Bit-4 of Frequency Domain-DCT Steganography Technique

Nedal M. S. Kafri[1] and Hani Y. Suleiman[2]
Department of Computer Science & IT,
Al Quds University, Palestine.
[1]nkafri@science.alquds.edu
[2]hani@itce.alquds.edu

Abstract

Steganography is the art and science of concealing secret information in such a way that no one apart from the sender and intended recipient even realize that there is hidden information. In this paper, we describe a new method of steganography based on embedding message bits in the 4^{th} *bit of the coefficients of a transform domain, such as the DCT and Wavelet, of an image. The proposed technique utilizes the idea of SSB-4 technique in modifying the other bits (i.e,* 1^{st}, 2^{nd}, 3^{rd} *and/or* 5^{th}*), to obtain the minimum variation between the original and the modified coefficient. Since this approach uses significant bit, the hidden message resides in more robust areas, spread across the entire stego image, and provides better resistance against steganalysis processes. The obtained experimental results also indicate that; the proposed method will be a good and acceptable steganogaphy scheme.*

1. Introduction

Security of information is one of the most important factors of information technology and communication. Security of information often lies in the secrecy of its existence and/or the secrecy of how to decode it. Cryptography techniques often use the worst approach assuming that only one of these two conditions holds [5]. It was created as a technique for securing the secrecy of communication. Various methods have been developed to encrypt and decrypt data in order to keep the message secret. Unfortunately, it is not enough to keep the content of the information/message secret, it may also be necessary to keep the existence of the information secret. The technique used to implement this, is called steganography.

Steganography is the science and art of hiding information in another. The definition according to Neil Johnson "Steganography is the art of hiding information in a way that prevent the detection of hidden message" [9]. It is a useful tool that allows covert communication amongst acknowledged parties. The word steganography is derived from the Greek words "stegos" meaning cover/hidden/roof and "grafia" meaning writing defining it as "covered writing" and essentially means "to hide in plain sight" [5] [9]. In image steganography the information is hidden exclusively in images. Hiding messages by masking their existence is nothing new. Before the digital era, simple steganographic techniques have been in use for hundreds of years. However, with the emergence of networks and digital technologies and increasing use of communication and files in electronic format, new techniques for information hiding have become possible.

Steganography relies on hiding covert message in unsuspected text, protocols, images, and multimedia (audio/video) data which is generally used in secret communication between acknowledged parties. Also steganography is a method of encryption that hides data among the bits of a cover file, such as a graphic or an audio file. The technique replaces bits with the secret data. Steganography is widely used in image [4]. The most common image format include BMP, GIF and JPEG. Therefore, in this paper we focus on hidden message in this media type. Generally, the main idea of the proposed method is to hide message bits in a significant bit of a transform domain coefficients of an image and inverse it to spatial domain. The most well-known transform coding techniques used to implement lossy image compression are the discrete cosine transform (DCT) and Wavelet. It should be noted that the Wavelet technique recently replaced the DCT for the JPEG standard. Nevertheless, we utilized the DCT to evaluate the efficiency of the proposed method as an example.

The rest of this paper is organized as follows: Section 2 gives a background regarding the main schemes of steganography; spatial domain and frequency domain, and their evaluation techniques. While Section 3 introduces the proposed steganography algorithm. Section 4 presents and discusses the obtained experimental results. Finally, Section 5 concludes the paper.

978-1-4244-4615-5/09/$25.00 ©2009 IEEE

2. Background

Most of the camouflage processes use the redundant bits of an image to embed secret messages. Redundant bits are all bits that can be modified without perceptible change in the visual feature of a digital picture/image. It should be noted that pixels of most images are represented as triples (Red, Green, and Blue contributions), and the Blue one is the most imperceptible to human eye [1]. Each color is represented by a number of bits depending on the desired color of the final image. The number of bits of a color scheme is called the "bit depth" and refers to the number of bits used for each pixel. The typical bit depth used for grayscale and monochrome images is 8 bits, and for digital color image is 24 bits [13] [15]. In the case of a 24 bitmap, each base color (Red, Green, and Blue) in a pixel has eight bits. Note that grayscale images can be obtained when the values of the (Red, Green, and Blue) are equivalent (from $0\ 0\ 0$ to $255\ 255\ 255$).

