ISSN: 1991-8941

A Coding-Based Steganography Using Multiple Frequency Domains

Ahmed Tariq Sadiq* , Raoof Smko Assef, Omed Salim Khalind****
*** Computer Sciences Department, University of Technology, Baghdad**
**** Software Engineering Department, Salahddin University, Erbil**

Received:22/5/2008 Accepted:16/12/2008

Abstract:In this paper, a new technique for hiding text in a bitmap images will be present. The technique based on using an index of the dictionary representing the characters of the secret messages instead of the characters themselves. The technique uses multiple frequency domains for embedding these indexes in an arbitrary chosen bitmap image. By using discrete cosine transform DCT, discrete wavelet transform DWT, and a combination of both of them. A software package for implementing this technique are built and we got very good results in terms of capacity of hiding, imperceptibility which are the most two important properties of steganography, the time of hiding the text, and the security issues.

Key words: Steganography, JPEG, GIF, BMP, DCT, DWT, YAD, MSE, PSNR

Introduction

Steganography is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. The word steganography is derived from the Greek words "stegos" meaning "cover" and "grafia" meaning "writing" [1] defining it as "covered writing". In image steganography the information is hidden exclusively in images.

Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret [4]. Steganography and cryptography are both ways to protect information from unwanted parties but neither technology alone is perfect and can be compromised. Once the presence of hidden information is revealed or even suspected, the purpose of steganography is partly defeated [2]. The strength of steganography can thus be amplified by combining it with cryptography.

Two other technologies that are closely related to steganography are watermarking and fingerprinting [3]. These technologies are mainly concerned with the protection of intellectual property, thus the algorithms have different requirements than steganography.

In watermarking all of the instances of an object are "marked" in the same way. The kind of information hidden in objects when using watermarking is usually a signature to signify origin or ownership for the purpose of copyright protection [4]. With fingerprinting on the other hand, different, unique marks are embedded in distinct copies of the carrier object that are supplied to different customers. This enables the intellectual property owner to identify customers who break their licensing agreement by supplying the property to third parties [3].

This paper intends to offer a new approach for image Steganography using multiple frequency domain with dictionary based text embedding. The remainder of the paper is structured as follows: Section 2 gives the reader an overview of image steganography in general and different kinds of image steganography. In section 3 the new approach for image steganography are discussed. In section 4 the software package used for implementing the new approach is explained and the results discussed. In Section 5 a conclusions are reached were discussed.

Image steganography

There are many file types used as a cover for text steganography. Images are the most widely used in steganography techniques, because of the chances to get more free spaces for embedding texts, without changing the overall properties of the cover image.

Image Definition

To a computer, an image is a collection of numbers that constitute different light intensities in different areas of the image [5]. This numeric representation forms a grid and the individual points are referred to as pixels. Most images on the Internet consists of a rectangular map of the image's pixels (represented as bits) where each pixel is located and its color [6]. These pixels are displayed horizontally row by row.

The number of bits in a color scheme, called the bit depth, refers to the number of bits used for each pixel [7]. The smallest bit depth in current color schemes is 8, meaning that there are 8 bits used to describe the color of each pixel [7]. Monochrome and grayscale images use 8 bits for each pixel and are able to display 256 different colors or shades of grey. Digital color images are typically stored in 24-bit files and use the RGB color model, also known as true color [7]. All color variations for the pixels of a 24-bit image are derived from three primary colors: red, green and blue, and each primary colour is represented by 8 bits [5]. Thus in one given pixel, there can be 256 different quantities of red, green and blue, adding up to more than 16-million combinations, resulting in more than 16-million colors [16]. Not surprisingly the larger amount of colors that can be displayed, the larger the file size [6].

Image Compression

When working with larger images of greater bit depth, the images tend to become too large to transmit over a standard Internet connection. In order to display an image in a reasonable amount of time, techniques must be incorporated to reduce the image's file size. These techniques make use of mathematical formulas to analyze and condense image data, resulting in smaller file sizes. This process is called compression [6].

In images there are two types of compression: lossy and lossless [1]. Both methods save storage space, but the procedures that they implement differ. Lossy compression creates smaller files by discarding excess image data from the original image. It removes details that are too small for the human eye to differentiate [6], resulting in close approximations of the original image, although not an exact duplicate. An example of an image format that uses this compression technique is JPEG (Joint Photographic Experts Group) [5].

