Multistage Hiding Image Techniques

Nadia M. Mohammed
nadia.m.mohammed@uomosul.edu.iq
College of Computer Science and Mathematics
University of Mosul, Iraq

Received on: 14/9/2010 **Accepted on: 26/10/2010**

ABSTRACT

In the few recent years computer sciences have been widely developed especially in communication spaces and Internet, therefore, a great need appear for security and safety of our information. Steganography is the science of hiding information or data (like a secret message) in other cover (like a digital image) in such a way that a normal person can't sense it.

In this paper, four new methods were suggested in steganography systems to embed secret data in compressed images. Two methods are working in spatial domain, known as moving window and odd/even LSB, others are working in transform domain, known as odd/even DCT and DCT+DWT.

The comparison results present that new methods are better than traditional methods in many characteristics, like (efficiency, security level, imperceptibility and robustness). The work was implemented using Matlab.

Keywords- Steganography, LSB, DCT, DWT.

تقنيات إخفاء الصورة متعدد المراحل

نادية معن محمد

كلية علوم الحاسوب والرياضيات / جامعة الموصل

تاريخ الاستلام: 2010/09/14 **تاريخ القبول: 2010/10/26**

الملخص

بعيد التطور الذي طرأ على علوم الحاسوب في مجال الاتصالات والانترنيت، ظهرت الحاجة إلى إيجاد وسائل لغرض إيصال المعلومات والبيانات بصورة صحيحة ومحمية من الغير. فكانت الكتابة المخفية والتي هي علم إخفاء المعلومات والبيانات السرية في غطاء رقمي مثل الصور بحيث لا يمكن للشخص العادي أن يكتشفه أو يحسه.

في هذا البحث، تم استحداث أربع طرائق جديدة في مجال الكتابة المخفية للإخفاء في الصور المكبوسة. اثنين منهما يعملان في الحيز المكاني، يسميان (moving window and odd/even LSB)، و الاثنان الآخران يعملان في الحيز الترددي، يسميان (odd/even DCT and DCT+DWT).

أظهرت المقارنة أن الطرائق المقترحة أفضل من الطرائق القديمة في عدة مجالات، منها (الكفاءة، مستوى السرية، التشوه للصور المكبوسة، ومقاومة عوامل الإزالة). العمل نفذ باستخدام Matlab.

الكلمات المفتاحية: الكتابة المخفية، LSB، DCT، DWT.

1.Introduction

It has been said throughout time that "a picture is worth a thousand words." However, in this digital area, it could be said that, "a picture is worth a thousand *secrets*". With the development of Internet technologies, digital media can be transmitted conveniently over the networks. Therefore, how to protect secret message during transmission become an important issue. Steganography provide another layer of protection on the secret message, which will be embedded in another media such that the transmitted data will be meaningful and innocuous to everyone. [1][2]

2. What is Steganography?

The word Steganography is derived from Greek words, stego (meaning secret or hidden) and the graphy (meaning drawing or writing), that mean (steganography = secret writing or covered writing), its an ancient art used to hide secret messages under or inside a picture or a figure in a manner that unauthorized persons can not detect the secret embedded there. Digital steganography depends on the same ancient art of traditional steganography which is (a science of hiding digital data or information inside other digital data or information in such a way that other unauthorized parties can not know the presence of hidden message), the first part of those digital data is called secret message (may be text, image, movie, or digital audio), the second part is called cover that may be text, image, audio, or movie. [3]

3. Steganography in Images

Images have a very common use in computer environment and are shared easily on Internet. We can send an image by email without drawing any suspicions, not like other types of multimedia. There are several reasons for using images in steganography:-

- Images contain data may be not significant, changing value of those data has no effects on images functionality, while in other types of data may have effect like in text in which changing any bit will change a letter to another.
- Human visual system and its inability to distinguish minor changes in images color.
- Images normally contain some noise, hiding information acts like adding noise; therefore it is normal when we see that an image contains noise. [1][3]

4. Survey of Steganography Techniques

- For general Information Hiding subjects, see references ([1],[2],[3]).
- For Information Hiding with LSB subjects, see references ([4],[5],[6]).
- For Information Hiding with DCT subjects, see references ([7],[9]).
- For Information Hiding with DWT subjects, see references ([8],[10]).

5. Performance Measures

- Peak Signal-to-Noise Ratio (PSNR) is computed by the following formula:

$$\textbf{PSNR} = \frac{X*Y*\ \max x,y\ (P^2x,y)}{\sum x,y\ (\ Px,y - P'x,y\ )^2}$$

Bit Error Rate (BER), which measures

$$\textbf{BER} = \frac{\text{Error Extracted Bits}}{\text{Total Extracted Bits}}$$

ratio of error bits extracted from stego-cover, computed by:[2][3]

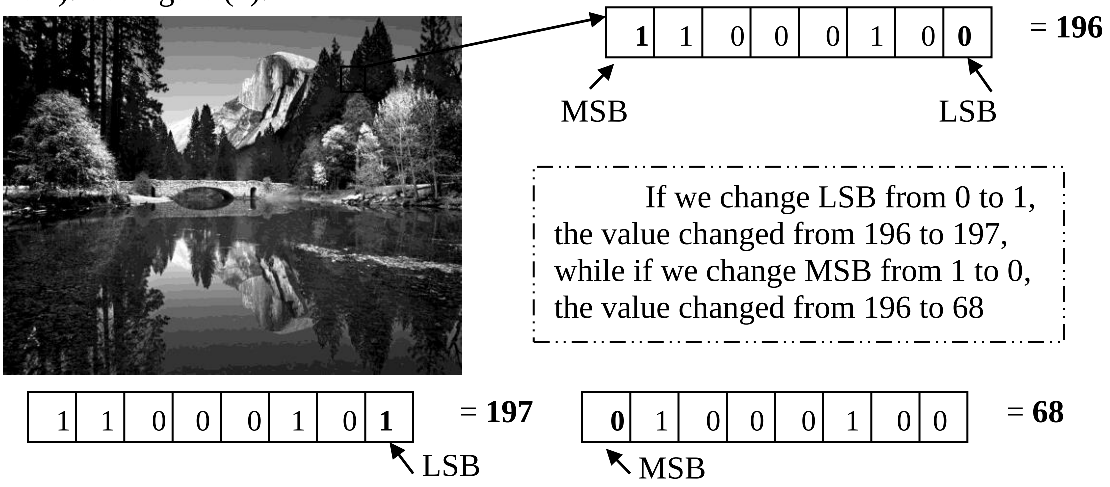
6. Spatial Domain Techniques

A high number of methods can be classified under this branch, ranging from LSB method to downgrading and cover region with parity bit and so on. All of those methods depend on substituting the least significant parts of cover by secret information bits.

