Session 5B

Image and Signal Processing

A Secure Image Steganography using LSB, DCT and Compression Techniques on Raw Images

#K.B.Raja[1], C.R.Chowdary[2], Venugopal K R[3], L.M.Patnaik [4]
[123] *Department of Computer Science Engineering, Bangalore University*
Bangalore - 560001, raja_kb@yahoo.com, kravi_c@yahoo.com, vkrajuk@vsnl.com
[4] *Microprocessor Applications Laboratory, Indian Institute of Science*
Bangalore – 560012, lalit@micro.iisc.ernet.in

Abstract

Steganography is an important area of research in recent years involving a number of applications. It is the science of embedding information into the cover image viz., text, video, and image (payload) without causing statistically significant modification to the cover image. The modern secure image steganography presents a challenging task of transferring the embedded information to the destination without being detected. In this paper we present an image based steganography that combines Least Significant Bit(LSB), *Discrete Cosine Transform*(DCT), *and compression techniques on raw images to enhance the security of the payload. Initially, the LSB algorithm is used to embed the payload bits into the cover image to derive the stego-image. The stego-image is transformed from spatial domain to the frequency domain using DCT. Finally quantization and runlength coding algorithms are used for compressing the stego-image to enhance its security. It is observed that secure images with low MSE and BER are transferred without using any password, in comparison with earlier works.*

1. INTRODUCTION

Steganography is derived from the Greek word *steganographic* which means covert writing. It is the science of embedding information into cover objects such as images that will escape detection and retrieved with minimum distortion at the destination. The rapid growth of internet coupled with high bandwidth and low cost computer hardware have propelled the explosive growth of steganography. The objective of modern steganography is to keep the payload(embedded information) undetected, but the steganographic systems, because of their invasive nature, leave behind the traces in the cover image. Steganography and cryptography are closely related. Cryptography provides confidentiality. Steganography on the other hand hides the message and there is no knowledge of the existence of the message. Steganography finds applications in watermarking, finger printing, and the modern multimedia message service; to name a few. The resultant image object obtained after embedding information into the cover image is called as stego object.

A famous example of steganography is "prisoner's problems" [1] where, two prisoners A and B wish to escape from the jail and their cellars are far apart. The only mode of communication is sending messages via the prison officer. Before they are arrested, they agree upon a stego system that describes the way the secret message is embedded into the covert text. If the prison officer detects conspiracy, the security will be further tightened. The prison officer can deliberately modify the stego text to foil the prisoners' escape.

In modern image stenography which exploits the advantages of the present day digital media, the earlier examples appear simple but the concepts are similar. This is largely due to the fact that, multimedia objects which generally permits the addition of significantly large amount of payload by means of simple modifications that preserve the perceptual content of the underlying cover image. Hence multimedia objects have been found to be perfect candidates for use as cover messages [2]. A steganographic technique is said to be ε–secure if the relative entropy of the probability distribution of cover images and stego-objects is less than or equal to ε. A steganography technique is perfectly secure if ε is zero [3].

Some of the well-known steganography methods are the following: LSB, masking and filtering and transform technique. In the LSB approach, the basic idea is to replace the Least Significant Bits (LSB) of the cover image with the Most Significant Bits (MSB) of the image to be hidden without destroying the statistical property of the cover image significantly. The LSB-based technique is the most challenging one as it is difficult to differentiate between the cover-object and stego-object if few LSB bits of the cover-object are replaced. In masking and filtering techniques two signals are embedded into each other in such a manner that only one of the signals is perceptible to the human eye. This is mainly used in watermarking techniques. In the transform-based method, the spatial domain is transformed to frequency domain using DCT, Fast Fourier Transforms(FFT), and Wavelets etc., [4].

