

# SM10

## Text Hiding Using Discrete Cosine Transformation (DCT)

Razis bin Shamsudin<sup>1</sup>, Nik Shahidah Afifi binti Md Taujuddin<sup>2</sup>

Faculty of Electric and Electronic Engineering,  
Universiti Tun Hussein Onn Malaysia  
Email : razis.shamsudin@yahoo.com<sup>1</sup>, shahidah@uthm.edu.my<sup>2</sup>

**Abstract**-Steganography comes from the Greek words steganos, roughly translating as “covered writing”. Steganographic techniques allow one party to communicate information to another without a third party even knowing that the communication is occurred. The ways to deliver these “secret messages” is vary greatly. “The goal of steganography is to hide messages inside other harmless messages in a way that it does not allow any enemy to even detect that there is a second secret message present.” Steganography is a process that involves hiding a message in an appropriate carrier for example an image or an audio file. The carrier can then be sent to a receiver without anyone else knowing that it contains a hidden message. There are various technique being used in hiding the message using steganography practice, namely digital watermarking, least significant bit (LSB), fractal codes, masking and filtering, algorithms and transformations, Discrete wavelet transformation, Discrete Cosine Transformation and many more. This project will implement Discrete Cosine Transformation (DCT) as steganography technique in hiding text into an image. The process start with divides up the image into 8 by 8 pixel blocks, and then calculates the discrete cosine transform (DCT) of each block. A quantizer rounds off the DCT coefficients according to the quantization matrix. This step produces the “lossy” nature of Joint Photography Expert Group (JPEG) , but allows for large compression ratios. JPEG's compression technique uses a variable length code on these coefficients, and then writes the compressed data stream to an output file (.jpg). The purpose of this project is to develop a prototype software system that can hide a text into an image using Discrete Cosine Transformation (DCT) technique. It means that various algorithms and transformations are applied on the image to hide the text or messages. The system is build by using Matlab 7.6.

Keywords: Discrete Cosine Tansformation (DCT), JPEG (Joint Photographic Expert Group), steganography, image compression.

## 1. Introduction

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from Greek, literally means “covered writing.” It includes a vast array of secret communications methods that conceal the message’s very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications. Steganography and cryptography are cousins in the spycraft family. Cryptography scrambles a message so it cannot be understood. Steganography hides the message so it cannot be seen. A message in cipher text, for instance, might arouse suspicion on the part of the recipient while an “invisible” message created with steganographic methods will not.

## 2. Literature Review

### A. History of Steganography

Throughout history, people have hidden information by a multitude of methods and variations (D. Kahn, 1967). For example, ancient Greeks wrote text on wax-covered tablets. To pass a hidden message, a person

would scrape wax off a tablet, write a message on the underlying wood and again cover the tablet with wax to make it appear blank and unused. Another ingenious method was to shave the head of a messenger and tattoo a message or image on the messenger's head. After the hair grew back, the message would be undetected until the head was shaved again (B. Norman, 1973).

Invisible inks offered a common form of invisible writing. Early in World War II, steganographic technology consisted almost exclusively of these inks (D. Kahn, 1967). With invisible ink, a seemingly innocent letter could contain a very different message written between the lines (H.S. Zim, 1948).

Document layout may also reveal information. Documents can be marked and identified by modulating the position of lines and words (J. Brassilet, 1995). Message detection improved with the development of new technologies that could pass more information and be even less conspicuous.

Microdots are photographs the size of a printed period having the clarity of standard-sized typewritten pages, which permits the transmission of large amounts of data, including drawings and photographs (D. Kahn, 1967).

### *B. Steganography Process*

The data to be concealed is compressed and hidden within another file. The first step is to find a file which will be used to hide the message (also called a carrier or a container.) The next step is to embed the message one wants to hide within the carrier using a steganographic technique. Two different techniques commonly used for embedding are replace the least significant bit of each byte in the [carrier] with a single bit for the hidden message and select certain bytes in which to embed the message using a random number generator, resampling the bytes to pixel mapping to preserve color scheme, in the case of an image, hiding information in the coefficients of the discrete cosine, fractal or wavelet transform of an image and applying mimic functions that adapt bit pattern to a given statistical distribution.