Image steganography schemes can be divided into two groups: Spatial/image Domain and Frequency/Transform Domain. Spatial domain techniques embed messages in the intensity of the pixels directly, while in transform domain, images are first transformed and then the message is embedded in the image [5].

2.1. Spatial domain steganography

Least Significant Bit (LSB) is the first most famous and easy spatial domain steganography technique. It embeds the bits of a message in a sequential way in the LSB of the image pixels [2] [14]. But the problem of this technique is that if the image is compressed then the embedded data may be destroyed. Thus, there is a fear for damage of the message that may have sensitive information [13]. Moreover, these kinds of methods are easy to attack by steganalysis techniques. LSB has been improved by using a Pseudo-Random Number Generator (PSNR) and a secret key in order to have private access to the embedded information [7]. The embedding process starts with deriving a seed for a PRNG from the user password and generating a random walk through the cover image that makes the steganalysis hard. Another recent improvement based on random distribution of the message was introduced by M. Bani Younes and A. Jantan [20]. In this method they utilize an encryption key to hide information about horizontal and vertical blocks where the secret message bits are randomly concealed.

Although those spatial hiding methods enable us to embed a great amount of information, they are not robust against attacks. The embedding process can be made in the LSB1, LSB2 or even in more significant bits such as System of Steganography using Bit-4 (SSB-4) by using the fourth bit of the pixel image [18].

SSB-4 steganography approach introduced by Rodrigues, Rios and Puech in 2003 [18]. This approach is based on the observation that a small variation in the channel color value of a colored image (e.g., RGB-24 image) is imperceptible to human eye [1]. SSB-4 steganography technique talks about changing the 4^{th} bit of a pixel in the original image according to the bit message. Then modify the other bits (1^{st}, 2^{nd}, 3^{rd} and/or 5^{th}) to minimize the difference between the new/changed pixel value and the original one. Note that the 4^{th} digit is a significant bit and if the image is compressed the embedded information will not be destroyed [6]. The authors in [18] argued that the difference must be equal or less than four (i.e., ± 4). The 4^{th} bit was chosen because it satisfies that changing of ± 4 units in the channel color value is imperceptible to human eyes, and it is the most significant bit which provides the minimum change in the pixel values [5] [18]. Since changing the values smaller than 4 or greater than 251 can be perceptible to human eye, they usually are not employed to embed information.

2.2. Frequency domain steganography

Recalling that in spatial domain the data embed inside pixels directly, while in transform domain, images are first transformed and then the message is embedded in the image [5]. On the other hand, in the transform domain the embedding process can usually hide less information into pictures. There is no such an exact limit in the size of the embedded object as in the case of LSB insertion, where the number of pixels and the color depth determine the maximum size of the embedded data, while retaining the invisibility of occurred changes during embedding [12]. However, as noted in [19], "by imbedding data in the transform domain, the hidden data resides in more robust areas, spread across the entire image, and provides better resistance against signal processing".

There are many techniques used to transform image from spatial domain to frequency domain and lossy image compression can be thought of as an application of such transform coding. The most common frequency domain methods usually used in image processing are the 2D DCT and Wavelet [10] [12] [13]. In this work, we utilize the DCT as an example of the transform coding technique that can be used.

The DCT helps separate the image into parts of differing importance (with respect to the image's visual quality). In practical, DCT can be carried out by partitioning/sectioning the image into equally size 2D blocks i.e., $N \times N$ grids (e.g., 8×8 grid containing 64 pixels per grid). With each grid a DCT coefficient for every component in the pixel is calculated. The formula used to calculate the DCT coefficient $S(u, v)$ (for $u, v = 0, 1, 2 \dots N-1$) of an image grid of pixels $F(x, y)$ is given in Equation 1 [8] [11] [17]:

$$S(u,v) = \frac{2}{N}C(u)C(v)\left[\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}F(x,y) * cos\left(\frac{\pi u(2x+1)}{2N}\right)cos\left(\frac{\pi v(2y+1)}{2N}\right)\right], \quad (1)$$

where $C(k) = \frac{1}{\sqrt{2}}$, when $k = 0$; otherwise $C(k) = 1$, and each $F(x, y)$ pixel value has a level range from 0 to 255 in 8 bits monochromic image. It should be noted that for most images much of signal energy lies at low frequencies; these appear in the upper left corner of the grid of DCT coefficients. Note that since these techniques modify only nonzero DCT coefficients, message lengths are defined with respect to the number of nonzero DCT coefficients in the images [10].

