A proposed Technique for Information Hiding Based on DCT

[1]Dr. Fadhil Salman Abed, [2]Nada Abdul Aziz Mustafa
1, Iraq, Diyala, Technical Institute of Kalar,E-Mail: Fad_Sal_Abed@Yahoo.com
2, Iraq,Baghdad, University of Sulaimani, College of Basic Education
E-Mail: Fad1234567826@Yahoo.com
doi:10.4156/ijact.vol2. issue5.16

Abstract

The aim of this work is to design an algorithm which combines between steganography and cryptography that can hide a text in an image in a way that prevents, as much as possible, any suspicion of the hidden text

The proposed system depends upon preparing the image data for the next step (DCT Quantization) through steganographic process and using two levels of security: the RSA algorithm and the digital signature, then storing the image in a JPEG format. In this case, the secret message will be looked as plaintext with digital signature while the cover is a coloured image. Then, the results of the algorithm are submitted to many criteria in order to be evaluated that prove the sufficiency of the algorithm and its activity. Thus, the proposed algorithm for this research can be divided into two main parts: hiding the text of the sender, and extracting it by the receiver. Moreover, part can be divided into many procedures done by the program Delphi 5.

Keywords : *DCT Transformation, Public Key Encryptions, Image Processing , Hiding, Cryptography, Steganography*

1. Introduction

With the development of internet technologies, digital media can be transmitted conveniently over the internet. However, messages transmission over the internet still have to face all kinds of security problems.

Therefor, how to protect secret messages during transmission becomes an essential issue for the internet.The schemes include DES, AES and RSA**[1].** These methods scramble the secret message so that it cannot be understood However, it makes the messages suspicious enough to attract eavesdroppers attention Hence, this paper presents a novel two techniques are available to those wishing to transmit secrets using unprotected communications media. One is cryptography, where the secret is scrambled and can be reconstituted only by the holder of a key. The second method is steganography, where the secret is encoded in another message in a manner such that, to the casual observer, it is unseen. Steganography is often combined with cryptography to provide an additional layer of security.

The JPEG format is currently the most common format for storing image data. It is also supported by virtually all software applications that allow viewing and working with digital images. Recently, several steganographic techniques for data hiding in JPEG have been developed**[2].**

The proposed algorithm dependes on preparing the image data for next steps (DCT, quantization) through embedding processes and using two levels of security RSA algorithm, and digital signature, later the stego image is JPEG format. The secret message in this approach is plaintext, digital signature, while the cover is a coloured image. The algorithm results are submitted to many evaluating criteria to evaluate them, which prove their efficiency and activity. The proposed algorithm of this paper can be divided into two main parts, hiding and extracting, each of them can be further divided into a number of procedures**[3].**

2. DCT (Discrete Cosine Transformation) Technique

A more complex way of hiding a secret message inside an image comes with the use and modifications of discrete cosine transformations. Discrete cosine transformations(DCT) are used by

the JPEG compression algorithm to transform successive 8 x 8 pixel blocks of the image, into 64 DCT coefficients.

In this work one coefficient is used from each block (8 x 8) to hold the bit called DC coefficient in position (0,0);by preparing the value of pixels in the block until the DC coefficient becomes odd or even dependent on the bit which is wanted to be hidden. Figure (1) shows the general description of embedding messages and digital signatures in an image. This work includes the following steps :

1. Load a colour image (bitmap format 24 bits), and part the colours into the red, green, and blue.
2. Convert the image formula from RGB to YCbCr **[4]**, **[5]**

Y = (77/256) R + (150/256) G + (29/256) B
Cb = -(44/256) R – (87/256) G + (131/256) B + 128
Cr = (131/256) R – (110/256) G – (21/256) B +128

3. Separate the image components Y, Cb, Cr into blocks, each one consists of 64 pixels (8x8)
4. Transform each block (8x8) pixels to spatial frequency domain via the forward DCT

$$G_{ij} = \frac{1}{4} c_i c_j \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} P_{xy} \cos\left(\frac{(2x+1)i\pi}{16}\right) \cos\left(\frac{(2x+1)j\pi}{16}\right)$$

(P_{xy}) pixel in the block coefficients , (G_{ij}) DCT coefficient**[6]**.

5. Combine the stream of bits of the digital signature and the secret message.
6. Embed the stream of bits in the cover image. In each block embed, one bit in the DC element. This step will be described in details later on.
7. Quantize these blocks with quantization coefficients. The DCT coefficients are divided by their corresponding quantization coefficients (quantization table) and rounded to the nearest integer.
8. Quantize DCT coefficients by multiplying the same quantization tables that are used in a compression stage to obtain DCT coefficients.
9. Inversing DCT is applied in this step in each block**[6], [2]**

$$p_{xy} = \frac{1}{4} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} c_i c_j G_{ij} \cos\left(\frac{(2x+1)i\pi}{16}\right) \cos\left(\frac{(2y+1)j\pi}{16}\right), \text{ where } c_i c_j = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } i, j = 0 \\ 1 & \text{Otherwise} \end{cases}$$

10. Reconstruct the image by combining all the blocks.
11. Transform the image formula from YCbCr to RGB **[4] [5]**

R = Y + 1.371 (Cr – 128), G = Y – 0.698 (Cr – 128) – 0.336 (Cb –128)
B = Y + 1.732 (Cb – 128)