Generally substitution methods are not robust methods but they have a high capacity for embedding secret information, used in places where no attacks occur but need high capacities. Some of those methods will be explained in more detail. [4]

LSB

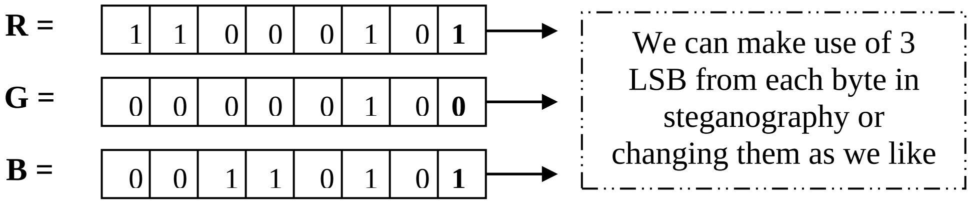
Least **S**ignificant **B**it (LSB) is the bit which has a least effect on pixel value compared with other bits. As we know that pixels may be represented by 8-bits or 24-bits, therefore in the 8-bit the bit no.1 is LSB while bit no.8 is **M**ost **S**ignificant **B**it (MSB). See figure (1).



If we change LSB from 0 to 1, the value changed from 196 to 197, while if we change MSB from 1 to 0, the value changed from 196 to 68

Figure (1): Explanation of LSB in 8-bit Image

In steganography the LSB is very important because changes in it, do not lead to big changes in image quality. Same thing is available in RGB true color as we say before that each color represented by 3 byte as shown in figure (2). [5][6]



We can make use of 3 LSB from each byte in steganography or changing them as we like

Figure (2): Explanation of LSB in 24-bit Image

7. Transform Domains Techniques

Transform domain methods hide message in significant areas of the cover image which makes them robust to attacks such as compression, cropping and other image processing methods.

Transform or frequency domains are excellent environments for hiding information because it gives stego-cover robustness and a high degree of undetectability, the only characteristic that will decrease is capacity. Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are two of the most famous transform techniques used in information hiding. [7][8]

a. DCT

Discrete Cosine Transform is the domain used for JPEG compression, therefore used for those steganography methods which want to resist JPEG compression.

The two dimensional DCT pair is given by: [3][7][9]

$$C(0,0) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y) \qquad \ldots(1)$$

$$C(u,v) = \frac{1}{2N^3}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\left[\cos(2x+1)u\pi\right]\left[\cos(2y+1)v\pi\right] \qquad \ldots(2)$$

For u,v =1,2,…,N-1, and the inverse DCT (IDCT) is given by

$$f(x,y) = \frac{1}{N}C(0,0) + \frac{1}{2N^3}\sum_{u-1}^{N-1}\sum_{v=1}^{N-1} C(u,v)\left[\cos(2x+1)u\pi\right]\left[\cos(2y+1)v\pi\right] \qquad ...(3)$$

b. DWT

The DWT and IDWT for a two dimensional image *F(m,n)* can be similarly defined by implementing the one dimensional DWT and IDWT for each dimension *m* and *n* separately as shown in figure (3) resulting in the pyramidal representation of the image shown in figure (4).

In fact the only way to construct these different levels of resolution is to cascade two channel filter banks, which are low pass filter and high pass filter. Low-pass filter do not allow high frequencies to pass through it while high pass filter do not allow low frequencies to pass through it.

As we know that images have two dimensions (height and width), so we must employ two dimensional wavelet. Filter operations are first performed on the rows of the image, and then filter operations are performed on the columns of the row transformed image. This process is iterated several times. At each iteration, a different level of resolution is represented. At each level all four combinations of low-pass and high-pass filters used on rows and columns of the image are performed. [3][8][10]

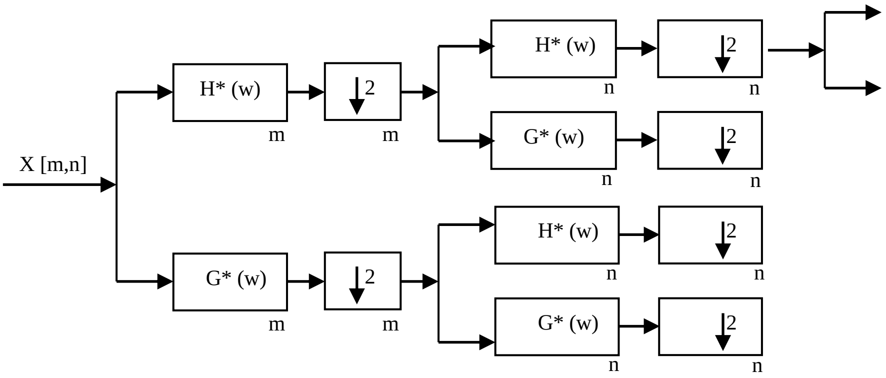
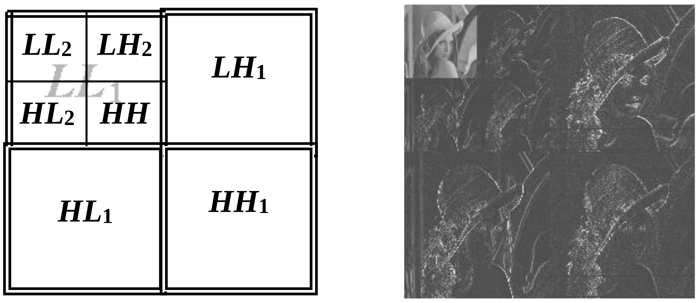


Figure (3): Two-Dimensional DWT.



(a) decomposition structure. (b) decomposed image.

Figure (4):
The pyramidal two-level decomposition of an image.

8. Enhancements And Developments

Enhancements four methods were developed to enhance the traditional methods. Two new methods work in spatial domain (moving window and odd/even LSB), while other two new methods work in transform domain (odd/even DCT and DCT+DWT). Following sections explain these methods in detail.