To reproduce a grid of image pixels $F(x, y)$, (for $x, y = 0, 1, 2 \dots N-1$), from the grid of DCT coefficients S(u,v), we can use the inverse of the DCT formula given in Equation 2:

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

2.3. Stego-image quality measures

Detecting an embedded message defeats the primary goal of steganography, that of concealing the existence of a hidden message. As steganography is based on obscurity, the most important tests are related to the human perception. These types of tests evaluate the invisibility or transparency. The most used tests are the Subjective and the Peak-Signal-to-Noise-Ratio PSNR in dB (decibel).

The subjective tests are carried out by people who look for visual differences between the images (original and stego image) trying to find which one of them is the original. If the percentage of success goes 50%, it can be concluded that the message is invisible. The subjective test's rules and recommendations are defined by the International Telecommunication Union [8] [18].

Unlike the subjective approach which is vulnerable to human vision, PSNR (Peak Signal to Noise Ratio) is a technical approach usually used to evaluate the real quality of stego image [3] [16]. This technique is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The PSNR is most commonly used to measure the quality of reconstruction in an image; by comparing the stego image with the original image. $PSNR$ can be calculated using the mathematical models/formulas in Equation 3, and 4 below. First we calculate MSE using Equation 3:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\|I(i,j) - K(i,j)\|^2, \quad (3)$$

Where MSE is the Mean Squared Error of $m \times n$ monochrome images I and K, where one of the images is considered a noisy approximation of the other, where lower is better. Thereafter, we can calculate PSNR using Equation 4:

$$PSNR = 10.log_{10}\left(\frac{MAX_i^2}{MSE}\right) = 20.log_{10}\left(\frac{MAX_i}{\sqrt{MSE}}\right), (4)$$

Where, MAX_i is the maximum pixel value of the image. In other words $MAX_i = 2^b - 1$, where b is the bit depth of the original image (e.g., $MAX_i = 255$ in the case of 8 bits depth grayscale images). Typical values for the PSNR in image and video compression are between 30 and 50 dB, where higher is better.

3. The proposed method

The challenge in this work was to find a way to camouflage a secret message in an image without perceptible degrading the image quality and to provide better resistance against steganalysis process. Therefore, we applied a combination of transform/frequency domain by means of DCT and the notion/idea of SSB-4 technique of spatial domain steganography.

The main idea of this method is to utilize significant bit (such as the 4^{th} bit) of the DCT coefficients of a cover image to hide message bits. This method modifies the 4^{th} bit of the coefficients while retaining the minimum difference between the original value and the modified one. In order to minimize this variation we can modify the 1^{st}, 2^{nd}, 3^{rd} and/or 5^{th} bits. The effect of this variation is distributed across the image by using the inverse of the discrete cosine transform (IDCT). This approach is illustrated in details in the following four steps (algorithm):

- Step 1: Applying 2D DCT on image pixels

  Herein the image is partitioned into 2D 8×8 grids/blocks. Thus each grid $F(x, y)$ consists of 64 values. If the image is 8 bits depth monochromic, $F(x, y)$ consists of the whole pixels' values. In the case of RGB 24-bits depth colored image, each grid $F(x, y)$ is constructed from only the least significant bytes (i.e., Blue color channel/contribution) of the successive pixels. This is because the Blue channel is the most imperceptible to human eye. Then we calculate the 64 DCT coefficients $S(u,v)$ of each grid $F(x,y)$ of the image using Equation 1 (in Sec. 2.2).

