



## Data Hiding in Image using least significant bit with cryptography

Mr. Vikas Tyagi

Assistant Professor, Computer Science, IMSEC, Ghaziabad (India)

itengg.vikas@gmail.com

**Abstract:** - To increase the security of messages sent over the internet steganography is used. This paper discussed a technique used on the LSB (least significant bit) and a new encryption algorithm. By matching data to an image, there is less chance of an attacker being able to use steganalysis to recover data. Before hiding the data in an image the application first encrypts it.

**Keywords:**-Steganography, LSB (least significant bit), Encryption, Decryption, Data Security.

### 1. Introduction

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. It is the art of concealing a message in a cover without leaving a remarkable track on the original message. It Pronounced "ste-g&-'nā-gr&-fE" and Derived from Greek roots "Steganos" = cover "Graphic" = writing Its ancient origins can be traced back to 440 BC. Histories the Greek historian Herodotus writes of a nobleman, Histaeus, who used steganography first time.<sup>3</sup>

The goal of Steganography<sup>1</sup> is to mask the very presence of communication making the true message not discernible to the observer. As steganography has very close to cryptography and its applications, we can with advantage highlight the main differences. Cryptography is about concealing the content of the message. At the same time encrypted data package is itself evidence of the existence of valuable information. Steganography goes a step further and makes the ciphertext invisible to unauthorized users.<sup>4</sup>

Two other technologies that are closely related to steganography are watermarking and fingerprinting. These technologies are mainly concerned with the protection of intellectual property. But steganography is concern with the hiding of text in information like image, text, audio, and video.

### 2. Type of steganography

There are 4 different types of steganography

- 1. Text
- 2. Image
- 3. Audio
- 4. Video
- 5. Protocol

**Text steganography** using digital files is not used very often since text files have a very small amount of redundant data.

**Audio/Video steganography** is very complex in use.

**Image steganography** is widely use for hiding process of data. Because this is quite simple and secure way to transfer the information over the internet. Image steganography has following types:

- \* Transform domain
  - 1) Jpeg
  - 2) Spread Spectrum
  - 3) Patch Work
- \* Image domain
  - 1) LSB and MSB in BMP
  - 2) LSB and MSB in JPG

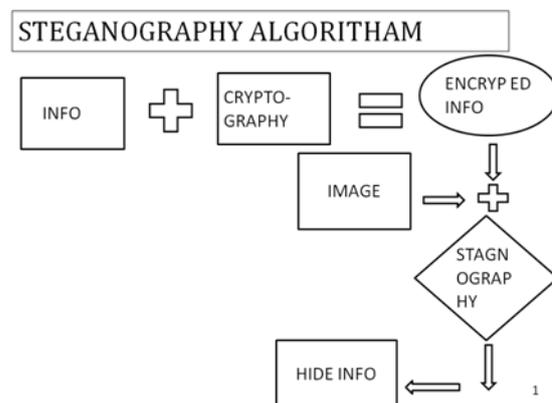


Fig.1 Description of Algorithm

### 3. CRYPTOGRAPHY ALGORITHM

Normal text message:- Saurabh

Key:-hello

- 1. Change the key and data in to ASCII format.

Eg. hello is changed in B[5]={ 8, 5, 9, 9, 13}

saurabh is changed in A[20]={ 19, 1,21,1,18,2,8}

- 2. Pad the Normal message according to the length of the key .

Eg. Saurabh has 7 char. In it and the key has 5 letters ,so first five letter of message will change according to the key but in the end we have only two letter left so we pad **p** letter (x or y or z) for padding to make exact length pairs.

Saurabh

Saura **bhxxx**

A[20]={19,1,21,1,18,2,8,24,24,24}

m = length of key

#### \*ENCRYPTION ALGORITHM

1. take two arrays flagtxt and flagkey of size of length of text and key and fill it with zeros.
2. Do this process till the length of key

#### A: Process for encryption of data by the key

```

for k=1 to m
  J=1
  for i=1 to n      ( n is length of padded text)
  {
    if( j>m)
    {
      j=1
      a[i]=a[i] + b[j]
      j++
    }

    else
      { a[i]=a[i] + b[j]
      j++
      }
  }

```

End for

#### B: Process of hiding of key

Do

```

for j=1 to m-1
  b[j]=b[j]+b[j+1]
end for
b[m-1]=b[m-1]+b[1]
End for

```

#### C: Change the array A and B in to character form

Eg.

```

For i=1 to n
while a[i]>256
a[i]=a[i]-256
flagtxt[i]+=1
end while
end for
for i=1 to m
while b[i]>256
b[i]=b[i]-256
flagkey[i]+=1
end if
end for

```

\*DECRYPTION ALGORITHM (this is reverse process of encryption)

#### A: Change the encrypted data in ASCII format

Eg

A[20]={20,143,29,231,256}

B[20]={10,2,230,19,23}

#### B: Decryption of data and key

```

for i=1 to n ( n is length of padded text)
while flagxt[i]!=0
a[i]=a[i]+256
flagxt[i]--
end while
end for

```

```

for i=1 to m
while flagkey[i]!=0
b[i]=b[i]+256
flagkey[i]--
end while
end for

```

```

for k=1 to m
b[m]=b[m]-b[1]

```

```

for j=m-1 to 1
b[j]=b[j]-b[j+1]

```

end for

```

j=1
for i=1 to n
if(j>m)
j=1
a[i]=a[i]-b[j]
end if
end for
end for

```

## 4. Pixel Processing

After the converting our information in secret code or encrypted form we need to patch that data in the image. We use least significant bit for the patching of data because of following reason.