2. RELATED WORK

Niels Provos [1] has explored a model to balance statistical properties of the cover image after embedding the pay load. Anderson et.al [2] have proposed a LSB based algorithm in which the quality of the retrieved image is poor. The two mathematical frameworks for steganography, i.e., informatic-theoretical model [3] and complex-theoretical view [5] give better mathematical foundations for applied steganography. A DCT co-efficient algorithm in which MSBs of hidden image are embedded into insignificant DCT coefficients of the cover image is presented in [6]. The usage, advantages, and limits of existing steganography techniques are analyzed in [7]. Aura [8] proposes that gray scale images are the best cover images.

0-7803-9588-3/05/$20.00 ©2005 IEEE

He observes that uncompressed scans of images obtained with a digital camera with good resolution are the safest image for steganography. Fredrich et.al, [9] conclude that the cover images stored in the JPEG format are a very poor choice for steganography that works in spatial domain, since very small modifications of the image can be reliably deleted by flipping the LSB of one pixel. Eggers et.al [10] observed that raw uncompressed format provides large space for secured steganography, but exchange of this uncompressed image is considered equivalent to cryptography by the same authors.

Pfitzmann and Westfield [11] proposed a practical algorithm for embedding JPEG images that would provide high steganographic capacity without sacrificing security. The F5 algorithm [12] embeds message bits into randomly chosen DCT co-efficient and employs matrix embedding that minimizes the necessary changes to embed a message of certain length. Marvel et.al [13] developed a high capacity model in which uncompressed raw image formats were used to embed payload bits. A Gaussian signal which is generated by a special non-linear transform together with the message bits is added to the cover image. A Secure Steganographic algorithm is presented in [14].

3. MODEL

In this section, we define the parameter of performance, describe the Least Significant Bit Algorithm, Discrete Cosine Transformation to obtain stego-image, quantization, and runlength coding for compression with examples. Here, the cover image is a carrier of embedded image; hidden image is an image to be embedded in the cover image and transported. LSB algorithm is used to hide an image in a cover image. Stego-image is the combination of cover image and hidden image. DCT is used to convert stego-object in spatial domain into stego-image in frequency domain. Quantization and runlength coding is applied for the compression of stego-image for enhanced security. The reverse process is carried out at the receiver end, where the hidden image is retrieved from the encoded stego-image using the inverse transform techniques like Decompression, encoding of runlength, dequantization and inverse DCT(IDCT). All the images are assumed to be in .raw format.

A. Least Significant Bit(LSB) Embedding

Digital images are mainly of two types (i) 24 bit images and (ii) 8 bit images. In 24 bit images we can embed three bits of information in each pixel, one in each LSB position of the three eight bit values. Increasing or decreasing the value by changing the LSB does not change the appearance of the image; much so the resultant stego image looks almost same as the cover image. In 8 bit images, one bit of information can be hidden. The cover image is shown in Figure 1(a) and a hidden image is shown in Figure 1(b). A stego-image is obtained by applying LSB algorithm on both the cover and hidden images (Figure 1(c)). The hidden image is retrieved from the stego-image by applying the reverse process (Figure 1(d)).

