Информационные процессы, Том 6, № 1, 2006, стр. 1–10
© *2006 Alkhraisat Habes.*

Information Hiding in BMP image Implementation, Analysis and Evaluation

Alkhraisat Habes

Saint Petersburg Institute for Informatics and Automation, Russian Academy of Sciences, Saint Petersburg, Russia

Received February 26, 2006

Abstract—Steganography comes from the Greek words steganos, roughly translating to "covered writing". Steganographic techniques allow one party to communicate information to another without a third party even knowing that the communication is occurring. The ways to deliver these "secret messages" vary greatly. This paper explores several methods in detail, and attempts to test them out in code, and in practice, through several examples.

"The goal of steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second secret message present."

1. INTRODUCTION

There are important distinctions to be made between cryptography and steganography. Cryptography changes the message such that even if it is overheard by a third party, it would be unintelligible. Steganography, on the other hand, involves hiding message in such a way that the casual observer should not be able to detect the hidden information. Steganography is a good choice in situations where it is a priority to disguise the occurrence of communication. Seemingly meaningless data can contain complex details, maps, or text. Steganography is often combined with cryptography to provide an additional layer of security.

The first recorded example of steganography occurred in ancient Greece. Persia was on the verge of invading Greece and Demeratus wanted to send warning to Sparta. To sneak word of the impending invasion out of Persia he carved a message into the wood underneath the wax in blank wax-tablets and had the tablets sent to Persia. To the casual observer the tablets appeared blank, but when the wax was removed the message was revealed (Mulhbauer). During World War II, it was imperative to find ways of sending messages covertly. Axis forces developed a method which produced a "microdot" which, when examined under a magnifying glass, would reveal vast amounts of text in a single dot. This was certainly an unpleasant surprise for the Allies. The development of the microdot caused the Allies to heavily inspect or restrict mail service throughout Europe.

Today, steganographic techniques are often used by copyright holders who wish to combat piracy and theft. Images, video, and music can be encoded with information that can be used to identify the work as being the property of an individual or corporation. These encodings are often called watermarks. Watermarked media can be distributed on the internet while allowing copyright holders to be able to maintain their intellectual property. Commercially available watermarking technologies use robust techniques to encode information that are resistant to a variety of attacks on the message. Such attacks may include cropping or distorting the image, or modifying color information to destroy any hidden information that could possibly exist in a given signal.

2. TECHNIQUE

Steganography works by replacing bits of useless or unused data in regular computer files (such as graphics, sound, text, HTML, or even floppy disks) with bits of different, invisible information. This hidden information can be plain text, cipher text, or even images and sound wave.

In the field of steganography, some terminology has developed. The adjectives cover, embedded and stego were defined at the Information Hiding Workshop held in Cambridge, England. The term "cover" is used to describe the original, innocent message, data, audio, still, video and so on. When referring to audio signal steganography, the cover signal is sometimes called the "host" signal.

The information to be hidden in the cover data is known as the "embedded" data. The "stego" data is the data containing both the cover signal and the "embedded" information. Logically, the processing of putting the hidden or embedded data, into the cover data, is sometimes known as embedding.

Occasionally, especially when referring to image steganography, the cover image is known as the container.

3. MANIPULATION IMPLEMENTATION

An extremely simple steganographic method is to take the individual pixels in an image. Each of these pixels in an image is made up of a string of bits. We can commandeer the 4-least significant bit of 8-bit true color image to hold 4-bit of our secret message (image) by simply overwriting the data that was already there. By experimental, we note that: The impact of changing the 4-least significant bits is almost always entirely imperceptible.

Figure 1 shows the general description of 4 least significant bits. Figure 2 describes 4-Least

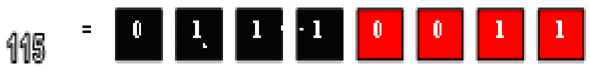


Figure 1. The number 115 and the corresponding binary value, the 4 LSB is highlighted in red. Only these value are allowed to change.

