Chapter 42
A Novel Distributed Image Steganography Method Based on Block-DCT

Rosemary Koikara, Dip Jyoti Deka, Mitali Gogoi and Rig Das

Abstract Distributed Image Steganography (DIS) is a method of hiding secret information in multiple carrier images, making it more difficult to trace than conventional steganographic techniques, and requiring a collection of affected images for the retrieval of the secret data. In this paper we concentrate on performing DIS on grayscale images using Block-DCT (Discrete Cosine Transformation). Distributed Image Steganography using Block-DCT adds to the security of DIS by embedding the secret data in the Frequency Domain. This makes the carrier images more immune to various steganalysis attacks as the secret data is more evenly distributed amongst the pixels of the carrier images making it more difficult to determine its existence. We use parity check in order to compensate for round-off errors that are typically associated with DCT.

Keywords Steganography · Distributed steganography · Block-DCT · PSNR

R. Koikara · D.J. Deka · M. Gogoi
Department of Computer Science and Engineering and Information Technology, Don Bosco College of Engineering and Technology, Guwahati 781017, Assam, India
e-mail: rosekoikara@gmail.com

D.J. Deka
e-mail: dipjyotideka123@gmail.com

M. Gogoi
e-mail: mitaligogoi@gmail.com

R. Das (✉)
Department of Computer Science and Engineering, National Institute of Technology, Rourkela 769008, Orissa, India
e-mail: rig.das@gmail.com

H.A. Sulaiman et al. (eds.), *Advanced Computer and Communication Engineering Technology*, Lecture Notes in Electrical Engineering 315,
DOI 10.1007/978-3-319-07674-4_42

42.1 Introduction

Steganography is the art of hiding information in ways that prevent the detection of hidden messages [1, 2]. It includes a vast array of secret communication methods that conceal the message's very existence. Some of the more common methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications [3]. Many conventional steganographic schemes hide the secret data in a single host image. These techniques include least significant bit (LSB) insertion or frequency domain embedding using the Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT) or Wavelet Transform (WT) [2, 4].

However, a common weakness of these techniques is that the secret data are all in a single information-carrier, and the secret data cannot be revealed completely [5], if the information-carrier is lost or crippled. Use of many duplicates may overcome the weakness but increase the danger of security exposure. Moreover conventional steganographic methods have restricted data hiding capacity.

Distributed Image Steganography overcomes these shortcomings by using a (k, n) threshold based image secret sharing technique for k ≤ n and allows large information payload embedding by generating n steganographic images. DIS allows (i) k or more steganographic images to reconstruct the secret image, and (ii) (k − 1) or fewer images cannot reveal the secret image [1]. Figure 42.1 shows

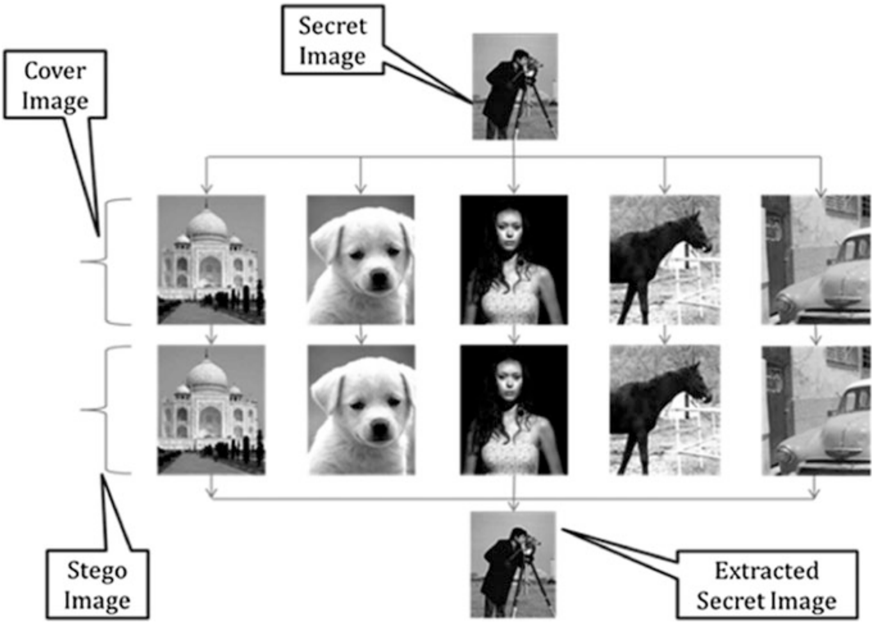


Fig. 42.1 The block diagram of a simple distributed steganographic system with (3, 5) threshold scheme

