RCC Institute of Information Technology
Canal South Road, Beliaghata, Kolkata – 700 015
[Affiliated to West Bengal University of Technology]

IMAGE STEGANOGRAPHY USING DISCRETE COSINE TRNASFORM ALGORITHM

PROJECT REPORT SUBMITTED FOR PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE DEGREE OF
BACHELOR OF TECHNOLOGY
In
INFORMATION TECHNOLOGY

Submitted by:

Chandratapa Roy
(IT/2014/013)
(11700214030)

Surela Saha
(IT/2014/014)
(11700214079)

Shipra Jha
(IT/2014/032)
(11700214064)

Under the Guidance of **Mr. Amit Khan**

(Assistant Professor, RCCIIT)

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Mr. Amit Khan of the department of Information Technology, whose role as project guide was invaluable for the project. We are extremely thankful for the keen interest he took in advising us, for the books and reference materials provided for the moral support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as our asset and to all non-teaching staff for the gracious hospitality they offered us.

Place: RCCIIT, Kolkata

Date: /05/2018

………………………………

Chandratapa Roy

………………………………

Surela Saha

………………………………

Shipra Jha

Department of Information Technology
RCCIIT, Beliaghata,
Kolkata – 700 015,
West Bengal, India

Approval

This is to certify that the project report entitled "Image Steganography In Frequency Domain Using Discrete Cosine Transform Algorithm" prepared under my supervision by CHANDRATAPA ROY (Roll No.: IT 2014/013) , SURELA SAHA (Roll No.: IT 2014/014), SHIPRA JHA (Roll No.: IT 2014/032) , be accepted in partial fulfillment for the degree of Bachelor of Technology in Information Technology.

It is to be understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

…………………………………
Dr. Abhijit Das
Associate Professor and Head,
Department of Information Technology,
RCC Institute of Information Technology

…………………………………
Mr. Amit Khan
Assistant Professor,
Department of Information Technology,
RCC Institute of Information Technology

INDEX

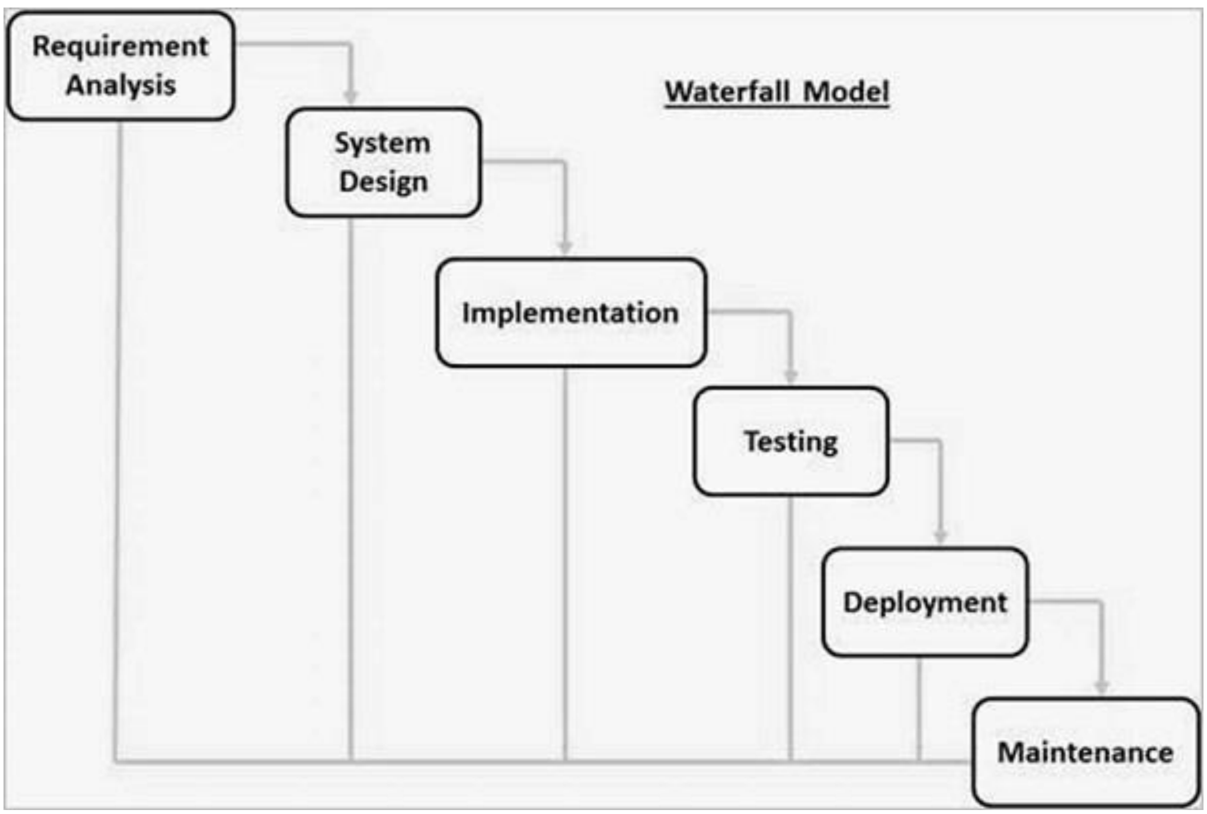
ABSTRACT

The purpose of this project is to automate the existing manual system with the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data or information can be stored for longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the clients. The required software and hardware are easily available and easy to work with.
Image Steganography, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means one need not be distracted by information that is not relevant, while being able to reach the information

LITERATURE STUDY

WATERFALL MODEL: The Waterfall Model was first Process Model to be introduced. It is very simple to understand and use. In a ***Waterfall model***, each phase must be completed before the next phase can begin and there is no overlapping in the phases. ***Waterfall model*** is the earliest SDLC approach that was used for software development.

In The Waterfall approach, the whole process of *software development* is divided into separate phases. The outcome of one phase acts as the input for the next phase sequentially. This means that any phase in the development process begins only if the previous phase is complete. The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance.



The sequential phases in Waterfall model

The sequential phases in Waterfall model are –

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

TYPES OF FEASIBILITY STUDY

There are four types of feasibility.

Operational feasibility:

• It is used for identify the importance of certain problem in project and how it is to be solved.

• It also measures how solution of the problems will work for any project or in any organization.

• It analysis the behaviour of the proposed system and whether the proposed system is easier than the existing system for the users of the system.

Technical feasibility:

• It is used to identify whether the technical resources are available to form the project or system. It suggests efficient input and output devices to manage large amount of data.