- Because the intensity of image is only change by 1 or 0 after hiding the information.
- Change in intensity is either 0 or 1 because the change at last bit .e.g.

11111000      11111001

The change is only one bit so that the intensity of image is not effected too much and we can easily transfer the data.

## 5. Steps To Insert Data In Image

- Take an input image.
- Find out the pixel values.
- Select the pixel on which we want to insert data.

This process of selection of pixel is done as user's choice he may choose pixel continuous or alternate or at a fixed distance.

- Insert the data values in pixels eg.  
For example a grid for 3 pixels of a 24-bit image can be as follows:

```
00101101 00011100 11011100
10100110 11000100 00001100
11010010 10101101 01100011
```

When the number **200**, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

```
00101101 00011101 11011100
10100110 11000101 00001101
11010010 10101100 01100011
```

**6. Results**



Fig. 2 Original Image

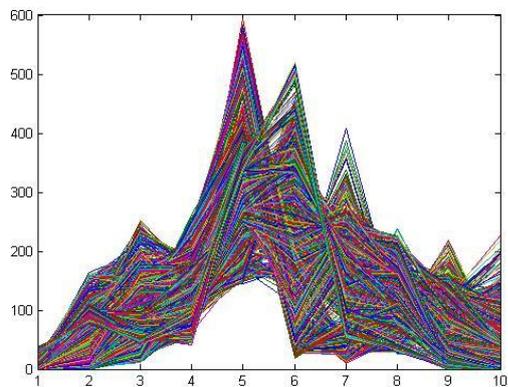


Fig. 3 Histogram of Original Image



Fig. 4 Output Image

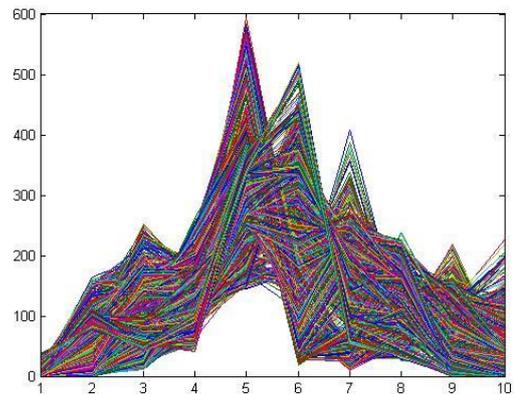


Fig.5 Histogram of Output Image

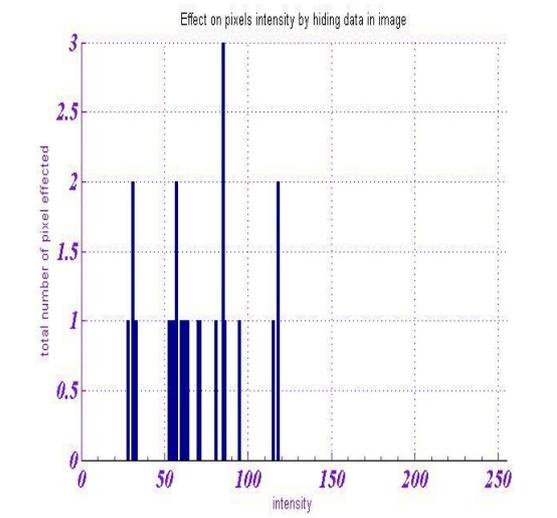


Fig. 6 Effect on intensity of pixels by hiding of data

security. Research in this field has already begun. Next to steganography, one of the most active fields of research is mass detection tools for hidden contents. The problems are really big. At first, known statistical tests are fragile and for many embedding schemes we still do not know which properties to test. At second, the today traffic in public networks is so overwhelming, that is too hard to rigorously check each file.

**Table 1**  
**Comparison between various steganography algorithms**

NO	BASE	Steganography without cryptography	Steganography with Other cryptography technique	Steganography with our cryptography technique
1.	Security	One level security	Two level security	Two level security
2.	Key size	No key present	Fixed size of key	Random size of key
3.	Steps involve in encryption of data	No step	Fixed step	Depend on the key size
4.	Brute force attack	No need	Can possible	Very hard to attack

**7. Conclusion**

This paper is a short introduction to the world of steganography. We have shown how the simplest methods work and how they can be explored. We have used symmetric encryption algo to provide more

**REFERENCES**

1. Eric Cole, Ronald D. Krutz, "Hiding in Plain Sight: Steganography and the Art of Covert Communication", Wiley Publishing Inc. (2003).
2. David Kahn, "The History of Steganography", Proc. of First Int. Workshop on Information Hiding,
3. Cambridge,UK, May30-June1 1996, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.), pp.1-7 Benderr, D. Gruhl, N. Morimoto and A.Lu, "Techniques for Data Hiding", IBM System's Journal, Volume 35, Issue 3 and 4, 1996, p.p., 313-336.
4. Artz, D, 'Digital Steganography: Hiding data within Data', IEEE Internet Computing, May/June 2001.
5. Wang, Y., Moulin, P. , 'Steganalysis of Block-DCT Image Steganography', BeckmanInstitute, CSL & ECE Department, University of Illinois at Urbana-Champaign, 2003.