the block diagram of a simple DIS system with (3, 5) threshold scheme that means the secret image will be distributed within 5 stego images and any 3 stego images are enough to regenerate the secret image.

Distributed Image Steganography in Block-DCT is refinement of the existing DIS technique in order to enhance the security of the secret data against various steganalysis attacks. In this paper, we have implemented Distributed Image Steganography by means of Block-DCT. We take the cover images and embed the shares of the secret data into the transformation domain of the cover images. During extraction, the shares need to be retrieved from the transformation domain of the stego images in order to recreate the original secret data. This whole process comprises of a series of computationally intensive operations and hence is impractical to deploy steganalysis for DIS in a massive scale.

This paper is organized as follows. Section 42.2 explains some related works of Distributed Steganography. The proposed novel embedding and extraction algorithms for DIS using Block-DCT are explained in Sect. 42.3. All experimental results are shown in Sects. 42.4 and 42.5 concludes.

42.2 Related Work

Different researchers employed different techniques for the purpose of distributing secret image over a set of stego images. Shamir's Secret Sharing Scheme and Thien and Lin's Secret Image Sharing Scheme are the two essential processes to protect secret image in DIS. Following are some of the related works carried out by some of the researchers.

42.2.1 Thien and Lin's Secret Image Sharing Scheme [6]

Suppose we want to divide the secret image D into n shadow images ($D_1, \ldots, D_n$), and the secret image D cannot be revealed without r or more shadow images. In the proposed method, they generate the $r - 1$ degree polynomial, by letting the r coefficients be the gray values of r pixels. Therefore, the major difference between Thien and Lin's method and Shamir's [7] is that they use no random coefficient. Because the gray value of a pixel is between 0 and 255, they took the prime number p be 251 which is the greatest prime number not larger than 255. To apply the method, it must truncate all the gray values 251–255 of the secret image to 250 so that all gray values are in the range 0–250. The image is divided into several sections. Each section has r pixels, and each pixel of the image belongs to one and only one section. For each section j, define the $r - 1$ degree polynomial as: $q_j(x) = (a_0 + a_1x + \cdots + a_{r-1}\, x^{r-1}) \bmod 251$, Where $a_0, \ldots, a_{r-1}$ are the r pixels of the section, and then evaluate $q_j(1), q_j(2), \ldots, q_j(n)$, The n output pixels $q_j(1) - q_j(n)$ of this section j are sequentially assigned to the n shadow images. Since for each given section (of r pixels) of the secret image, each shadow image receives one of the generated pixels; the size of each shadow image is 1 / r of the secret image. The reveal phase uses any r (of the n) shadow images, and the Lagrange's interpolation to extract the secret image.

42.2.2 An Estimation Approach to Extract Multimedia Information in Distributed Steganographic Images [1]

In this paper, a blind steganalysis technique is been proposed to attack DIS in which no host image is required for detecting and extracting hidden information. To develop this counter-measure for DIS they have put two assumptions: (i) One hidden image in a set of unsuspected steganographic images, (ii) Threshold value k is known.

The counter-measure process consists of three modules:

(i) *Detection Module (DM)* is responsible for detecting possible steganographic images
(ii) *Estimation Module (EM)* is responsible for extracting image shares embedded in steganographic images
(iii) *Reconstruction Module (RM)* is responsible for combining quantized image shares to reconstruct the secret image

42.3 Proposed Novel Method for Distributed Image Steganography Based on Block-DCT

As Thien and Lin's method is primarily based on Spatial Domain and there is also a Steganalysis method to counter the LSB based DIS [1], a novel method for DIS based on Block-DCT is been proposed in this paper which will add much more security to the secret image as the secret information is being embedded in frequency domain and extraction of the secret information is much more difficult than spatial domain based techniques.