• It measure whether the hardware and software of existing system to which extent it can support the proposed system.

• It checks the available technology is within the given constraints such as budget.

• For example: if the current computer is operating at the speed of 90% capacity, when it needs to run one or more applications at the same time. The system may get overloaded. This includes financial consideration to accommodate technical resources such as addition of hardware, the proposed system must support that enhancements.

Economic feasibility:

• It is used to determine the financial resources of the project.

• It measures all costs incurred in development of new system.

• It is called cost benefit analysis because it determines the total cost for development of new system and benefits derived from the new system

• Benefits of new system should be more that cost incurred to achieve profit from new system or for any organization.

Schedule feasibility:

• Scheduling is all based upon time; it measures whether there is available time to do the project.

• Some projects are needed to be completed in the given deadlines; feasibility determines whether the project can be completed within that deadline.

• This feasibility is used to allocate time for separate module development in proposed system.

INTRODUCTION

1.1.1 Terminology

The definition of steganography can be explained clearly by using adjectives like ‘cover’, ‘embedded’ and ‘stego’. Steganography simply grabs a piece of information and hides that information within another piece of information [Scribd 2007]. Many computer files such as audio files, text files, images contain few blocks of data that is either unused or not significant. Steganography takes advantage of these unused areas and hides the encrypted message.

1.1.2 Difference with Other Systems

The term “Information hiding” can be related to either steganography field or watermarking technology. Watermarking technology usually refers to various methods that conceal information in a data object so that the information is adjustable to future modifications [Wikipedia]. In essence, it should be impossible to remove the watermark without drastically modifying the quality of the object.

While on the other hand, steganography refers to hidden / concealed information that is fragile. Any small modification to the cover medium may destroy the concealed information. Also, the above mentioned two ways differ in one more way. In steganography, viewer or user must not know about the presence of the concealed information whereas in watermarking, this feature is optional.

1.1.3 History

The motive behind steganography is to communicate data secretly with somebody. An earlier account of steganography is found in a story by Herodotus, in which a slave sent by his master, Histiaeus, to the Ionic city of Miletus with a secret message tattooed on his scalp [Cox, Miller, Bloom 2001]. Once when the slave's hair had grown back and had successfully hidden the message, the slave, Herodotus was sent to warn of the Persian's impending invasion on the Greece.

Another method to secretly deploy ciphered messages was to modify ancient writing tablets [Wikipedia]. In this method, the messages that need to be hidden were written on the layer of wax covering the surface of the tablets. The enhanced version of this method was developed by Demeratus [Wikipedia]. Demeratus was by birth a Greek but he was exiled into Persia. He devised a master sketch to embed and thus hide a message by removing the layer of wax and writing directly on the underlying wood. Demeratus implemented this enhanced method and was successful in sending a warning message to Sparta that the Persians were planning an invasion. After the message was written directly on the writing tablets, they were then covered again with wax and appeared unused to the examiners of the shipment.

Another classic application of ancient steganography is the method of wrapping a ribbon around a wooden staff from top to bottom [Brainos II 2003]. This method is the best example of a null cipher that was mentioned above. The key to this method is to write across all over the ribbon and unravel it. By doing so, the clear text will be all over the wooden staff and only someone with the same size diameter wooden staff similar to the original one could read the hidden ciphered message. In this method, the most important feature is the fact that the ciphered message even existed will be hidden from the outsiders.

Coming to not so ancient steganographic applications, few methods were introduced during the two World Wars, especially World War II. During this period, the German military created microdots which are a breakthrough technology at that time. They leveraged microfiche technology to create these microdots. Microdots consisted of pictures and text messages which were shrunk down to the size of a period and used in the text of an otherwise innocent letter or memorandum [Wikipedia]. The big breakthrough for steganography has happened in the early nineties when governments, industries, general citizens and even some of the extremist organizations began using software applications to embed messages and photos into various types of media like digital photos, digital videos, audio files and text files.

Most of the techniques proposed in the literature use texts or images as covers. For embedding a message in a text document, apparently invisible coding techniques are used. However, for embedding a message in an image, a different set of techniques such as least-significant bit insertion, masking and filtering, and subtle transformation of the image are used. These techniques or transformations do not cause any visible changes in the cover image when viewed.

1.1.4 Components of a Steganographic Message

Before going deep into the steganographic process, first and foremost, we need to understand the various components of a steganographic message. The below list covers all the possible components that will be present in the steganographic message.

- Secret message
- Cover data
- Stego message

The *secret message* refers to the part of the message which is intended to be hidden. This message will later be encrypted to make it even more difficult for anyone who tries to break the security to get hold of the hidden informatic message. This is the crucial component in a steganographic message. Next part is the *cover data* component. This component refers to the container in which the secret message is hidden. This cover data component can be anything like digital photos, digital videos, audio files and text files. The final component is the *stego message* which is as crucial as the *secret message*. The *stego message* component refers to the final product.

1.1.5 Steganographic Approaches

There are various types of cream layer steganographic approaches. They are:

- Top-down approach
- Bottom-up approach

These approaches are again subdivided into sub layers. From a top-down approach, there exist three types of steganographic approaches. They are:

- Pure steganography
- Private key steganography
- Public key steganography

These categories convey the level of security with which the stego message is embedded, transmitted and read.

1.1.5.1 Pure Steganography

Pure steganography is defined as a steganographic system that does not require the exchange of a cipher such as a stego-key. This method of steganography is the least secure means by which to communicate secretly because the sender and receiver can rely only upon the presumption that no other parties are aware of this secret message [Brainos II 2003]. Using open systems such as the Internet, we know this is not the cause at all.

Pure steganography uses no keyed system to embed cleartext or null cipher text into the cover data in order to hide the existence of a secret message. Pure steganography is only secure in two aspects which are, the fact only the sending and receiving parties know of the secret message's existence and which steganographic algorithm was used to hide the message. In steganalysis, this type is the easiest to crack since once detected the message can only have been hidden in as many ways as the number of steganographic algorithms which exist.

The foremost difficult aspect is in the detection effort. The difficulty lies in the fact that the unlimited amount of screened data does not include pre-modification copies of themselves. For example, if any National Security Agency has to screen millions of Web pages for steganographic material, most of the authors of this material would not leave the original copies of the cover data in the Web site's directory or even on the computer which produced the stego message. In this way, the message will be virtually undetectable because of the fact that in the simplest form of the stego message, only the least significant bits of each byte representing a digital photo have been modified to carry the secret message. However, once detected, a pure stego message could be cracked very easily. This is the reason why the pure steganographic approach is the least secure method and thus least used.