- Step 2: Embedding message bits

  In this step, message bits are embedded one by one in the successive nonzero DCT coefficients of the low frequency region of the $S(u,v)$ (i.e., the upper left corner of the grid of the DCT coefficients): if the value of the 4^{th} bit and the message bit are equal, nothing should be made. Otherwise, the 4^{th} bit should be replaced by a message bit and modify the 1^{st}, 2^{nd}, 3^{rd} and/or 5^{th} bits to minimize the difference. This process is illustrated in the following piece of code:

  for $(u = 0; u \leq 7; u{+}{+})\{$
    for $(v = 0; v \leq 7 - u; v{+}{+})\{$
      $c = 4^{th}$ *bit of the* $S(u,v)$
      if($S(u,v) \neq 0$ *&& c* $\neq$ *message bit)*$\{$
        4^{th} *bit* $=$ *message bit and modify the*
        bits 1^{st}, 2^{nd}, 3^{rd} *and/or* 5^{th} $\}$
  $\}\}$

  This step produces an intermediate stego grid $S'(u,v)$ that conceals the message bits. It should be noted that using the low frequency region of the DCT coefficients to conceal message bits is to limit the degrading of the visual quality of the produced stego image after applying the next steps (i.e., Step 3 and 4).

- Step 3: Apply the IDCT

  Now, we apply Equation 2 (i.e., the inverse of DCT or IDCT) on the stego grid $S'(u,v)$ generated by Step 2. The result of this process will be the stego grid $F'(x,y)$ of the stego image bytes.

- Step 4: Construct the stego image

  Finally, the stego image is constructed by replacing each original image grid $F(x,y)$ by the proper stego grid $F'(x,y)$.

It should be noted that the produced variation/differnce in a DCT coefficient by Step 2 can be from 0 to ± 8 units. This is depending on the message bit, the replaced bit, and the values of the 5 lest significant bits of a coefficient. For example, consider a coefficient with least significant 5 bits equal to $(\ldots 00001)_2$ and a bit message equals to "1". After applying Step 2, the value becomes $(\ldots 01000)_2$. It is clear that the resulting difference/variation is 7 units.

Table 1 shows more examples showing the least significant bytes of the original binary values of DCT coefficients before the embedding process (pointed by the letter "A"), and the associated values after insertion a message bit and the best modification on the remainder bits to achieve minimum difference (pointed by the letter "B"). It is clear that the produced difference as a result of embedding message bits "0", "1", and "0" in the 4^{th} bit of the coefficients 120, 83, and 208, respectively, are 1, 4, and 8 units.

Table 1. A ⇒ Original value, B ⇒ Best modification of the reminder bits.

		Binary Value							
	Decimal	8	7	6	5	**4**	3	2	1
A ⇒	120	0	1	1	1	1	0	0	0
B ⇒	119	0	1	1	**1**	**0**	**1**	**1**	**1**
A ⇒	83	0	1	0	1	0	0	1	1
B ⇒	79	0	1	0	**0**	**1**	**1**	**1**	**1**
A ⇒	208	1	1	1	0	0	0	0	0
B ⇒	216	1	1	1	**1**	**0**	**0**	**0**	**0**

Unlike the SSB-4 technique, this change will have no significant effect on the stego grid of the image bytes $F'(x,y)$ generated in step 3, because it will be spread across the entire image grids. To reduce this change (e.g., to ± 4), as an improvement to the proposed method, we have to avoid using the coefficients with specific values to hide the message bits. Consequently, such improvement reduces the amount of information that can be embedded in an image. Furthermore, an extra process and memory should be utilized to keep track of the used coefficients.

4. Experimental Results

Since the visual detection of stego images is depending on the nature of the image [4], variety of image categories were utilized in the experiments. In order to have significant results in the assays, we have divided the images to 5 categories relevant to the human perception; trees, flowers, mountains, people, and buildings image category. In other words, the images were classified depending on the existence of areas with contiguous homogenous/same luminosity and grayscale level. For example, people and building images have such areas more than trees. An appropriate collection of 75 BMP 24-bits grayscale images, 15 images for each category (downloaded from websites similar to webshots.com), were used in the experiments. Several randomly selected messages with different short lengths were utilized to be covert and retrieved. We focused on short messages because they are the most challenging to detect [4]. In addition to the proposed steganography technique, for comparison purposes, we utilized two other well-known techniques: the LSB and the LSB with DCT.

In order to evaluate the quality of the stego images generated by the compared techniques, and apart from the expensive and inaccurate subjective tests/judgments which are based on human perception (such as the Subjective test), instead we utilized – the commonly used metric– Peak Signal to Noise Ratio PSNR. Therefore, in this paper the stego image qualities are represented by PSNR (Equation 4, which is calculated from the root mean squared error MSE Equation 3), introduces in Section 2.3.