Significant Bet system, which we are going to used to embed secret message in cover image. The message may be a few thousand bits (often at 7 or 8 bits per text character) embedded in millions of other bits. Probably the most typical use is digital images. Digital images are commonly stored in either 24-bit or 8-bit files. If an 8-bit image is viewed as a grid and the grid is made up of cells, these cells are called pixels. Each pixel consists of an 8-bit binary number (or a single byte), and each 8-bit binary number refers to the color palette (a set of colors defined within the image). All color variations for the pixels are derived from three primary colors: red, green, and blue. Each primary color is represented by 1 byte (= 8 bits).

3.1. hiding image inside other image

Block diagram (Figure 3) describes the general steps of concealing image in container image.

The following steps describe the details of 4-LSB steganography, in order to concealing secret image inside cover image (Figure 4).

Preparing secret image

1. Convert secret image to stream of binary bits.

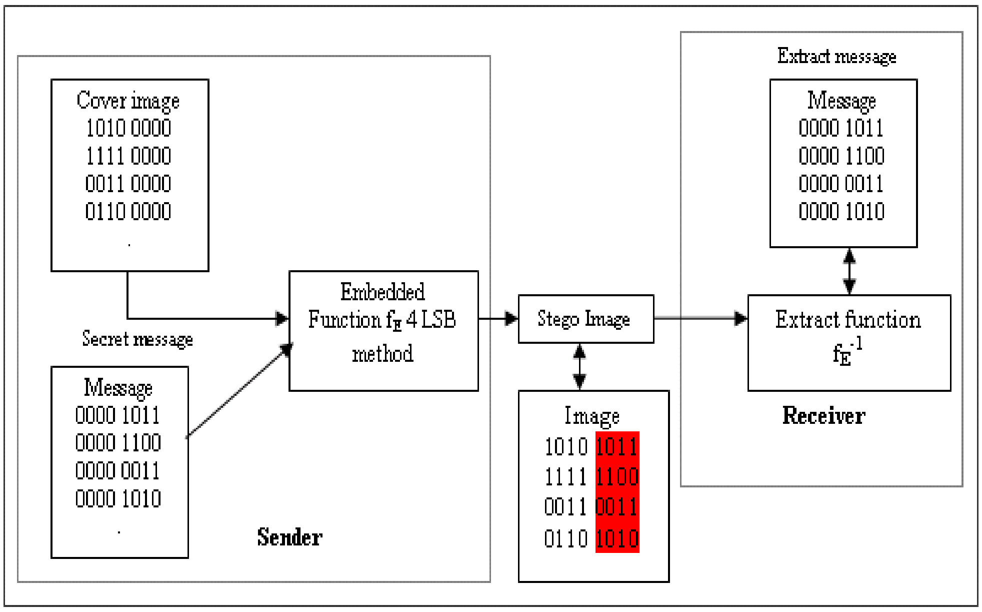


Figure 2. Graphical Version of the LSB method.

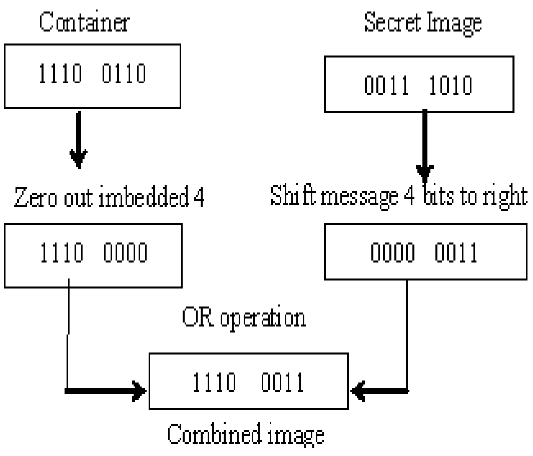


Figure 3. Block Diagram of image hiding.

