ISSN 2348 – 8034
Impact Factor- 3.155

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

IMAGE STEGANOGRAPHY: XOR MASKING IN LSB

Souma Pal[*1] and Prof. Samir Kumar Bandyopadhyay[2]
[*1]Department of Computer Science & Engineering, University of Calcutta, India
[2]Department of Computer Science & Engineering, University of Calcutta, India, skb1@vsnl.com

ABSTRACT

Steganography is going to achieve more importance in our advanced civilization due to maintain secrecy of communication over the Internet. It is an advanced stage of cryptography. In cryptography, a message which is to be sent to the destination is embedded into an another message using encryption algorithm, resulting the encrypted message is transmitted to the receipient(s) over the Network and receiver(s) decrypt(s) the original message using decryption algorithm. In steganography, the same technique is followed but in more advanced way. The encrypted message is embedded with another text, image, audio, video or protocol and then transmitted to the receipient(s) through the Internet in such a way that none except the receipient(s) can understand the existence of message within the image, they simply observe the image nothing more. This paper highlights the Image steganography by using xor masking in LSB of an image.

Keywords- *Image steganography, XOR masking, Least Significant Bit(LSB).*

I. INTRODUCTION

Steganography is a technique of information hiding into another information/message in such a way that everyone except the receipient(s) is unaware of the existence of original message and sends the embedded information to the internet. Though there are different kinds of steganography methods like text, image, audio etc, but the key point of this paper is Image Steganography [1-2].

Generally, image steganography is method of information hiding into a cover image and yields a stego-image. This stego-image are sent to the destination over the network. The other parties except the recipient (s) are totally unaware the existence of message within the stego-image. They cannot guess about the message(s) that is/are to be hidden within the stego-image. After the receiving the stego-image, the receiver(s) extract(s) the original image with using the reverse method.

Methodology:

(a)Message:- Data/information that is to be sent.

(b)Cover image:- Any image(JPEG format) in which message is hidden.

(c)Stego-image:- Message + cover image.

(d)Stego-key:- Reverse procedure for extracting the original message from the stego-image.

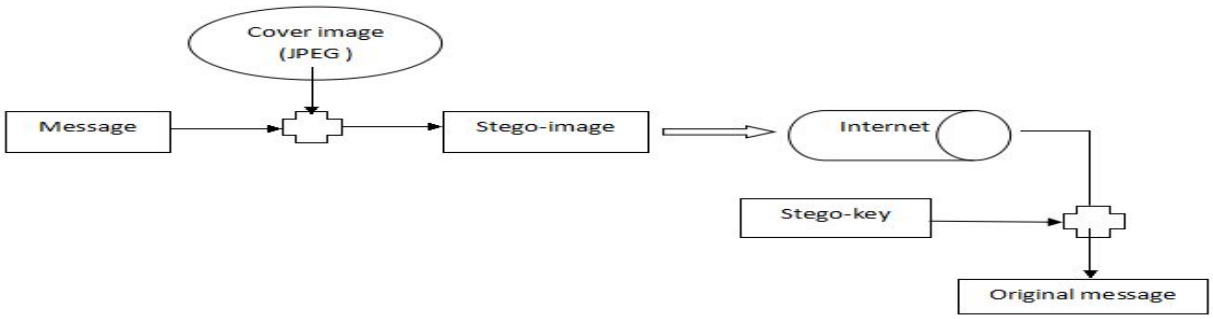

Fig: Procedure of Image Steganography

ISSN 2348 – 8034
Impact Factor- 3.155

Pixel Processing: The data value will be hidden within the image. Least significant bit is used for patching of data within image because the intensity of image can be changed simply by changing 0 to 1or 1to 0.

11110001 11110000 or 10111000 10111001

Least significant bit is selected because the change is only one bit ,i.e., LSB so that the intensity of image is not effected too much and the other parties cannot recognize the change of picture. So the picture is transmitted.The procedure is written below,-

a) Input 8 bit data value and cover image(JPEG format).

b) Select pixel region (continuously or discretely) of the cover image.

c) Perform bit-wise XOR masking.

d) Change the LSB of the selected pixels.

e) Create modified image.

Data value is always 8 bit because ASCII range is 0-255.

Data value is 240 whose binary representation is 11110000. It is embedded into the particular pixel of the cover image using XOR masking in LSB method.

Program:

```
%Create Stego-image using Data value and cover image%

%Read cover image%

>>o=imread('originalroseimage.jpg');

%Select Pixel region%

>> r=[401 256 95];

>> c=[220 136 415];

>> pixel=impixel(o,c,r)                    %Pixel value in decimal form%

pixel =

   21    36    13

   31    58    17

   46    78    37

>> pixel1=pixel;

>> binary=dec2bin(pixel1)                  %Convert into binary form%

binary =

0010101

0011111

0101110
```

ISSN 2348 – 8034
Impact Factor- 3.155

```
0100100

0111010

1001110

0001101

0010001

0100101

>> bin=[00010101 00011111 00101110 00100100 00111010 01001110 00001101 00010001 00100101;];

>> %Calculate LSB%

>> x_lsb=mod(bin,10)

x_lsb =

    1    1    0    0    0    0    1    1    1

>> %Enter 8-bit data value%

>> m=240;

>> mbin=dec2bin(m)                    %Convert it into binary form%

mbin =

11110000

>> t1=[];                             %String to number conversion%

>> for i=1:length(mbin)

s=str2num(mbin(i));

t1=[t1 s];

end

>> t1

t1 =

    1    1    1    1    0    0    0    0

>> %XOR Masking%

>> d1=[];

>> for i=1:length(mbin)

d=bitxor(x_lsb(1,i),t1(1,i));

d1=[d1 d];

end
```