Lossless compression, on the other hand, never removes any information from the original image, but instead represents data in mathematical formulas [6]. The original image's integrity is maintained and the decompressed image output is bit-by-bit identical to the original image input [1]. The most popular image formats that use lossless compression is GIF (Graphical Interchange Format) and 8-bit BMP (a Microsoft Windows bitmap file) [5].

Compression plays a very important role in choosing which steganographic algorithm to use. Lossy compression techniques result in smaller image file sizes, but it increases the possibility that the embedded message may be partly lost due to the fact that excess image data will be removed [8]. Lossless compression though, keeps the original digital image intact without the chance of lost, although is does not compress the image to such a small file size [5]. Different steganographic algorithms have been developed for both of these compression types and will be explained in the following sections.

Image and Transform Domain

Image steganography techniques can be divided into two groups: those in the Image Domain and those in the Transform Domain [9]. Image – also known as spatial – domain techniques embed messages in the intensity of the pixels directly, while for transform – also known as frequency – domain, images are first transformed and then the message is embedded in the image [10].

Image domain techniques encompass bit-wise methods that apply bit insertion and noise manipulation and are sometimes characterized as "simple systems" [11]. The image formats that are most suitable for image domain steganography are lossless and the techniques are typically dependent on the image format [12].

Steganography in the transform domain involves the manipulation of algorithms and image transforms [11]. These methods hide messages in more significant areas of the cover image, making it more robust [2]. Many transform domain methods are independent of the image format and the embedded message may survive conversion between lossy and lossless compression [12]. Figure (1) below shows the kinds of image steganography.

Discrete Cosine Transform DCT

Transform coding constitutes an integral component of contemporary image/video processing applications. Transform coding relies on the premise that pixels in an image exhibit a certain level of correlation with their neighboring pixels. Similarly in a video transmission system, adjacent pixels in consecutive frames2 show very high correlation. Consequently, these correlations can be exploited to predict the value of a pixel from its respective neighbors. A transformation is, therefore, defined to map this spatial

(correlated) data into transformed (uncorrelated) coefficients. Clearly, the transformation should utilize the fact that the information content of an individual pixel is relatively small i.e., to a large extent visual contribution of a pixel can be predicted using its neighbors [13].

Like other transforms, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded ndependently without losing compression efficiency. This section describes the DCT and some of its important properties.

The most common DCT definition of a 1-D sequence of length N is

$$c(u) = a(u)\sum_{x=0}^{n-1} f(x)\cos\left[\frac{p(2x+1)u}{2N}\right] \qquad (1)$$

for u= 0,1,2, , N-1 . Similarly, the inverse transformation is defined as

$$f(x) = \sum_{u=0}^{N-1} a(u)C(u)\cos\left[\frac{p(2x+1)u}{2N}\right] \qquad (2)$$

for x = 0,1,2, ,N-1

In both equations (1), and (2), we have:

$$a(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{For u = 0} \\ \sqrt{\frac{2}{N}} & \text{For u} \neq 0 \end{cases} \qquad (3)$$

It is clear from (1) that for u = 0, $C(u=0) = \sqrt{\frac{1}{N}}\sum_{x=0}^{N-1} f(x)$.

Thus, the first transform coefficient is the average value of the sample sequence. In literature, this value is referred to as the *DC Coefficient*. All other transform coefficients are called the *AC Coefficients.*

Varying values of *u* is shown in Figure 2. In accordance with our previous observation, the first the top-left waveform (u = 0) renders a constant (DC) value, whereas, all other waveforms (u = 1, 2, …,7) give waveforms at progressively increasing frequencies [14].

These waveforms are called the *cosine basis function*. Note that these basis functions are orthogonal. Hence, multiplication of any waveform in Figure 2 with another waveform followed by a summation over all sample points yields a zero (scalar) value, whereas multiplication of any waveform in Figure 2 with itself followed by a summation yields a constant (scalar) value.

Orthogonal waveforms are independent, that is, none of the basis functions can be represented as a combination of other basis functions [15].

If the input sequence has more than *N* sample points then it can be divided into sub-sequences of length *N* and DCT can be applied to these chunks independently. Here, a very important point to note is that in each such computation the values of the basis function points will not change. Only the values of *f(x)* will change in each sub-sequence. This is a very important property, since it shows that the basis functions can be pre-computed offline and then multiplied with the sub-sequences. This reduces the number of mathematical operations (i.e.,) multiplications and additions) thereby rendering computation efficiency [13].