12. Save the stego image in JPEG format.

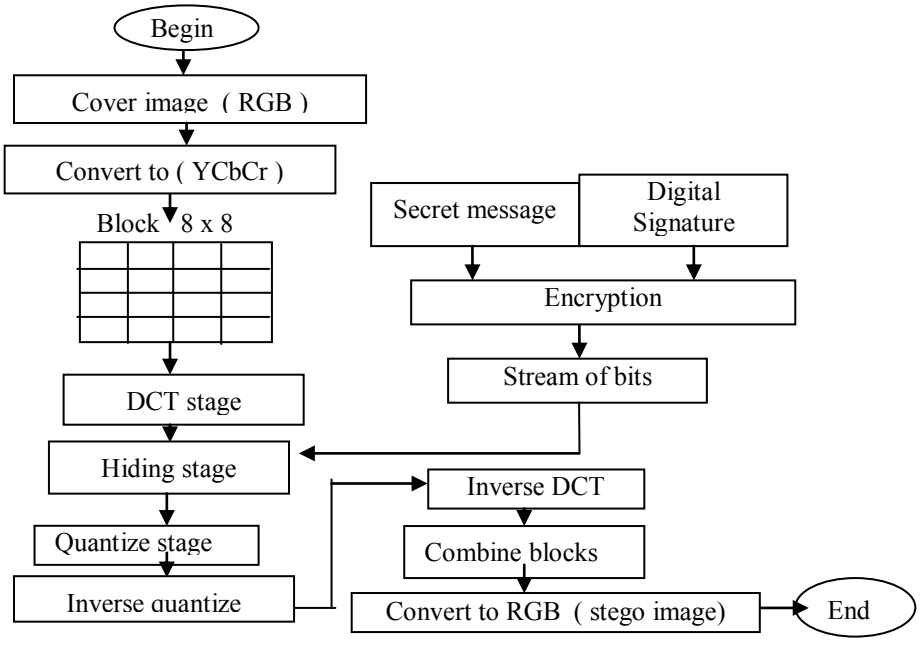


Figure 1. Encoding and hiding digital signatures and texts

2.1. Hiding Bits

In this step, the text bits are embedded in the cover-image. After inputting the text in the system, convert each letter in the text to a decimal number and encrypt it and convert each one to a binary form. Figure (2) shows the block diagram of hiding one bit in the block. Some steps are implemented to embed the text bits. These steps are:

1. From each block (8x8) one DCT coefficient is chosen to hold the bit. This coefficient is in position (0,0). Compute the quantize DC coefficient (**dc**) .

$$dc = round(block(0,0)/16)$$

2.If (**dc**) is an odd number and the bit ('1'), or (**dc**) is an even number and bit ('0'), no change happens in the original block pixels. Bring a new block and a new bit to continue or work and hide them.

3 .If (**dc**) is an odd number and the bit ('0'), or (**dc**) is an even number and bit ('1'), there must be a change in the original block coefficients until (**dc**) value satisfies the relationship in point (2). How can be satisfied in this work.

❖ Compute the new Quantize DC coefficient without rounding (**dc1**).

$$dc_1 = (block(0,0)/16$$

If (**dc > dc1**) then find the different (**df**) between them. To determine the number of pixels (**np**) that must be changed by **subtracting one** to the original value through comparing (**np**) with table (1). Table (1) shows the number of pixels that must be changed, subtracted or added depending on the number of difference.

Df=dc-dcl

Np=0.5-df

If (dc1 > dc) then find the different (***df***) between them. To determine the number of pixels (***np***) that must be changed by ***adding one*** to the original value through comparing (***np***) with a table which contains how many numbers that must be subtracted or added corresponding to the difference between the two results.