8.1. Spatial Domain Enhancement Methods

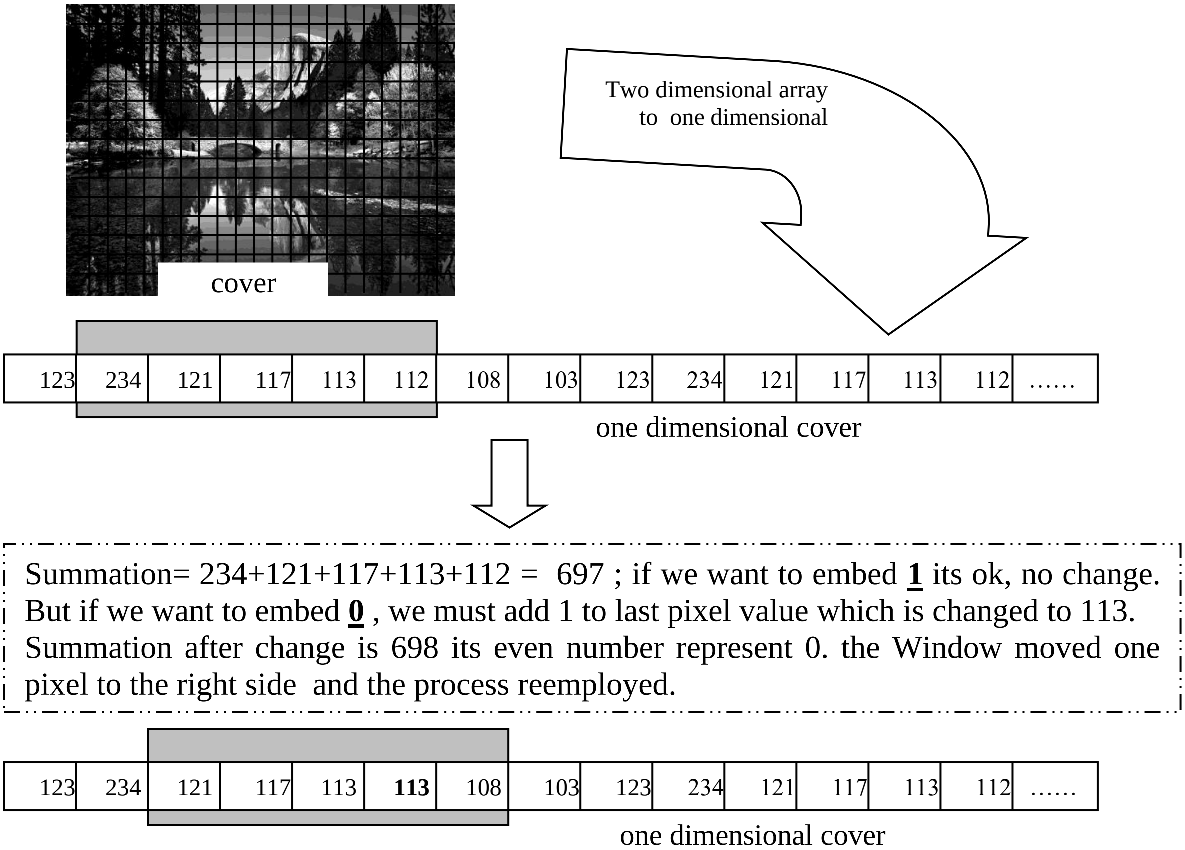
Here, an attempt to enhance the method of hiding was done. In other words, in this domain, new ways for embedding process were provided and called moving window and odd/even LSB methods.

a. Moving Window Method

In this new method, the secret bit was not embedded to a single pixel like in traditional LSB method, but to a higher number of pixels related with each other without any decrease in cover embedding capacity. Figure (5) explains the embedding process:

Embedding steps:

- Convert secret message or secret image to bits.
- Converting two dimensional cover image to one dimensional array.
- Selecting window with width 3,4,5,6,7,……
- Moving this window to one dimensional array sequentially
- In each movement, value of all pixels in the window are summed with each other and then check if the summation is odd or even number.
- If our secret bit to hide is 0 and result of summation is even its ok no changes is required, but if it is odd, we must add 1 to the last pixel in the window. The same thing done for 1 secret bit but in reverse order.
- After embed process is done, window is moved one pixel to the right until it covers all pixels in the array.



123	234	121	117	113	112	108	103	123	234	121	117	113	112	……

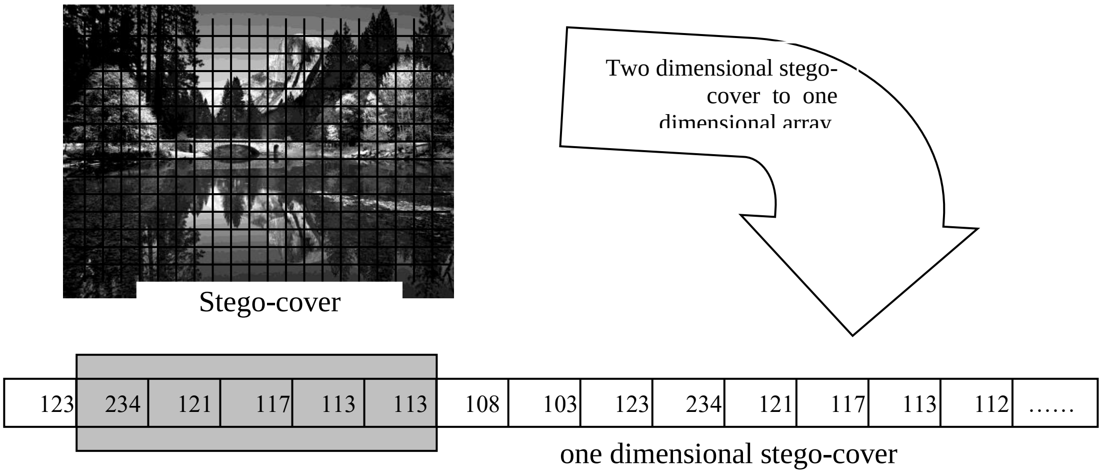
Summation= 234+121+117+113+112 = 697 ; if we want to embed **1** its ok, no change. But if we want to embed **0** , we must add 1 to last pixel value which is changed to 113. Summation after change is 698 its even number represent 0. the Window moved one pixel to the right side and the process reemployed.

123	234	121	117	113	**113**	108	103	123	234	121	117	113	112	……

Figure (5): Explanation of moving window method in embed process

Extracting steps:

- Converting image produced in the embed process from two dimensional stego-cover to one dimensional array.
- Selecting window with the same width as in embed process 3,4,5,….
- Moving this window on the one dimensional array sequentially
- In each movement, value of all pixels in the window are summed with each other and check if it is odd or even number.
- If result of summation is odd that means secret bit=1, but if its even secret bit= 0.
- After each extraction done, window moved one pixel to the right until it moves above all pixels in the array.
- We can rearrange secret bits with each other and reconstruct message or secret image. See figure (6).


