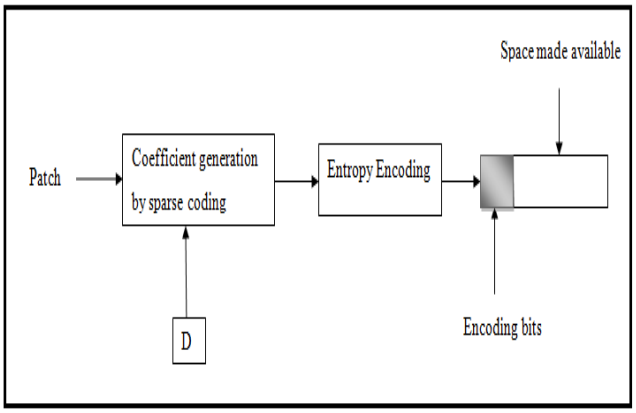

Fig.2. Sparse coding technique

Fig.2 shows the sparse coding technique which is the core part of the proposed work. As we already aware of that for most of the computer applications, the image can be analyzed and processed at the patch level instead of at the individual bit or pixel level. Patches or a group of patches are having type of similar information within them. Hence they have advantages in terms of computation and generalization. Such patches or regions allow the user to correlate the information in any image in a certain way within a limited local searching range. Here using an over-complete dictionary designed **D** that contains K prototype signal atoms, the image patch **y** is represented by sparse linear combinations of these atoms as:

$$y = round\left(D\tilde{x}\right) + \tilde{e} \qquad (1)$$

It shows that only a small number of coefficients $\tilde{x}$ and the corresponding residual error $\tilde{e}$ caused by sparse representation require space to record. Hence they can be heavily compressed, and thus they result in a large hiding room.

The output of this block will be the cover image represented by sparse coefficients.

2. Coefficient encoding and residual error self embedding: After the cover image is represented by sparse coefficients, the corresponding coefficients and reconstructed residual errors are directly encoded for the given selected patches. To losslessly recover the

cover image the calculated residual error for each patch is reversibly embedded within the non-selected patches of the cover medium. For this embedding process, a standard RDH algorithm is used. The trained dictionary is also embedded into the encrypted image for further use ie. for lossless recovery of cover image process.
The output of this block will be selected patches with a space reserved for further data hiding.

3) Image Encryption: From the room preserved self embedded image Ic, we generate the encrypted image Ie by a stream cipher RC4. This algorithm uses Boolean functions to generate the encrypted version. It simply performs the bit exor operation by using keys provided by user. If the eight bits of the pixel $p_{i,j}$ (i = 1, 2,... N1, j = 1, 2,... N2).

Thus $b_{i,j,k} = \left[\frac{p_{ij}}{2^m}\right] \bmod 2$, m = 0, 1,..., 7. (2)

Then, the encrypted bit stream can be expressed as $b'_{i,j,m} = b_{i,j,m} \oplus r_{i,j,m}$ where m = 0, 1,..., 7
The encrypted version of cover image Ie is further subjected to data embedding process performed by data hider.
4. Secret data embedding: Once the encrypted image is received, the data hider starts the secret data embedding process for management or authentication requirement of the application. Separate key is used by data hider for strong authentication which is called as data hiding key. The standard RDH algorithm is used for this data hiding. After embedding the secret data, the position of the first selected patch for data hiding and the size of the hiding room for each patch are also embedded into the encrypted image containing additional embedded data with RDH algorithm. Finally this encrypted image with hidden data will be transmitted to the receiver.

Receiver: The data extraction and image decryption are processed separately at the receiver side. With the encrypted image containing additional embedded data, the receiver faces three situations depending on whether the receiver has encryption key and/or data hiding key. These 3 cases are discussed below.

1. Data Extraction With Only Data Hiding Key:
For the receiver who only has data hiding key K_d. It will first extract and compute the starting position and the hiding room size for each patch and divides the received image into non-overlapped N × N patches. Then, data extraction will be finished by checking the last n^d (parameter bits) bits for the selected patches in the received image. After that, all original hidden data are extracted and recovered with the data hiding key K_d.

2. Image Decryption With Only Encryption Key:
In this case, the receiver will have only encryption key K_e. After extraction the position of the first selected patch by RDH algorithm, all the selected patches will be identified one by one. In addition, the dictionary D is also obtained by extraction algorithm. After patch segmentation of the received image, the decryption procedure will be performed and it will include two cases as unselected patch decryption and selected patch decryption.