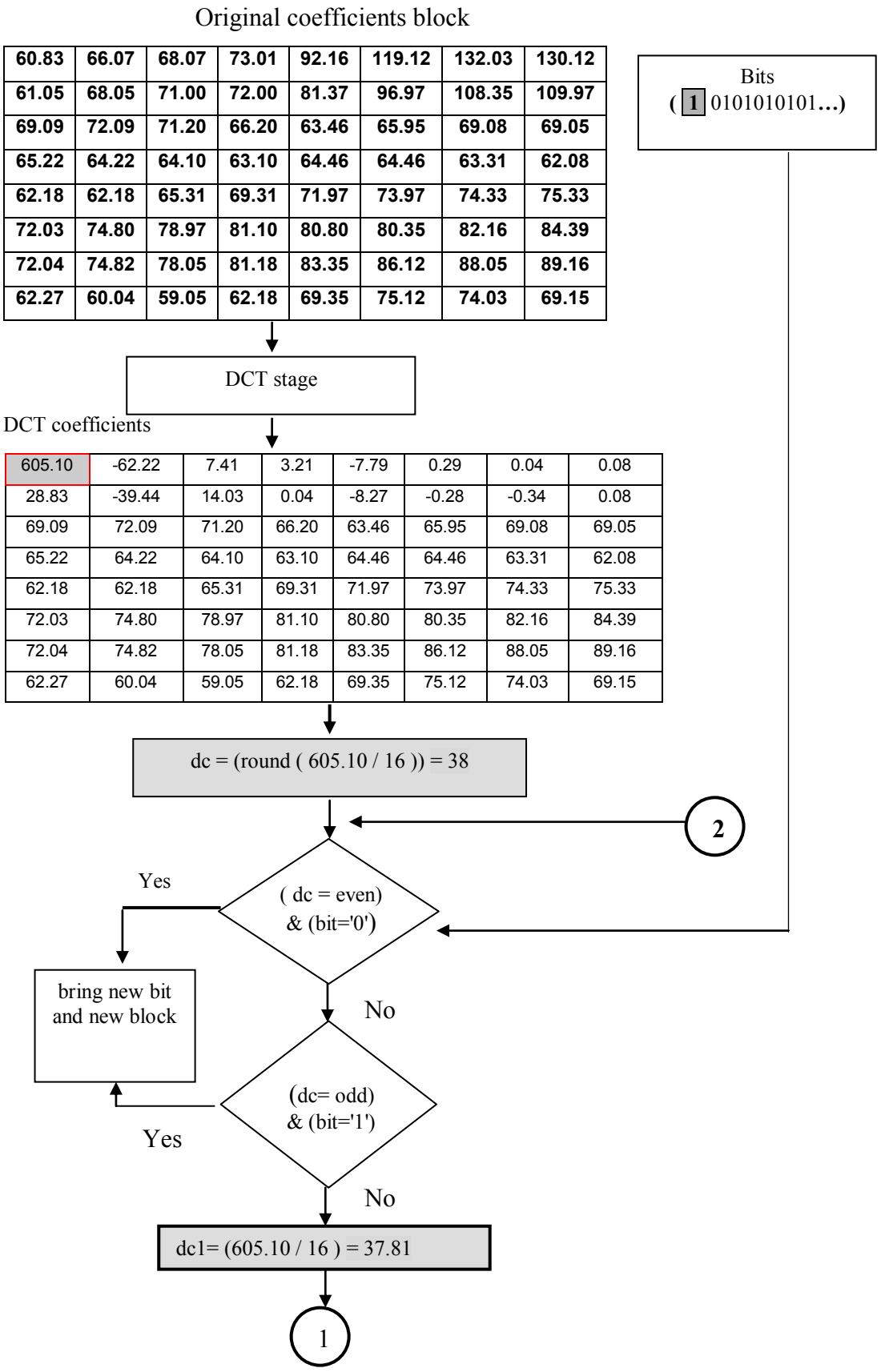
df=dc1-dc , np=0.5-df

Table 1. Shows the amount added or subtracted of each block

Difference	Amount
0.000-0.063	8
0.064-0.125	16
0.126-0.188	24
0.189-0.250	32
0.251-0.313	40
0.314-0.375	48
0.376-0.437	56
0.438-0.500	64

❖ After subtracting or adding from the original coefficients block work, the research employs the DCT in the same block again.

4. Steps (1-3) continue with each block until hiding all the bits.


Original coefficients block
60.83 66.07 68.07 73.01 92.16 119.12 132.03 130.12
61.05 68.05 71.00 72.00 81.37 96.97 108.35 109.97
69.09 72.09 71.20 66.20 63.46 65.95 69.08 69.05
65.22 64.22 64.10 63.10 64.46 64.46 63.31 62.08
62.18 62.18 65.31 69.31 71.97 73.97 74.33 75.33
72.03 74.80 78.97 81.10 80.80 80.35 82.16 84.39
72.04 74.82 78.05 81.18 83.35 86.12 88.05 89.16
62.27 60.04 59.05 62.18 69.35 75.12 74.03 69.15
Bits
(1 0101010101…)
DCT stage
DCT coefficients
605.10 -62.22 7.41 3.21 -7.79 0.29 0.04 0.08
28.83 -39.44 14.03 0.04 -8.27 -0.28 -0.34 0.08
69.09 72.09 71.20 66.20 63.46 65.95 69.08 69.05
65.22 64.22 64.10 63.10 64.46 64.46 63.31 62.08
62.18 62.18 65.31 69.31 71.97 73.97 74.33 75.33
72.03 74.80 78.97 81.10 80.80 80.35 82.16 84.39
72.04 74.82 78.05 81.18 83.35 86.12 88.05 89.16
62.27 60.04 59.05 62.18 69.35 75.12 74.03 69.15
dc = (round (605.10 / 16)) = 38
2
Yes
(dc = even)
& (bit='0')
bring new bit
and new block
No
(dc= odd)
& (bit='1')
Yes
No
dc1= (605.10 / 16) = 37.81
1