Stego-cover
Two dimensional stego-cover to one dimensional array
one dimensional stego-cover

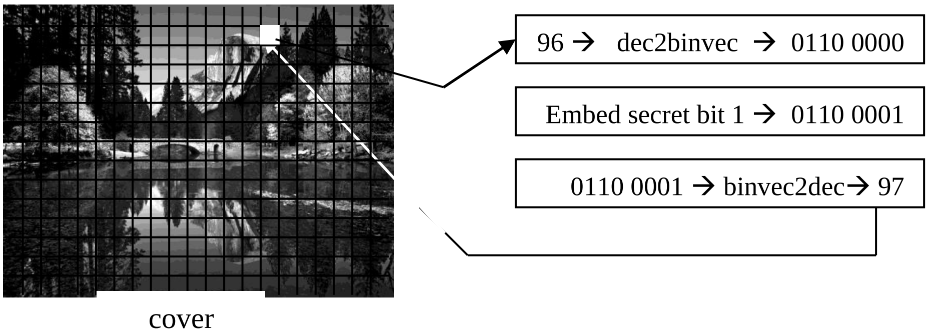
123	234	121	117	113	113	108	103	123	234	121	117	113	112	……

Summation= 234+121+117+113+113 = 698; it is even number which means that the secret bit is **0**. Window is moved one pixel to right side and the process reemployed**.**

Figure (6): Explanation of moving window method in recovery process

b. Odd/Even LSB Method

LSB traditional method is not efficient and very slow in execution because of the amount of processes done in each step. For each pixel we must change, the value of pixel from decimal to binary (in Matlab the code is dec2binvec which takes relatively long time) and then change will done in least significant bit. After embed we must reconvert binary representation to decimal (in Matlab the code is binvec2dec which takes long time too) all of those processes are explained in figure (7).



96 → dec2binvec → 0110 0000

Embed secret bit 1 → 0110 0001

0110 0001 → binvec2dec→ 97

Figure (7): How LSB method is executed

Therefore to avoid time waste and because efficiency is one of the requirements of steganography, another way was suggested which is more efficient, more simple and easier for employing and is called odd /even LSB method.

Embedding steps:

- Cover image consists of pixels. For embedding secret bit random pixels are selected using secret key.
- Simply calculate pixel value to determine if it is odd or even.
- If secret bit is 0, pixel value must be even, if not, we must add 1 to it. On the other hand if secret bit is 1, pixel value must be odd, if not, 1 is added to it too.

- After embedding is done, pixel value is returned to its position in the cover image, with a very small change in color intensity.
- Another pixel is selected and the same process is employed, until all bits are embedded, and the cover called stego-cover. See figure (8).

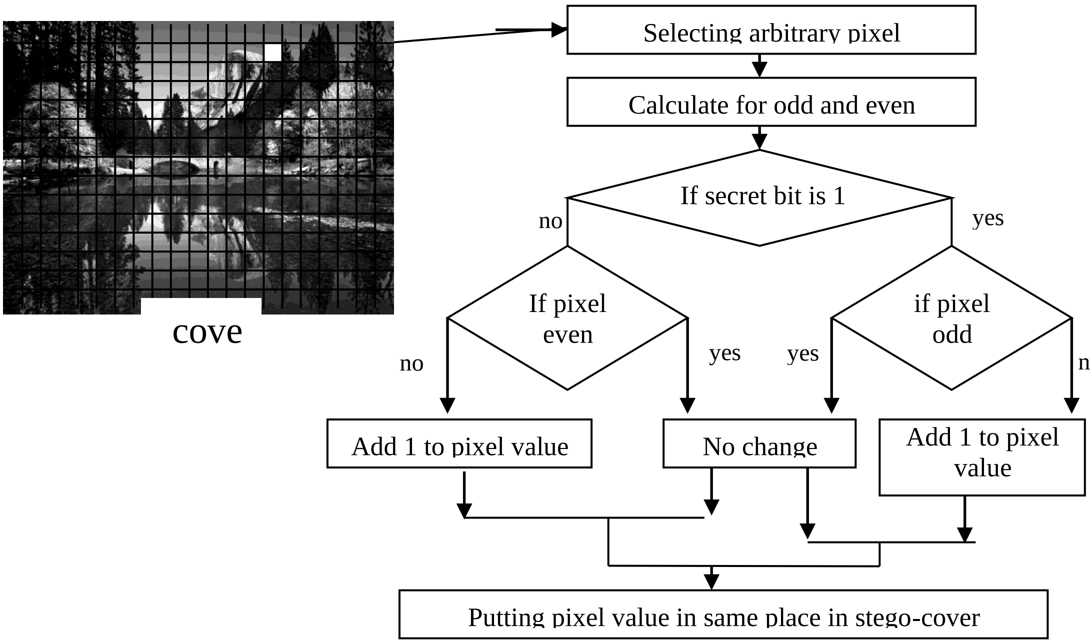


Figure (8): Embedding process of Odd/even LSB method

Extracting steps:

- Selecting pixels is accomplished randomly like in embed process (same key).
- Simply calculate pixel value to determine if it is odd or even.
- If pixel value is even→ secret bit is 0; if pixel value is odd → secret bit is 1.
- Another pixel is selected and the same process is employed, until all bits are recovered, and then secret message is reconstructed. See figure (9).

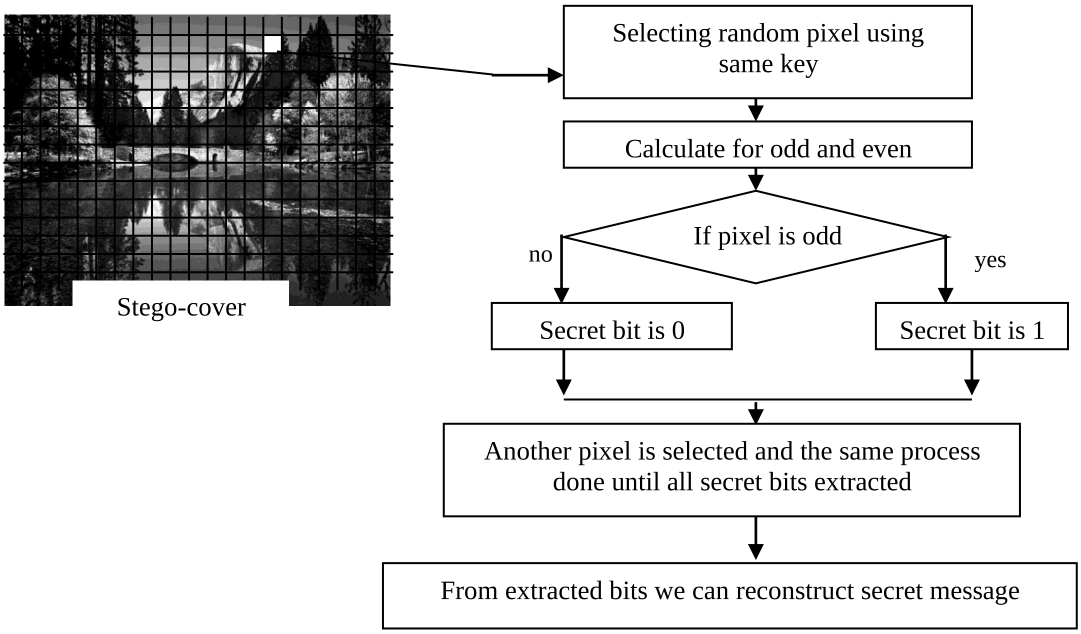

Stego-cover
Selecting random pixel using same key
Calculate for odd and even
If pixel is odd
no
yes
Secret bit is 0
Secret bit is 1
Another pixel is selected and the same process done until all secret bits extracted
From extracted bits we can reconstruct secret message