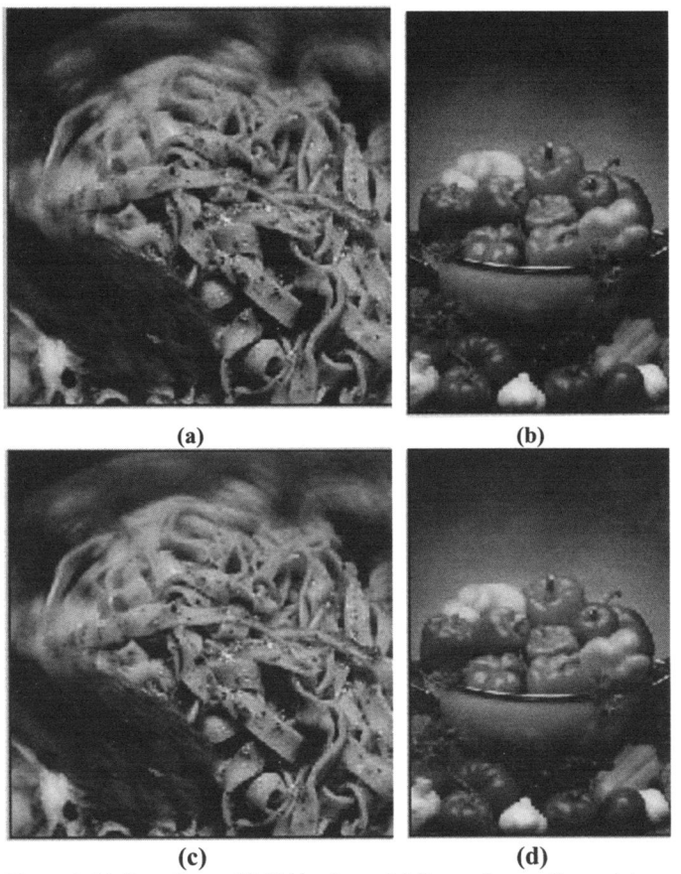

Figure 1: (a) Cover Image (b) Hidden Image (c) Stego – Image after applying LSB (d) Retrieved Image

B. Discrete Cosine Transform (DCT)

In the transform-based method there are two types (*i*) the large number of coefficients are modified slightly to accommodate data of the payload, (*ii*) replacing the smaller number of insignificant coefficients by the data of the payload. Here the data is embedded into the cover image by changing the coefficients of a transform of an image such as discrete cosine transform (DCT) coefficients. There are mainly three transformation techniques (i) Fast Fourier Transform (FFT) (ii) Discrete Cosine Transform (DCT) and (iii) Discrete Wavelet Transform (DWT). FFT introduce round off errors; so this technique is not suitable for hidden communication. The two dimensional DCT is applied on blocks of 8x8 pixels. This transforms 8x8 pixels blocks into 64 DCT coefficients, modifying one coefficient affects all the 64 image pixels.

DCT:

$$F(u,v) = \frac{\Delta(u)\Delta(v)}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16} . f(i,j)$$

IDCT:

$$\hat{f}(i,j) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} \Delta(u)\Delta(v) . \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16} . F(u,v)$$

$$\Delta(\varepsilon) = \begin{cases} 1/\sqrt{2}, & for\, \varepsilon = 0 \\ 1 & otherwise \end{cases}$$

C. Compression

Since images require large bandwidth, compression is useful to reduce bandwidth. Here, compression is achieved using quantization and run length coding of the transformed coefficients.

D. Error Computation

(*i*) *Bit error rate (BER)*: Here we compute the BER for two equal size images that is cover image and stego-image. BER is more accurate for error analysis when compared to MSE, because in BER we compute the actual number of bit positions which are replaced in the stego image.

(*ii*) *Mean square error (MSE)*: The MSE is computed by performing byte by byte comparisons of the two images, since a pixel is represented by 8 bits and hence 256 levels are available to represent the various gray levels. The MSE will result in a meaningful value only when each byte of an image is compared with the corresponding byte of another image. Let c and s be the cover image and stego-image respectively. Let $n*n$ be the total number of pixels. The computation of MSE can be performed as follows,

$$MSE = \frac{1}{n*n} \sum_{i,j=0}^{n-1} (c(i,j) - s(i,j))^2$$

E. Security

In the JPEG, BMP and GIF image formats, the header contains most of the image information. This leads to the problem of insecurity and therefore the payloads from such images can be easily identified. In our work, the hidden image(h) and the cover image(c) are considered to be raw images of different sizes. Therefore, the sender and the receiver should have a prior decision on the size of the images for image identification and retrieval. For example, consider a 400*600 RGB raw image with size 400*600*3. If the receiver is not aware of the dimensions of the image that has been sent from the source, then the default size of the image at the receiver end will be considered as 600*600*2. Therefore, when the receiver is unaware of the original dimension of the source image from the sender, leads to retrieval of a distorted image. Hence embedding raw images in cover image is more secure than other techniques.