The implementations of the compared techniques (LSB, DCT with LSB, and the proposed one) and the PSNR tests were carried out using C# programming language on a PC running on MS Windows XP.

4.1. Comparison with other methods

The obtained results of the experiments are summarized in the following table and figure by means of the average PSNR in dB (decibel) values, of the 75 images of the 5 categories. Recall that the tested techniques are the proposed technique (Bit-4 with DCT), LSB, and LSB with DCT.

Table 2 shows some of the obtained results: the average PSNR of the different image categories (trees, flowers, mountains, people, and building) that conceal a messages of 40 bytes. However, the table shows more precisely the decreasing of the PSNRs of stego images as the size of the embedded message increases. From the tables we can see that all of the tested techniques produce acceptable reconstruction of the covering image i.e., greater than 30 dB.

Table 2. Average PSNR with 40 bytes message

	Average PSNR		
Image Type	LSB	LSB&DCT	Bit-4&DCT
Trees	40.344	50.991	48.596
Flowers	44.071	50.863	47.878
Mountains	40.673	50.597	47.758
People	42.696	50.836	45.311
Buildings	39.097	51.045	43.360

The results are clearly depicted in Figure 1, where the x-axis represents the categories of the used images by the three tested techniques while the y-axis represents the obtained average PSNRs in dB. Since the higher the SPNR value the better the quality of the stego image, it is clear that the LSB with DCT is the best amongst the tested techniques by means of the PSNR (quality of stego image). However, it is more vulnerable to the loss of information in lossy compression processes and easy to attack by steganalysis techniques. We can see that this technique is almost insensitive to the image category. In addition to its sensitivity to the compression processes by means of the loss of information and the easiest to attack by steganalysis, the LSB technique has the lowest PSNR values. Furthermore, we can see that the stego image produced by LSB technique is sensitive to the covert image category.

Results show that the proposed technique produces almost closed stego image quality to LSB with DCT technique when applied on images with higher frequency changing colors (trees, flowers, and buildings). In general our technique performs better than LSB and slightly lower than LSB with DCT. On the other hand, we can consider the proposed technique the best amongst the tested approaches by means of its robustness against stegalysis attacks and lossy compression methods that may ignore lower bits. This property is due to the utilization of a higher significant bit and the distribution of the effect of the embedded information in the frequency domain across the stego image grid by applying the inverse DCT process as stated in step 3 of the described algorithm.

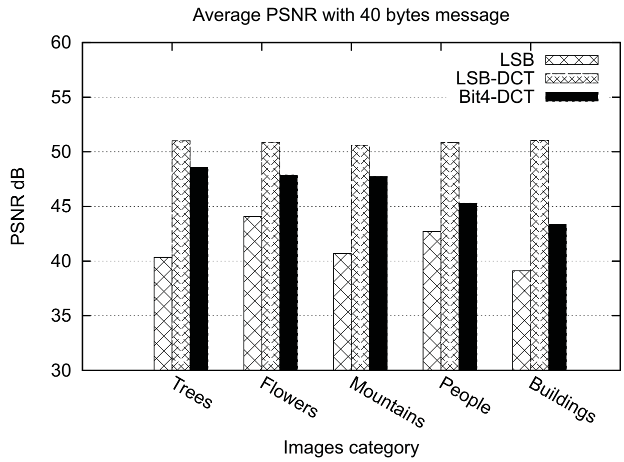

Figure 1. Average PSNR with 40 bytes message.

As a visual example, Figure 2 shows the original well-known image (Lena) and its stego images generated by the various tested techniques with its PSNR values (LSB, LSB with DCT, and the proposed technique Bit-4 of DCT).

5. Conclusion

In this research we proposed a mixed approach that applies the spatial domain with the frequency domain steganography techniques. The idea is to utilize a significant bit (4^{th} bit) of the DCT coefficients of a cover image to hide message bits. Thereafter, the information and the variation of the coefficients, affected by the embedding process, are spread in the stego image by utilizing the inverse of the DCT process. The obtained experimental re-

Original image

After insertion using LSB
(PSNR = 41.526 dB)

After insertion using
LSB with DCT
(PSNR = 49.935 dB)

Using the proposed
method Bit-4 with DCT
(PSNR = 45.312 dB)

Figure 2. Original Lena Image, stego images using LSB, LSB with DCT, and the proposed scheme Bit-4 of DCT steganography.

sults indicate that, the proposed method will be a good and acceptable steganogaphy scheme. Furthermore, by imbedding information in the main significant bits of the DCT domain, the hidden message resides in more robust areas, spread across the entire stego image, and provides better resistance against stiganalysis process than other techniques. However, there is more work to do. Therefore, our future work will focus on the improvement and further development this technique.