When using DCT with images, we need to use two dimensions DCT, 2-D DCT is a direct extension of the 1-D DCT and is given by:

$$C(u,v) = a(u)a(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{p(2x+1)u}{2N}\right]\cos\left[\frac{p(2y+1)v}{2N}\right] \qquad (4)$$

For *u, v* = 0, 1, 2, …, *N-1*, and α*(u)* and α*(v)* are defined in (3). The inverse transform is defined as:

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} a(u)a(v)C(u,v)\cos\left[\frac{p(2x+1)u}{2N}\right]\cos\left[\frac{p(2y+1)v}{2N}\right] \qquad (5)$$

For x, y = 0, 1, 2, …, N-1. The 2-D basis functions can be generated by multiplying the horizontally oriented 1-D basis functions with vertically oriented set of the same functions [14]. It can be noted that the basis functions exhibit a progressive increase in frequency both in the vertical and horizontal direction.

The New Approach

In this paper, we used a new technique based on working with Bitmap images (BMP); we used

various transform techniques in this paper to be more efficient and have more valuable results.
The steps of the approach used for embedding the text in the cover image to obtain the stego image, is shown in figure (3). These steps are programmed in efficient way in the proposed system, which is designed specially for this new approach.
The contributions reached in this technique can be classified in different viewpoints. We shall discuss them in the next articles.

Transfer domain

Two different types of transferring techniques are used in the approach, which they are as following:

Discrete Cosine Transform:

The embedding process begins with separating the (RGB) component of the cover image and putting them in a 3D array, after this we divide it into (8x8) regions (blocks), then we take the DCT of each block, by this the image is transformed to frequency domain and now we have an array contains all (8x8) blocks of DCT coefficients. Now we embed each bit of the secret message in one (8x8) block by selecting two points with high frequencies (to make the change be more imperceptible; which is the most important requirement of any steganographic systems). If the secret message bit is '1'; we should make (p1>p2) by swapping them if they not, and if the secret message bit is '0'; we should make (p1<p2), and setting the difference between the two selected points to 4, to make the secret message recoverable without error (or Bit Error Rate, BER = 0) as it is important for extraction process, because we have looses in transformation and rounding process. After embedding process we take the Inverse DCT and recreating image from the 3D array (RGB).
This process is done on the layers (B, R, and G) respectively because they are not equally affecting the luminosity histogram, as shown below:

- B 11%
- R 30%
- G 59%

And this order is useful when we have a secret message less than the maximum capacity of the image (reduces the effect). This can serve us when attackers want to analyze the luminosity histogram between the cover and stego image.
And the extraction process is done in the same way, but by comparing the two selected points. If (p1>p2) then the extracted bit has the value of '1', and when (p1<p2) then the extracted bit will be '0'.

Discrete Wavelet Transform with DCT:

Here for both Haar and Daubechies-4, after separating (RGB) and saving them in 3D array, we take the DWT of the 3D array and we pass the selected quarters (HH, HL, LH, HH+HL, HH+LH, or HH+HL+LH; but not LL because it affects the image significantly) to the DCT, by this we applied multiple or hybrid frequency domain, and embedding the secret message with the same steps we have previously talked about. Except in haar wavelet we did it with a small change; setting the difference between the selected two points to 9, this is because we have extra looses with haar wavelet, but for Daubechies-4 remains the same 4.

After taking the IDCT we take the IDWT too for recreating the image.

The Security in the new approach

In the security point of view, we used what we called it a Stego Key. Here we add a facility of having a Stego Key (or Password), which makes the secret message to be password protected, and for extracting the secret message from stego image the correct password should be entered otherwise no message will be recovered.

Also we encrypted the data before hiding it in the cover image; a special dictionary used for encoding the secret message characters by getting the address of the character in the dictionary and not its well known character codes, which is used in both embedding and extraction process that makes the secret message extraction impossible by Stegano analysis even if the embedding process is known.
Other techniques are used for efficiency, like Data compression; such that before the encoding process, the secret message goes through a white space remover process (for example removing extra contiguous repeated spaces, tabs, and end lines) and also goes through a filter process that transform all characters to lower case (this is useful for saving space in the dictionary too by having just lower case characters there).
Payload encoding; to make our extraction process perfect and well retrieval of the secret message, the payload header is 99-bit length that holds these information as shown in figure (4).

The characters encoded in a way such that each character in the secret message with 6-bits instead of 8-bits, for improving the capacity, because steganography in frequency domain suffers from having a low capacity, especially hybrid method (DWT+DCT).

