International Journal of Advanced Research and Development
ISSN: 2455-4030
Impact Factor: RJIF 5.24
www.advancedjournal.com
Volume 3; Issue 2; March 2018; Page No. 953-958

Analysis and implementation of high efficient steganography using patch level sparse representation

Sonali M Pawar[1], Vikas Marathe[2]
[1] P.G. Student, Department of E & TC Engineer, N.B. Navale Sinhgad College of Engineering, Solapur, Maharashtra, India
[2] Assistant Professor, Department of E & TC Engineering, N.B. Navale Sinhgad College of Engineering, Solapur, Maharashtra, India

Abstract
In data communication the main problem is with security and confidentiality of the contents of communication. Hence, the field of Steganography has been attracted considerable attention from last two decades. Recently Steganography has become most important as it maintains the best property to recover the original data while extracting the contents of an image. Reversible data hiding is a steganographic technique in which the image containing the hidden data is losslessly recovered. Here introduced a new technique which uses sparse representation for reversible data hiding in encrypted images. Compressing the signals and then representing them using dictionary atoms is done by using sparse representation. The proposed work focuses on the patch level sparse representation instead of pixel level for hiding secret data in the encrypted images, because the neighboring pixels (patch) are highly correlated to each other. By using this method, large space for hiding secret message can be achieved.

Keywords: steganography, reversible data hiding, sparse representation, K-SVD

1. Introduction

Steganography is a secret communication process where a part of information (a secret message) is hidden into another part of information called as cover in such a way that the existence of the secret information is undetectable by usual vision. Steganography is classified into several sub classes such as text, audio, video or image Steganography, depending upon which media is used as the cover medium. Among these available forms, digital image Steganography is more popular form as images are widely used as medium of communication for data transmission.

The basic concept of Steganography is to hide the very presence of communication by embedding message into the cover objects ie. it is a method to hide (embed) additional message into some distortion free cover media which further aims to recover both the embedded secret information and the original cover image. Most of the data hiding technique embeds the message into the cover image by modifying the LSB's of the cover image, so they cause some distortion to cover image and hence make it difficult to reconstruct the original image. In most of the fields, small amount of signal distortion caused by embedding process are allowed. But in applications like medical or military imagery, even a small distortion is not allowed. Hence in such type of application fields it is desirable to design a different kind of data hiding method, which is popularly known as reversible data hiding (RDH). Reversible Data Hiding (RDH) is used to embed a piece of data into an image to generate a marked image and after extracting process the original image can be recovered from the marked image. RDH is also called as invertible or lossless data hiding technique. The embedding rate and quality of the decrypted image are important parameters for the measurement of the performance of the RDH algorithm as the increase in the hiding rate causes more distortion in the cover image content.

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.

Depending upon the previous work in steganography field, some other methods also divide the cover image into patches to perform data hiding, but their main aim is to basically consider the correlation of pixels within the selected cover patch. Therefore, they are kind of the pixel-level compressive methods essentially. On the other hand, here the patches are considered as a whole for processing and represented them using a small number of sparse coefficients. Hence this method is beyond the traditional pixel-level case. And thus a high capacity room is made available which in turn increases the efficiency of our algorithm.

2. Related Work

In steganography, 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 designed. 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] represent different methods for VRAE technique. For RRBE, there are two methods 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 decrypts the encrypted image, and then extracts 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 scheme 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 is not separate from the content decryption. From this point of view, Zhang [6] proposed a novel scheme for separable RDHEIs which include advancement of cryptography with steganography. In this, the original uncompressed image is encrypted by using an encryption key to create a sparse space to hide some additional data, 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*. proposed a novel scheme of RDHEIs in [7] which is mainly based on lossless compression of encrypted data by 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 steganography method. Moreover, Yin *et al*. proposed a RDHEI scheme in [8] which offers 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 is applied to those blocks. Depending on smoothness of blocks the data hider selects the several smoother blocks for data embedding process.

As we know the entropy of the image increases due to encryption, the room vacating is more difficult work in the entire process. The schemes in [4]–[8] can achieve a small amount of payloads 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. Hence this fact invokes that losslessly vacating room from encrypted images is relatively difficult task and sometimes it is inefficient. To overcome this disadvantage, the RRBE methods [2], [3] are kept forward. Zhang *et al*. in [2] estimated some pixels before encryption and then the additional data are embedded in the estimating errors instead of embedding data in encrypted images directly. In advancement of this scheme, Ma *et al*. [3] designed an effective scheme by RRBE. In his proposed method, 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 kept forward by Xiaochun Cao in [1]. His proposed method inherits the merits of RRBE technique based on patch level sparse representation. The proposed method not only separates the data extraction from image decryption but also achieve excellent performance 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.