### *C. Steganography Tools*

S-Tools is one of the most popular steganography tools. This program is Windows 95/98 compatible. It has the ability to conceal files within BMP, GIF and WAV files. Allows you to simply point and click your way to hiding files. It also has the ability to hide multiple files in one container. It has been updated each year and can be easily download by anyone.

<http://members.tripod.com/steganography/stego/software.html>

Listed below is some of the steg software available:

- i. EZStego(StegoOnline,StegoShareware,RomanaMachado) – Java based software program which supports only GIF and PICT formats.
- ii. Gif-It-Up v1.0 founded by Lee Nelson – A stegoprogram for Windows 95 that hides data in GIF files.
- iii. Hide and Seek founded by Colin Maroney – can hide any data into GIF images.
- iv. JPEG-JSTEG founded by Derek Upham – can hide data inside a JPEG file
- v. MP3Stego founded by Fabien A.P. Petitcolas, Computer Laboratory, University of Cambridge – can hide data inside MP3 sound files; (<http://www.sans.org/rr/whitepapers/stenganography/550.php>)
- vi. Steganos founded by Demcom, Frankfurt, Germany – encrypts files and then hides them within BMP, DIB, VOC, WAV, ASCII and HTML files.

The website of this software is:

<http://members.tripod.com/steganography/stego/software.html>

### *D. Concealment in Digital Images*

Information can be hidden many different ways in images. To hide information, straight message insertion may encode every bit of information in the image or selectively embed the message in “noisy” areas that draw less attention those areas where there is a great deal of natural color variation. The message may also be scattered randomly throughout the image. Redundant pattern encoding “wallpapers” the cover image with the message. A number of ways exist to hide information in digital images (P. Wayner,1996). Common approaches include:

- i. least significant bit insertion (LSB).
- ii. masking and filtering.
- iii. algorithms and transformations.

Each of these techniques can be applied, with varying degrees of success, to different image files.

#### *i. Least Significant Bit Insertion*

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover file. Unfortunately, it is vulnerable to even a slight image manipulation. Converting an image from a format like GIF or BMP, which reconstructs the original message exactly (lossless compression) to a JPEG, and then back could destroy the information hidden in the LSBs. 24-bit images. To hide an image in the LSBs of each byte of a 24-bit image, you can store 3 bits in each pixel. A 1,024 x 768 image has the potential to hide a total of 2,359,296 bits (294,912 bytes) of information (Kessler, 2002).

#### *ii. Masking and Filtering*

Masking and filtering techniques, usually restricted to 24-bit and gray-scale images, hide information by marking an image, in a manner similar to paper watermarks. Watermarking techniques may be applied without fear of image destruction due to lossy compression because they are more integrated into the image. Visible watermarks are not steganography by definition.

#### *iii. Algorithms and transformations*

LSB manipulation is a quick and easy way to hide information but is vulnerable to small changes resulting from image processing or lossy compression. Such compression is a key advantage that JPEG images have over other formats. High color quality images can be stored in relatively small files using JPEG compression methods thus, JPEG images are becoming more abundant on the Internet.

### *E. Discrete Cosine Transformation (DCT)*

Like other transforms, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency.

### *F. Type of Image File*

#### *i. JPEG*

A JPEG (stands for Joint Photographic Experts Group) is a compressed file format. It's a file format designed to balance image quality with file size and it's usually the right choice of format for pictures on the Web. A JPEG's file size can be a tenth or less of the file size of the original image. Advantages of JPEG are, it produce a small image size, perfect for most images and uses millions of colors. While the disadvantages are it leads to high compression loses quality and every time a JPEG is saved, it loses more and more pixel of the picture

### *G. JPEG Image Compression*

#### *i. Lossless Image Compression*

Lossless Image Compression is a class of image compression algorithms that allows the exact original image to be reconstructed from the compressed image.

#### *ii. Lossy Image Compression*

Lossy encoding is based on the concept of compromising the accuracy of the reconstructed image in exchange for increased compression.