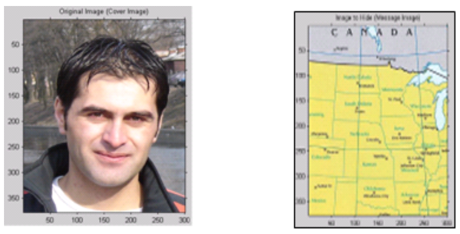


Figure 4. Concealing USA map in my portrait.

255 255 255 255 255 255 255 255 255 255 ....

2. Shift secret image over 4 bits to right (Figure 5).
   15 15 15 15 15 15 15 15 15 15 15 15
   (15)10 = (0000 1111)2

Preparing container image

3. Convert cover image to stream of binary bits.
   120 117 113 113 112 109 112 112 109 ......
   Example First pixel: (120)10 = (0111 1000)2
   * The red bits are the place where we put our secret image.
4. Zero out embedded bits in cover image.
   112 112 112 112 112 96 112 112 96 96 96
   Example First pixel: (112)10 = (0111 0000)2

Applying Embedded function

5. Apply OR operation between secret image and cover image, the result is stego image(Figure 5).
   Example:

M0: (15)10 = (1111)2
Logical OR
C0: (112)10 = (0111 0000)2
____
Stego: (127)10 = (0111 1111)2

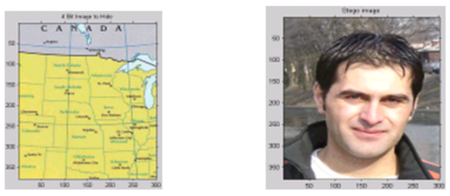


Figure 5. shift message image (left) over 4 bits to right, Stego message (right) that contains the map.

Figure 6 shows another experiment of hiding secret image inside container image.

Reconstruction the secret image

Reconstruction of the secret message is performed by reversing the process taken to insert the secret image message in the container. To reconstruct the secret message is simply done by shifting the Stego image 4 bits to left (Figure 7 and Figure 8).

3.2. Hiding text message inside image

The following steps show in details the procedure of hiding secret text inside cover image Block diagram (Figure 9).

Preparing container image

1. Convert cover image to streams of binary bits.
2. Use two adjacent bits to hide one character.

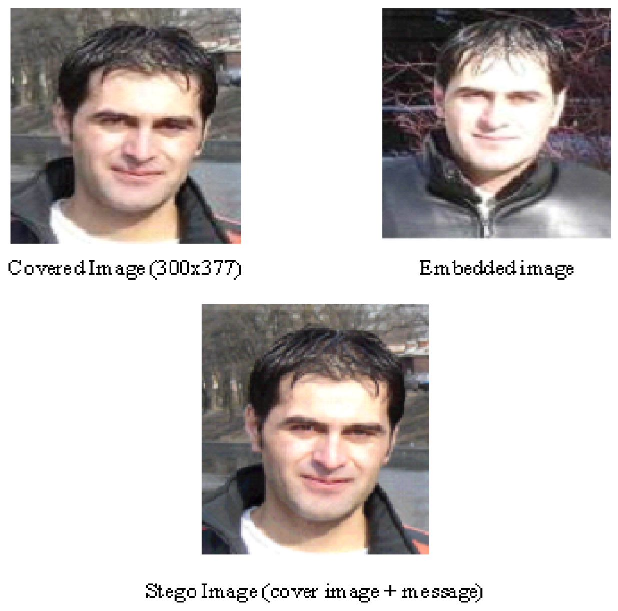


Figure 6. Another example of using 4-LSB.

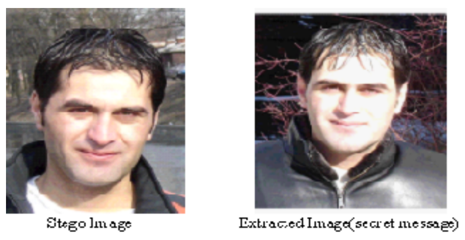


Figure 7. Extract the right (map) image from the left (Stego image).