Experiment Results

The experiments done on Lena image, the image is colored bitmap 512x512 pixel, and the size of 768 KB. The Results of the experiment

included in a comparison table as shown in table (1). The comparison is used for comparing different techniques used in the experiments, in terms of the quarter used in the transformation, the maximum capacity (measured in characters), the secret message size (measured in characters), the time taken in embedding the message in the cover image (the time is measured according to the specification of the computer used in the running of the program), and finally the number of bits that changed comparing the cover image (before embedding) and stego image (after embedding). The experiments implemented using the package have been developed specially for the new approach used in steganography. The proposed system developed using Microsoft J# 2005 Express Edition, which it produce more flexible for this work in the designing a GUI for the application, and it is fast enough since most of the image processing program needs a long time for execution.

All the experiments executed using PC with 2.0 GHz CPU speed, 1 GByte RAM, 120 GByte HDD, and 512 MByte graphic card, so all the results that take the time in account are based on these specifications.

Figure (5) shows the graph representing the maximum capacity (measured by characters) for the techniques used in the approach. The frequency domain steganography suffers from having a low capacity; for example in DCT we can embed one bit in 64-pixels (8x8), but there is some improvement in our approach which is the use of 6-bits for each character instead of 8-bit, and embedding the message in three layers (RGB).

The relation between used technique and embedding duration time is shown in Figure (6); regarding the capacity parameter, the "DCT only" technique has the minimum embedding duration time, because other techniques are used with DCT that means their time is added with DCT time. And for DWT+DCT, in general it increases with the number of selected quarters like; one quarter < two quarters < three quarters for both Haar and Daub-4. But Daub-4 is faster than Haar, has less duration time, when used with DCT.

By taking the best values Figure (7) shows that, the DWT (Daub-4) + DCT has the highest imperceptibility (highest PSNR) among all techniques in average regardless to capacity (because DCT has higher capacity than other technique), then DWT (Haar) + DCT technique comes, and "DCT only" has the minimum imperceptibility (lowest PSNR).

In general DWT (Daub-4) + DCT has higher PSNR than the DWT (Haar) + DCT, but PSNR for both DWT + DCT techniques (Haar and Daub-4) depends on the selection of quarters (filter bands) as it is shown in Figure (8) and Figure (9); for example HH filter band has the highest PSNR and HH+HL+LH has the lowest PSNR. The sequence is like the following:

HH < LH < HL < HH+LH < HH+HL < LH+HL < HH+HL+LH

The embedding duration time is directly proportional with the capacity; the higher capacity cover image has the higher embedding duration time (if we fill it with characters). And if we don't take the worst case, then the larger secret message has the larger embedding duration time as shown in Figure (10).

The PSNR is adversely proportional with the capacity; the higher capacity cover image has the lower PSNR (if we fill it with characters). And if we don't take the worst case, then the larger secret message has the lower PSNR as shown in Figure (11).

The extraction duration time of the secret message in a stego image is nearly equal to the embedding duration time, because they do the same operations but in the reverse order. Here the extraction duration time is not mentioned, but you can think of it as embedding duration time for all; its relation with capacity, used technique, and selected quarter(s).

Conclusions

In all techniques and quarter selections the PSNR=121 as an average; which shows that there is no distortion in the image at all but we measured a bit difference as an extra parameter for this and by experiment and recording various results we saw that the best technique is Daubechies-4 with DCT, then Haar with DCT, and then DCT only. And best quarter for embedding in DWT+DCT is (HH)..

There is no noticeable change in the luminosity histogram between cover and stego image; which prevents the stego image from attacks, when attackers want to see the change in luminosity histogram.

For duration time of embedding; best technique is Daubechies-4 with DCT, then DCT, and then Haar with DCT.

The embedded secret message is 100% recoverable without error (BER=0), which is a good result and great challenge in frequency domain that holds many looses in transform and rounding values during both embedding and extraction processes.

The capacity is improved by 1.333, by encoding each character of the secret message with 6-bits. And also improving it by 3 due to embedding message in all layers (RGB).

Since Haar wavelet depends on the differencing and averaging for high and low pass filters respectively it is simpler than Daubechies-

4 to deal with, but it has more looses in transform process. While Daubechies-4 is more complex and difficult, but nearly it has no loose in transform process.

References

[1] Moerland, T., "**Steganography and Steganalysis**", *Leiden Institute of Advanced Computing Science*, www.liacs.nl/home/ tmoerl/privtech.pdf.

[2] Wang, H & Wang, S, "**Cyber warfare: Steganography vs. Steganalysis**", *Communications of the ACM*, 47:10, October 2004.