(*i*) *Entropy*: Entropy is a measure of security for steganographic system which is computed as follows. Let e_1, e_2, ... e_m be m possible elements with probabilities $P(e_1)$, $P(e_2)$, ... $P(e_m)$; the entropy is given by,

$$H(e) = -\sum_{i=0}^{m-1} P(e_i) \log_2 P(e_i)$$

The above equation provides an estimate of the average minimum number of bits required to encode a string of bits based on the frequency of the symbol.

4. ALGORITHM

Problem definition: Given a cover image c and the image to be embedded (payload) h; the objective is,

(i) to embed the payload in the cover image by replacing LSB bits of cover image by the image of the payload. The combined image is called stego-object(s).

(ii) to transform the stego-object from spatial domain to frequency domain using DCT.

(iii) to compress the frequency domain stego-object using quantization and runlength coding to generate a secure stego-object.

Assumptions:

(i) Cover and payload objects are raw images of arbitrary size.

(ii) The LSBs of the cover image is utilized to embed the payload to minimize distortion in the cover image.

(iii) Stego-object is transmitted over the noiseless channel.

Let c be the cover image, h be the hidden image, and s be the stego image. Let P be the number of bytes in cover image which are used to store one byte of the hidden image. Let *ifile* be the input file and *cfile* be the output file. The algorithm Secured Steganography using LSB, DCT and Compression techniques (SSLDC) is given below.

Algorithm SSLDC:

Input: Cover Image (c) and a Hidden Image (h)

Output: Encoded Stego Image(s)

Repeat

Step 1: *Read* the first byte of c and h into temporary locations *cb* and *hb* respectively.

Step 2: Run *LSB*()

Step 3: Compute *DCT*()

Step 4: Perform *Quantization*()

Step 5: Apply R*unlength Coding*() on each block.

Step 6: Copy the output as a Stego Image.

Until (EOF)

(i) *LSB*( )

Read n;

Repeat

if (P = 1) then

replace last four bits in *cb* by first four bits of *hb*

else if (P = 2) then

read x; /* 3 <x <8 */

replace last x bits in *cb* by first x bits in *hb*

copy *cb* into *ifile*

read next byte of c into *cb*

replace last 8–x bits of *cb* by last 8 – x bits of *hb*

else if (P = 4) then

for (i =1; i <= 4; i++)

Replace last two bits in *cb* by the i^{th} two bits of *hb*

else if (P = 8) then

for (i =1; i <= 8; i++)

Replace the last bit of *cb* by the i^{th} bit of *hb*

Write *cb* into *ifile*.

Read the next byte of c into *cb*

Read the next byte of c and h into *cb* and *hb*

Until (*hb* != EOF)

(ii) DCT( )

- Read a block of $n*n$ pixels
- Compute $c = \Delta(u)\,\Delta(v) / 4$
- *for*($i = 0; i \leq 7; i$++)
- *for*($j = 0; j \leq 7; j$++)
- Compute
  $F(u, v)=c*\cos((2i+1)u\pi/16)*\cos((2j+1)v\pi/16)*f(i, j)$

(iii) *Quantization*( )

- Read $n*n$ matrix
- Read a down scale factor x
- Scale down the matrix by a factor x

(iv) *Runlength Coding*( )

- Read the input string y
- Count the similar consecutive characters.
- Replace the substring of similar consecutive characters by their count suffixed by the character.

The algorithm SSLDC works as follows. By applying LSB algorithm on c and h, we get a stego image s. We then apply DCT followed by quantization and runlength coding on the stego image to obtain encoded stego image. The reverse procedure is adopted at the receiver end. In this technique we use four different types of LSB transformations. In the first case, one byte of cover image is used to store one byte of hidden image. In the second case, we use two bytes of cover image to store one byte of hidden image. In case three, four bytes of cover image are used to store one byte of hidden image. In case four, eight bytes of cover image are used to store one byte of hidden image.