DIS using Block-DCT is a refinement of DIS. Here we use a polynomial equation to create shares of the secret image and embed these shares into multiple cover images that have been transformed using Block-DCT. The schematic/block diagram of the whole process is given in Figs. 42.2 and 42.3.

Our novel algorithm for DIS based on Block-DCT is based on (k, n) Threshold Scheme and has two parts, one for Embedding the Secret Image inside n–Cover Images and another for Extracting the Secret Image from k–Stego Image. The Secret Image is divided into several sections. Each section has k pixels, and each pixel of the secret image belongs to one and only one section. For each section j, we define the following k − 1 degree polynomial:

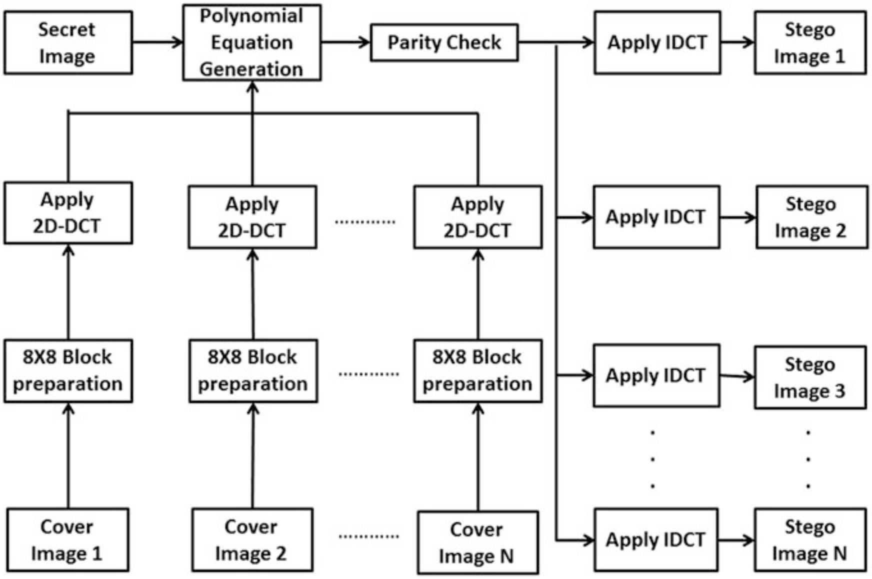


Fig. 42.2 Insertion of a secret image inside n–cover images

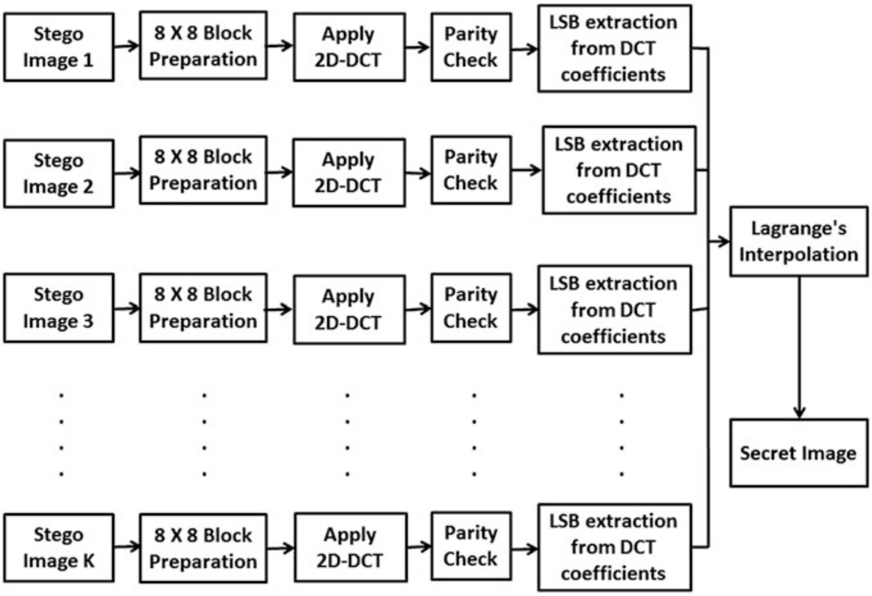


Fig. 42.3 Extraction of secret image from k–stego images

$$p_j(x) = \left(a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}\right) \bmod 256 \tag{42.1}$$