[3] Anderson, R.J. & Petitcolas, F.A.P., "**On the limits of steganography**", *IEEE Journal of selected Areasin Communications*, May 1998.

[4] Marvel, L.M., Boncelet Jr., C.G. & Retter, C., "**Spread Spectrum Steganography**", *IEEE Transactions on image processing, 8:08*, 1999.

[5] Johnson, N.F. & Jajodia, S., "**Exploring Steganography: Seeing the Unseen**", *Computer Journal*, February 1998.

[6] "**Reference guide: Graphics Technical Options and Decisions**", http://www.devx.com/projectcool/Article/19997.

[7] Owens, M., "**A discussion of covert channels and steganography**", *SANS Institute*, 2002.

[8] Dunbar, B., "**Steganographic techniques and their use in an Open-Systems environment**", *SANS, Institute*, January 2002.

[9] Silman, J., "**Steganography and Steganalysis: An Overview**", *SANS Institute*, 2001.

[10] Lee, Y.K. & Chen, L.H., "**High capacity image steganographic model**", *Visual Image Signal Processing*, 147:03, June 2000.

[11] Johnson, N.F. & Jajodia, S., "**Steganalysis of Images Created Using Current Steganography Software**", *Proceedings of the 2nd Information Hiding Workshop*, April 1998.

[12] Venkatraman, S., Abraham, A. & Paprzycki, M., "**Significance of Steganography on Data Security**", *Proceedings of the International Conference on Information Technology: Coding and Computing*, 2004.

[13] Syed Ali Khayam "**The Discrete Cosine Transform (DCT): Theory and Application**", *ECE 802 – 602:* 2003.

[14] W. B. Pennebaker and J. L. Mitchell, "**JPEG – Still Image Data Compression Standard**", Newyork: International Thomsan Publishing, 1993.

[15] G. Strang, "**The Discrete Cosine Transform**", *SIAM Review*, Volume 41, Number 1, pp. 135-147, 1999.