1.1.5.2 Private Key Steganography

Private key steganography is also called as secret key steganography [Brainos II 2003]. This secret key steganography is defined as a steganographic system that requires the exchange of a secret key prior to communication. Secret key steganography takes a cover message and embeds the secret message inside of it by using a secret key. This secret key is also called the stego key. Only the parties who know the secret key can reverse the process and read the secret message.

Unlike pure steganography where a perceived invisible communication channel is present, secret key steganography exchanges a stego key, which makes it more susceptible to interception [Anderson, Petitcolas 1998]. The benefit to secret key steganography is even if it is intercepted; only parties who know the secret key can extract the secret message.

This private key steganography method uses a mutual key for encrypting then hiding the secret message within the cover data. As in traditional encryption, the private key system is only as robust as the knowledge of the key. Since the private key system requires both parties to know the key, once it is compromised the entire stego message is non-secure.

1.1.5.3 Public Key Steganography

Public key steganography can be defined as a steganography system that uses a public key and a private key to secure the communication between the parties wanting to communicate secretly [Brainos II 2003]. The sender will use the public key during the encoding process and only the private key, which has a direct mathematical relationship with the public key, can decipher the secret message.

Public key steganography provides a more robust way of implementing a steganographic system because it can utilize a much more robust and researched technology in public key cryptography. It also has multiple levels of security in that unwanted parties must first suspect the use of steganography and then they would have to find a way to crack the algorithm used by the public key system before they could intercept the secret message. This public key encrypted steganography uses the key pair system to add a layer of robustness to the process. As in public key encryption, the public key of the recipient is used to encrypt the secret message and only that user's private key may decrypt it after extracting it from the cover data. This method is the most secure type of steganography. This approach is recommended since it combines the benefits of hiding the existence of a secret message with the security of encryption.

PROBLEM DEFINITION

In frequency domain, the image is first transformed to its frequency distribution. Unlike in the spatial domain where changes are made to pixel values directly, in frequency domain the rate is dealt at which the pixel values change in spatial domain. Whatever processing is to be done is carried in frequency domain and the resultant image is subjected to inverse transform to obtain the required image. Discrete cosine transform (DCT), discrete fourier transform (DFT), discrete wavelet transform (DWT) etc are the examples of frequency domain. Stegnography process in transform domain proposed entropy based technique using block level entropy thresholding. In this method, cover image was divided into 8×8 non overlapping blocks. After selecting block DCT was computed for selected block. Secret message was embedded on block by middle frequency selection. This method gave much preferable robustness, good PSNR results and provides high security presented frequency domain steganographic method based on entropy thresholding scheme. In this method, large volume of data was embedded in image. After computing 64 DCT coefficients for each non overlapping block, entropy of four most significant bits and least significant bits was computed. This proposed technique was data hiding method with which one can adjust quality factor and embedding capacity dynamically.

PLANNING

To compress an image into JPEG format, the RGB colour representation is first converted to a YUV representation. In this representation the Y component corresponds to the luminance (or brightness) and the U and V components stand for chrominance (or colour). According to research the human eye is more sensitive to changes in the brightness (luminance) of a pixel than to changes in its colour . This fact is exploited by the JPEG compression by downsampling the colour data to reduce the size of the file. The colour components (U and V) are halved in horizontal and vertical directions, thus decreasing the file size by a factor of 2. The next step is the actual transformation of the image.
For JPEG, the Discrete Cosine Transform (DCT) is used, but similar transforms are for example the Discrete Fourier Transform (DFT). These mathematical transforms convert the pixels in such a way as to give the effect of "spreading" the location of the pixel values over part of the image. The DCT transforms a signal from an image representation into a frequency representation, by grouping the pixels into 8 × 8 pixel blocks and transforming the pixel blocks into 64 DCT coefficients each . A modification of a single DCT coefficient will affect all 64 image pixels in that block. The next step is the quantization phase of the compression. Here another biological property of the human eye is exploited: The human eye is fairly good at spotting small differences in brightness over a relatively large area, but not so good as to distinguish between different strengths in high frequency brightness . This means that the strength of higher frequencies can be diminished, without changing the appearance of the image. JPEG does this by dividing all the values in a block by a quantization coefficient. The results are rounded to integer values and the coefficients are encoded using Huffman coding to further reduce the size. JPEG steganography Originally it was thought that steganography would not be possible to use with JPEG images, since they use lossy compression which results in parts of the image data being altered. One of the major characteristics of steganography is the fact that information is hidden in the redundant bits of an object and since redundant bits are left out when using JPEG it was feared that the hidden message would be destroyed. Even if one could somehow keep the message intact it would be difficult to embed the message without the changes being noticeable because of the harsh compression applied. However, properties of the compression algorithm have been exploited in order to develop a steganographic algorithm for JPEGs. One of these properties of JPEG is exploited to make the changes to the image invisible to the human eye. During the DCT transformation phase of the compression algorithm, rounding errors occur in the coefficient data that are not noticeable . Although this property is what classifies the algorithm as being lossy, this property can also be used to hide messages. It is neither feasible nor possible to embed information in an image that uses lossy compression, since the compression would destroy all information in the process. Thus it is important to recognize that the JPEG compression algorithm is actually divided into lossy and lossless stages. The DCT and the quantization phase form part of the lossy stage, while the Huffman encoding used to further compress the data is lossless. Steganography can take place between these two stages. Using the same principles of LSB insertion the message can be embedded into the least significant bits of the coefficients before applying the Huffman encoding. By embedding the information at this stage, in the transform domain, it is extremely difficult to detect, since it is not in the visual domain.