### 3. Methodology

Figure 4 shows the general lossless coding system that consist of encoder and decoder. The encoder takes as input an image and generates as a output as a compressed bit stream. Then the decoder will take the compress bit stream to be decode to perform the original image. The processes conducted in these coding system are transformation, data to symbol mapping and lossless symbol coding.

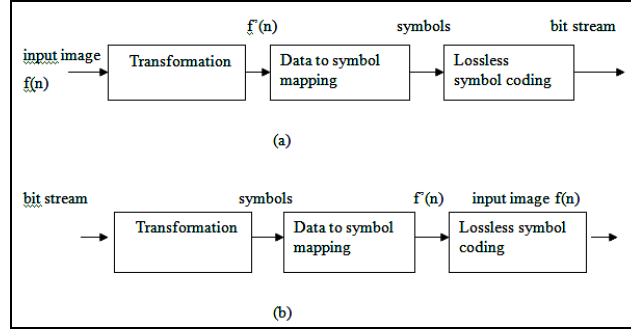


Figure 4: General lossless coding system: (a) encoder (b) decoder.

#### ii. Transformation

In this project, the steganography technique that will be use is transformations. DCT (Direct cosine transformation) is one such method. DCT is used by the JPEG compression algorithm to transform successive 8 x 8 pixel blocks of the image, into 64 DCT coefficients each. Steganography tools can use the LSB of the quantized DCT coefficient can be used to hide information. In addition to DCT, images can be processed with fast Fourier transformation and wavelet transformation..

This stage applies a reversible (one-to-one) transformation to the input image data. The purpose of this stage is to convert the input image data  $f(n)$  into a form  $f'(n)$  that can be compressed more efficiently. For this purpose, the selected transformation can aid in reducing the data correlation (interdependency, redundancy), alter the data statistical distribution and/or pack a large amount of information into few data samples or sub band regions. So for this stage, discrete cosine transform (DCT) technique is used.

#### iii. Discrete Cosine Transform (DCT)

The discrete cosine transform (DCT) is a technique for converting a signal into elementary frequency components. It is widely used in image compression. These functions illustrate the power of Mathematical in the prototyping of image processing algorithms. For this project, 2-D DCT is used. For analysis of two-dimensional (2D) signals such as images, we need a 2D version of the DCT.

#### iv. Transfer Domain Technique

The LSB method has the defect of being extremely vulnerable to even small cover modifications. Transform domain methods hide messages in the significant areas of the cover image.

There are many transform domain techniques, the most common being using the DCT. In digital image processing the 2D DCT is usually used:

$$\begin{aligned}
 S(u, v) &= \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} s(x, y) \cos\left(\frac{\pi u(2x+1)}{2N}\right) \cos\left(\frac{\pi v(2y+1)}{2N}\right) \\
 s(x, y) &= \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) S(u, v) \cos\left(\frac{\pi u(2x+1)}{2N}\right) \cos\left(\frac{\pi v(2y+1)}{2N}\right)
 \end{aligned}$$

Figure 5: Two dimensional DCT

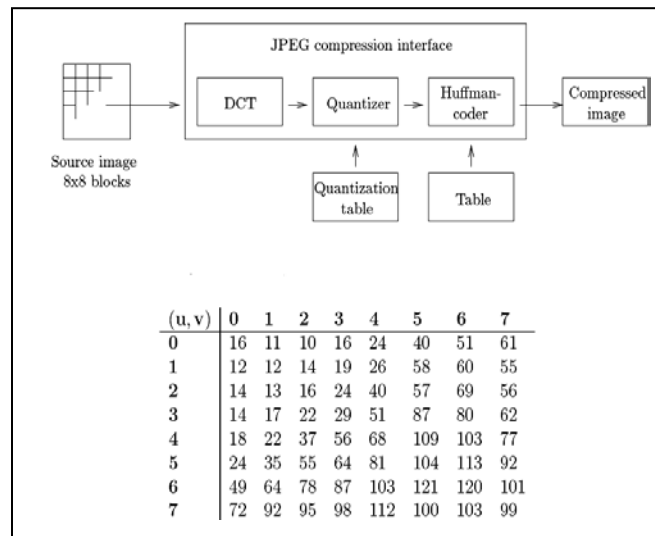


Figure 6: JPEG process outline &amp; Quantization values used in JPEG

The basic way JPEG compression works is outlined above. In particular the influence of the highest DCT coefficients is reduced: they are usually dominated by noise and not expected to contribute much to the picture details.