Case 1: In L1 transformation, one byte of cover image is used to store one byte of hidden image. Last four bits(LSBs) of cover image are replaced by the first four bits(MSBs) of the hidden image. This is a lossy transformation.

Case 2: L2, L3, L4 and L5 are the stego images obtained after the subsequent transformations using two bytes of the cover image. In L2 transformation, seven MSB bits of the hidden image are embedded in seven LSB bits of first byte of the cover image and the last bit of the hidden image is embedded into the last bit of the second byte of the cover image. In L3 transformation, six MSB bits of the hidden image are embedded in six LSB bits of first byte of the cover image and last two bits of the hidden image are embedded into the last two bits of second byte of the cover image. In L4 transformation, five MSB bits of the hidden image are embedded in five LSB bits of first byte of the cover image and last three bits of hidden image are embedded in last three bits(LSBs) of second byte of the cover image. In L5 transformation, four MSB bits of the hidden image are embedded in four LSB bits of first byte of the cover image and last four bits of the hidden image are embedded into last four bits(LSBs) of second byte of the cover image.

Case 3: In the L6 transformation, four bytes of the cover image is used to store one byte of the hidden image. The last two bits(LSBs) of first byte of the cover image are replaced by first two bits(MSBs) of the hidden image. Similarly, the subsequent bits of the hidden image are embedded in last two bit locations of second, third and fourth byte of the cover image respectively.

Case 4: In the L7 transformation, eight bytes of the cover image is used to store one byte of the hidden image. The last bit of eight bytes of the cover image is replaced by consequent bits of the hidden image.

5. Performance Analysis

We consider the cover image as shown in Figure 1(a) for all the experiments and analysis performed in this paper. A color image and its gray scale image of an *Eagle* are considered as the payload(image to be hidden). The seven different transformations of the LSB are applied to obtain the corresponding stego-images L1 - L7 as shown in Figure 2. The images L1 – L6 are obtained by embedding the color image of an eagle (hidden image) in the cover image and L7 is obtained by embedding the gray scale image of an Eagle in the cover image(as it requires a large size cover image).

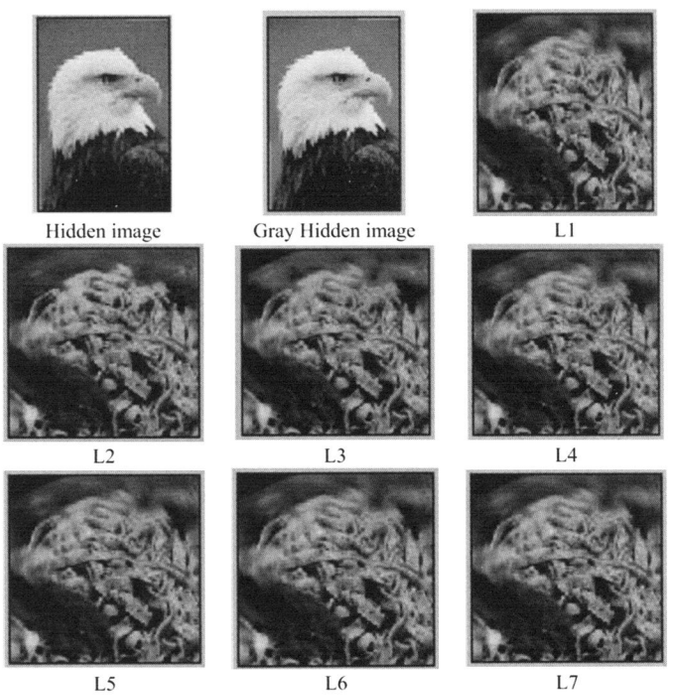

Figure 2: The hidden image (eagle) and stego-images (L1 – L7) after applying LSB algorithms

In Figure 3, we apply the reverse process on DCT applied L1-L7 to retrieve the payload from the stego images to obtain the payload images R_1- R_7 respectively. R_3 is the best retrieved image by which we can conclude that L3 is the best transformation.

