**Steganography**

Made By-

Tejasram Ramesh,A046

Soham Nanavati,A039

**Aim**

To hide a text inside an image using LSB steganography.

**Introduction**

Steganography, security under obscurity.

Steganography is the practice of concealing information under information like image, audio and video.

This repository contains a simple MATLAB implementation for using Steganography with images.

**LSB method**

The Least Significant Bit method of using Steganography involves hiding the secret message under the least significant bit of the pixels of the image. For a 24 bit color image, this would mean that 3 bits (one for each- Red, Green, Blue) contain bits from the secret message.

**Explanation**

Let the first few pixels of the original image (24bit color) be-

10101011.10001011.010101000, 10101001.00001100.01011110, 10100111.11001101.01011001

Let the secret message be 'A'.  
'A' translates to '01000001' in (8 bit) ASCII.

To *conceal* the secret message inside the image, we change the last bits of the image like so-

**A = 0 1 0 0 0 0 0 1**

1010101**0**.10001011.010101000, 1010100**0**.00001100.01011110, 1010011**0**.11001101.01011001

(**Bold** digits show changed bits)

The LSB Method does not require a key to be shared prior to the message being transferred between the sender and the recipient.

In addition, a delimiter needs to be used to stop decoding after the message length has been reached. An alternate approach is to specify the size of the message before the actual message.

**Usage**

**Encoding**

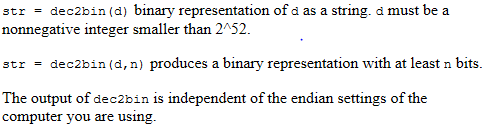
* Change message.txt to store your secret message.
* Replace image.png with the image you want to use.
* Execute the encode.m script with PATH having the corresponding directory in MATLAB.

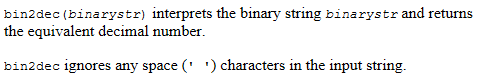
This would produce stego.png which is the new image with the hidden secret text.

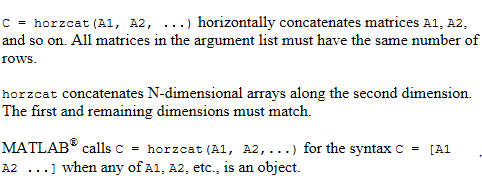
**Decoding**

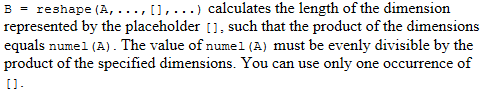
* With stego.png existing, execute the decode.m script with PATH having the corresponding directory in MATLAB.
* The secret text is displayed in the command window and saved inside the secret.txt file.