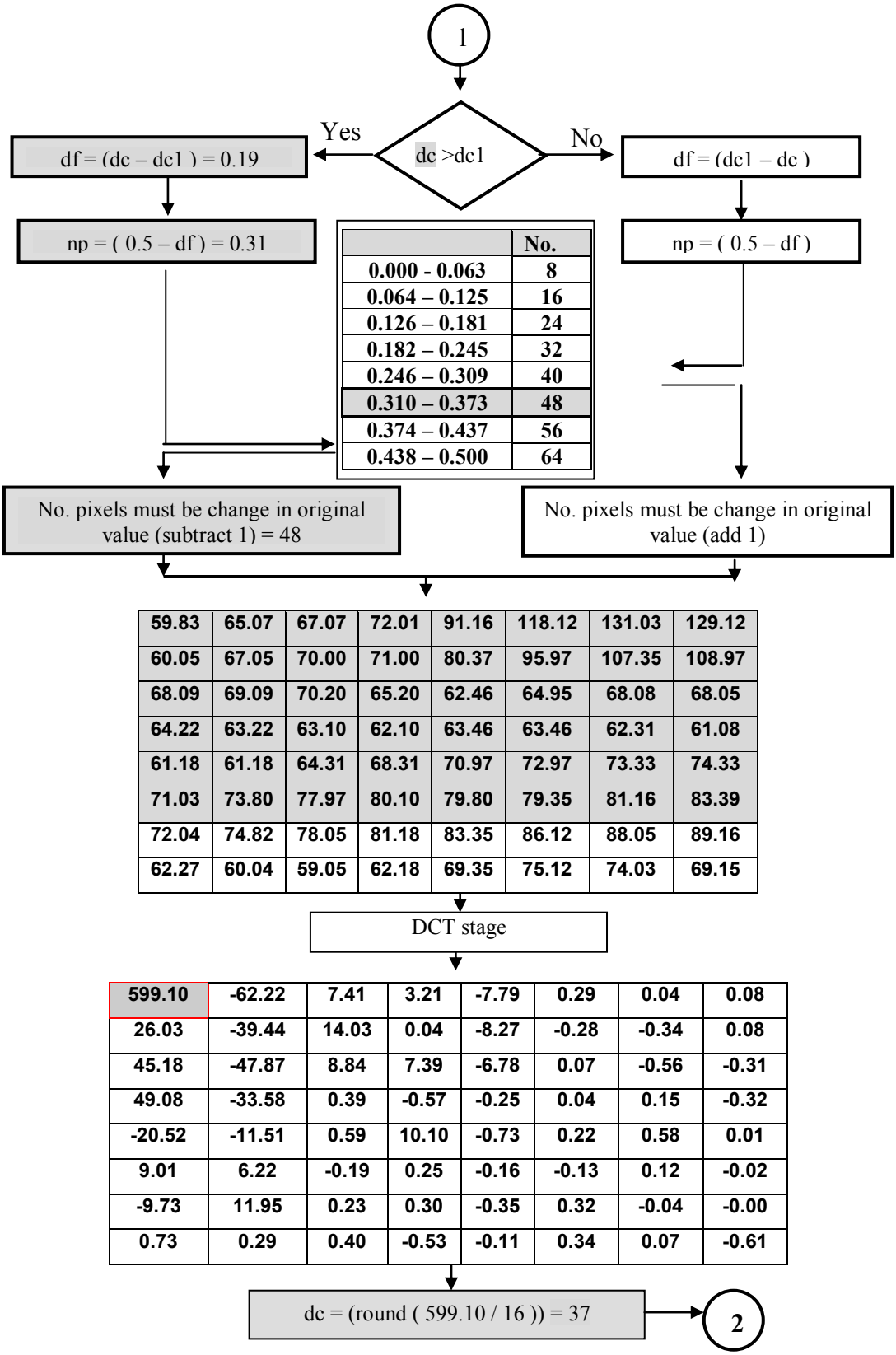


Figure 2. Shows block outline to hiding one bit in the block

2.2. Information about the Extracting Stage

This stage includes two parts, first extract bits from a stego image and convert each group of bits (12 bits) to a decimal number, second decrypt the decimal number to find the digital signature and the message. Figure (3) shows the general description of extracting messages and digital signatures from images**[7]**. The extracting message and digital signature stage include the following steps :

1. Load the stego image (bitmap 24 bits). This image contains the digital signature and the secret message.
2. Convert the image formula from RGB to YC_bC_r.
3. Separate image components into blocks, each one consists of (8 x 8) pixels.
4. Transform each block (8 x 8) pixels to spatial frequency domain via the forward DCT. This step is executed on the Y components only.
5. The fifth step includes:

- Extract the digital signature bits and convert each group (12 bits) to decimal numbers, then decrypt it by using a public key of the sender and a private key of the receiver presented in algorithm(1).
-

Algorithm 1. Convert each number to 12 bits

1- I = 1→ length of a secret message (nmtxt)
2- S = " ; B = " ; C = $nmtxt_i$
3- If (C mod 2) = 0) then S = S + '0'
4- If (C mod 2) = 1) then S = S + '1'
5- C = C div 2
6- If (C >0) then go to 3
7- J =1 → (12 – (length of (S))) S = S + '0'
8- J =12 → 1 B = B + S_j
9- Cph_i = B
10- Go to 1
11- End

- Convert each number after decryption to a corresponding letter until extracting all digital signatures.
- Extract the cipher text bits and convert each group (12 bits) to a decimal number then, decrypt it by using a private key of the RSA**[8]**.
- Convert each number after decryption to corresponding letter until extracting all plain text. This step will be described in detail.

6. Print the secret message and the digital signature.

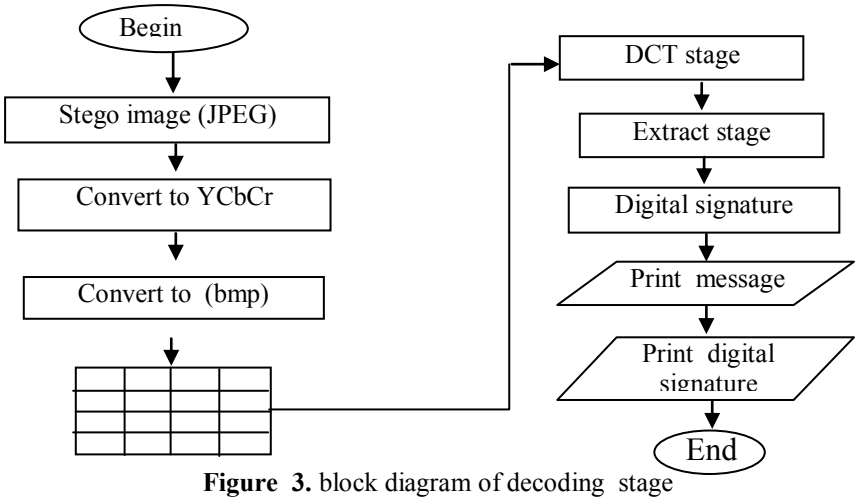


Figure 3. block diagram of decoding stage

2.3. Extracting Bits

1. In this stage, the bits are extracted from the stego-image. After converting the stego-image from JPEG to BMP, convert the image form RGB to YCbCr, separate the image components into blocks, each one consists of (8 x 8) pixels and transform each block (8 x 8) pixels to spatial frequency domain via the forward DCT. Figure (4) shows the block diagram for extracting bits from the stego-image. Some steps are implemented to extract the bits :
2. From each block the same DCT coefficients are used in embedding stage choice to extract the bit. This coefficient is in position (0, 0) in each block.
3. Divide the value in position (0,0) by 16 and round the result called (**dc**) and inspect the result.