Figure 8. Another example.

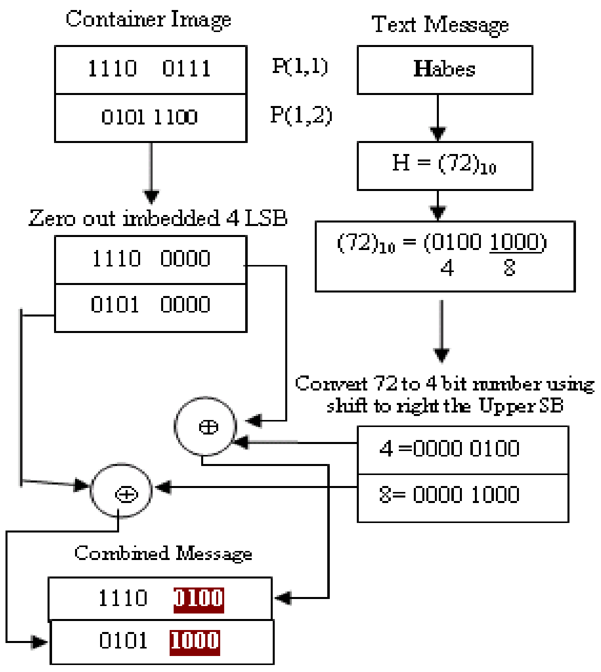


Figure 9. Block Diagram of concealing text inside image container.

Preparing secret text message

3. Convert each character of the secret message to decimal number. Example H = (72)10 = (0100 1000)2
   (a) We take the 4 least significant bits alone; we can do that by perform AND operation: (72)10 AND (15)10 = (0100 1000)2 AND (0000 1111)2 = (0000 1000) = (8)10.
   (b) We take the 4 upper significant bits alone; we can do that by perform shift operation by 4: (72)10 Shift to right be 4 = (0000 0100)2 = (4)10
4. Now we can add the secret message to the cover image by applying OR operation.

As shown in the block diagram (figure 9), to hide each character of secret message we need two pixels. So the number of character that we can hide in (n x n) image is given by the following equation:

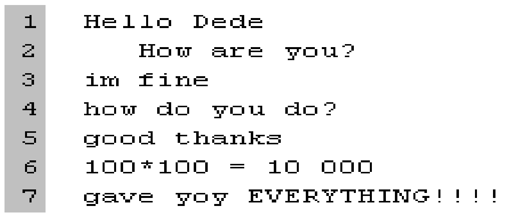
$$Numberofcharacters \leq (n \cdot n) \div 2 - n \tag{1}$$

In equation (1), we subtract n pixels because we don't set secret text in the first row of cover image; we start setting data from the second row of cover image. The first row of covered image used to store specific data, like position of last pixel in the covered image that contains secret data. The following two equations show how to calculate the pixel that determine of secret text data:

$$Ypos = length(1^{st} rowofimage) mod length(secretmessage) \times 2 \tag{2}$$

$$Xpos = (length(secretmessage) - Ypos) \div length(1^{st} rowofimage) \tag{3}$$

Figure 10 and Figure 11 show an example of hiding of text file in image file.



```
Hello Dede
   How are you?
im fine
how do you do?
good thanks
100*100 = 10 000
gave yoy EVERYTHING!!!!
```

Figure 10. Text to be hid inside image.

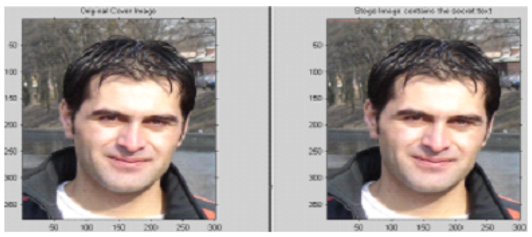


Figure 11. Cover image (left), Stego image that contains the secret text data (right).