The popular way of doing steganography in the DCT domain is to modulate the relative size of two or more DCT coefficients within one image block. The basic algorithm is described as splitting the image into 8x8 blocks and calculating the DCT of the block. Then two middle-frequency (so that they are not to altered by the quantization/compression which will take place in JPEG) are chosen and agreed upon by both send and receive parties. A block encodes a 1 if  $DCT(a,b) > DCT(c,d)$  otherwise it encodes a 0. In the encoding step the coefficients are swapped if their relative size does not match with the bit to be encoded. Since the JPEG compression can affect the relative size of the coefficients the algorithm ensures that  $abs(DCT(a,b) - DCT(c,d)) > x$  where  $x$  is a value which represents the tradeoff between image quality and robustness.

#### v. Data-to -Symbol mapping

This stage converts the image data  $f'(n)$  into entities called symbols that can be efficiently coded by the final stage. The conversion into symbols can be done through partitioning or run length coding (RLC), for example the image data can be partitioned into blocks by grouping neighboring data samples together; in this case each data block is a symbol. Grouping several data units together allows the exploitation of any correlation that might be present between the image data and may result in higher compression ratios at the expense of increasing the coding complexity.

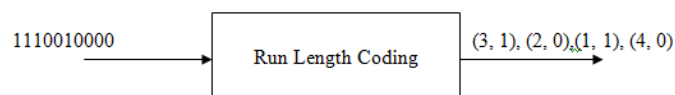


Figure 7: Illustration of RLC for a binary input sequence

The basic idea behind RLC is to map a sequence of numbers into a sequence of symbol pairs (run, value), where value is the value of a data sample in the input data sequence and run or run length is the number of times that data samples is contiguously repeated.

#### vi. Lossless Symbol Coding

This stage generates a binary bit stream by assigning binary code-words to the input symbols. Lossless symbol coding, is commonly referred to as noiseless coding or just lossless coding since this stage is where the actual lossless coding into the final compressed bit stream is performed. The first two stages can be regarded as preprocessing stages for mapping the data into a form that can be more efficiently coded by this lossless coding stage.

Lossless compression is usually achieved by using variable-length code words, where the shorter code words are assigned to the symbols that occur more frequently. This variable-length code words assignment is known as variable-length coding (VLC) and also as entropy coding. Entropy coding such as Huffman and arithmetic coders attempt to minimize the average bit rate (average number of bits per symbol) needed to represent a sequence of symbols, based on the probability of symbols occurrence. So for this stage, Huffman Coding was selected (Petitcolas, 2003).

## 4. Project Design

The baseline JPEG encoding follows some rules as predefined in figure 9.

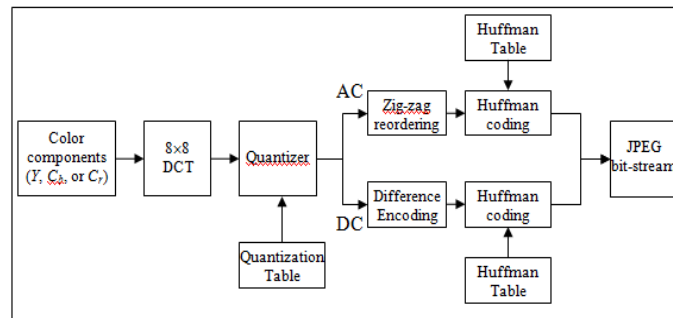


Figure 9: Baseline JPEG Encoding

### i. The color components (Y, Cb, Cr)

The first step in baseline JPEG encoding is transform the input image to the color components (Y,Cb,Cr). First, the image should be converted from RGB into a different color space called YCbCr. It has three components Y, Cb and Cr. The Y component represents the brightness of a pixel, the Cb and Cr components represent the chrominance (split into Blue and Red components). The YCbCr color space conversion allows greater compression for the same image quality (or greater image quality for the same compression). Figure 10 show the original image.