3. Data Extraction and Image Recovery With Both Data Hiding and Encryption Keys:
If the receiver has both the data hiding key K_d and encryption key Ke, the data extraction and image recovery will achieve full reversibility. On the one hand, with the data hiding key K_d, one can extract the hidden secret data without any error. On the other hand, with the encryption key Ke, user will first perform direct image decryption and then the corresponding coefficient for selected patches will be obtained. After that, the residual errors will be extracted from the non-selected patches & the recovery patches will be computed.

4. EXPERIMENTAL ANALYSIS AND RESULTS

In this section, we conduct several experiments to evaluate the proposed algorithm which include: choice of dictionary parameters, image encoding, and performance analysis on public available standard images.

A. Choice of Dictionary Parameters:

Our dictionary training is based on patch size 8*8 taken from image database. For the training process, we adopt K-SVD [9] as the trainer. In our implementation, the maximal number of iterations, T, of K-SVD is set to 50. The output dictionary has the size of 16×K. The dictionary coefficients are computed using orthogonal matching pursuit (OMP) algorithm with a fixed maximal number of non zero coefficients L. K and L are selected by referring to the performance comparison of our algorithm. For making a best choice of dictionary parameters which represent the higher embedding rate, we compute the average position bits ($n - p$), value bits ($n - v$), and error bits ($n - e$) for all the patches in the training set. figure3 shows demonstration of trained dictionary in matlab 2013a.

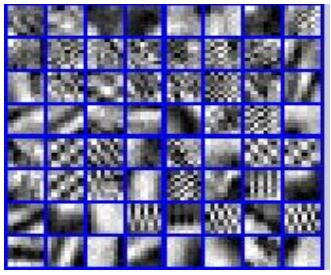

Fig.3: Dictionary overview

B. Image Encoding:

Once the well-trained dictionary is obtained, the given image can be represented by the sparse coding according to this dictionary. Our encoding strategy allows the user to learn which part of the image or which type of image can be easily represented by sparse coding. It can be seen clearly that patches with smoother textures, such as backgrounds or plain clothes, are simpler to represent than complex ones. Analysis shows that the patches containing higher frequency elements have the higher residual errors. According to the embedding requirement, some patches are selected for data hiding, and its corresponding residual errors are needed to be self-embedded within the cover media. Fig.4 shows the encoded version of the cover image.

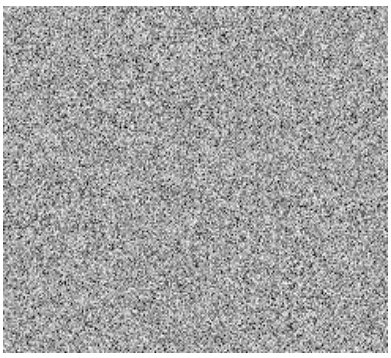

Fig.4: Encrypted image

C. Reversible Data Hiding:

Once the data hiders acquire the encrypted images, they can embed particular amount of data for some particular demands like authentication. Fig.5 shows the results of our proposed method with ER = 1.33 bits per pixel. Fig. 5(a) shows the original cover image and 5(b) show the encrypted image with embedded data. For the receivers that only have data hiding key Kd, they can extract the hidden data losslessly. Moreover, the receiver with encryption key Ke can directly decrypt the image with higher image quality. When both of the keys Kd and Ke are available, we can losslessly recover the original image by decrypting and decoding the corresponding reconstructed coefficients and residual errors. The directly decrypted image is shown in Fig.5(c) and reconstructed/recovered image is shown in Fig. 5(d). From experimental results, it can be said that the recovery version is identical to the original image visually, with higher image quality.