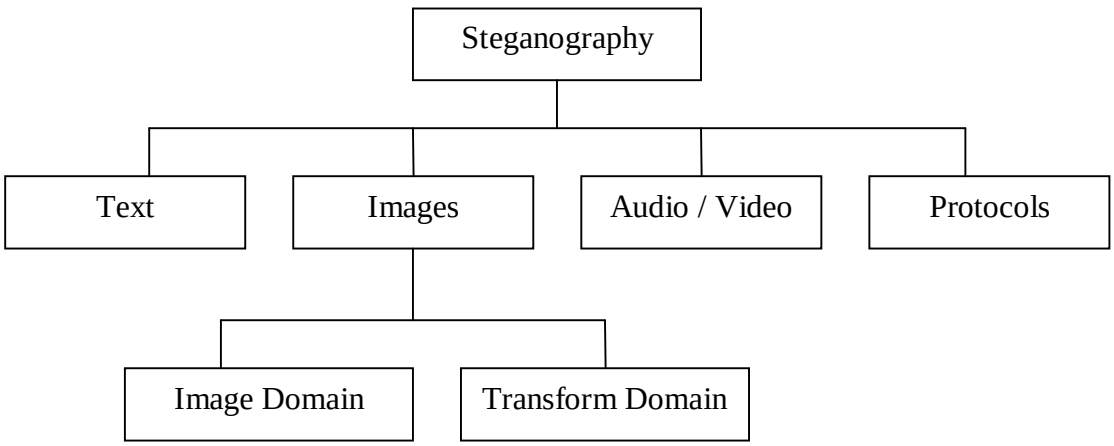

Figure (1) Categories of Image Steganography

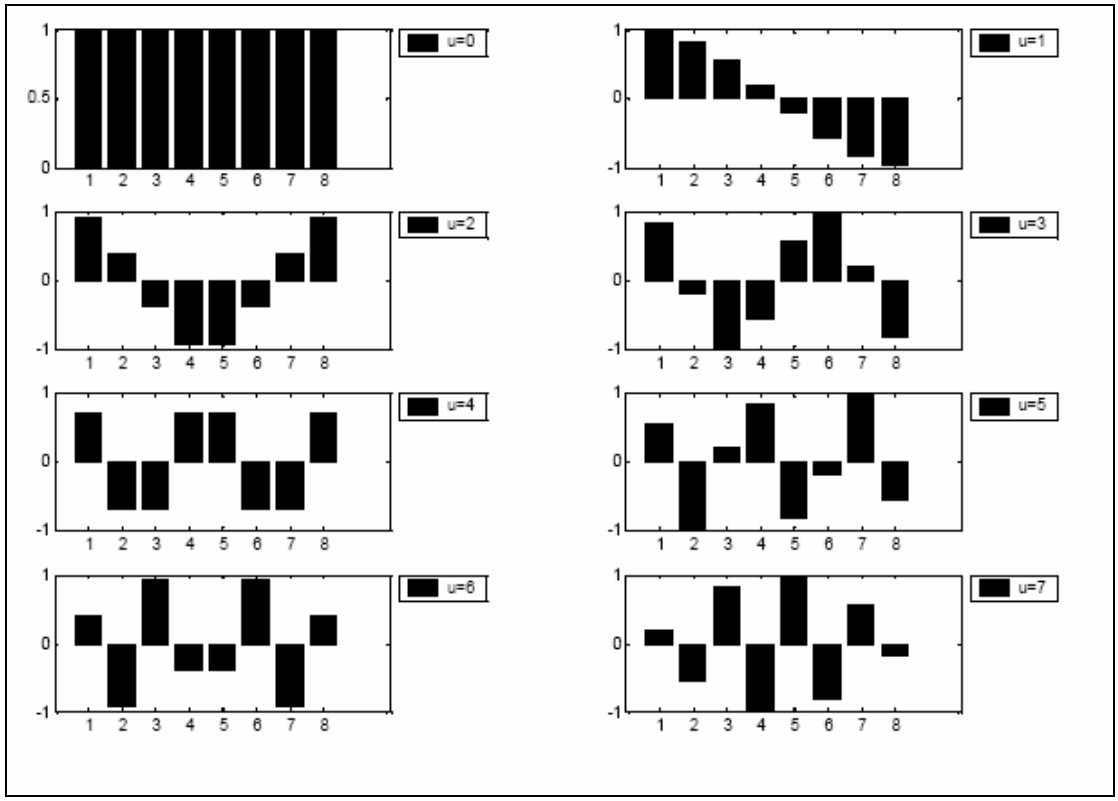

Figure (2) One dimensional cosine basis function (*N*=8)

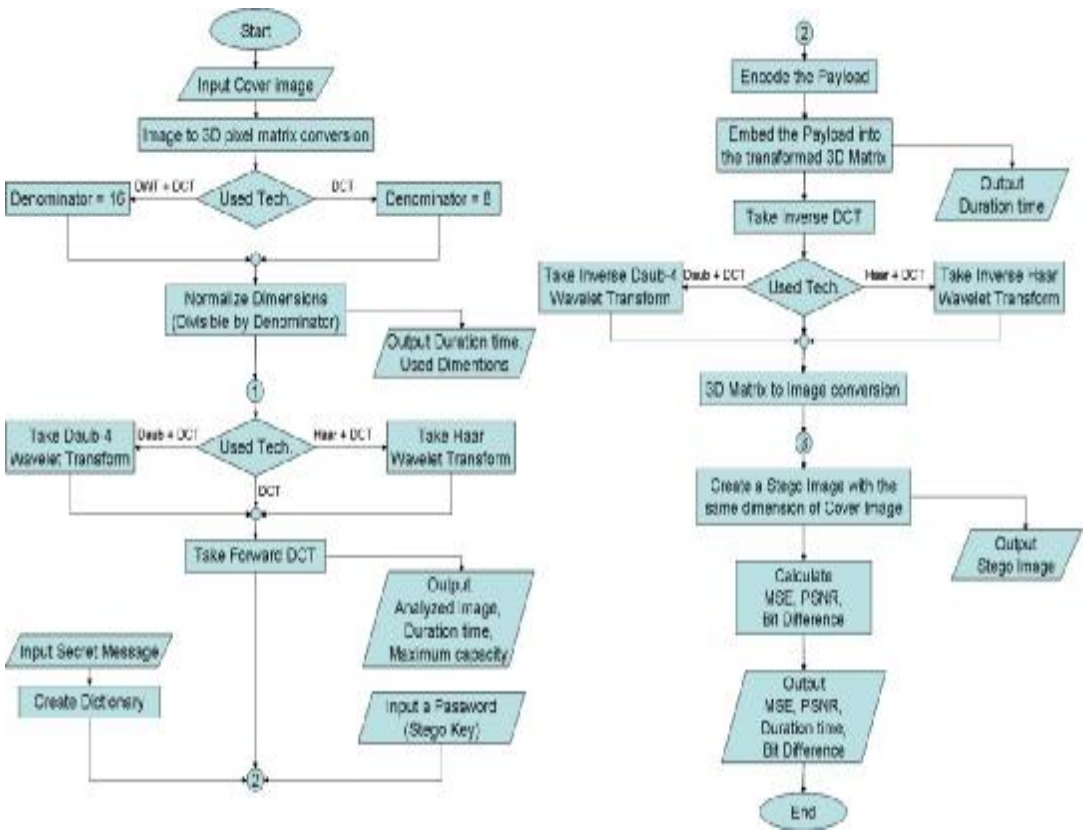

Figure (3) The steps in the embedding stage

ID	Message length	Password	Used Tech	Quarter	Message
42-bit	16-bit	36-bit	2-bit	3-bit	Message length / 6 - bits