Reconstruction the secret text file

Reconstruction of the secret text message is performed by reversing the process used to insert the secret message in the container image. The following steps describe the details of reconstruction the hidden text file (Figure 12):

1. Take two adjacent pixels from the stego image.
2. Shift the first pixel by 4 to right 1110 0100 shift to right by 4 = (0100 0000)2
3. Perform AND operation with 15 to the second pixel (0101 1000) AND (00001111)2 = (0000 1000)2
4. ADD the result of step 2 and 3 together and we get (0100 0000)2 + (0000 1000)2 = (0100 1000) = (72)10 = H.

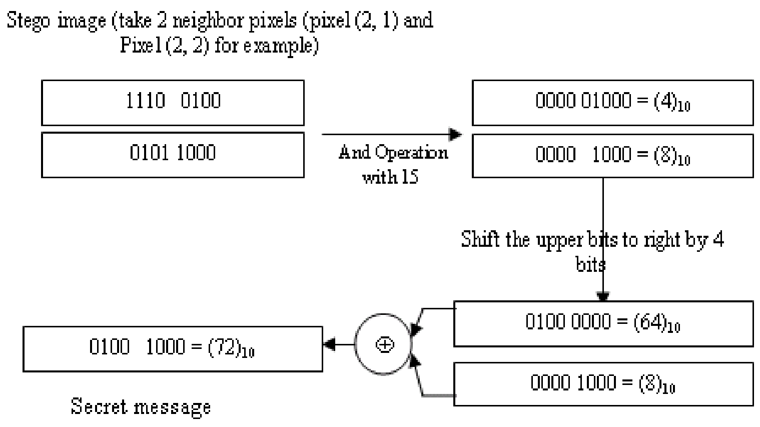


Figure 12. Block diagram diagram of reconstruction hidden text file.

4. FREQUENCY MANIPULATION

The process to encode a message involves 3 essential steps, preparing the container image, preparing the secret message (image or text file), and adding the messages together.

The first step in preparing a container image is to simply select a good candidate. The container should be long enough to fit the entire secret message. It should also be contain a large amount of information in order to mask. The exact method of preparation for the secret message depends on its nature (sound, image, etc.).In our work we prepare the secret image message by convert it from 9 bits image to 4 bits, and prepare the text file by represent each character of text file as two numbers of 4 bits.

Once both the container and the secret message have been prepared, they can be simply added together to get the encoded image. In 4-least significant method we use spatial domain technique for scattering the secret data inside covered message, but we try to imply some self-similarity function to distribute the secret message throughout the covered message in order to increase the degree of security for this method.

5. RESULTS

To test the 4 least significant bits method we encoded the map of united state in my face portrait and vise versa as shown in (Figure 3), and we encoded text file in my face portrait as shown in (Figure 9 and 10). We were able to encode and extract the data with no loss of data integrity.

In the above algorithm we take in our account that the receiver does not have the carrier image, in order to give the sender freedom to choose any picture as carrier image just with specific properties (like width and height of image) that he/she agree on with the receiver. The above algorithm can be implemented by using XOR operation, in order to do that the sender and receiver must be agreeing on specific carrier image in advance. If the sender want to use another container it is necessary to inform the receiver about and send him/her the new container in order to give him an ability to recover the secret message. The sender implements XOR operation between container image and the secret image to create stego image for example:

- Carrier image (120)10 = (0111 1000)2
- Secret message (15)10 = (1111)2
- Stego image (0111 1000)2 XOR (1111)2 = (0111 0111)2
  To recover the secret message using XOR method, the receiver should have the container image, then implement XOR operation between a container and Stego image to get the secret message for example:
- Carrier image (120)10 = (0111 1000)2
- Stego image (0111 0111)2

Secret message (0111 1000)2 XOR (0111 0111)2 = (0000 1111)2 this message that we hide it in the above example. We write programs using Mat Lab to test the result.

6. DISCUSSION

Each steganographic method has its own advantages and disadvantages. There are tradeoffs between ease of implementation, resistance to attack, felicity of the secret message, and maximum ratio of secret message size to carrier size.