Figure (9): Recovery process of Odd/even LSB method

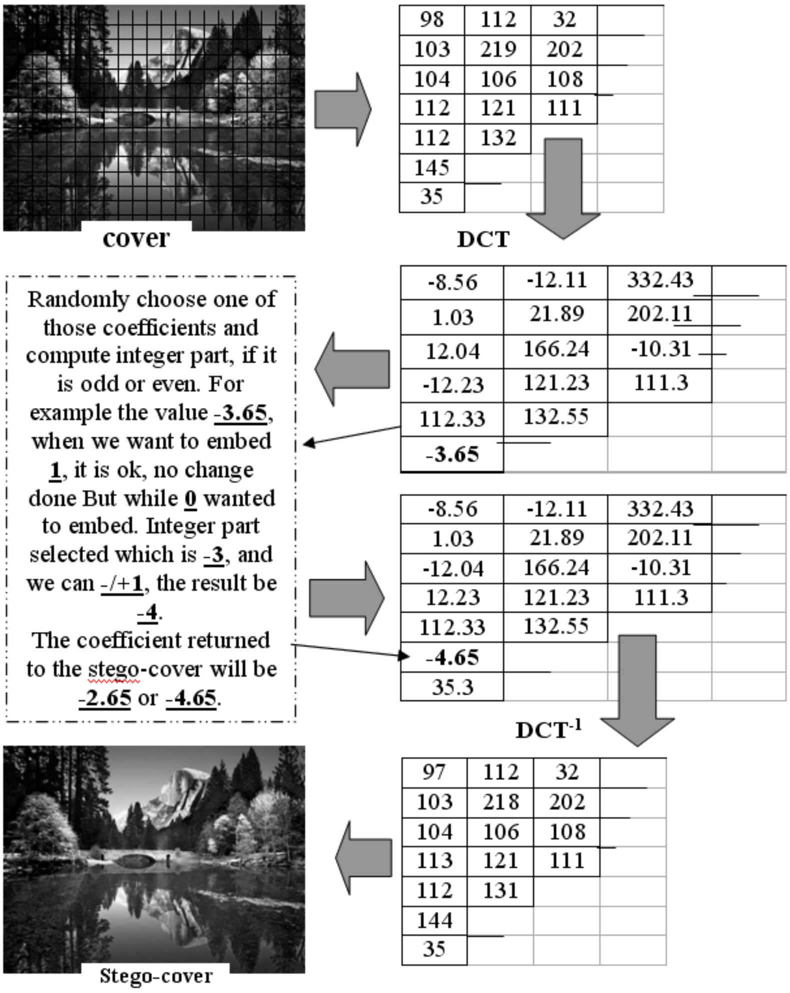
8.2. Transform Domain Enhancement Methods

Methods working in transform domain are usually more robust than those working in spatial domain especially against jpeg lossy compression, the most useful transform domains used in information hiding are discrete cosine transform (DCT) and discrete wavelet transform (DWT).

a. Odd/Even DCT Method

Embedding process for this method is as follows:

- Converting secret data to binary representation which may be a secret message or a secret image.
- Apply DCT to cover image to produce DCT coefficients.
- Choosing positions in cover image coefficients using a secret key.
- For embedding each bit we need three coefficients. Because 3 is an odd number, if one of those three coefficients is recovered wrong, then recovered bit can be determined by the two other remaining points.
- Determine if the integer part of coefficient is odd or even. Odd representing 1, even representing 0.
- If secret bit is not like integer part, only a small change is done by adding 1 to integer part.
- After all changes are done, inverse DCT operation is employed for the whole image, and the result is called stego-cover. See figure (10).



98	112	32	
103	219	202	
104	106	108	
112	121	111	
112	132		
145			
35			

-8.56	-12.11	332.43	
1.03	21.89	202.11	
12.04	166.24	-10.31	
-12.23	121.23	111.3	
112.33	132.55		
-3.65			

Randomly choose one of those coefficients and compute integer part, if it is odd or even. For example the value **-3.65**, when we want to embed **1**, it is ok, no change done But while **0** wanted to embed. Integer part selected which is **-3**, and we can **-/+1**, the result be **-4**.
The coefficient returned to the stego-cover will be **-2.65** or **-4.65**.

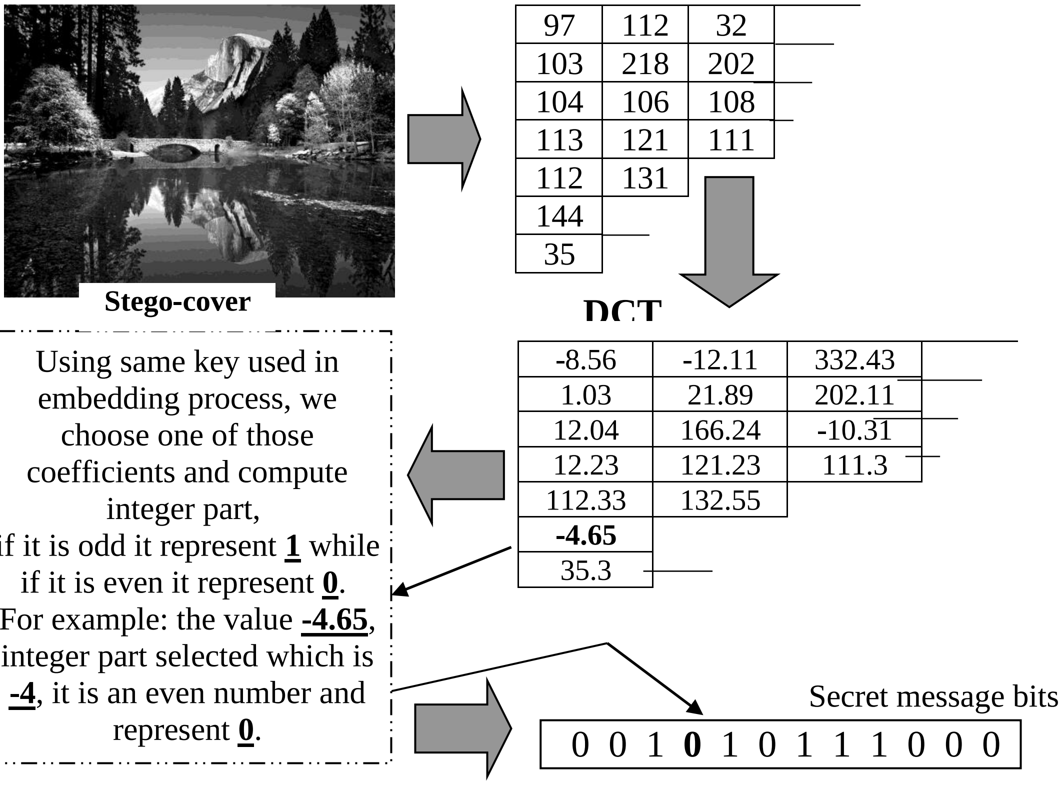
-8.56	-12.11	332.43	
1.03	21.89	202.11	
-12.04	166.24	-10.31	
12.23	121.23	111.3	
112.33	132.55		
-4.65			
35.3			