With reference to the earlier works on transform-based techniques for steganography, the experiments are conducted by applying only DCT/IDCT on the cover image and the corresponding stego-image is shown in Figure 4(a). The

retrieved image from Figure 4(a) is shown in Figure 4(b). The stego-image and the corresponding best retrieved image of the proposed algorithm SSLDC are shown in Figure 4(c) and 4(d). From Figure 4, we can conclude that the proposed algorithm SSLDC gives better results with respect to stego and retrieved images when compared with the transform based techniques.

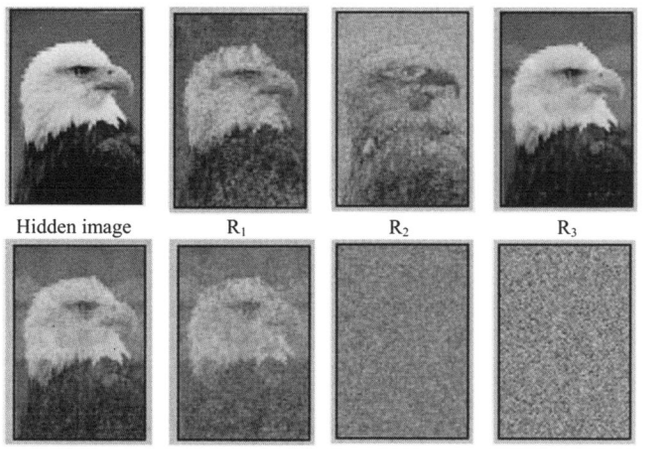

Figure 3: Retrieved images R_1- R_7 from respective stego-images.

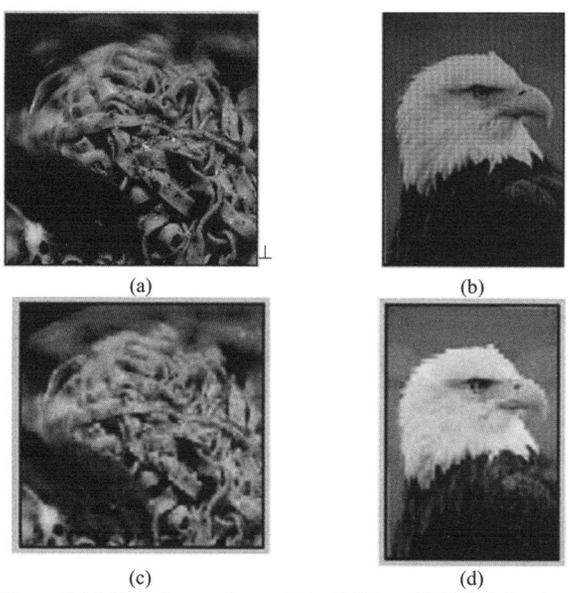

Figure 4: (a) Stego image after applying DCT and IDCT (b) Retrieved image from Fig. 5(a); (c) Stego Image after applying SSLDC (d) Retrieved image from Fig. 5(c).

Error Analysis: The experimental results obtained are subjected to various statistical techniques to evaluate the performance parameters of the steganographic images viz., (i) Bit Error Rate (ii) Mean Square Error (iii) Entropy.

A. Bit Error Rate

The BER is computed for various values of Depth of Hiding shown in Figure 5. It is observed that the BER is lower as when six bits of LSB in the first byte and two bits of LSB in the second byte of the cover image are utilized for embedding the payload(for color images). The BER is lowest when eight bytes of cover image are used to store one byte of the hidden image, where LSB of the eight bytes of cover image is replaced by a byte of consequent bits of the hidden image.