Implementation of Mechanisms Existing Techniques Basic techniques

- ❖ A novel technique for image steganography based on Block-DCT and Huffman Encoding
- ❖ High Capacity Image SteganographyusingWavelet Transform and Genetic Algorithm
- ❖ A Secure Image Steganography using LSB, DCT and Compression Techniques on Raw Images
- ❖ Labeling method
- ❖ JPEG and particle swarm optimization

- ❖ Quantized-frequency Secure Audio Steganography algorithm
- ❖ Integer Transform based Secure Audio Steganography algorithm

Block-DCT (Discrete Cosine Transform)

Let I(x,y) denote an 8-bit grayscale cover-image with x = 1,2,…?,M1 and y = 1,2,…?,N1. This M1 cover-image is divided into 8x8 blocks and two-dimensional (2- D) DCT is performed on each of L = M1×N1 / 62 blocks. The mathematical definition of DCT is:

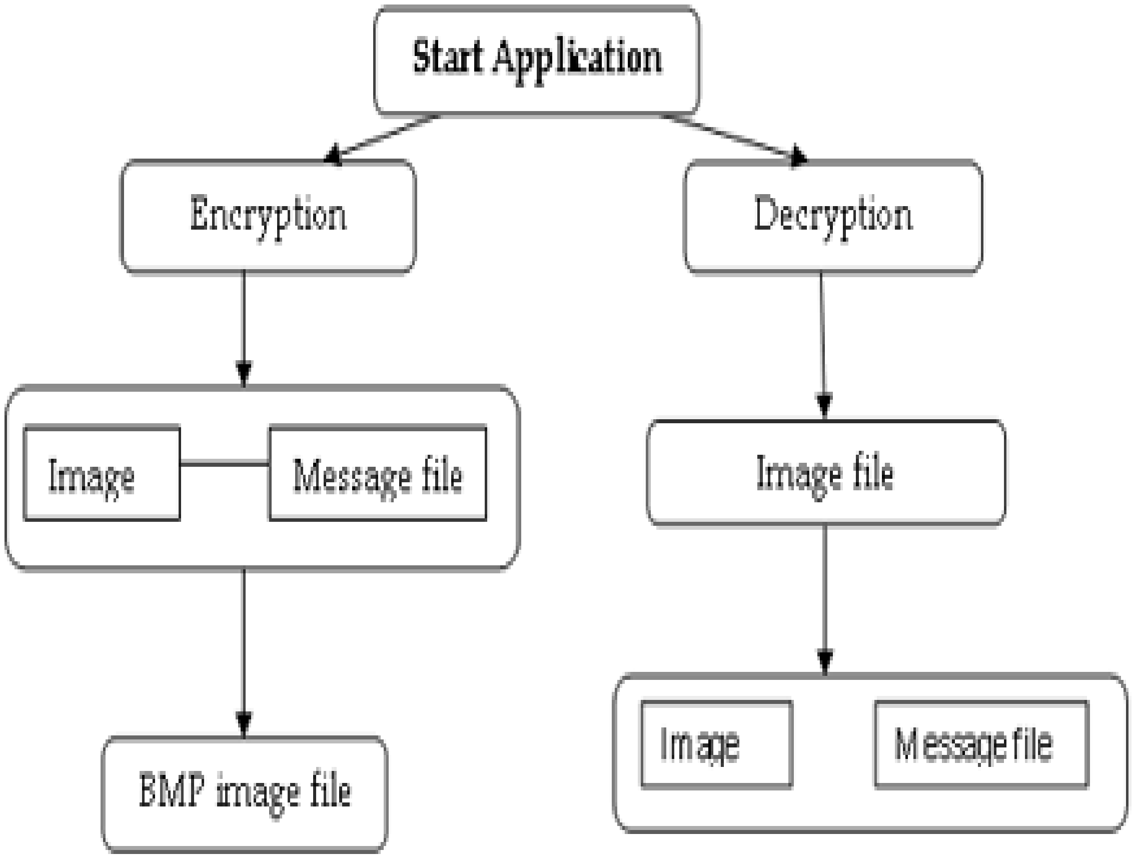
Forward DCT:

$$F(u,v) = \frac{1}{4} C(u)C(v) \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos\left[\frac{\pi(2x+1)u}{16}\right] \cos\left[\frac{\pi(2y+1)v}{16}\right]$$
$$\text{for } u = 0,\dots,7 \text{ and } v = 0,\dots,7 \tag{1}$$
$$\text{where } C(k) = \begin{cases} 1/\sqrt{2} & \text{for } k = 0 \\ 1 & \text{otherwise} \end{cases}$$

Inverse DCT :

$$f(x,y) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C(u)C(v)F(u,v) \cos\left[\frac{\pi(2x+1)u}{16}\right] \cos\left[\frac{\pi(2y+1)v}{16}\right] \tag{2}$$
$$\text{for } x = 0,\dots,7 \text{ and } y = 0,\dots,7$$

DESIGNING



Algorithm for Encryption:

Step-1: Write text message.(original message).
Step-2: Select cover image.
Step-3: The cover image is broken into 8×8 block of pixels.
Step-4: Use DCT to transform each block Oi into DCT coefficient matrix.
Step-5: Calculate LSB of each DC coefficient and replace with each bit of secret message.
Step-6: Write stego image.

Algorithm for Decryption:

Step-1: Read the stego image.
Step-2: Divide the stego image into 8×8 block of pixels.
Step-3: DCT is applied to each block.
Step-4: Calculate LSB of each DC coefficient.
Step-5: Get original message.

Project code for Encryption:-

```
function [embimg,p]=wtmark(im,wt)
% wtmark function performs watermarking in DCT domain
% it processes the image into 8x8 blocks.

% im     = Input Image
% wt     = Watermark
% embimg = Output Embedded image
% p      = PSNR of Embedded image

% Checking Dimnesions and then converting into grayscale.
im=imread('b.jpg');
if length(size(im))>2
   im=rgb2gray(im);
end

im        = imresize(im,[512 512]); % Resize image
w1 = imresize(wt,[64 64]);% Resize and giving the watermark image a possible size
watermark = im2bw(w1,0.7);% Converts the image into a binary image

x={}; % empty cell which will consist all blocks
dct_img=blkproc(im,[8,8],@dct2);% DCT of image using 8X8 block
m=dct_img; % Source image in which watermark will be inserted

k=1; dr=0; dc=0;
% dr is to address 1:8 row every time for new block in x
% dc is to address 1:8 column every time for new block in x
% k is to change the no. of cell

%%% To divide image in to 4096---8X8 blocks %%%
for i=1:8:512 % To address row -- 8X8 blocks of image
   for j=1:8:512 % To address columns -- 8X8 blocks of image
      for i1=i:(i+7) % To address rows of blocks
         dr=dr+1;
         for j1=j:(j+7) % To address columns of block
            dc=dc+1;
            z(dr,dc)=m(i,j);
         end
         dc=0;
      end
      x{k}=z; k=k+1;
      z=[]; dr=0;
   end
end
nn=x;
```

```
%%% To insert watermark in to  blocks %%%
i=[]; j=[]; w=1; wmrk=watermark; wlm=numel(wmrk); % welem - no. of elements
for k=1:4096
   kx=(x{k}); % Extracting block into kx for processing
   for i=1:8 % To address row of block
      for j=1:8 % To adress column of block
         if (i==8) && (j==8) && (w<=wlm) % Eligiblity condition to insert watremark
            % i=1 and j=1 - means embedding element in first bit of every block
             if wmrk(w)==0
                kx(i,j)=kx(i,j)+10;
            elseif wmrk(w)==1
                kx(i,j)=kx(i,j)-10;
             end
         end
      end
   end
   w=w+1;
   x{k}=kx; kx=[]; % Watermark value will be replaced in block
end

%%% To recombine cells in to image %%%
i=[]; j=[]; data=[]; count=0;
embimg1={}; % Changing complete row cell of 4096 into 64 row cell
for j=1:64:4096
   count=count+1;
   for i=j:(j+63)
      data=[data,x{i}];
   end
   embimg1{count}=data;
   data=[];
end



% Change 64 row cell in to particular columns to form image
i=[]; j=[]; data=[];
embimg=[];  % final watermark image
for i=1:64
   embimg=[embimg;embimg1{i}];
end
embimg=(uint8(blkproc(embimg,[8 8],@idct2)));
imwrite(embimg,'out.jpg')
p=psnr(im,embimg);
```

