Impact of the Nibble Substitution for Image Steganography - An Enhanced Lsb Algorithm

Information Technology

KEYWORDS : Steganography, Stego, Efficiency, intensity

Viral Patel Institute of Technology, Nirma University

Dheeraj Dadhich Institute of Technology, Nirma University

ABSTRACT

Steganography can be used to hide the information inside the various objects like image, sound, video and text. Image Steganography can be well understood by the LSB algorithm. By using LSB algorithm it will allows us to hide the information inside the image but for hiding very few bits we need large bytes of image as a carrier image. This is the limitation of the image Steganography when we are using LSB algorithm. We have tried to propose the method for enhance LSB algorithm which will be four times efficient in terms of hiding the information.

INTRODUCTION

Steganography can be used to hide the information inside the object. Here we can use various types of objects like image, audio and text. The basic concept of the steganography is shown in the figure.

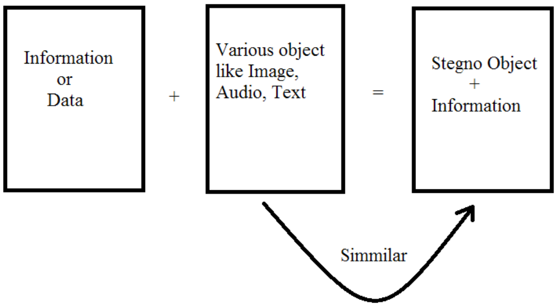

Fig - 1 : Steganography Basic Principle

There are various types of Steganography,

Text Steganography [1] : Information is hidden within the text document using various stenographic techniques like,

[1] Character Marking
[2] Invisible Ink
[3] Pin Puncture
[4] Type writer correction ribbin

Audio and Video Steganography [2]: Secret information is hidden by altering the physical properties of the audio & video file.

[1] LSB Algorithm Method
[2] Spread Spectrum
[3] Echo Signal Technique
[4] Phase Shift method

Image Steganography:

The physical property of the image to hide the information inside the image.

[1] LSB Algorithm method [1]
[2] Matching method technique [3]

Our focus of the work is based on enhancing the rate of hiding the secret information in an image file. The main objective is to hide maximum number of bits in minimum size of image.

LITERATURE SURVEY

Any image is represented by the combination of pixels and every pixel is a combination of Red, Green and Blue (RGB) Color with its hexadecimal Value. As an example 24 Bit representation for single pixel then each color (R, G, B) is represented by 8 Bits. As shown in figure-2 pixel with Red color is represented and Blue and Green color value is Zero.

From figure we can say Red color with 255 (1111 1111) and 254 (1111 1110) is looking almost same as human eyes can't detect it as a two different colors.

(R-255,G-0,B-0)
(R-254,G-0,B-0)

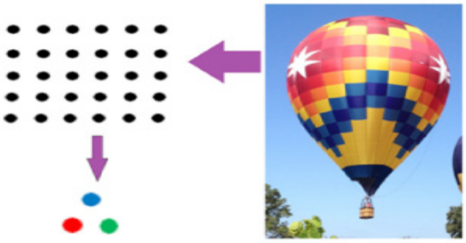

Fig-2: Working Diagram of LSB Algorithm

And hence if we alter the last bit of the image with our information bit, we can store it in the pixel and at receiver side, we can decode it in a reverse manner. Hence we can hide the 1 bit in each color as LSB bit. Thus 3 bits can be hidden at each pixel.

Example:

The letter "A" with the binary value "10000011" can be embedded in three pixels (9 bytes) with the following binary value (refer to Table 1): [3]

Table: 1 Original Bit in Pixel

Pixel Number	Red	Green	Blue
Pixel No. 1	10000111	11101001	11001000
Pixel No. 2	10000111	11001000	11101001
Pixel No. 3	11001000	10000111	11101001

Inserting the binary value for the letter "A" in the three selected pixels would result in figure 2.As shown by Table-1, the underlined bits are the actually used eight bits in the stego implementation.

Table: 2 Modified Bit in Pixel

Pixel Number	Red	Green	Blue
Pixel No. 1	10000111	11101000	11001000
Pixel No. 2	10000110	11001000	11101000
Pixel No. 3	11001000	10000110	11101001

Disadvantage of this method is, for hiding few bits we need large size of the carrier image. To overcome this problem we introduce one method that is enhancement to the LSB Algorithm Method.

PROPOSED METHOD

Any color image is represented with 24 bits and in this pixel every color is of 8 bits. Here, the color value is more reflected with its higher bit and hence if we change the lower order bits in the color then also it will not differ then its original color representation.

So if we are using 24 Bit representation for single pixel then each color (R, G, B) is represented by 8 Bits.