Figure 10: Original Image

### ii. RGB to Grayscale

By referring to Figure 11, it should be noted that the image is transform to gray color ranging from value 0 for pure black and 255 for pure white (Katenbeisser and Petitcoles,2000). . Therefore, each individual pixel could be represented in 256 shade of gray colour.



Figure 11: Grayscale image

### iii. The 8 x 8 DCT blocks logic

By referring to Figure 12 and Figure 13, an image is usually consisting of hundred or even thousands of 8 x 8 blocks of pixel. Each pixel will do the same process individually. The image is then need to be blocked in multiple 8 x 8 array. Because of the image is 2D signal, the 2D version of DCT is applied. The 2D DCT is calculated as 1D DCT applied to each row of S and to each column of the result.

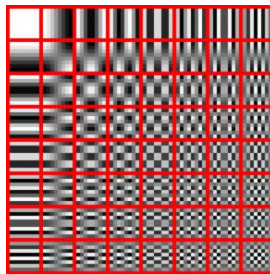


Figure 12: DCT transform (64 pixels)

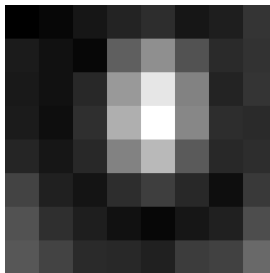


Figure 13: 8 x 8 subimages

### iv. Quantization

quanttable1=1*[	16	11	10	16	24	40	51	61;
	12	<u>12</u>	14	19	26	58	60	55;
	14	13	16	24	40	57	69	56;
	14	17	22	29	51	87	80	62;
	18	22	37	56	68	109	103	77;
	24	35	55	64	81	104	113	92;
	49	64	78	87	103	121	120	101;
	72	92	95	98	112	100	103	99];

Figure 14: Quantization Table

Next, the image is needed to perform quantization process using quantization table as figure 14. Quantization is a process of reducing the number of possible value. Hence, it will reduce the number of bit used. For example, we have number ranging from 0 till 7, means that it has 8 real numbers. Rounding to the nearest integer it can be represented as 2.78452, thus it can be rounded to 3. So, it can be represented as 3 bit.

0, 1, 2, 3, 4, 5, 6, 7 (8 value)  
 Nearest integer = 2.78452  
 Rounded = 3  
 Prove:  $2^3 = 8$

Image with higher spatial frequency is almost unnoticeable. Thus, the quantization factor needs to be large as possible as the higher frequency. The Figure 14 shows the 'de facto' matrix (Quantization Table).

#### v. Zig Zag ordering

In Zig Zag ordering, DC coefficient is treated separately from 64 AC coefficient. DC coefficient is a measure of the average value of 64 image. The DC value is calculated using Differential Pulse Code Modulation (DPCM). The remaining AC coefficients are ordered into Zig Zag sequence, which help to facilitate entropy coding to set the low frequency coefficient before high frequency coefficient. The output of DPCM and Zig Zag ordering is then will be encoded using entropy coding separately.

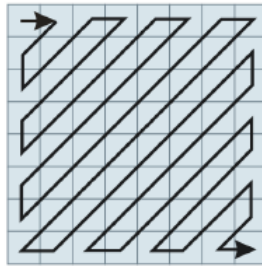


Figure 15: DC and AC coefficient (Zig Zag ordering)

#### vi. Entropy coding

An entropy encoding is a coding scheme that assigns codes to symbols to match code lengths with the probabilities of the symbols. Typically, entropy encoders are used to compress data by replacing symbols represented by equal-length codes with symbols represented by codes proportional to the negative logarithm of the probability.

Entropy of an image is a process of decreasing the number of bits required to represent an image. So, the lower value of entropy means lower number of bit required to represent an image.

#### B. Baseline Decoding

##### i. Decoding Text

Example:

The Huffman code for a, b, c, d is:

a = 0  
 b = 1 1  
 c = 1 0 0  
 d = 1 0 1

Now the Huffman code for the string "aaabbbcd" will be 0001111100101 – total 13 bits. ASCII code that needs 56 bits (7 character x 8 bit/character) for the string 'aaabbbcd'.