   $$dc = round(block(0,0)/16)$$

   a. If the **dc** value equals odd numbers, this means one is hidden in it.
   b. If the **dc** value equals even numbers, this means zero is hidden in it.
4. Convert each 12 bits to decimal numbers.
5. Decrypt the decimal number by using the receiver key of the digital signature. Convert the result to a corresponding letter until finding (.) that refers to end the digital signature.
6. After finding all the characters of the digital signature, decrypt the decimal number by a private key (RSA algorithm) and convert the number after the decryption stage to a corresponding letter.
7. Steps one, two , three, four and five continue until finding the (#) character that refers to the end of the text.

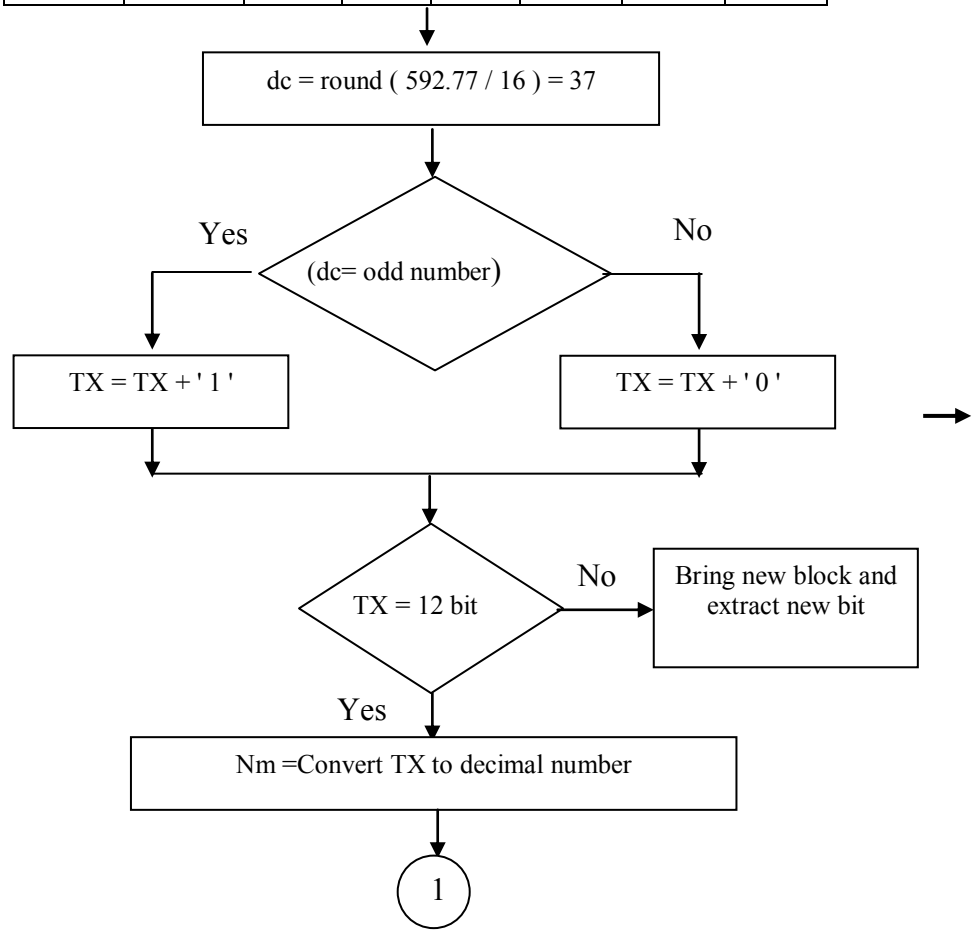
Original coefficients block

56.52	58.41	63.76	74.54	91.31	111.78	131.56	142.03
68.27	67.86	67.75	71.41	78.76	88.54	98.42	104.90
68.48	67.25	65.73	64.50	63.86	64.63	66.41	67.41
58.82	59.12	60.59	61.37	61.73	61.08	60.27	59.86
60.93	63.12	66.59	70.95	73.73	74.50	73.97	73.86
73.37	74.95	77.84	81.50	83.86	84.63	85.41	85.29
71.27	71.86	73.63	76.10	78.76	81.54	83.42	84.90
56.52	57.41	58.88	62.54	66.90	71.78	76.56	79.44

DCT stage

DCT coefficients

592.77	-66.03	9.64	-0.10	0.67	-0.00	-0.16	-0.10
23.84	-35.93	14.22	-0.26	-0.24	-0.05	0.07	0.06
41.91	-51.84	15.85	-0.28	-0.21	-0.11	-0.10	-0.14
56.29	-34.23	0.14	0.21	0.08	-0.08	0.00	-0.02
-17.88	-22.07	-0.09	0.05	-0.01	-0.12	0.12	-0.01
0.21	0.06	-0.01	-0.17	-0.23	0.26	0.42	0.18
-0.29	0.19	-0.22	-0.32	-0.08	0.32	-0.24	0.10
0.00	-0.08	-0.17	-0.00	-0.08	-0.01	0.11	-0.18