$$q_j(x) = \text{floor}\left[\left(a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}\right)/256\right] \tag{42.2}$$

where, value of x ranges from 1 to n and a_0 to a_{k-1} are Secret Image's k number of pixel's intensity values. These intensity values changes sequentially e.g., first we take 1–8 pixels' intensity values then 9–16 pixels' intensity values and so on. Modulus and Floor operations are performed using 256 as divisor because a grey level image has 256 different intensity levels. Equation (42.1) finds the Remainder Value after performing Modulus operation and (42.2) finds the Quotient value after performing Floor operation. This method consists of several phases as explained below.

42.3.1 Block-DCT

We have created blocks of size 8 × 8 of the cover images and perform DCT operation on each of these blocks. This makes sure that the secret image is evenly distributed amongst all the pixels on the 8 × 8 blocks.

42.3.2 Polynomial Equation

A polynomial equation is used to divide the secret image into its shares. The polynomial equation used:

$$S_x(i,j) = I(i \times k + 1, j) + I(i \times k + 2, j)x + \cdots + I(i \times k + k, j)x^{k-1} \tag{42.3}$$

where, S_x is the share of the secret image I and j denote the pixel positions, I is the secret image, k is the threshold of the secret sharing scheme. We calculate both the remainder and the quotient of this equation as follows:

$$M_x(i,j) = \left(I(i \times k + 1, j) + I(i \times k + 2, j)x + \cdots + I(i \times k + k, j)x^{k-1}\right) \bmod 256 \tag{42.4}$$

$$Q_x(i,j) = \text{floor}\left((I(i \times k + 1, j) + I(i \times k + 2, j)x + \cdots + I(i \times k + k, j)x^{k-1}\right)/256) \tag{42.5}$$

where, M_x is the remainder of the polynomial equation, Q_x is the quotient of the polynomial equation, mod is the operation that calculates the remainder floor gives the nearest integer that is less than or equal to the real value. We need the values of both the remainder and the quotient to reconstruct the image during extraction.

42.3.3 Lagrange's Interpolation

Lagrange's interpolation is the Nth degree polynomial approximation formula to the function P(x), which is known at discrete points x_i, i = 0, 1, 2, …, Nth

$$\begin{aligned}P(x) = &\frac{(x - x_2)(x - x_3)\ldots(x - x_n)}{(x_1 - x_2)(x_1 - x_3)\ldots(x_1 - x_n)}y_1 + \frac{(x - x_1)(x - x_3)\ldots(x - x_n)}{(x_2 - x_1)(x_2 - x_3)\ldots(x_2 - x_n)}y_2 \\ &+ \cdots + \frac{(x - x_1)(x - x_2)\ldots(x - x_{n-1})}{(x_n - x_1)(x_n - x_2)\ldots(x_n - x_{n-1})}y_n\end{aligned} \tag{42.6}$$

42.3.4 Parity Check

This is an error detection and correction code used to eliminate the rounded-off errors that may arise due to transformation into the frequency domain. Before performing Block-DCT over the cover image, the cover image is changed into DOUBLE format. After performing Block-DCT the pixels intensity values gets changed into frequency domain which has negative fraction values. Now as the LSB of these negative fraction values can't be modified to insert the Secret Image's data, it should have to be changed into positive integer number. After modifying the LSB of the cover images and performing Inverse 2-D DCT, to generate the stego image it is required to convert these pixels intensity value into UINT8 format from DOUBLE format (as the original cover image was in UINT8 format). Now at the time of change from DOUBLE to UINT8 format the fraction value gets rounded off. But the modification on LSB has been done in those fraction part itself. For example if a pixel's intensity is 169.78 then it gets rounded off into 170 and if it is 152.21 then rounded off to 152. So the value of every LSB gets changed. Thus a small amount of change can change the actual Secret Image into havoc [8]. To reduce this loss of information an algorithm is devised to detect and correct this error. The algorithm we have formulated is a modified form of parity check. This method does not completely eliminate rounded-off error; it just reduces to a certain extent. Following is the devised Parity Checking Algorithm.