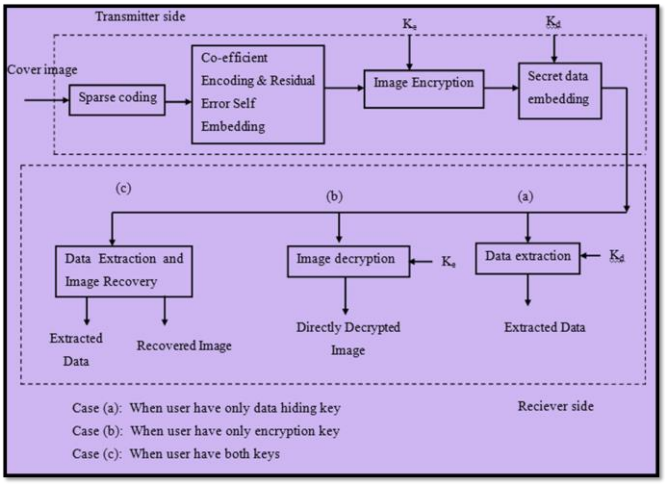

Fig 1: Overview of proposed system.

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.

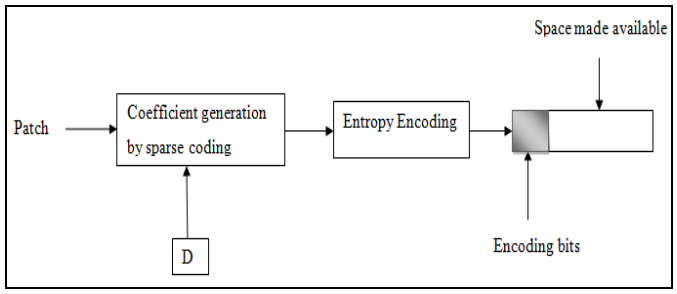

Fig 2: Sparse coding technique

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:

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

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).

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

Then, the encrypted bit stream can be expressed as

$$b'_{i,j,m} = b_{i,j,m} \oplus r_{i,j,m} \text{ where } m = 0, 1,..., 7$$

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 result

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. Figure 3 shows demonstration of trained dictionary in matlab 2013a.

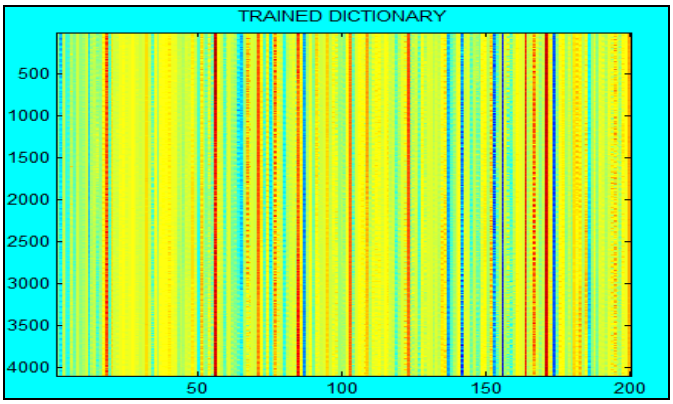

Fig 3: Dictionary overview

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.

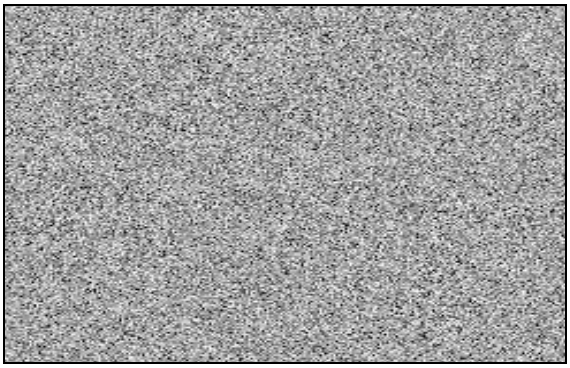

Fig 4: Encrypted image

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(a) (b) (c) (d): Results demonstration of proposed method with ER 1.33 bits per second. (a) Original image. (b) Encrypted image. (c) Directly decrypted 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:

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

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

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

Then, we rewrite the above equation as

$$C = \left\lceil \frac{M + n_a}{8N2 - U} \right\rceil$$

Where

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

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

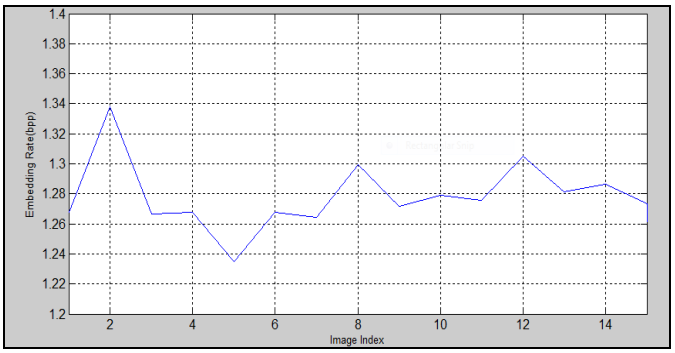

Fig 6: Graph of ER (bpp) value v/s Image Index

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:

$$PSNR = 10 \log_{10} \frac{R^2}{MSE}$$

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

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

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

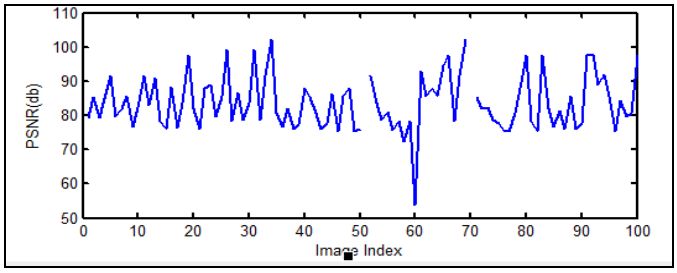

Fig 7: Graph of PSNR value v/s Image Index for reconstructed images

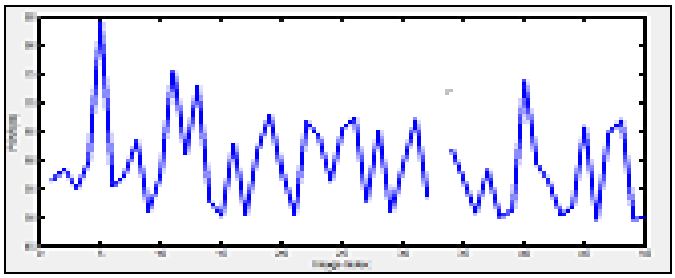

Fig 8: Graph of PSNR value v/s Image Index for directly decrypted images

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.

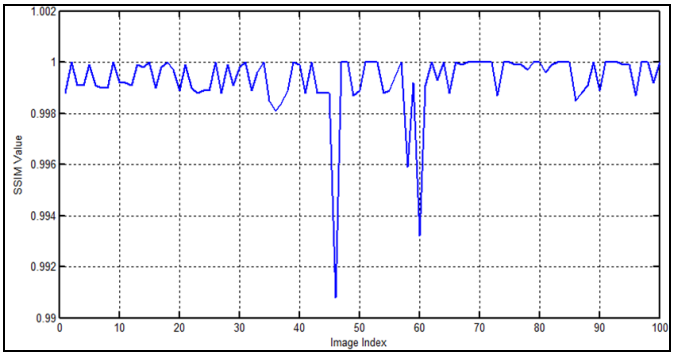

Fig 9: Graph of SSIM value v/s Image Index

5. Conclusion

In this paper, a new 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.

6. References

1. Ma K, Zhang W, Zhao X, Yu N, Li F. Reversible data hiding in encrypted images by reserving room before encryption, IEEE Trans. Inf. Forensics Security. 2013; 8(3):553-562.
2. Zhang W, Ma K, Yu N. Reversibility improved data hiding in encrypted images, Signal Process. 2014; 94:118-127.
3. Xiaochun Cao X, Ling Du, Xingxing Wei, Dan Meng, Xiaojie Guo. High Capacity Reversible Data Hiding In Encrypted Images By Patch Level Sparse Representation IEEE Transactions on cybernetics, 2016, 46(5).
4. Zhang, Reversible data hiding in encrypted image, IEEE Signal Process. Lett. 2011; 18(4):2550-258.
5. Hong W, Chen TS, Wu H. An improved reversible data hiding in encrypted images using side match, IEEE Signal Process. Lett. 2012; 19(4):199-202.
6. Zhang X. Separable reversible data hiding in encrypted image, IEEE Trans. Inf. Forensics Security. 2012; 7(2):826-832.
7. Zhang X, Qian Z, Feng G, Ren Y. Efficient reversible data hiding in encrypted images, J. Vis. Commun. Image Represent. 2014; 25(2):322-328.
8. Yin Z, Luo B, Hong W. Separable and error-free reversible data hiding in encrypted image with high payload, Sci. World J. 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. 2006; 54(11).
10. Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, Eero Simoncelli P. Image Quality Assessment: From Error Visibility to Structural Similarity, IEEE Image Processing, 2004, 13(4).