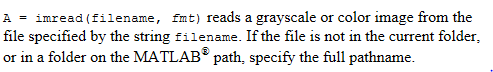
**FUNCTIONS**



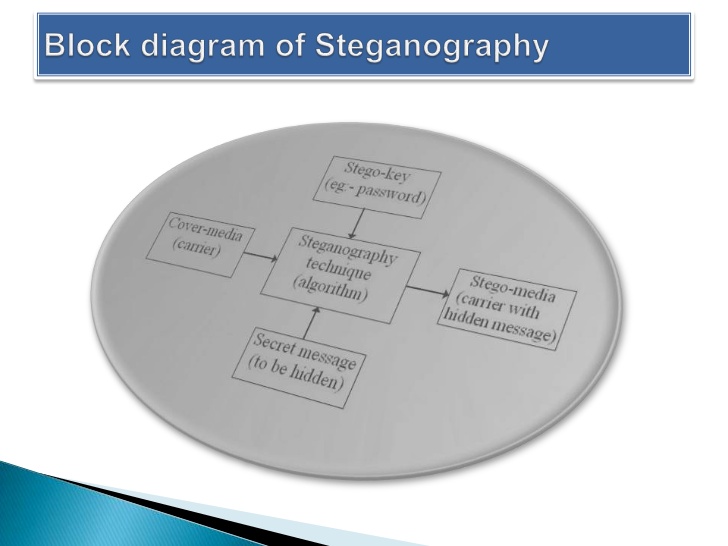




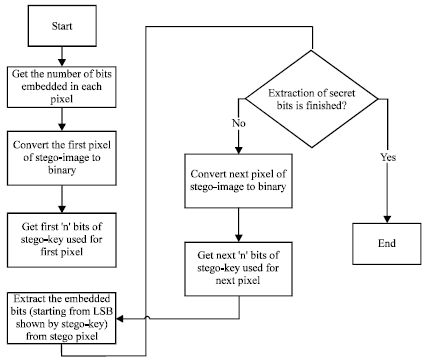




**Block diagram**



**FLOWCHART**



**ENCODE**

clc

im1 = imread('image.png');

message = fileread('message.txt');

message\_binary = reshape(dec2bin(message, 8)', 1, []);

message\_size = length(message\_binary);

[l,w,b] = size(im1);

cover\_size = l\*w\*b;

if (message\_size > cover\_size)

disp('Message too large for image.');

end

im2 = im1;

i = 1; j = 1; k=1;

for a = 1 : length(message\_binary)

if (k > 3)

k = 1;

j = j + 1;

end

if (j > w)

i = i + 1;

j = 1;

end

temp\_pixel = dec2bin(im1(i, j, k), 8);

LSB = temp\_pixel(8);

SM = message\_binary(a);

if (LSB ~= SM)

if (LSB == '1')

im2(i, j, k) = im1(i, j, k) - 1;

else

im2(i, j, k) = im1(i, j, k) + 1;

end

end

k = k + 1;

end

imwrite(im2, 'stego.png');

figure;

subplot(1,2,1), imshow(im1), title('Original Image');

subplot(1,2,2), imshow(im2), title('Secret Image');

**DECODE**

delimiter = '%';

clc

im1 = imread('stego.png');

% figure(1), imshow(im1); title('Original Image');

[l,w,b] = size(im1);

cover\_size = l\*w\*b;

message = '';

char\_stream = '';

i = 1; j = 1; k=1;

while (true)

if (k > 3)

k = 1;

j = j + 1;

end

if (j > w)

i = i + 1;

j = 1;

end

if (i > l)

break;

end

temp\_pixel = dec2bin(im1(i, j, k), 8);

LSB = temp\_pixel(8);

char\_stream = strcat(char\_stream, num2str(LSB));

if length(char\_stream) == 8

new\_char = char(bin2dec(char\_stream));

message = horzcat(message, new\_char);

if (new\_char == delimiter)

break;

end

char\_stream = '';

end

k = k + 1;

end

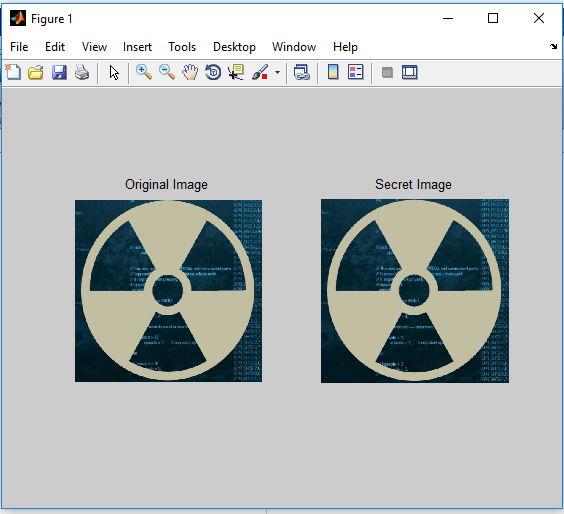
secret\_file = fopen('secret.txt','wb');

fwrite(secret\_file, message);

fclose(secret\_file);

disp(message);

**OUTPUT**

**Of encode:**

**Of decode:-**