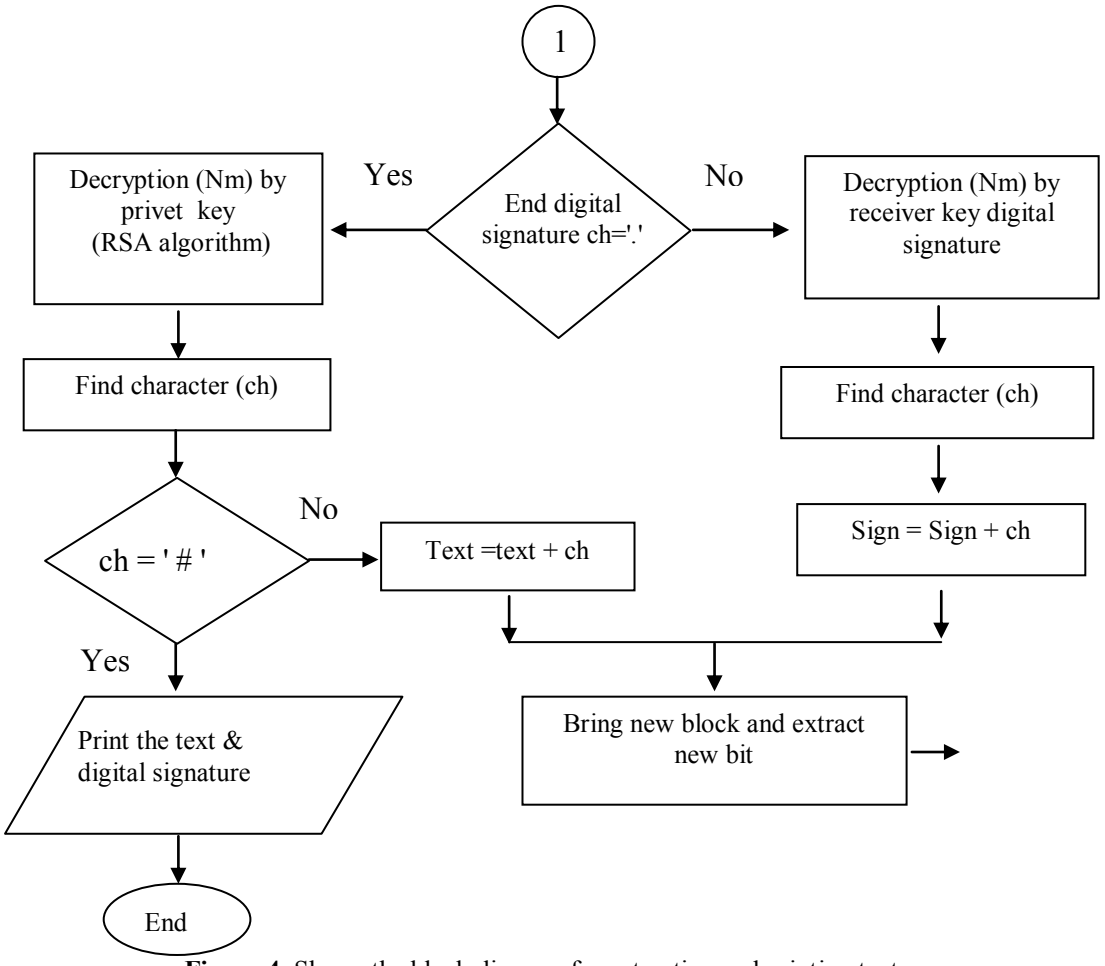


Figure 4. Shows the block diagram for extracting and printing text

Where Text : string contains the plain tex, Sign : string contains the digital signature.
dc : real value refers to the hiding bit, TX : string [12] bits (obtain the extract bit from each block).
Nm : integer number, Ch : character.

2.4. System Implementation

The goal of this system is to embed the text and the digital signature in a cover-image (BMP format) to produce the stego-image (JPEG format).

System implementation accepts six inputs in the embedding stage:

1. Input the file (BMP format), cover-image.
2. Input the secret message.
3. Input the public key to encrypt the text.
4. Input the digital signature.
5. Input the public key to encrypt the digital signature.
6. Output file (JPEG format), stego-image.

System implementation accepts three inputs in the extracting stage:

1. Input the file (BMP format),which contains the secret message.
2. Input the privet key to decrypt the text.
3. Input the privet key to decrypt the digital signature.