97	112	32	
103	218	202	
104	106	108	
113	121	111	
112	131		
144			
35			

Figure (10): Embedding process of Odd/even DCT method

Extracting steps:

- DCT is employed on a stego-cover and coefficients produced.
- Randomly choose positions in stego-cover coefficients using the same key of embedding process.
- Simply compute if the integer part is odd or even. Odd = 1, even = 0.
- As, a three bits are extracted from each coefficient; therefore we consider the extracted bit 1 if those three bits are 1 or at least two of them are 1. And the extracted bit considered 0 if those three bits are 0 or at least two of them are 0.
- After all secret bits are extracted from stego-cover they are combined with each other to reconstruct the secret information. See figure (11).



97	112	32
103	218	202
104	106	108
113	121	111
112	131	
144		
35		

-8.56	-12.11	332.43
1.03	21.89	202.11
12.04	166.24	-10.31
12.23	121.23	111.3
112.33	132.55	
-4.65		
35.3		

Using same key used in embedding process, we choose one of those coefficients and compute integer part, if it is odd it represent **1** while if it is even it represent **0**. For example: the value **-4.65**, integer part selected which is **-4**, it is an even number and represent **0**.

0 0 1 **0** 1 0 1 1 1 0 0 0

Figure (11): Recovery process of Odd/even DCT method

b. DCT+DWT Method

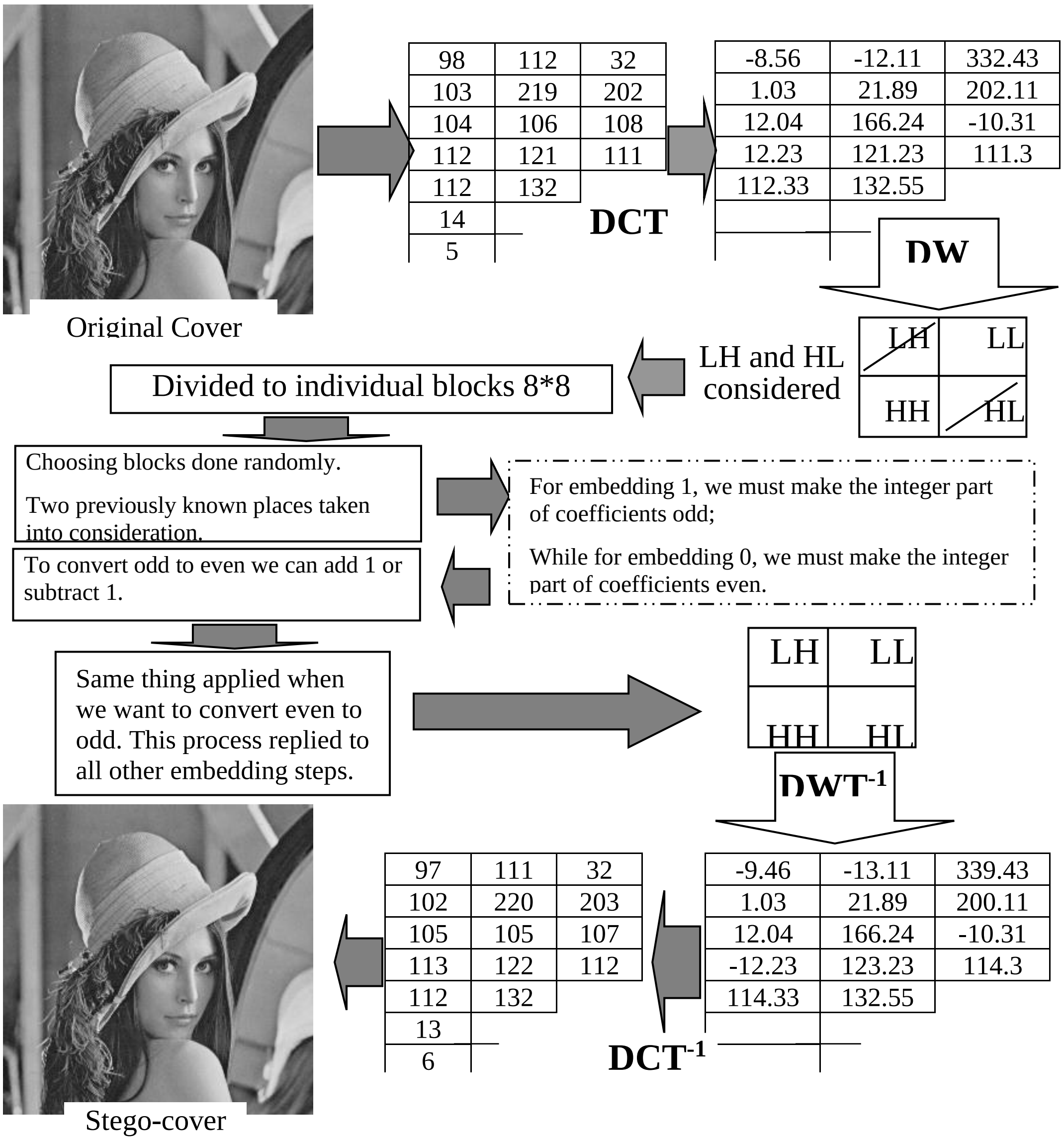
This method aimed to combine two transform domains which are (DCT) and (DWT) and using them to increase robustness and imperceptibility of hiding information.

This method is very robust and very imperceptible, because those two methods are very robust, DCT have a great resistance to jpg compression while DWT give us ability to choose places which are most suitable for hiding information. Therefore combining these two methods with each other provides advantages of both. Generally cracking or analyzing DCT+DWT method is impossible. The only disadvantage in using this method is capacity.