### C. Graphical User Interface (GUI)

Firstly to perform the encoding process, the size of the message and image must be calculated. Message size is used to mark the end of the message that means how many round to run the program.

The image size is used to calculate the maximum capacity that can be used to embed the message. Then, each character of the message will be converted into binary format according to the ASCII (American Standard Code for Information Interchange) code and combined all binary in one group.

Then, each bit from binary group will be embedding to the Least Significant Bit of every pixel of the image. One byte of image can only hide 1 bit of message.

One character will present 8 bit. So to embed one character, 8 pixels of image data will be needed. Figure 16 show the GUI (Graphical User interface) how encoding process be done on Matlab 7.6.

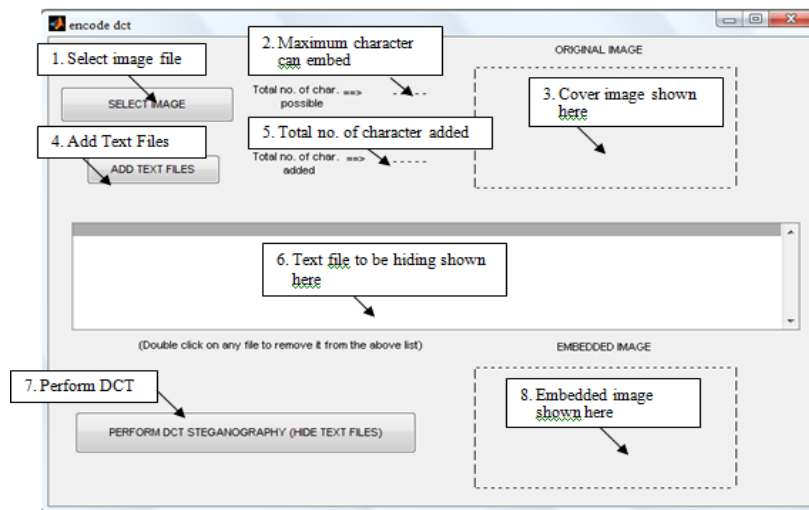


Figure 16: GUI for encode DCT using Matlab 7.6

## 5. Result and Analysis

This system only use one types of image as the cover image, namely Joint Photographic Experts Group (JPEG) format. The JPEG images are used as cover image are fruit.jpg, chick.jpg and torch.jpg. While the text use to be hidden in the image are jpg.txt (2630 character) and christmas\_carol.txt (1882 character).

### A. Analysis on Images dividing into Blocks Logic

Figure 17, 18,19 shows 8 x 8 block value for fruit.jpg, chick.jpg and torch.jpg respectively. The values obtain is relay the input image. The grayscale color of the image is in integer values. Therefore to get the 8 x 8 blocks logic the integer should be minus with 128. As a result the values are in the range of -256 to +256.

[illegible]

Figure 17: The values for fruit.jpg 8 x 8 block logic

	1	2	3	4	5	6	7	8	9	10
1	32	32	32	32	32	32	32	32	32	
2	32	32	32	32	32	32	32	32	32	
3	32	32	32	32	32	32	32	32	32	
4	32	32	32	32	32	32	32	32	32	
5	32	32	32	32	32	32	32	32	32	
6	32	32	32	32	32	32	32	32	32	
7	32	32	32	32	32	32	32	32	32	
8	32	32	32	32	32	32	32	32	32	
9										
10										

Figure 18: The values for chick.jpg 8 x 8 block logic

	1	2	3	4	5	6	7	8	9	10
1	127	127	127	127	127	127	127	127	127	
2	127	127	127	127	127	127	127	127	127	
3	127	127	127	127	127	127	127	127	127	
4	127	127	127	127	127	127	127	127	127	
5	127	127	127	127	127	127	127	127	127	
6	127	127	127	127	127	127	127	127	127	
7	127	127	127	127	127	127	127	127	127	
8	127	127	127	127	127	127	127	127	127	
9										
10										

Figure 19: The values for chick.jpg 8 x 8 block logic

## B. DCT Output

The DCT value is acquired block by block individually. The aim of DCT process is for domain transform. As the data used is an image, 2D DCT is applied. Figure 20 show the DCT output of the fruit.jpg, chick.jpg and torch.jpg image. Clearly can be seen that the DCT output is the same for all images because they have the same 512 x 512 size of image.