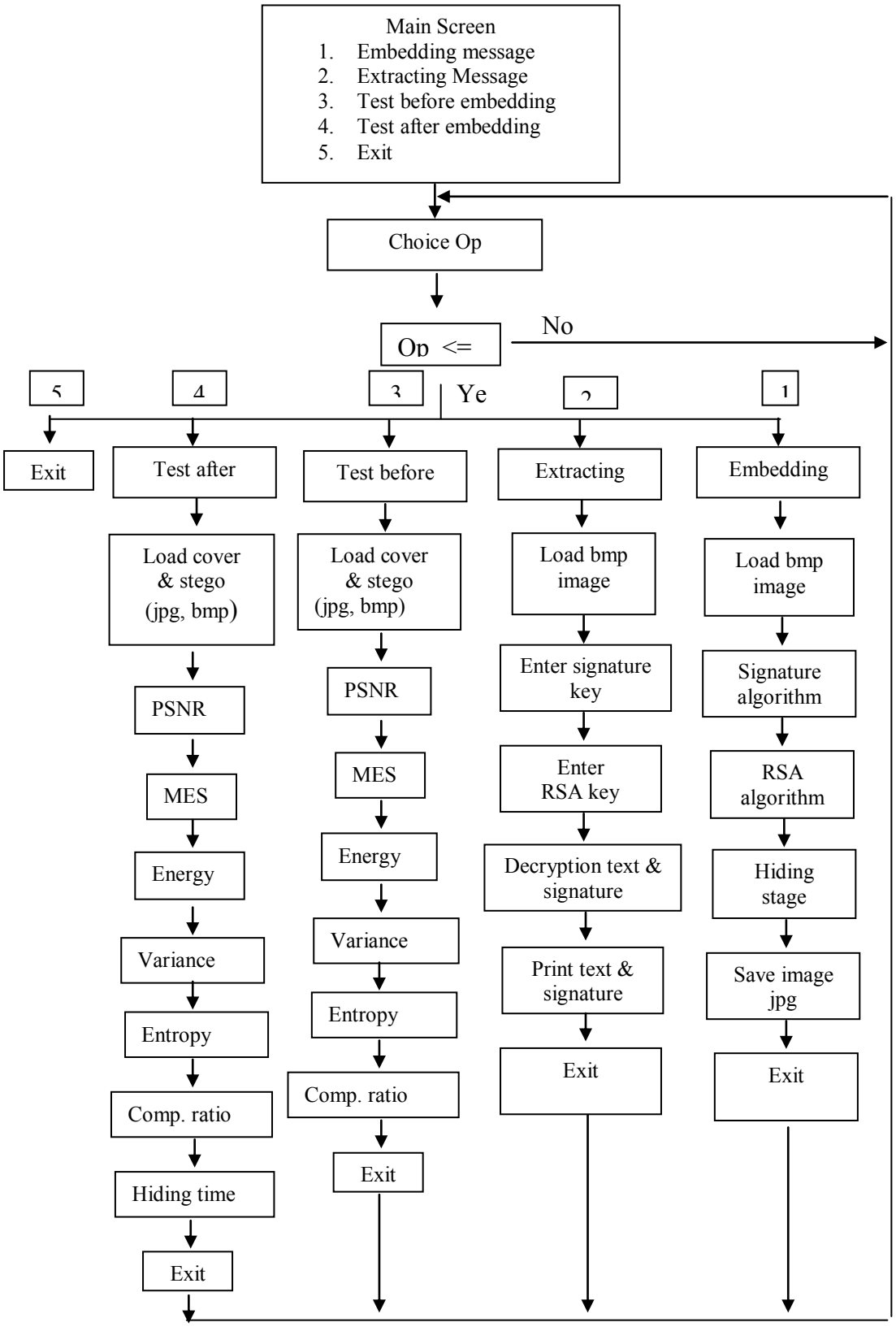


Figure 5. The block diagram of the system

2.4.1. Test Before Embedding Message

The third step in this proposed system is testing before embedding a message. This will reveal the results between the cover-image and stego-image without embedding the message and the digital signature by using PSNR, MES, Entropy , Variance, Energy and compression ratio. The test is executed for each colour and the compression ratio is computed for the stego-image, figure (6a), illustrated the result.

2.4.2 Test After Embedding the Message

The fourth step in the proposed system is doing the test after embedding the message. This step will reveal the results between the cover-image and stego-image after embedding the message and the digital signature by using PSNR, MES, Entropy , Variance, Energy, compression ratio and embedding time. The test is executed for each colour and the compression ratio is computed for the stego-image, **figure(6b) illustrated the result.**

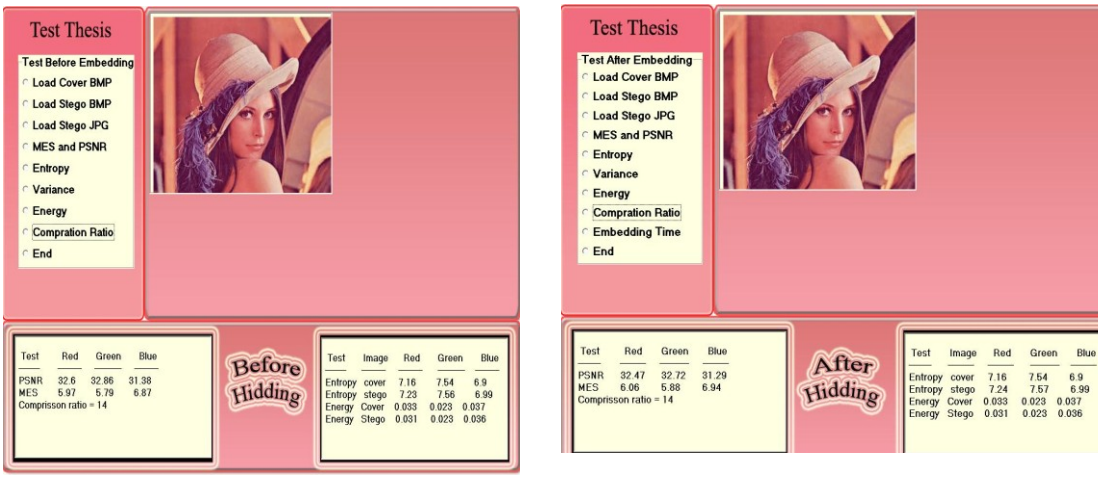


figure 6 (a) **figure 6 (b)**

Figure 6. Before and after embedding the message

2.5. Experimental Result

In these experimental, the secret message, the digital signature and encrypting the message are used via the RSA algorithm, a second method of communication, called Steganography offers data protection in a somewhat different manner #. In table (2) and table (3) the secret message(109) characters, encryption prevents an unauthorized party from discovering the contents of a communication # , with different images tests Lenna, Barrow and Lion bitmap images of size 352*288 are applied, table (4) table (5) applied with the above images and the secret message(93) characters . Table(6) Show the Comparison between 2-LSB(two bit Least Significant Bit) and the proposed hiding DCT method.