Embedding steps:

- DCT process employed to cover image, result is DCT coefficients.

- DWT process employed to result of step 1, DWT will divide coefficient array to four regions according to frequency of coefficients. Two middle regions choosed to hide information in it which are (high low) and (low high), while the other two (high high) and (low low) are ignored, because (high high) region contains high frequency coefficients which are very easy to be attacked or removed, while (low low) region contains low frequency coefficients in which any change will make changes in original cover perceptible.
- After taking those two middle frequency regions in consideration, they are divided to 8×8 blocks.
- Using secret key, a random 8×8 block is choosed.
- Three previously known places in the block are considered, if we want to hide 1, we must make integer part of these three points odd, while for hiding 0 we must make integer part of those three points even by adding 1 or subtracting 1 to coefficient values.
- After embedding all secret bits in choosed blocks, inverse DWT process is employed on the stego coefficients for the whole image.
- Employing inverse DCT is the last step in embedding process, and a robust stego-image is produced. See figure (12).



98	112	32
103	219	202
104	106	108
112	121	111
112	132	
14		
5		

-8.56	-12.11	332.43
1.03	21.89	202.11
12.04	166.24	-10.31
12.23	121.23	111.3
112.33	132.55	

Choosing blocks done randomly.

Two previously known places taken into consideration.

For embedding 1, we must make the integer part of coefficients odd;

While for embedding 0, we must make the integer part of coefficients even.

To convert odd to even we can add 1 or subtract 1.

Same thing applied when we want to convert even to odd. This process replied to all other embedding steps.

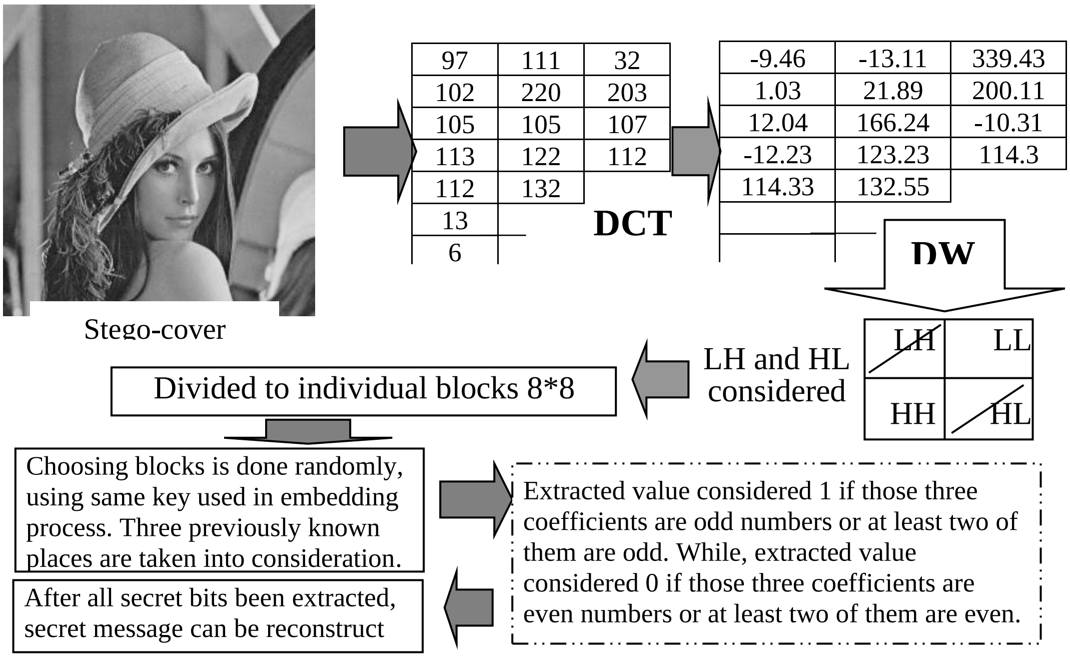
-9.46	-13.11	339.43
1.03	21.89	200.11
12.04	166.24	-10.31
-12.23	123.23	114.3
114.33	132.55	

97	111	32
102	220	203
105	105	107
113	122	112
112	132	
13		
6		

Figure (12): Embedding process of DCT+DWT method

Extracting steps:

- DCT process is employed to cover image, the result is DCT coefficients.
- DWT process is employed to the result of step 1, DWT will divide coefficient array to four regions according to frequency of coefficients. Two middle regions choosed to hide information in it which are (high low) and (low high), while the other two (high high) and (low low) are ignored.
- After taking those two middle frequency regions in consideration, we will divide them to 8×8 blocks.
- Using same secret key used in embedding process, a random 8×8 block is choosed.
- Three previously known places in the block taking into consideration, if 1st point is greater than 2nd point that means secret bit is 1, while if 2nd point is grater than 1st point means secret bit is 1.
- After extracting all secret bits, we can reconstruct secret message or secret image embedded. See figure (13).



97	111	32
102	220	203
105	105	107
113	122	112
112	132	
13		
6		

-9.46	-13.11	339.43
1.03	21.89	200.11
12.04	166.24	-10.31
-12.23	123.23	114.3
114.33	132.55	

~~LH~~	LL
HH	~~HL~~

Divided to individual blocks 8*8

Choosing blocks is done randomly, using same key used in embedding process. Three previously known places are taken into consideration.

Extracted value considered 1 if those three coefficients are odd numbers or at least two of them are odd. While, extracted value considered 0 if those three coefficients are even numbers or at least two of them are even.

After all secret bits been extracted, secret message can be reconstruct

Figure (13): Recovery process of DCT+DWT method

9. Results

Results of moving window method and odd/even LSB method are compared with traditional LSB method [5], while results of odd/even DCT method and DWT+DCT method will be compared to traditional DCT method [11].

We must note that cover image used in traditional LSB, moving window and odd/even is same cover of a gray scale (256×256) pixels called (view.PNG). The cover image used in traditional DCT, DWT+DCT and odd/even DCT method is (lena_orig.JPG) gray scale (256×256) pixels.

Secret information comes in two general manners; the first one is secret text message while the second type is secret image in two colors black for writing and white as a back ground, in other words writing in image manner.

9.1. Moving Window Method

In this method, secret bit embedded not to a single pixel like in LSB traditional method but to a higher number of pixels related with each other without any decrease in cover embedding capacity using moving window technique. See table (1).