Figure (4) The format of Payload Encoding

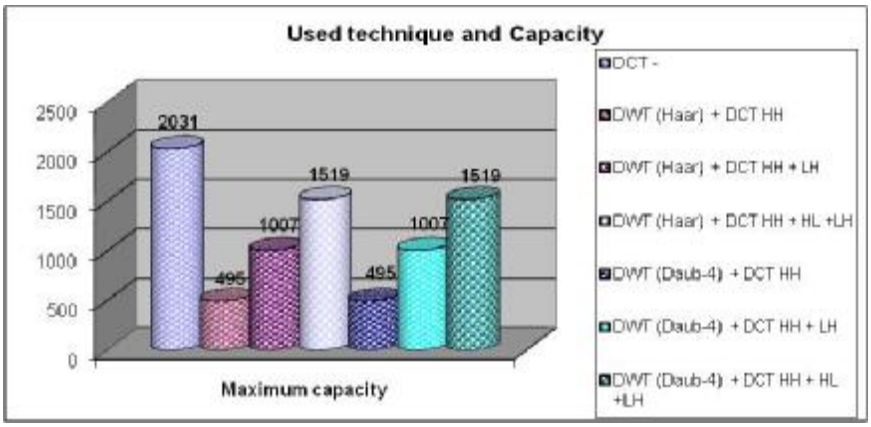

Figure (5) The maximum capacity in characters for the techniques used

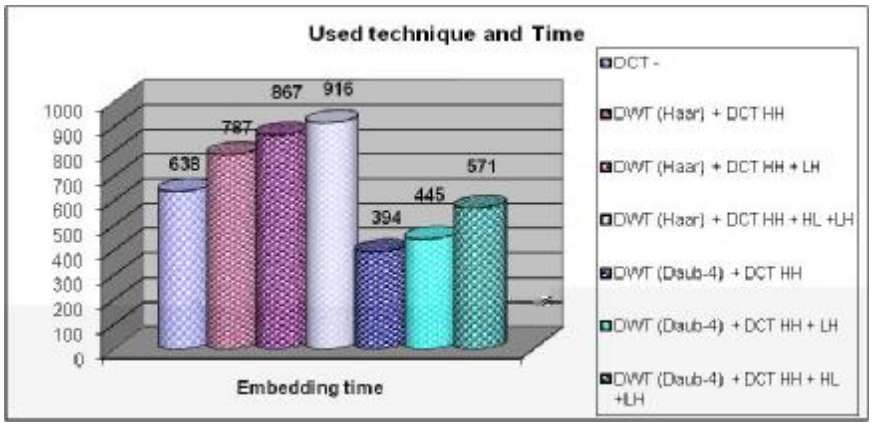

Figure (6) Techniques and duration time chart

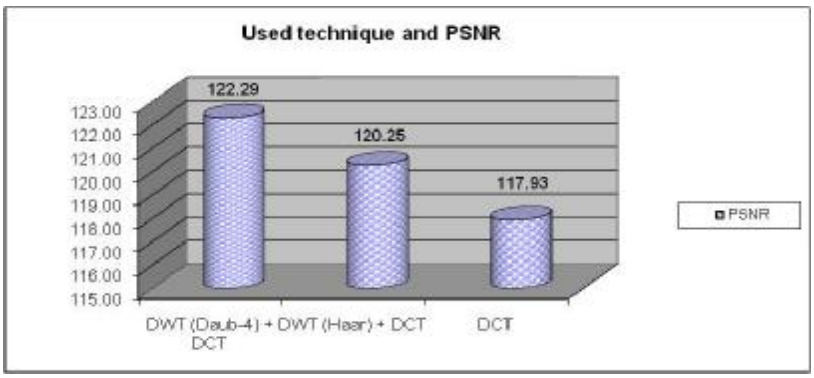

Figure 7 Techniques and PSNR chart

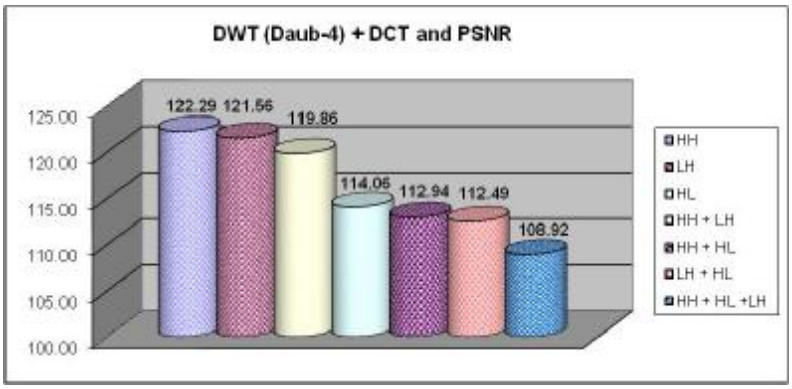

Figure 8 DWT (Daub-4) + DCT technique quarters and PSNR chart

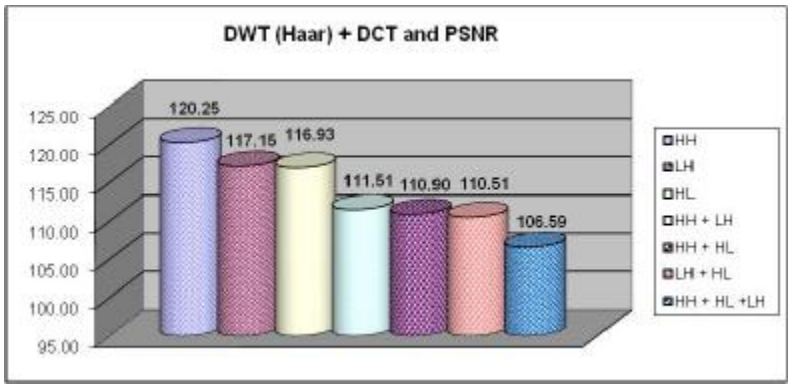

Figure 9 DWT (Haar) + DCT technique quarters and PSNR chart

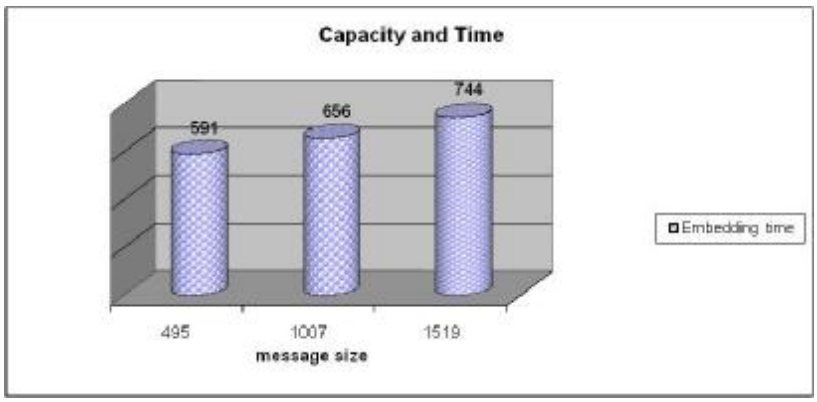

Figure 10 Message size and embedding duration time chart

الإخفاء المعتمد على الترميز باستخدام التعددية في المجال الترددي

أحمد طارق صادق رؤوف سمكو آصف أوميد سليم خلند

E.mail: scianb@yahoo.com

الخلاصة

قي هذا البحث سيتم تقديم تقنية جديدة لاخفاء النص قي صور من نوع Bitmap . التقنية تعتمد على استخدام قهرس القاموس لتمثيل الاحرف في الرسائل المشفرة بدلا من الاحرف نفسها. وهذه التقنية تستخدم التعددية في المجال الترددي لتضمين هذه الفهرسة في الصورة المختارة. باستخدام تحويل الجيب تمام المتقطع و تحويل الدالة المويجية المتقطع وكذلك الدمج بينهما. تم تنفيذ حقيبة نظام برمجي سميت (YAD Stego) وتم الحصول على نتائج جيدة جدا وخصوصا فيما يتعلق بالمساحة المتوفرة للاخفاء والغموض اللتان تعتبران اهم خواص علم الاخفاء وكذلك تم الاخذ بنظر الاعتبار عاملا الوقت والامنية.