42.3.4.1 Algorithm for Parity Check During the Insertion of Secret Image

Input: Stego-image

Output: Stego-image with parity check information embedded into its 1st and 2nd LSB Position

Step-1: Calculate even parity and embed it into the 1st LSB position

Step-2: Calculate odd parity and embed it into the 2nd LSB position

42.3.4.2 Algorithm for Parity Check During the Extraction of Secret Image

Input: Stego Image
Output: Stego Image with Corrected pixel value

```
Step-1: Let x = Pixel value of the Stego Image.
Step-2: Calculate even parity of x and embed it in x's 1st LSB
Step-3: Calculate odd parity of x and embed it in x's 2nd LSB
Step-4: If x is equal to the Pixel value then it is correct or else we need to
        increment or decrement the pixel value respectively.
Step-5: Repeat Step-2 and Step-3 till x becomes equal to the pixel value
```

42.3.5 Proposed Novel Algorithm for DIS Based on Block-DCT

Embedding Algorithm

Input: *n* number of M × N *Carrier Images* and a P × Q *Secret message/Image*
Output: *n* number of M × N S*tego-Images*

```
Step-1: Read Secret Image and Cover Images
Step-2: Divide the Cover Image into non overlapping blocks of size 8 × 8 and
        apply 2-D DCT on each of the blocks of the cover image.
Step-3: Sequentially take k–number of not-shared-yet pixels of the Secret Image
        and use (42.1) and (42.2) to find the n number of Remainder and
        n number of Quotient value.
Step-4: Change the 3rd LSBs of DCT transformed n Cover Images to insert
        each set of Remainder and Quotient values found in Step-4 in each of
        the Cover Images.
Step-5: Add parity bit information into 1st and 2nd LSBs of every pixel of n-
        Cover Images using the algorithm proposed in Sect. 42.3.4.1.
Step-6: Repeat Step-3, Step-4 and Step-5 until all the pixels of the Secret Image
        are embedded into n-Cover Images.
Step-7: Write all the n Stego Images into the disk.
```

Extraction Algorithm

Input: *k* number of M × N S*tego-Images*
Output: A P × Q *Secret Image*

Step-1: Read k number of *Stego Images*

Step-2: Divide the Stego Image into non overlapping blocks of size 8 × 8 and apply 2-D DCT on each of the blocks of the Stego image.

Step-3: Extract k number of Remainders and k number of Quotients from the k *Stego Images* by extracting the 3rd LSBs of the pixels.

Step-4: Use Parity bit Checking as described in Sect. 42.3.4.2 to reduce the rounded off error (as described in Sect. 42.3.4).

Step-5: Use Lagrange's Interpolation to retrieve k number of pixel's intensity.

Step-6: Repeat Step-3 and Step-4 until the total number of pixels of the *Secret Image* are processed.

Step-7: Write the Extracted Secret Image into the disk.

42.4 Simulation and Results

In this section, some experiments are carried out on our proposed algorithm for Distributed Image Steganography (DIS).The measurement of the quality between the cover image f and stego-image g of sizes M × N is done using PSNR (Peak Signal to Noise Ratio) value and the PSNR is defined as:

$$PSNR = 10 \times log\left(255^2/MSE\right) \tag{42.7}$$

where,

$$MSE = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}(f(x,y)-g(x,y))^2/(M \times N) \tag{42.8}$$

$f\,(x,\ y)$ and $g\ (x,\ y)$ means the pixel intensity value at position $(x,\ y)$ in the cover-image and the corresponding stego-image respectively. The PSNR is expressed in dB. The larger PSNR indicates the higher the image quality i.e., there is only little difference between the cover-image and the stego-image. On the other hand, a smaller PSNR means there is huge distortion between the cover-image and the stego image.