References

[1] Ismail Avcibas N.M. and B. Sankur, "Steganalysis using image quality metrics", In *IEEE Transactions on Image Processing*, vol. 12, No. 2., February 2003.

[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for Data Hiding", *I.B.M. Systems Journal*, 35(3-4): pages 313-336, 1996.

[3] C. C. Chang, T. S. Chen, and L. Z. Chung, "A steganographic method based upon JPEG and quantization table modification" *Information Sciences*, Volume 141, Issues 1-2, Pages 123-138, March 2002.

[4] L. Davidson, and P. Goutam, "Locating secret message in images", In ACM SIGKDD international conference on Knowledge discovery and data mining, (Seattle, Washington, Aug. 22-25). ACM 1-58113-888-1, 2004.

[5] S. Dickman, An Overview of Steganography, Research Report JMU-INFOSEC-TR -2007-002, James Madison University, July, 2007.

[6] J. Fridrich, and M. Goljan, "Practical steganalysis: stateofthe-art", In Proceeding of SPIE Photonics West, Electronic Imaging 2002, volume 4675, pages 1-13, 2002.

[7] J. Fridrich, M. Goljan, Steganalysis of JPEG Images: Breaking the F5 Algorithm, Publisher: *Springer Berlin, Heidelberg,* Lecture Notes in Computer Science, pp 310-323, Vol. 2578/ 2003.

[8] International Telecommunication Union, "Information Technology - Digital Compression and Coding of Continuous-Tone Still Images - Requirements and Specifications Recommendation T.81", *ITU* Sept., 1992

[9] N. F. Johnson, S. Jajodia, "Exploring Steganography: Seeing the Unseen," *IEEE Computer*, 31(2): 26-34 (1998) available at: http://www.jjtc.com

[10] M. Kharrazi06, H. Sencar, and N. Memon, "Performance study of common image steganography and steganalysis techniques," *Communications of the SPIE and IS&T*, 15, No.4, 1017-9909, Oct-Dec., 2006.

[11] R. Krenn, "Steganography and Steganalysis," 2004, available at: http://www.krenn.nl/univ/cry/steg

[12] J. Lenti, "Steganographic methods," periodic polytechnica ser.el.eng 44, No.3-4 (Jun 2002), pp 249-258.

[13] T. Morkel, J. Eloff, and M. Olivier, "An overview of image steganography", In Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005), (Sandton, South Africa, Jun/Jul. 2005).

[14] N. Nikolaidis, and I. Pitas, "Robust Image Watermarking in the Spatial Domain", *Signal Processing*, 66(3):385-403, 1998.

[15] M. Owens, "A Discussion of Covert Channels and Steganography", SANS [Online] (2002, March 19), Available at: http://www.sans.org/rr/whitepapers/covert/677.php

[16] G. Pavlidis, A. Tsompanopoulos, N. Papamarkos, and C. Chamzas, "JPEG2000 over noisy communication channels thorough evaluation and cost analysis", *Signal Processing: Image Communication*, Volume 18, Issue 6, July 2003, Pages 497-514,

[17] N. Provos, and P. Honeyman, "Hide and Seek: An introduction to Steganography", *Security & Privacy, IEEE*, Vol.1, Issue 3, pp 32-44, May-June 2003.

[18] J. Rodrigues, J. Rios, and W. Puech "SSB-4 System of Steganography using bit 4", In International Workshop on Image Analysis for Multimedia WIAMIS, (Montreux, May 2005).

[19] H. Wang, and S. Wang, "Cyber warfare: Steganography vs. Steganalysis", *Communications of the ACM* 47, No.10, pages 76-82, Oct., 2004.

[20] M. A. Bani Younes, A. Jantan, "A New Steganography Approach for Image Encryption Exchange by Using the Least Significant Bit Insertion", *IJCSNS, International Journal of Computer Science and Network Security*, Vol. 8 No. 6, June 2008.