Project code for Decryption :-

```
function [wm]=exwmark(embimg)
```

```
% exwmark will extract the watermark which were
% embedded by the wtmark function

% embimg  = Embedded image

% wt  = Extracted Watermark

[row clm]=size(embimg);
m=embimg;

%% To divide image in to 4096---8X8 blocks %%%
k=1; dr=0; dc=0;
% dr is to address 1:8 row every time for new block in x
% dc is to address 1:8 column every time for new block in x
% k is to change the no. of cell
for i=1:8:row % To address row -- 8X8 blocks of image
   for j=1:8:clm % To address columns -- 8X8 blocks of image
      for i1=i:(i+7) % To address rows of blocks
         dr=dr+1;
         for j1=j:(j+7) % To address columns of block
            dc=dc+1;
            z(dr,dc)=m(i,j);
         end
         dc=0;
      end
      x{k}=z; k=k+1;
      z=[]; dr=0;
   end
end
nn=x;

%%Extract water mark %%

wm=[]; wm1=[]; k=1; wmd=[]; wmd1=[]
while(k<4097)
   for i=1:64
   kx=x{k}; % Extracting Blocks one by one
   dkx=blkproc(kx,[8 8],@dct2); % Applying Dct
   nn{k}=dkx; % Save DCT values in new block to cross check


   %% Change me for pixel location
   wm1=[wm1 dkx(8,8)]; % Forming a row of 32 by 8,8 element

   % Extracting water mark without dct
    wmwd1=[wmwd1 kx(8,8)];
     k=k+1;
   end
```

```
    wm=[wm;wm1]; wm1=[]; % Forming columns of 32x32
    wmd=[wmd;wmd1]; wmd1=[];
end

for i=1:64
    for j=1:64
        diff=wm(i,j);
        if diff >=0
            wm(i,j)=0;
        elseif diff < 0
            wm(i,j)=1;
        end
    end
end

wm=wm';
imwrite(wm,'wex.jpg')
```

PSNR code for performance measurement :

```
% Peak to signal ratio=PSNR
% original- Initial source image
% processed- Embedded image after watermark function being performed
% Calculates the difference in the original and processed image

function y=psnr(processed,original)
processed=im2double(processed);
original=im2double(original);
[m n]=size(original);

%mserror
error=processed - original;
se=error.*error;
sumse=sum(sum(se));
mse=sumse/(m*n);
%mserror

ma=max(max(processed));
y=10*log10(ma*ma/mse);
```

UI for the process:-

```
function varargout = wmark_enc(varargin)
% WMARK_ENC MATLAB code implemented on for  wmark_enc.fig
%      WMARK_ENC, by itself, creates a new WMARK_ENC or raises the existing
%      singleton*.
%
```

```
%      H = WMARK_ENC returns the handle to a new WMARK_ENC or in order to  handle to
%      the existing singleton*.
%
%      WMARK_ENC('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in WMARK_ENC.M with the given input arguments as specified.
%
%      WMARK_ENC('Property','Value',...) creates a new WMARK_ENC or raises the
%      existing singleton*.  .  All inputs are passed to wmark_enc_OpeningFcn via varargin.
%



% initialization part
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @wmark_enc_OpeningFcn, ...
                   'gui_OutputFcn',  @wmark_enc_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization part.

% --- Executes just before wmark_enc is made visible.
function wmark_enc_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to wmark_enc (see VARARGIN)

% Choose default command line output for wmark_enc
handles.output = hObject;
axes(handles.iim); axis off
axes(handles.oim); axis off
axes(handles.omsg); axis off
axes(handles.demsg); axis off
set(handles.emmsg,'Enable','off')
set(handles.extmsg,'Enable','off')
set(handles.msg,'Enable','off')
```

```
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes wmark_enc wait for user response
% Outputs from this function are returned to the command line.
function varargout = wmark_enc_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% Get default command line output from handles structure
varargout{1} = handles.output;



% Will execute on button press in exit.
function exit_Callback(hObject, eventdata, handles)
% eventdata  reserved - to be defined in a future version of MATLAB
close wmark_enc

% Will execute on button press in extmsg.
function extmsg_Callback(hObject, eventdata, handles)
% hObject    handle to extmsg (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
embimg=handles.embimg;
wm=exwmark(embimg);
axes(handles.demsg); imshow(wm); title('Extracted MSG')
handles.wm=wm;
guidata(hObject,handles)



% Will execute on button press in emmsg.
function emmsg_Callback(hObject, eventdata, handles)
% hObject    handle to emmsg
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
img=handles.img; msg=handles.msg;
h=warndlg('Wait....','Processing');
[embimg,ps]=wtmark(img,msg);
handles.embimg=embimg;
axes(handles.oim); imshow(embimg); title('Embedded Image')
set(handles.impsnr,'String',ps)
set(handles.extmsg,'Enable','On')
close(h)
guidata(hObject,handles)
```