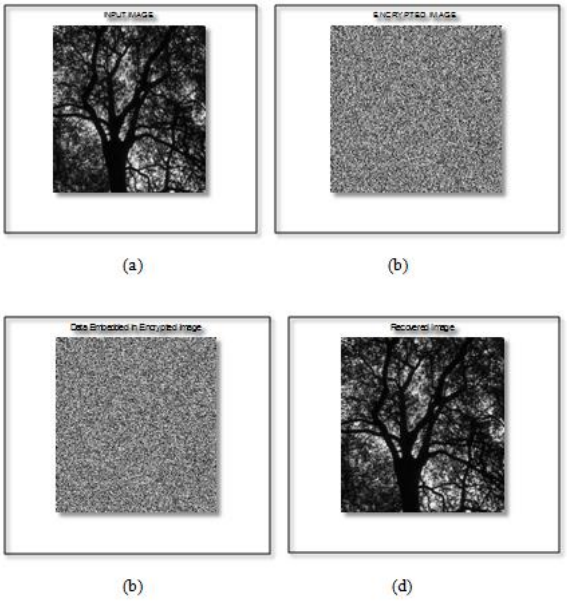

Fig.5. Figure 9. Results demonstration of proposed method with ER 1.33 bits per second. (a) Input image. (b) Encrypted image. (c) Data embedded in encrypted image (d) Reconstructed image.

Performance analysis parameters:

1. Embedding Rate: Embedding rate is the main parameter which is taken into account while designing this algorithm. Calculating the embedding rate of the image is done by computing the different bits as follows. If we assume the selected patch number is denoted as C, our MER for the data hider is computed as:

$$MER = \frac{C \times (8N^2 - L(n_p + n_v) - n_b) - n_a}{N1 \times N2} \quad (3)$$

Where C is the selected patch number of the cover image, N is patch size, L is the pre determined no of non zero entries in the dictionary, n_p is position bits, n_v is value bits, n_b is parameter bits and n_a is the dictionary size and is fixed for our algorithm.

If we assume the size of the data to be hidden is denoted as M, the relationship between selected patch number C, data hidden size M and room preserving per patch n_d is

$$C = \frac{M}{n_d} = \left[\frac{M}{8N^2 - L(n_p + n_v) - n_b - n_a} \right] \quad (4)$$

Then, we rewrite the above equation as

$$C = \left[\frac{M + n_a}{8N2 - U} \right] \quad (5)$$

Where

$$C = \frac{M}{n_d} = \left[\frac{M}{8N^2 - L(n_p + n_v) - n_b - n_a} \right] \quad (6)$$

Then, we rewrite the above equation as

$$C = \left[\frac{M + n_a}{8N2 - U} \right] \quad (7)$$

Where

$$U = L([\log_2 K] + 11) + \left(\left[\log_2 \frac{N1}{N} \right] + \left[\log_2 \frac{N2}{N} \right] \right) \quad (8)$$

The graph of embedding rate for different size of images is shown below.

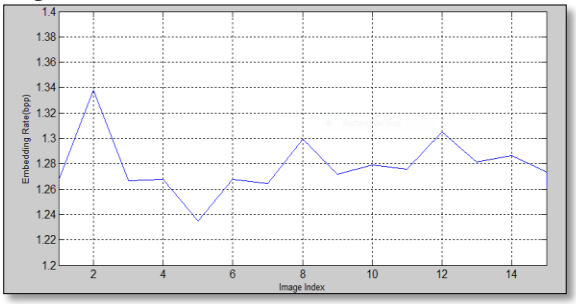

Fig.6: Graph of ER (bpp) value v/s Image Index

2. Peak Signal to Noise ratio (PSNR):

To quantitatively measure the performance of our proposed method, we also computed PSNR values of directly decrypted images and reconstructed images. PSNR is very common in image processing. It is used to measure the quality of reconstruction of lossy and lossless compression techniques. Here a simple use of PSNR is in the comparison between the original image and the reconstructed/directly decrypted image. It is measured by using MSE (mean square error) value. The simple formula to measure the PSNR value of any RGB image is as follows:

$$PSNR = 10 \log_{10} \frac{R^2}{MSE} \quad (9)$$

Where R is the max no of pixel intensity value and MSE ie. Mean Square Error is given by,

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N} \quad (10)$$

The graphs of PS NR values for reconstructed images and directly decrypted images is as shown below.