B. Mean Square Error

The MSE is computed for various values of Depth of Hiding as shown in Figure 6. It is observed that the MSE is lower when six bits of LSB in the first byte and two bits of LSB in the second byte of cover image are utilized for embedding the payload(for color images). The MSE and BER values computed are lower than those obtained from the transform based techniques. It is evident from Table 1 and Table 2, that these values are computed after applying IDCT.

C. Entropy

The values of entropy are close to zero when 1 bit of the cover image is utilized to embed the hidden gray level image. It is also observed that in Figure 7 the entropy is nearly zero when six bits of LSB in the first byte and two bits of LSB in the second byte of the cover image are utilized for embedding the payload. In contrast when seven bits of LSBs of the first byte and 1 bit of LSB of the second byte of the cover image is utilized to embed the payload. The entropy rises very steep; this is on account of fact that most of the bits of the cover image are replaced by the bits of the hidden image.

Table 3 presents the comparison of BER, MSE, MAE and Relative Entropy for various depths of hiding. From all these statistics, it is clearly evident that the best way to embed the color image (payload) into the cover image is to utilize six bits of LSB in the first byte and two bits of LSB in the second byte of the cover image.

Table 1: Comparison of Mean Square Error(MSE) of SSLDC and Transformed-based Technique(TBT)

(Image: *Eagle*)	SSLDC	TBT
Cover vs. stego-image	0.020218	0.027348
Hidden vs. retrieved	0.015455	0.024892

Table 2: Comparison of Bit Error Rate(BER) of SSLDC and Transformed-based Technique(TBT)

(Image: *Eagle*)	SSLDC	TBT
Cover vs. stego-image	16.244791	18.190277
Hidden vs. retrieved	28.653473	29.069792

Table 3: Statistical Comparison of BER, MSE, MAE and Relative Entropy

Depth of Hiding	BER	MSE	MAE	Rel. Entropy
1	2.780208	0.000980	0.222430	0.00036
2	8.330555	0.036652	0.840565	0.007497
3	8.315001	0.027001	0.978210	0.008876
4	8.326388	0.020798	1.767254	0.010254
5	8.347917	0.011505	2.243344	0.010371
6	8.265972	0.007807	3.460004	0.001964
7	8.359446	0.005473	5.855214	0.041230