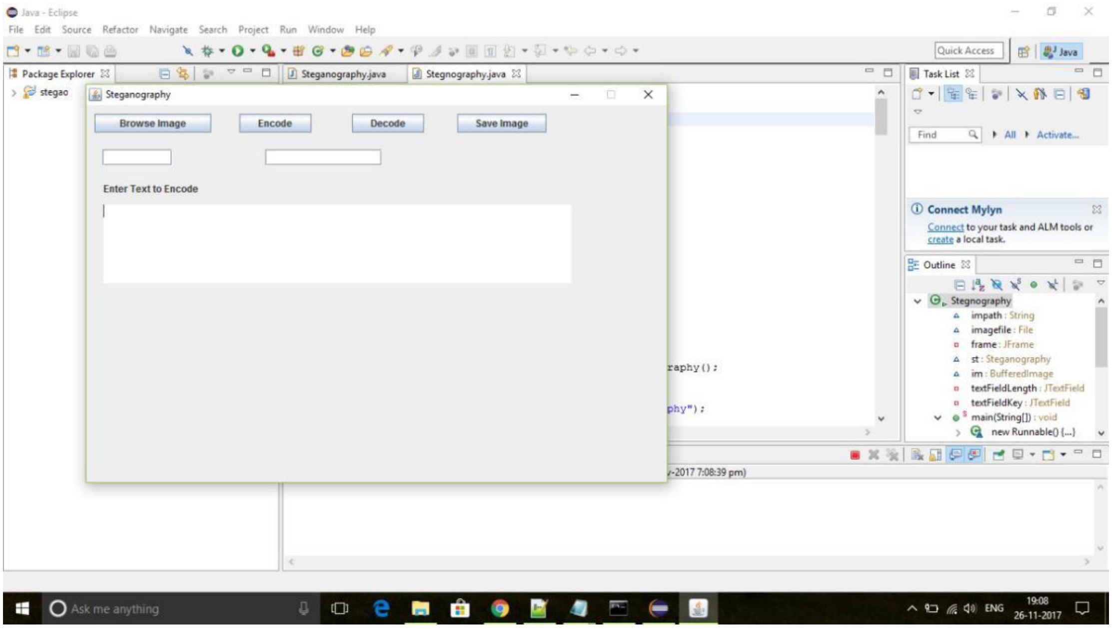
```
% Executes on button press in inp.
function inp_Callback(hObject, eventdata, handles)
% hObject    handle to with respect  inp
% handles    structure with handles and user data.
[fname path]=uigetfile({'*.jpg';'*.bmp';,'*.jpeg';'*.tiff';'*.png'},'Browse Image');
if fname~=0
   img=imread([path,fname]);
   if length(size(img))>2
      img=rgb2gray(img);
   end
   axes(handles.iim); imshow(img);
   title('Orignal Image')
   handles.img=img;
   set(handles.msg,'Enable','on')
else
   warndlg('Please Select Image File');
end
guidata(hObject,handles);

% Executes on button press in msg.
function msg_Callback(hObject, eventdata, handles)
% hObject    handle to msg
[fname path]=uigetfile({'*.jpg';'*.bmp';,'*.jpeg';'*.tiff';'*.png'},'Browse Image');
if fname~=0
   msg=imread([path,fname]);
   if length(size(msg))>2
      msg=rgb2gray(msg);
   end
   axes(handles.omsg); imshow(msg);
   title('MSG')
   handles.msg=msg;
   set(handles.emmsg,'Enable','on')
else
   warndlg('Please Select Image File');
end
guidata(hObject,handles);
```

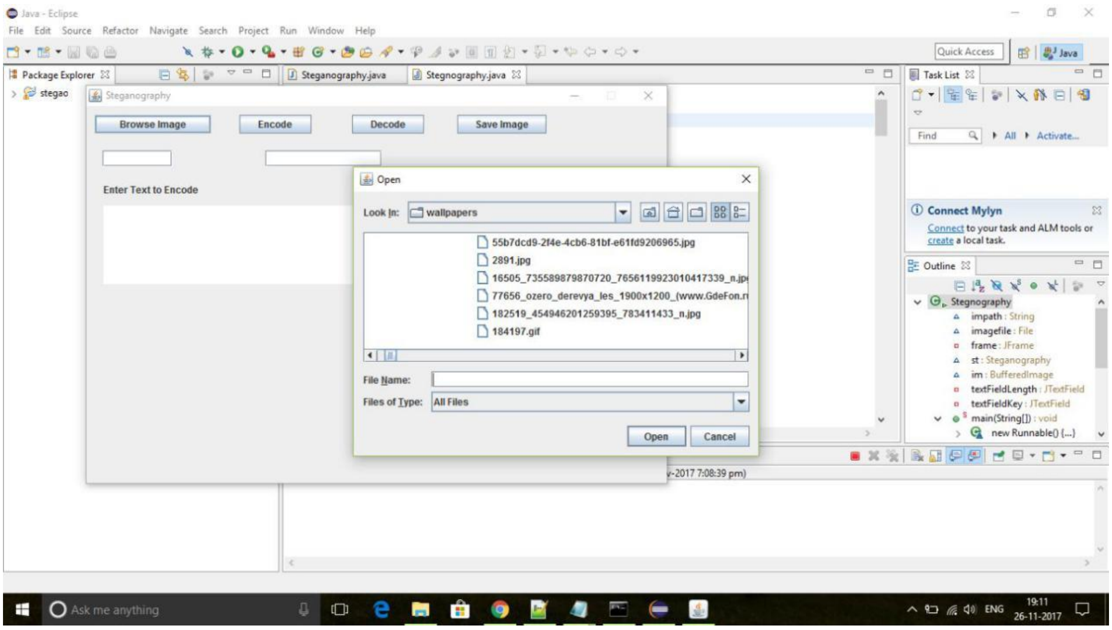
RESULT AND DISCUSSION

ENCRYPTION:

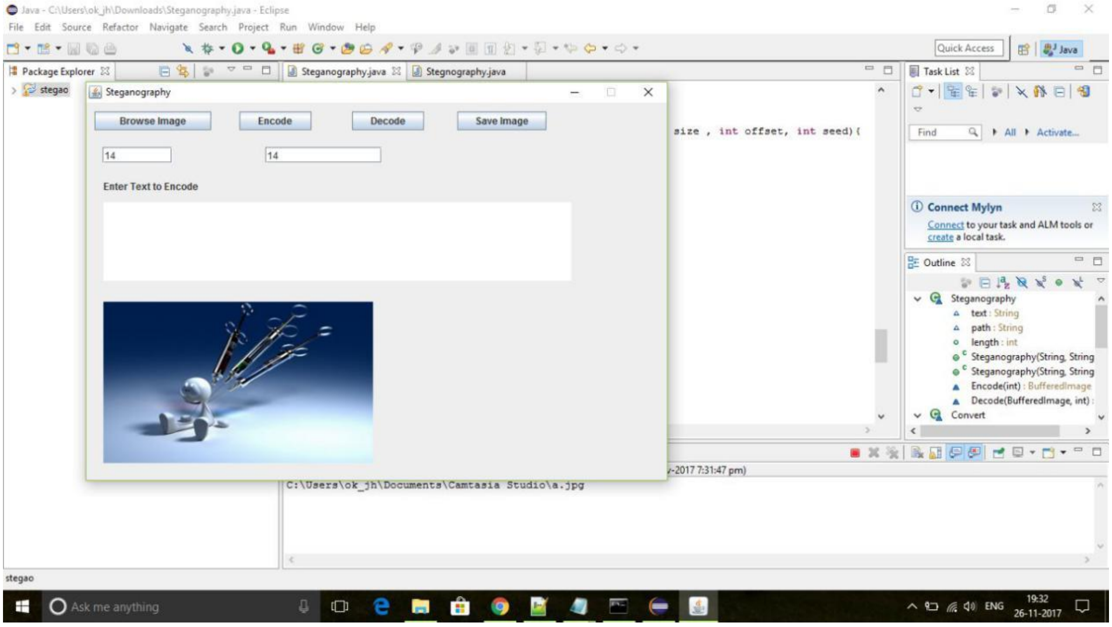
1.



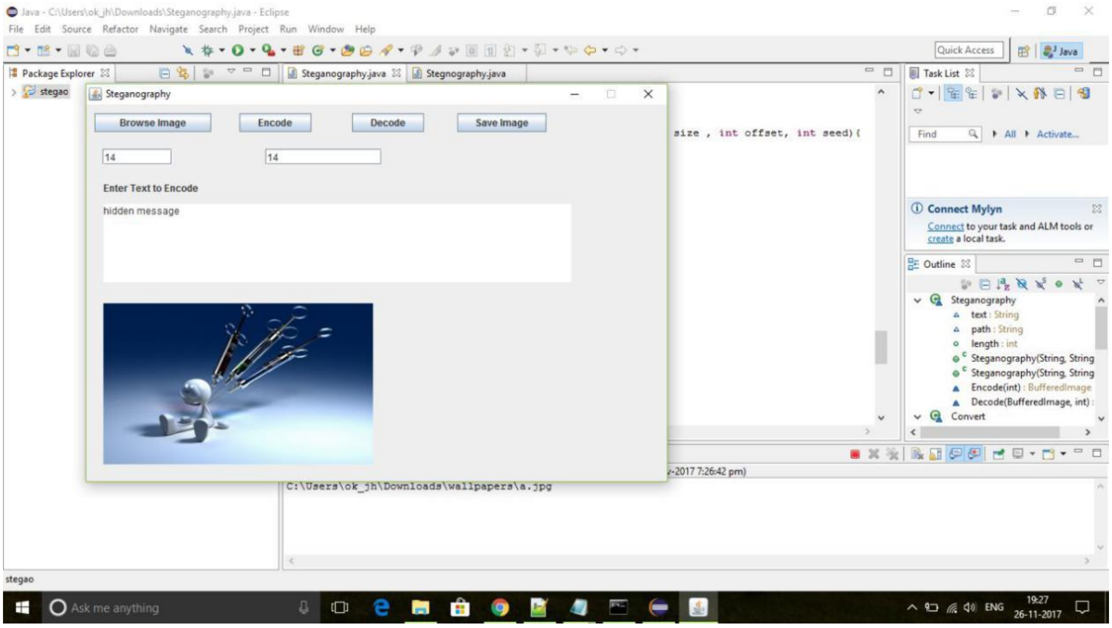
2.



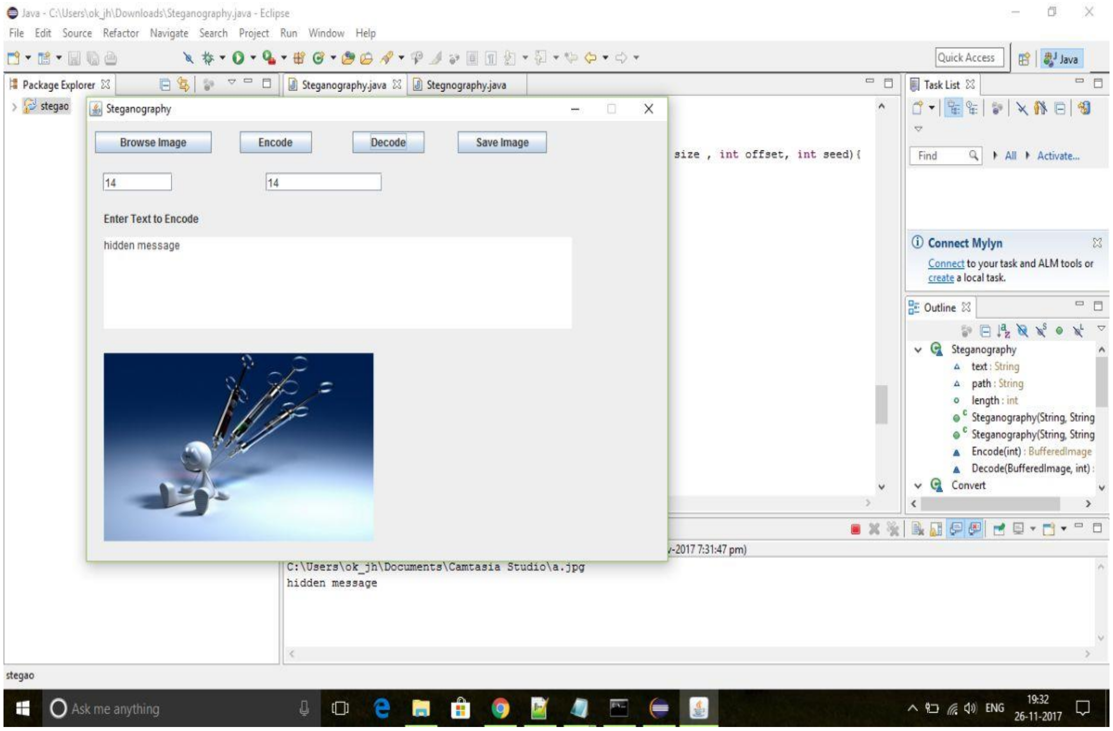
3.


Java - C:\Users\ok_jh\Downloads\Steganography.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
stegao
Steganography.java
Stegnography.java
Steganography
Browse Image
Encode
Decode
Save Image
14
14
Enter Text to Encode
size , int offset, int seed) {
Task List
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
Steganography
text : String
path : String
length : int
Steganography(String, String
Steganography(String, String
Encode(int) : BufferedImage
Decode(BufferedImage, int) :
Convert
C:\Users\ok_jh\Documents\Camtasia Studio\a.jpg
Ask me anything
ENG
19:32
26-11-2017

4.