All the simulation has been done using the MATLAB 7 program on Windows XP platform. Three different sets of 8-bit grayscale TIFF images of size 1024 ×1024 and 256 × 256 are used as the cover-images and secret image respectively to form the stego-images. Three different sets of cover images are considered to evaluate our results. Each set consists of five cover images. A single secret image was used to embed into all the three sets. Figure 42.4(1)–(5) shows the first set of original cover (carrier) images, Fig. 42.5(1)–(5) shows second set of cover images. Figure 42.6(1)–(5) shows third set of cover images of the proposed DIS method based on (4, 5) threshold scheme. Figure 42.7 shows the Original Secret Image and Fig. 42.8 shows three extracted secret images from Set 1, Set 2 and Set 3 of the cover images.

Fig. 42.4 Cover images set 1, (**1**)–(**5**) five cover images of proposed DIS method

Fig. 42.5 Cover images set 2, (**1**)–(**5**) five cover images of proposed DIS method

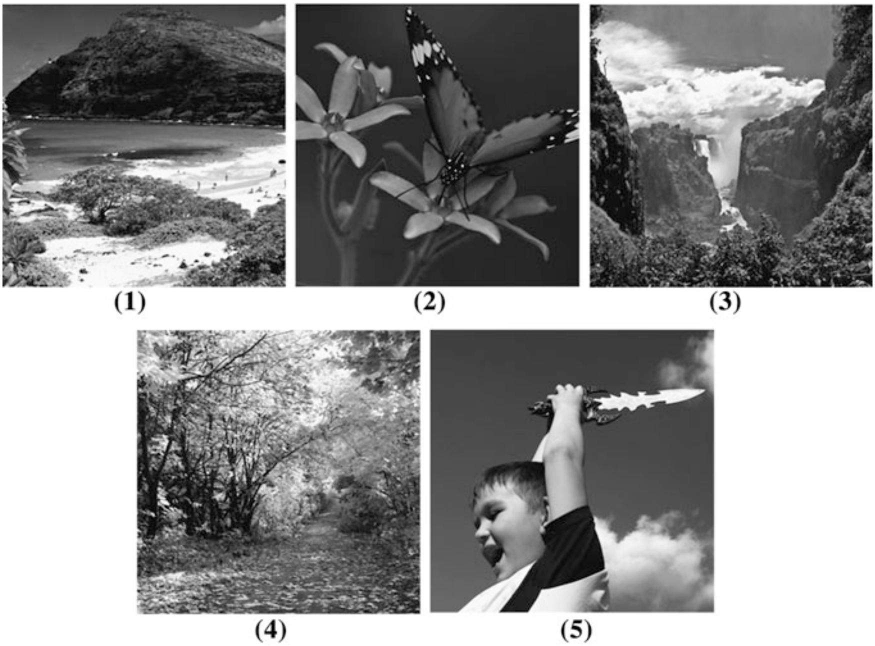
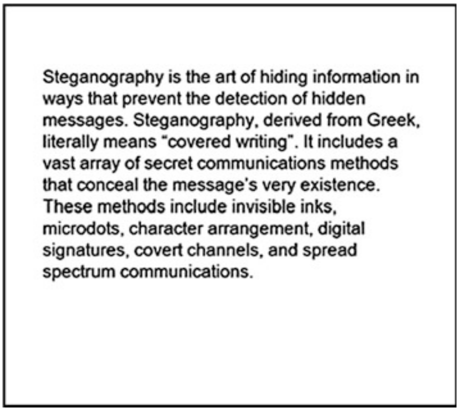


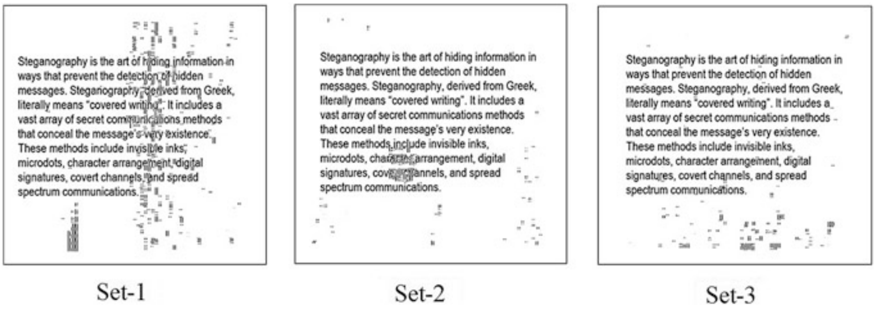
Fig. 42.6 Cover images set 3, (**1**)–(**5**) five cover images of proposed DIS method

Fig. 42.7 Original secret images



Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from Greek, literally means "covered writing". It includes a vast array of secret communications methods that conceal the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications.