Table (1): Comparison between traditional LSB [5] and moving window_7

	secret is *text* message		secret is *image* message	
Properties	**Normal LSB**	**Moving Window_7**	**Normal LSB**	**Moving Window_7**
PSNR (dB)	51.06	51.086	51.08	52.02
Capacity	65,536 bits	65,536 bits	65,536 bits	65,528 bits
BER	0 %	0 %	0 %	0 %
No. of Secret Key	1	2 (window-size)	1	2 (window-size)

9.2. Odd/Even LSB Method

In traditional LSB when we want to embed a pixel we must change value of pixel from decimal to binary (Matlab code = dec2binvec which take relatively long time) and then embed change will done in least significant bit. After embed we must reconvert binary representation to decimal (Matlab code = binvec2dec which takes long time too), while in odd/even LSB all those further works are eliminated and instead only we test if its odd or even and we make it by adding 1 or subtracting 1 to pixel value which give same result in shorter time as shown in table (2).

Table (2): Comparison of execution time needed between traditional LSB [5] and odd/even LSB

	secret is *text* message		secret is *image* message	
Properties	**Normal LSB**	**odd/even LSB**	**Normal LSB**	**odd/even LSB**
PSNR (dB)	51.06	51.086	51.08	52.02
Capacity	65,536 bits	65,536 bits	65,536 bits	65,528 bits
BER	0 %	0 %	0 %	0 %
Execution time	605.1800 sec.	44.9350 sec.	643.8660 sec.	66.4260 sec.

From above table (1st part), we note that execution time is reduced nearly 12 times or (12:1) in embed program only and same thing in extracting program. While (2nd part), present that the time needed to execute an embed program is reduced 10.5 times (10.5:1).

9.3. Odd/Even DCT Method

With this section, a comparison between odd/even DCT and traditional (Zhao) DCT [11] is presented, the results are shown in table (3).

Table (3): Comparison between Zhao DCT[11] and odd/even DCT, Secret is *text* message

	PSNR (dB)		*BER*		*File Size*	
Comparison	**Zhao DCT**	**Odd/ even DCT**	**Zhao DCT**	**Odd/ even DCT**	**Zhao DCT**	**Odd/ even DCT**
BMP no comp.	**26.12**	**46.37**	**0.59 %**	**0 %**	**65.0 KB**	**65.0 KB**
JPG q=100	**26.12**	**46.37**	**0.39 %**	**0 %**	**58.0 KB**	**57.5 KB**
JPG q=95	**26.12**	**46.37**	**3.13 %**	**3.02 %**	**35.4 KB**	**34.9 KB**

As it is clear that, PSNR is changed from 26 to 46. For BER, we notice that, the new method always have smaller values than traditional DCT.

9.4. DCT+DWT Method

The same chance is given to both traditional DCT and DCT+DWT method by using the same (cover, secret message, size of secret message, ratio of attacks). Results are explained in table (4).

Table (4): Comparison between Zhao DCT [11] and DCT+DWT, Secret is *text* message

	PSNR (dB)		*BER*		*File Size*	
Comparison	**Zhao DCT**	**DCT + DW T**	**Zhao DCT**	**DCT + DW T**	**Zhao DCT**	**DCT + DW T**
BMP no comp.	**26.12**	**46.29**	**0.59 %**	**0 %**	**65.0 KB**	**65.0 KB**
JPG q=100	**26.12**	**46. 29**	**0.39 %**	**0 %**	**58.0 KB**	**57.5 KB**
JPG q=95	**26.12**	**46. 29**	**3.13 %**	**3.02 %**	**35.4 KB**	**34.9 KB**

As it is clear that, PSNR is changed from 26 to 46. For BER, we notice that, the new method always have smaller values than traditional DCT.

10. Conclusions

In this paper, traditional existing methods are enhanced in two main branches of steganography which are spatial and transform domain techniques. Four new methods are proposed. It is clear from comparable tables, that new enhanced methods are better than exists traditional methods.

All of these works are done on a :

- Lossless compressed image cover (PNG) for spatial domain.
- Lossy compressed image cover (JPG) for transform domain.

The conclusion can be summarized as follows:

a. Lossless image formats better than lossy image formats for steganography.
b. Employing spatial domain techniques are easier than employing transform domain techniques (in both cost and execution time).
c. The capacity of spatial domain techniques for hiding secret bits is higher than the capacity of transform domain techniques.
d. In many cases, transform domain techniques are more robust than spatial domain techniques.
e. Odd/even LSB method is more efficient than traditional LSB between (1:10) to (1:12), see table (2).
f. Moving window method gives steganography systems another additional security key, which is the window size.
g. Odd/even DCT is better than traditional Zhao DCT method, because it makes imperceptibility less, see value of (PSNR).
h. DCT+DWT method is more imperceptible than traditional Zhao DCT, see value of (PSNR).

REFERENCES

[1] Mihcak, M., (2002), “Information Hiding Codes and Their Applications to Images and Audio”, Ph.D. Thesis at Electrical Engineering Department, the Graduate College, the University of Illinois at Urbana-Champaign.

[2] Le, Tri V., (2004), “Information Hiding”, Ph.D. Dissertation at Dept of Computer Science, College of Arts and Sciences, Florida State University.

[3] Nori, A. Sami, (2006), "An Investigation for Steganography in Moving Pictures", Ph.D., Thesis submitted to the Computer Science Dept., College of Computer Science & Mathematics Sciences, University of Mosul.

[4] Koppola, R. R., (2009), "A High Capacity Data-Hiding Scheme in LSB-Based Image Steganography", M.Sc. Thesis Presented to Akron University.

[5] Juneja, M., et. al., (2009), "Application of LSB Based Steganographic Technique for 8-bit Color Images", World Academy of Science, Engineering and Technology.

[6] Yu, L., et. al., (2010), "Improved Adaptive LSB Steganography Based on Chaos and Genetic Algorithm", EURASIP Journal on Advances in Signal Processing, Volume Article ID 876946, 6 pages.

[7] Lin, C., and Rd, C., (2010), "High Capacity Data Hiding Scheme for DCT-based Images", Journal of Information Hiding and Multimedia Signal Processing, Volume 1, Number 3, July.

[8] Sue, V. K., (2004), "Digital Image Compression Using Wavelets", M.Sc. Thesis, Chinese University of Minnesota Duluth.

[9] Kumar, K. B. S., et. al., (2010), "Bit Length Replacement Steganography Based On Dct Coefficients", International Journal of Engineering Science and Technology, Vol. 2(8), 3561-3570.

[10] Al-Haj, A., and Mohammad, A., (2010), "Digital Audio Watermarking Based on the Discrete Wavelets Transform and Singular Value Decomposition", European Journal of Scientific Research, Vol.39, No.1, pp.6-21.

[11] Zhao, J., and Koch, E., (1995), “Towards Robust and Hidden Image Copyright Labeling”, IEEE Workshop on Nonlinear Signal and Image Processing, June, pp. 452-455