Java - C:\Users\ok_jh\Downloads\Steganography.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
stegao
Steganography.java
Stegnography.java
Steganography
Browse Image
Encode
Decode
Save Image
14
14
Enter Text to Encode
hidden message
size , int offset, int seed) {
Task List
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
Steganography
text : String
path : String
length : int
Steganography(String, String
Steganography(String, String
Encode(int) : BufferedImage
Decode(BufferedImage, int) :
Convert
C:\Users\ok_jh\Downloads\wallpapers\a.jpg
Ask me anything
ENG
19:27
26-11-2017

DECRYPTION:



ADVANTAGES:

Improvement in security & image quality
A good invisibility
Less distortion after embedding process
Expected to be practical
Provides three layers of security

DISADVANTAGES:

Robustness is not achieved
Can be distorted by unintended users

CONCLUSION

What can we take from these findings? Well, the simple point is that steganography is not easy;with knowledge of the algorithm/method (which we should always assume), it is very hard to hide messages in an undetectable way. This difficulty increases with the size of the message and the desired robustness of the scheme — a single bit could be hidden trivially (and not robustly) by changing a random LSB of the image to alter the parity of the image's bits, but once we want to encode enough pseudorandom data to make a statistical attack possible, things swiftly become more difficult. The flaw in the systems discussed is that they assumed certain parts of an image (either least significant bits of LSBs or DCT coefficients) were pseudorandom when they in fact are not. A possible approach to future techniques is to investigate ways of finding pseudorandom data in cover works, possibly by applying focussed tests such as the chi-square test, and inserting information in those parts of the image. Unfortunately, the amount of such data in most cover works is likely to be small, as natural data tends to not be truly random, and good compression schemes will destroy such pseudorandom data, as it carries no important perceptual information that cannot be recreated. Even if, for example, we hide data in the thermal noise in a digital photo, this may change or destroy properties of the sensor fingerprint, and by examining other images from the same camera, the modification may be detected.

BIBLIOGRAPHY

[1] Eric Cole, Ronald D. Krutz, Consulting Editor (2003), Hiding in Plain Sight, Steganography and the Art of Covert Communication, Wiley Publishing, Inc.
[2] Stefan Katzenbeiser & Fabien A.P.Petitcolas(1999), Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Computer Security series, Boston, London.
[3] Fabien A.P.Petitcolas, Ross J.Anderson and Markus G.Kuhn, (1999) "Information Hiding – A Survey", Proceedings of the IEEE, special issue on protection of multimedia content, pp.1062-1078.
[4] Mamta Juneja and Parvinder Singh Sandhu, (2013) "A New Approach for Information security using an Improved Steganography Technique", Journal of Info.Pro.Systems, Vol 9, No:3, pp.405-424.
[5] P.Thiyagarajan, V.Natarajan, G.Aghila, V.Pranna Venkatesan, R.Anitha, (2013) "Pattern Based 3D Image Steganography", 3D Research center, Kwangwoon University and Springer 2013, 3DR Express., pp.1-8.
[6] Shamim Ahmed Laskar and Kattamanchi Hemachandran, (2013) "Steganography Based On Random Pixel Selection For Efficient Data Hiding", International Journal of Computer Engineering and Technology, Vol.4, Issue 2, pp.31-44.
[7] S.Shanmuga Priya, K.Mahesh and Dr.K.Kuppusamy, (2012) "Efficient Steganography Method To Implement Selected Least Significant Bits in Spatial Domain", International Journal of Engineering Research and Applications,, Vol2, Issue 3, pp. 2632-2637.
[8] B. Sharmila and R.Shanthakumari, (2012) "Efficient Adaptive Steganography For Colour Images Based on LSBMR Algorithm", ICTACT Journal on Image and Video Processing, Vol. 2, Issue:03, pp.387-392.
[9] Shweta Singhal, Dr.Sachin Kumar and Manish Gupta, (2011) "A New Steganography Technique Based on Amendment in Blue Factor ", International Journal of Electronics Communication and Computer Engineering, Vol.2, Issue 1, pp.52-56.
[10] Fahim Irfan et. Al. 's (2011) "An Investigation into Encrypted Message Hiding through Images Using LSB ", International Journal of EST,
[11] Rajkumar Yadav, (2011) "A Novel Approach For Image Steganography In Spatial Domain Using Last Two Bits of Pixel Values", International Journal of Security, Vol.5 Iss. 2 pp. 51-61.
[12] M.B.Ould MEDENI and El Mamoun SOUIDI, (2010) "A Generalization of the PVD Steganographic Method", International Journal of Computer Science and Information Security, Vol.8.No.8, pp156-159
[13] Weiqi Luo, Member, IEEE, Fangjun Huang, Member, IEEE, and Jiwu Huang, Senior Member, IEEE, (2010) "Edge Adaptive Image Steganography Based on LSB Matching Revisited", IEEE Transactions on Information Forensics and Security, Vol.5.No.2, pp.201-214.
[14] C.-H. Yang and M.-H. Tsai, (2010) "Improving Histogram-based Reversible Data Hiding by Interleaving Predictions", IET Image Processing, Vol.4. Iss. 4 pp. 223-234.
[15] Venkata Abhiram.M, Sasidhar Imadabathuni, U.Padmalochini, Maheedhar Imadabathuni and RamyaRamnath (2009), "Pixel Intensity Based Steganography with Improved Randomness", International Journal of Computer Science and Information Technology, Vol 2, No 2, pp.169-173.
[16] G.Sahoo & Rajesh Kumar Tiwari (2009) "Hiding Secret Information in Movie Clip: A Steganographic Approach", International Journal of Computing and Applications, Vol. 4, No.1, pp 103-110.
[17] Jaswinder Kaur, Inderjeet & Manoj Duhan, (2009) "A Comparative Analysis of Steganographic Techniques", International Journal of Information Technology and Knowledge Management, Vol. 2, No. 1, pp 191-194.
[18] Hao-Tian Wu and Jean-Luc Dugelay , (2009) "Steganography in 3D Geometrics and Images by Adjacent Bit Mapping", EURASIP Journal on Information Security, Vol. 2009, Article ID 317165, pp1-10.International Journal of Computer Science & Engineering Survey (IJCSES) Vol.4, No.6, December 2013