Table 42.1 exhibit the PSNR comparison of Stego Images with their corresponding Cover Images for (4, 5) threshold scheme for all three sets of images. From Table 42.1 it is observed that for threshold scheme (4, 5) PSNR is greater than 40 dB for all the cases, so the quality of the stego image is quite acceptable and the deterioration in quality due to embedding of secret image cannot be distinguished by naked eye. Best result is achieved in case of Set 2 of Images, as



Set-1

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from Greek, literally means "covered writing". It includes a vast array of secret communications methods that conceal the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications.

Set-2

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from Greek, literally means "covered writing". It includes a vast array of secret communications methods that conceal the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications.

Set-3

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from Greek, literally means "covered writing". It includes a vast array of secret communications methods that conceal the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications.

Fig. 42.8 Secret image extracted from cover images set *1*, *2* and *3*

Table 42.1 PSNR comparison of cover images and stego images for (4, 5) threshold scheme for three different sets of cover images and their corresponding stego images

Cover Image Set	PSNR (DB) between cover image and stego image					
	Threshold scheme (4, 5)					
	Cover image-1 and stego image-1	Cover image-2 and stego image-2	Cover image-3 and stego image-3	Cover image-4 and stego image-4	Cover image-5 and stego image-5	Original secret image and extracted secret image
Set 1	+48.92	+44.86	+50.43	+47.79	+48.50	+40.37
Set 2	+46.64	+48.53	+48.96	+46.01	+47.41	+45.08
Set 3	+47.59	+48.36	+47.27	+47.99	+47.21	+42.97

the PSNR is greater than +45 dB for the extracted secret image. If we pass the extracted secret image through a median filter then there will be further improvement of the extracted secret image quality.

42.5 Conclusion

Our proposed novel Distributed Image Steganographic method based on Block-DCT which uses (k, n) Threshold Scheme improves the security and the quality of the Stego Images. According to the simulation results the Stego Images of our proposed algorithm are very difficult to distinguish from the Cover Images. We have achieved a quite satisfactory quality of the extracted Secret Image for any set of chosen Stego Images for (4, 5) threshold scheme. Distributing the Secret Image among n–number of Cover Images keeps the Secret Image away from stealing; destroying by any unintended users and Block-DCT adds to the security of the secret image as the secret image embedding is done in frequency domain. Hence the proposed method may be more robust against brute force attack.

References

1. Bai, L., Biswas, S., Blasch, P.E.: An estimation approach to extract multimedia information in distributed steganographic images. In: Proceedings of the 10th International Conference on Information Fusion, Quebec, Canada, 9–12 July 2007
2. Jhonson, F.N., Jajodia, S.: Exploring steganography: seeing the unseen. In: Proceedings of the IEEE paper of Feb 1998
3. Cheddad, A., Condell, J., Curran, K., Kevitt, M.P.: Digital image steganography: survey and analysis of current methods. J. Sign. Proces. **90**, 727–752 (2010)
4. Li, B., He, J., Huang, J., Shi, Y.Q.: A survey on image steganography and steganalysis. J. Inf. Hiding Multimedia Sign. Proces. **2**(2), 142–172 (2011)
5. Provos, N., Honeyman, P.: Hide and seek: an introduction to steganography. IEEE Secur. Priv. **1**(3), 32–44 (2003)
6. Thien, C.C., Lin, J.C.: Secret image sharing. J. Comput. Graph. **26**(5), 765–770 (2002)
7. Shamir, A.: How to share a secret. Commun. ACM. **22**(11), 612–613, (1979)
8. Das, R., Tuithung, T.: A review on "A novel technique for image steganography based on block-DCT and Huffman encoding". In: Proceedings of the 4th International Conference on Computer Graphics and Image Processing, ICGIP-2012, 6–7 Oct, Singapore, SPIE 2012