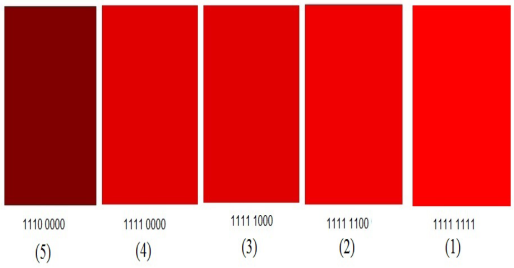

Fig -3: Different Value of Red Color

Here the right most red color is shown with the binary value 1111 1111 (hex value 255).If we alter the last 2 LSB bits with 0 then the

Figure-2 shows this color with hexadecimal value 252. Now if the last 3 LSB bits with 0 it is shown in figure 3. This color with hex value 248. The last 4 LSB bit with 0 color is shown in figure 4 with hexadecimal value 240. There are nominal differences between every image and it is difficult to detect with naked eyes.

Now if we alter the 5th bit then red color shown in figure 5 with hex value 224. Here the color is change such a way that it will alter the original image and hence for hiding the information we can alter up to 4 bits in the image. So by using this method for the image Steganography we can hide 4 times more bits then the original LSB algorithm method.

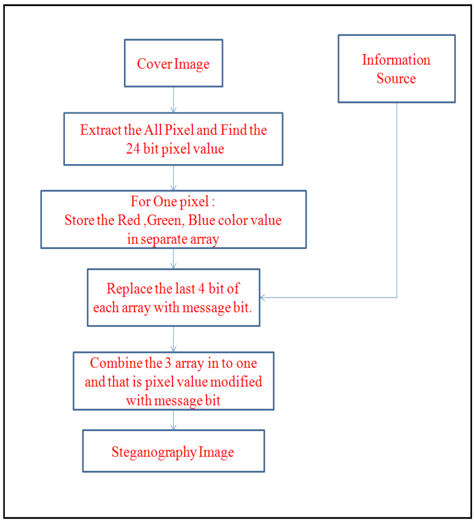
Cover Image
Information Source
Extract the All Pixel and Find the 24 bit pixel value
For One pixel :
Store the Red ,Green, Blue color value in separate array
Replace the last 4 bit of each array with message bit.
Combine the 3 array in to one and that is pixel value modified with message bit
Steganography Image

Fig – 4: Proposed Method Algorithm

Example:

The letter "ABCD" with the binary value "00000001 00000010 00000011 00000100 " can be embedded in three pixels (9 bytes) with the following binary value (refer to figure 1):

Table 3: Original Bit Representation

Pixel Number	Red	Green	Blue
Pixel No. 1	10000111	11101001	11001000
Pixel No. 2	10000111	11001000	11101001
Pixel No. 3	11001000	10000111	11101001

Inserting the binary value for the letter "ABCD" in the three selected pixels would result in figure 2. As shown by Table - 4, the underlined bits are the only eight actually used bits in the stego implementation.

Table 4: Modified Bit Representation

Pixel Number	Red	Green	Blue
Pixel No. 1	10000000	11100001	11000000
Pixel No. 2	10000010	11000000	11100011
Pixel No. 3	11000000	10000100	11101001

IMPLEMENTATION

Simulation Parameter:-

Let we have 6 * 6 inches image with 100 dpi value. So total number of pixels that we have in width and height is 600. Also assume that each pixel is represented using the 24 bit.

LSB Algorithm:

Each pixel can hide the 3 bit. Total number of pixel in the image is 600 * 600.So total amount of information that can be hidden 600 * 600 * 3 bit = 135 Kbyte.

Propose method:

Each pixel can hide 12 bit (3 bit in each color).Total number of pixel in the image is 600 * 600.So total amount of information can be hidden is 600 * 600 * 12 bit = 540 Kbytes. Hence the propose method is 540/135 = 4 times higher than storing the information inside the image.

SIMULATION

Before Stego

After Stego

The proposed method is implemented to measure the performance. In our implementation, we used 600 * 600 bit map image file to hide data of various size. The proposed method was coded in C#, and run on a computer with 2.0 GHz Intel Core(TM)2 Duo CPU and 4GB RAM.

CONCLUSION

Experimental results shows that the Implementation for the Nibble Substitution on image leads to the visible difference as intensity is changed of the stego image but alternatively it is observed that if the image are very dark (Black or White) it is impossible for a naked eye to detect the original & Steganography image. Hence for hiding the large amount of data inside small size of image can be possible by choosing black or white image as a carrier image.

REFERENCE

[1] Cryptography and network security , 4th | Edition by William Stalling. | [2] Soumyendu Das , Subhendu Das,Bijoy | Bandyopadhya,Sugata Sanyal "Steganography and Steganalysis: Different | Approaches" | [3] Mohammad Ahmad Alia and Abdelfatah A. Yahya - "Public Key Steganography Based | on Matching Method " European Journal of | Scientific Research ISSN 1450-216X | Vol.40 No.2 (2010), pp.223-231 | [4] Shawn D. Dickman. "An Overview of | Steganography" | [5] T. Morkel,J.H.P. Eloff,M.S. Olivier "An | overview of image steganography" | [6] Arvind Kumar and Km. Pooja – | "Steganography- A Data Hiding Technique" | [7] Neil F. Johnson and Sushil Jajodia – | "Exploring Steganography:Seeing the | Unseen" | [8] Andreas Westfeld and Andreas Pfitzmann – | "Attacks on Steganographic Systems |