Table 2. Rresults of embedded differential length text

Colour	PSNR	MSE	Text length	Compression ratio	Hiding time
Red	29.94	8.11	109 character	10	00:00:04
Green	30.77	7.37			
Blue	29.24	8.79			

Table 3. Results of embedded differential length text

Colour	Entropy		Variance		Energy	
	cover	stego	Cover	stego	cover	stego
Red	7.77	7.76	174.23	185.72	0.043	0.044
Green	7.51	7.51	138.66	132.3	0.056	0.055
Blue	7.19	7.24	181.89	174.92	0.078	0.076
Average	7.49	7.50	164.92	164.31	0.059	0.058

Table 4. Results of embedded differential length text

Colour	PSNR	MSE	Text length	Compression ratio	Hiding time
Red	28.48	9.6	93 character	10	00:00:04
Green	28.93	9.12			
Blue	27.63	10.59			

Table 5. Results of embedded differential length text

Colour	Entropy		Cariance		Energy	
	Cover	Stego	Cover	Stego	Cover	Stego
Red	23.22	23.19	404.59	428.06	0.173	0.175
Green	21.86	21.87	324.39	314.38	0.257	0.258
Blue	20.8	20.88	490.51	510.22	0.331	0.331
Average	21.96	21.98	406.49	417.55	0.25	0.25

Table 6. Comparison between 2-LSB and the proposed hiding DCT method

Items	2-LSB	The proposed hiding DCT method
Hiding time	Very short	Depends on the size of an image
Security	Weak	Strong
Detection	Suspicious	Very difficult to be suspected
Size	Large	Small
Transmitting time	Very slow	Very fast

3. Conclusions

The proposed system provides the JPEG method with the digital signature and RSA cipher and hopes for an embedding text in an image. A number of conclusions are derived from this study:-

1. we used cover images, size 352*288 and a secret message with different lengths. The results explain :

- Compression ratio is different from an image to another because of the different data given for each image.
- The PSNR value for the colour of images is between (27- 33).
- The PSNR value increases with the increase of the secret message length.
- The MSE value of the colour of image is between (6-10).
- The hiding time of this image (352*288) is 4 seconds. The hiding time increases with large images.

2. Steganography is not intended to replace cryptography but rather to supplement it. If a message is encrypted and hidden with a steganographic method it will provide an additional layer of protection and reduce the chance of the hidden message being detected.

3. The proposed system can be defined as asymmetric key Steganography since it uses two keys: a secret key and a public key between the sender and the receiver, in this system there is no need for the knowledge of the original cover in the extraction process.

4. In this system, we prove that if you hide information inside an image file (BMP) and that file is converted to another image format(JPEG), the hidden information will not be lost.

5. LSB in BMP is most suitable for applications where the focus is on the amount of information to be transmitted and not on the secrecy of that information, because LSB in BMP images are surely the suspicious ones that might arise from a very large BMP image being transmitted between parties. So we use JPG, because it is suitable for images that have to be communicated over an open system environment like the Internet.

6. From the implementation we conclude that the proposed system is very rapid in performing the extraction process and the size of the embedded text does not affect the speed of the system very much.

7. The proposed system does not affect the image quality; we can say it is not noticeable for human eyes. To prove this we show the cover-image and the stego –image to a team of 15 persons to take their opinion if there is any difference between the stego-image and the cover-image and their answer that there is no difference between both images.

4. References

[1] A.Nag, S. Biswas, D.Sarkar, P.P.Sakar," A noval Technique fir Image Steganography based on Block_DCT and Huffman Encoding",Information Jornal of Computer Scince and Information Technology, Volume 2, Number 3,2010

[2] Alawy S., "Robust Information Hiding Techniques Using JPEG" University of Almusstnsry, Ms.c thesis in computer Science, 2004.

[3]Nada Abdul Aziz Mustafa," Design and Implementation proposed Encoding and Hiding Text in an Image", University of Sulaimani, Ms.c Thesis, 2010.

[4] Bushra Kassim Al-Abudi, "Colour Image Data Compression Using Multilevel Block Trunsaction Coding Technique", Phd Thesis, College of Science, University of Baghdad, 2002.

[5] Sua'd Kakil Ahmad, "Image in Image Hiding System using Iterated Function System (IFS)", Msc Thises, University of Sulaimani, 2009.

[6] Shih T.Y. & Liu J.K., "On the Performance of JPEG2000 for Aerial Photo Compression", Department of Civil Engineering National Chiao-Tung University, 2001.

[7] Trappe W. & Washington L., " Introduction to cryptography with coding theory". New Jersey: Prentice Hall, Rivest R. MD5.1992. Algorithm [Online]. Available: http://www.kleinschmidt.com/edi/md5.htm.

[8] William Stallings," Cryptography and Network Security Principles and Practice", Second Edition, United states of America Prentice Hall,1999