ISSN 2348 – 8034
Impact Factor- 3.155

```
>> d1

d1 =

    0    0    1    1    0    0    1    1

>> %Change the LSB%

bin

bin =

  Columns 1 through 5

      10101      11111      101110      100100      111010

  Columns 6 through 9

    1001110       1101       10001      100101

>> binarray=[0 0 0 1 0 1 0 1;0 0 0 1 1 1 1 1;0 0 1 0 1 1 1 0;0 0 1 0 0 1 0 0;0 0 1 1 1 0 1 0;0 1 0 0 1 1 1 0;0 0 0 0 1 1 0
1;0 0 0 1 0 0 0 1;0 0 1 0 0 1 0 1;]

binarray =

    0    0    0    1    0    1    0    1

    0    0    0    1    1    1    1    1

    0    0    1    0    1    1    1    0

    0    0    1    0    0    1    0    0

    0    0    1    1    1    0    1    0

    0    1    0    0    1    1    1    0

    0    0    0    0    1    1    0    1

    0    0    0    1    0    0    0    1

    0    0    1    0    0    1    0    1

>> for i=1:length(mbin)

binarray(i,8)=d1(1,i);

end

>> binarray

binarray =

    0    0    0    1    0    1    0    0

    0    0    0    1    1    1    1    0

    0    0    1    0    1    1    1    1
```

ISSN 2348 – 8034
Impact Factor- 3.155

0 0 1 0 0 1 0 1

0 0 1 1 1 0 1 0

0 1 0 0 1 1 1 0

0 0 0 0 1 1 0 1

0 0 0 1 0 0 0 1

0 0 1 0 0 1 0 1

%String convertion%

>> binarray1=[00010100 00011110 00101111 00100101 00111010 01001110 00001101 00010001 00100101;];

>> binstring=[00010100;00011110;00101111;00100101;00111010;01001110;00001101;00010001;00100101;];

>> binstring=num2str(binstring)

binstring =

10100

11110

101111

100101

111010

1001110

1101

10001

100101

%Binary to Decimal conversion%

>> decimal=bin2dec(binstring)

decimal =

20

30

47

37

58

78

13

ISSN 2348 – 8034
Impact Factor- 3.155

```
   17
   37
%Change the pixel value%
>> i=1;
>> for j=1:3
for k=1:3
if(i<=9)
pixel1(k,j)=decimal(i,1);
i=i+1;
end
end
end
>> %Original Pixel value%
>> pixel
pixel =
   21    36    13
   31    58    17
   46    78    37
>> %Modified Pixel value%
>> pixel1
pixel1 =
   20    37    13
   30    58    17
   47    78    37
%Change the pixel value within the image%
>> modi=o;
>> r=[401 256 95];
>> c=[220 136 415];
>> for i=1:3
x=r(i);
```

ISSN 2348 – 8034
Impact Factor- 3.155

```
y=c(i);
j=1;
for k=1:3
if(j<=3)
modi(x,y,j)=pixel1(i,k);
j=j+1;
end
end
end
>>%Write it into a new file%
>> imwrite(modi,'modifiedroseimage.jpg','Mode','Lossless');
%Decrypt the original Data using Stego-key%
%Read cover image file and stego-image file%
>> ori=imread('originalroseimage.jpg');
>> modi=imread('modifiedroseimage.jpg');
>> r=[401 256 95];                          %Read selected region%
>> c=[220 136 415];
>> ori_pixel=impixel(ori,c,r)               %Pixel value of original image%
ori_pixel =
   21    36    13
   31    58    17
   46    78    37
>> modi_pixel=impixel(modi,c,r)             %Pixel value of stego-image%
modi_pixel =
   20    37    13
   30    58    17
   47    78    37
>> ori_binary=dec2bin(ori_pixel)            %Binary conversion%
ori_binary =
0010101
```

ISSN 2348 – 8034
Impact Factor- 3.155

```
0011111

0101110

0100100

0111010

1001110

0001101

0010001

0100101

>> ori_bin=[00010101 00011111 00101110 00100100 00111010 01001110 00001101 00010001 00100101;];

>> x_lsb_ori=mod(ori_bin,10)

x_lsb_ori =

    1    1    0    0    0    0    1    1    1

>> modi_binary=dec2bin(modi_pixel)

modi_binary =

0010100

0011110

0101111

0100101

0111010

1001110

0001101

0010001

0100101

>> modi_bin=[00010100 00011110 00101111 00100101 00111010 01001110 00001101 00010001 00100101;];

>> x_lsb_modi=mod(modi_bin,10)

x_lsb_modi =

    0    0    1    1    0    0    1    1    1

>> %Calculate the original Data value%

>> data1=[];

>> for i=1:8
```

ISSN 2348 – 8034
Impact Factor- 3.155

```
s=bitxor(x_lsb_ori(1,i),x_lsb_modi(1,i));
data1=[data1 s];
end
>> data1
data1 =
    1     1     1     1     0     0     0     0
>> data_bin=[11110000;];
>> data_str=num2str(data_bin);
>> data=bin2dec(data_str)
data =
   240
```

Output:

Cover-image:-

Fig:originalroseimage.jpg

ISSN 2348 – 8034
Impact Factor- 3.155

Stego-image:-

Fig: modifiedroseimage.jpg

CONCLUSION

This paper introduces a brief idea about Image steganography. This procedure is simple and secure. This technique has hidden the data from outside world. In this paper, XOR masking is done in LSB of the cover image(JPEG).

REFERENCES

1. *Mr. Vikas Tyagi, "Data Hiding in Image using least significant bit with cryptography."*
2. *Mehdi Hussain and Mureed Hussain, "A Survey of Image Steganography Techniques."*