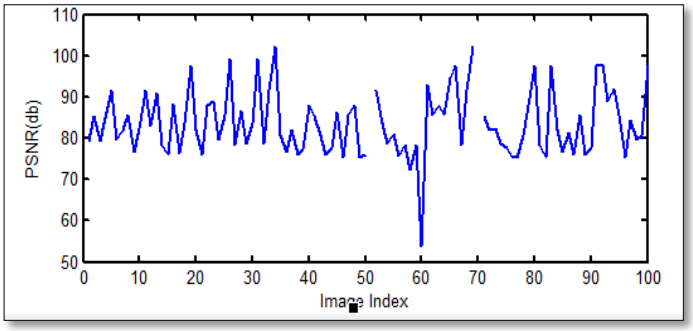

Fig.7: Graph of PSNR value v/s Image Index for reconstructed images

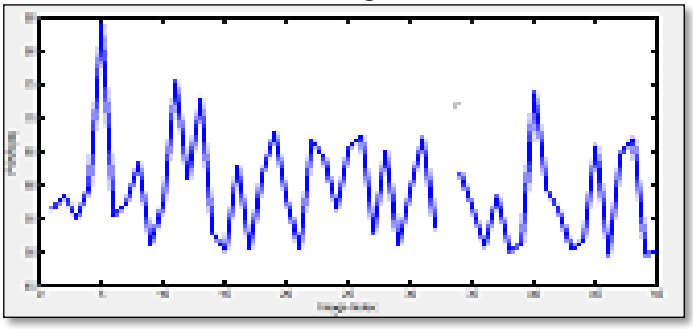

Fig.8: Graph of PSNR value v/s Image Index for directly decrypted images

3. Structural Similarity Index (SSIM): SSIM is another perceptual image quality metric that indicates image quality degradation caused by processing such as data compression or degradation caused by losses in data transmission process. SSIM is a full reference metric that requires *two* images from the same image capture. One is a reference image and another is a processed image. In most of the cases, the processed image is typically compressed using any compression technique. SSIM cannot judge which of the two images is better since it actually measures the perceptual difference between two similar images. That difference must be known to user by identifying which is the original image and which has been subjected to additional processing such as data compression. Unlike PSNR, SSIM is based on visible structures in the cover image[10]. Here SSIM index is calculated for 100 images and the graph of it is as follows.

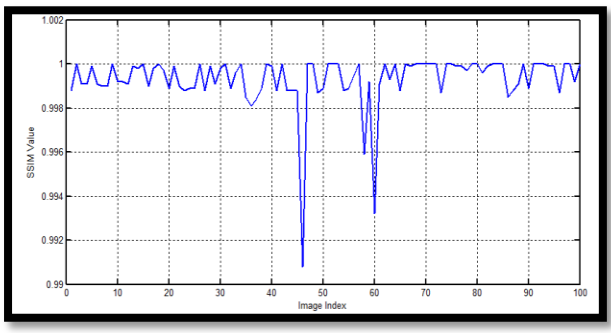

Fig.9: Graph of SSIM value v/s Image Index

5. CONCLUSION

In this paper, an improved framework for separable and reversible data hiding in encrypted images is praposed. This method consists of phases like image encryption, data hiding and data-extraction/image recovery. The major advantage of this method is that a large embedding rate is achieved by this method by increasing the size of the vacating space per patch. Also the operations of data extraction and cover image recovery are separable depending upon keys and they are free of any error.

Acknowledgments

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank Mr. S.S.Shirgan, for providing me an opportunity to do the project work on reversible data hiding and giving me all support and guidance which made me complete the project duly. I am extremely thankful to him for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.

I owe my deep gratitude to our project guide Mr. V.S. Marathe sir who took keen interest on my project work and guided me all along, till the completion of my project work by providing all the necessary information for my project.

REFERENCES

[1] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 3, pp. 553–562, Mar. 2013

[2] W. Zhang, K. Ma, and N. Yu, "Reversibility improved data hiding in encrypted images," Signal Process., vol. 94, pp. 118–127, Jan. 2014.

[3] X. Xiaochun Cao, Ling Du, Xingxing Wei, Dan Meng and Xiaojie Guo, "High Capacity Reversible Data Hiding In Encrypted Images By Patch Level Sparse Representation" IEEE Transactions on cybernetics, VOL. 46, NO. 5, MAY 2016

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

[5] W. Hong, T. S. 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.

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

[7] X. Zhang, Z. Qian, G. Feng, and Y. Ren, "Efficient reversible data hiding in encrypted images," J. Vis. Commun. Image Represent., vol. 25, no. 2, pp. 322–328, Feb. 2014.

[8] Z. Yin, B. Luo, and W. Hong, "Separable and error-free reversible data hiding in encrypted image with high payload," Sci. World J., vol. 2014, Mar. 2014, Art. ID 604876.

[9] Michal Aharon, Michael Elad.Alfred Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation" IEEE Signal Process, vol. 54, no. 11, Nov. 2006.

[10] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh and Eero P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity", IEEE IMAGE PROCESSING, VOL. 13, NO. 4, APRIL 2004