	1	2	3	4	5	6	7	8	9	10
1	0.0000	11.3137	11.3137	11.3137	11.3137	11.3137	11.3137	11.3137	11.3137	
2	11.3137	16	16	16	16	16	16	16	16	
3	11.3137	16	16	16	16	16	16	16	16	
4	11.3137	16	16	16	16	16	16	16	16	
5	11.3137	16	16	16	16	16	16	16	16	
6	11.3137	16	16	16	16	16	16	16	16	
7	11.3137	16	16	16	16	16	16	16	16	
8	11.3137	16	16	16	16	16	16	16	16	
9										
10										

Figure 20: DCT Output

## C. Quantization Output

Figure 21 show the Quantization Output for fruit.jpg, chick.jpg and torch.jpg image. Image with higher spatial frequency is almost unnoticeable. Thus, the quantization factor needs to be large as possible as the higher frequency. The 'de facto' matrix (Quantization table) was applied.

	1	2	3	4	5	6	7	8	9	10
1	0.5000	1.0205	1.1314	0.7071	0.4714	0.2020	0.2210	0.1055		
2	0.9428	1.3333	1.1429	0.8421	0.6154	0.2759	0.2667	0.2909		
3	0.8081	1.2309		1	0.6667	0.4000	0.2007	0.2319	0.2057	
4	0.8081	0.9412	0.7273	0.5517	0.3137	0.1839	0.2000	0.2581		
5	0.6285	0.7273	0.4324	0.2857	0.2353	0.1468	0.1553	0.2078		
6	0.4714	0.4571	0.2909	0.2500	0.1975	0.1538	0.1416	0.1739		
7	0.2309	0.2500	0.2051	0.1039	0.1553	0.1322	0.1333	0.1584		
8	0.1571	0.1739	0.1684	0.1633	0.1429	0.1600	0.1553	0.1616		
9										
10										

Figure 21: Quantization Output

#### D. Zig Zag Output

Figure 22 show the Zig zag output for fruit.jpg, chick.jpg and torch.jpg image. In Zig zag ordering, DC coefficient is treated separately from 64 AC coefficients. DC coefficient is a measure of the average value of 64 images. The DC value is calculated using Differential Pulse Code Modulation (DPCM). The remaining AC coefficients are ordered into Zig zag sequence, which help to facilitate entropy coding to set the low frequency coefficient before high frequency coefficient.

0.5	1.0285	0.9428	0.8081	1.3333	1.1313	0.7071	1.1429
1.2308	0.8081	0.6285	0.9412	1	0.8421	0.4714	0.2828
0.6154	0.6667	0.7273	0.7273	0.4714	0.2309	0.4571	0.4324
0.5517	0.4	0.2759	0.2218	0.1855	0.2667	0.2807	0.3137
0.2857	0.2909	0.25	0.1571	0.1739	0.2051	0.25	0.2353
0.1839	0.2319	0.2909	0.2857	0.2	0.1468	0.1975	0.1839
0.1684	0.1633	0.1553	0.1538	0.1553	0.2581	0.2078	0.1416
0.1322	0.1429	0.16	0.1333	0.1739	0.1584	0.1553	0.1616

Figure 22: Quantization Output

#### E. Huffman and Entropy Text

Table 1 show the information of text file format after applied the Huffman and entropy coding. The jpeg.txt file has 4.5747 entropy, 4.5945 average length and 0.43313 redundancy. While, the christmas\_carol.txt file has 4.4976 entropy, 4.5302 average lengths and 0.72602 redundancy.

Table1: Information of text file format

Text file name	Entropy	Average length	Redundancy
jpeg.txt	4.5747	4.5945	0.43313
christmas_carol.txt	4.4976	4.5302	0.72602

## 6. Analysis

Table 2 shows the comparison between three (3) different image file (JPEG) used fruit.jpg, chick.jpg and torch.jpg. All the images are using JPEG format and in the weight and height of 512 x 512. The size of the image is differ depend on the furnishing occurred to the image.