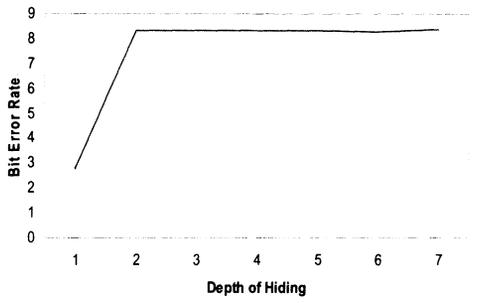

Figure 5: Depth of Hiding versus BER

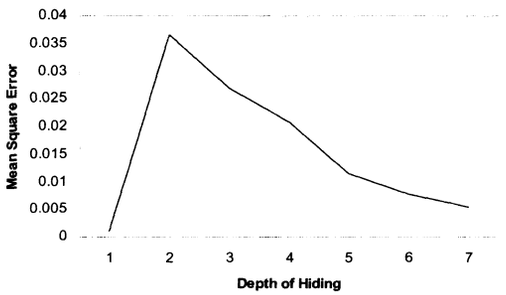

Figure 6: Depth of Hiding versus MSE

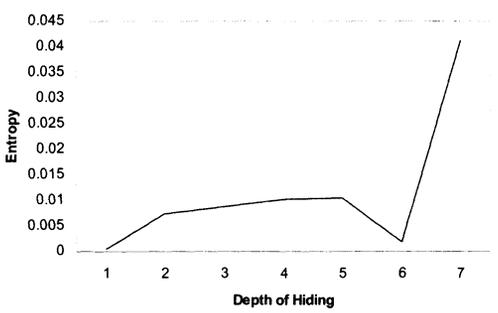

Figure 7: Depth of Hiding versus Entropy

6. CONCLUSIONS

In this paper we have used the combination of LSB algorithms, DCT transformation, and compression using quantization and runlength coding on raw images to obtain secure stego-image. The LSB technique has been used to accommodate maximum payload. The entire payload is embedded into the cover image to obtain the stego-object. The stego-object in the spatial domain is transformed into frequency domain by applying DCT. The stego-object is further compressed using quantization and runlength coding to derive a secure stego-object. An exactly reverse procedure is followed to retrieve the payload at the receiver. The integrated approach of combining LSB, DCT and compression techniques enable secure transfer of payload with low BER and MSE compared to earlier techniques.

References

[1] Niels Provos, "*Defending against statistical steganalysis*", In Proceedings of the 10th USENIX Security Symposium, August 2001, pp. 323-335.

[2] R.Anderson and F. Petitcolas, "*On the limits of steganography*" IEEE Journal of Selected Areas in Communications, Vol. 16, No. 4, May 1998, pp. 474-481,.

[3] Christian-Cachin, "*An information-theoretic model for steganography*," Lecture Notes in Computer Science, 1998, 1525:306-318.

[4] L.M. Marvel, C.G. Boncelet, and C.T. Retter, "*Reliable Blind Information Hiding for Images*" In Information Hiding: Second International Workshop, LNCS, Vol. 1525. Springer-Verlag, New York, 1998, pp. 48-61.

[5] Imre Csiszar. "The method of types". IEEE TIT:IEEE Transactions on Information Theory, 1998.

[6] Luis von Ahn and Nicholas J. Hopper, "*Public-key steganography.*" In Lecture Notes in Computer Science, volume 3027/2004 of Advance in Cryptology-EUROCRYPT 2004, Springer-Verlag Heidelberg, 2004, pp. 323-341.

[7] Niels Provos, Peter Honeyman, "*Hide and Seek: An Introduction to Steganography*," IEEE computer society, 2003, pp. 32-44.

[8] Aura, T, "*Practical Invisibility in Digital Communication*," In Information Hiding: First International Workshop. Lecture Notes in Computer Science, Vol. 1174. Springer-Verlag, Berlin Heidelberg New York 1996, pp. 265-278.

[9] Fridrich. J., Goljan. M., and Du. R: Steganalysis Based on JPEG Compatibility. Proc. SPIE Multimedia Systems and Applications IV, Vol. 4518, Colorado, 2001, pp. 275-280.

[10] Eggers J. J, Baumal R., and Girod. B, "*A Communications Approach to Image Steganography*," Proceedings of SPIE: Electronic Imaging 2002, Security and Watermarking of Multimedia Contents, Vol. 4675, California 2002.

[11] Pfitzmann and Westfeld. A, "*High Capacity Despite Better Steganalysis,*" (F5-A Steganography and Watermarking-Attacks and Countermeasures. Kluwer Academic Publishers, Boston Dodrecht London, 2000.

[12] Jessica Fridrich, Miroslav Goljan, and Dorin Hogea, "*Steganalysis of JPEG Images: Breaking the F5 Algorithm*," Fifth Information Hiding Workshop, Noordwijkerhout, the Netherlands, 2002, pp. 310-323.

[13] L.M. Marvel, C.G. Boncelet, and C.T. Retter, "*Reliable Blind Information Hiding for Images*" In Information Hiding: Second International Workshop, LNCS, Vol. 1525. Springer-Verlag, New York, 1998, pp. 48-61.

[14] K B Raja, Venugopal K R and L M Patnaik, "*A Secure Stegonographic Algorithm using LSB, DCT and Image Compression on Raw Images*", Technical Report, Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, December 2004.