After applying DCT technique on the image and inserting the secret text into the image it is clearly seen that the size of the image is compressed efficiently without any quality degradation noticeable by human eyes. This is because the program merely replaced the LSB with the secret message to be hidden, not add in. Replacing the LSB with the secret message will not harm the quality of the image.

The size of image after embedding for jpeg image is lower than the jpeg original image. This is because this project is use lossless image compression. The lossless image compression is to represent signal with the smallest loss of any information, thereby speeding up transmission and minimizing storage requirements.


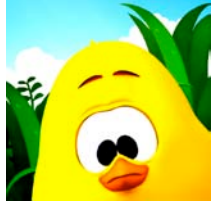


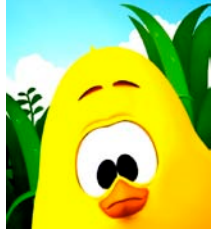

Original image can be reproduced exactly from the compressed data. Lossless compression is possible because, in general there is significant redundancy present in image signals. This redundancy is proportional to the amount of correlation among the image data samples. For example, in a natural still image, there is usually a high degree of spatial correlation among neighboring image samples.

Entropy of an image is a process of decreasing the number of bits required to represent an image. So, the lower value of entropy means lower number of bit required to represent an image. So, by using entropy technique, the image will be compressed effectively.

Lossless symbol coding, is commonly referred to as noiseless coding or just lossless coding since this stage is where the actual lossless coding into the final compressed bit stream is performed. Entropy coding such as Huffman and arithmetic coders attempt to minimize the average bit rate (average number of bits per symbol) needed to represent a sequence of symbols, based on the probability of symbols occurrence. For the width and height of the image, it's still same with the original image and after embedding.

Table 2: Result embed using three (3) different image

Name of the cover image		fruit.jpg	chick.jpg	torch.jpg
Text file name that has been hidden		Christmas_carol.txt	jpeg.txt	Christmas_carol.txt and jpeg.txt
Image file format	Before embedding	jpg	jpg	jpg
	After embedding	jpg	jpg	jpg
The width and height of the cover image	Before embedding	512x512	512x512	512x512
	After embedding	512x512	512x512	512x512
Size of the image	Before embedding	162KB	50.6KB	49.9KB
	After embedding	40.6KB	20.3KB	22.6KB

Image quality	Before embedding			
	After embedding			
	Observation	No visible quality degradation	No visible quality degradation	No visible quality degradation

## 7. CONCLUSION

The aims and objective of this project have been achieved where it can increase the security of the data transferred using Discrete Cosine Transformation (DCT) technique. The whole system is function properly.

In this project, "Text Hiding using Discrete Cosine Transformation", for a given input image, dividing into blocks, level shifting by -128, DCT, Quantization, Zig Zag Ordering are done one after the other. The text file which was implemented using the coding tool "MATLAB 7.6" has been successfully embedded on the cover images named 'fruit.jpg', 'chick.jpg' and 'torch.jpg'.

All the input and output images of all the operations are displayed. The difference of the original image and the compressed image can be observed.

## References

- B. Norman, (1973) "Secret Warfare", AcropolisBooks, Washington, D.C.
- D. Kahn,(1967) "The Codebreakers", Macmillan, New York.
- H.S. Zim,(1948) "Codes and Secret Writing", William Morrow, New York.
- J. Brassilet et al.,(1995) "Document Marking and Identification using Both Line and Word Shifting," Proc. Infocom95, IEEE CS Press, Los Alamitos, Calif.
- P. Wayner, (1996)"Disappearing Cryptography", AP Professional, Chestnut Hill, Mass.
- Kessler,G.C.(2004)"Steganography: Hiding Data Within Data" Windows & NET Magazine.

Katenbeisser, S and Petitcolas, F.A.P.(2000). "Information Hiding Techniques for Steganography and Digital Watermarking." Norwood, MA: Artech House.  
Petitcolas,A.P (2003) "Information Hiding" Berlin:Springer.