The advantages of the 4-LSB method are clear. It is very simple to implement, and generally quite difficult to detect. If the encoded image is transmitted perfectly with no errors, then when it is decoded there will be no data lost in the secret message. The disadvantages of the 4-LSB method, if the form of the carrier data is changed in any way the entire secret message could be lost. For example an encoded image is resized or changed from a bitmap to a jpeg then there is virtually no chance of recovering the secret message.

Steganography's strength lies in the sheer amount of information that changes hands every day. It is very simple using digital technology to conceal any given digital information within other information, so virtually anything could contain a hidden meaning. There is no practical way to check it all. However, none of steganographic methods we examined could resist a concerted attack if someone knew that there was a message in a given document. For the greatest level of secrecy, a combination of both steganography and cryptography is necessary.

7. EVALUATION

Our evaluation for this method based on the following assumptions:

- 8-bit true carrier image of size 256x256.
- 8-bit true secret image.
- 8-bit ASCII code for each character.

For embedding secret image inside container image we get the following result comparing with other methods:

Method Name	number of bit that we can embed	Distortion of cover image
1-LSB	256*256	Weakness
2-LSB	256*256*2	Weakness
4-LSB	256*256*4	weekness

Table 1. Comparison between different LSB method based on secret image.

For embedding text message inside container image we get the following result comparing with other methods:

Method Name	number of bit that we can embed	Distortion of cover image
1-LSB	(256/8)*256	Weakness
2-LSB	(256/4)*256	Weakness
4-LSB	(256/4)*256	weekness

Table 2. Comparison between different LSB method based on secret text message.

8. CONCLUSION

After studying number of techniques for hiding information in digital media, as well as touching on the limitations and possibilities of each. In this paper we focus our concern in image because of it's widely used in Internet and also in mobile system.

And based on our study, 4-LSB substitutions is a good method for embedding an acceptable amount of data, that's because size of embedded message to carrier's size.4 LSB embedded data can easily be implemented and do not visually degrade the image to the point of being noticeable. Furthermore the encoded message can be easily recovered and even altered by a 3rd party. It would appear that 4 LSB is good method of steganography due to its tremendous information capacity. Using 4-LSB method we can exchange secret message s over public channel in a safe way.

9. FUTURE WORK

We are trying to implement and evaluate stego methods to hide data inside wave files; and in this way the person can hide his/here secret information inside it.Ttherefore no one except him can extract the secret information.

10. ACKNOWLEDGMENTS

I would like to convey my gratefulness to my supervisor Prof. Viktor V. Alexandrov for his valuable response to my queries. I profusely thank him for constant support, interesting discussions, and strange questions. I hope I am able to live up to his standard of advising.

REFERENCES

1. Lenti, Jozsef, Steganographic Methods. Budapest: Budapest University of Technology and Economics, 2000.
2. Marvel, Lisa M, et al, Hiding Information in Images. Aberdeen Proving Ground, MD: US Army Research Laboratory, 1998.
3. Johnson, N. F., Duric, Z., Jajodia, S. Information Hiding: Steganography and Watermarking- Attacks and Countermeasure. Kluwer Academic Press. Norwrll, MA, New York, The Huague, London, 2000.
4. An Evaluation of Image Based Steganography Methods Kevin Curran, Internet Technologies Research Group, University of Ulster Karen Bailey, Institute of Technology, Letterkenny, Ireland
5. R.C. Gonzalez, R. E. Woods, "Digital Image processing," Upper Saddle, River, New Jersy, Prentice Hall, Inc., 2002.
6. Information Hiding—A Survey. Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn. Proceedings of the IEEE, special issue on protection of multimedia content, 87(7):10621078, July 1999.
7. A Survey of Techniques for Digital Watermarking Chris Shoemaker Independent Study EER-290.Prof Rudko Spring 2002.
8. www.eng.istate.edu/cpre592/

This paper was recommended for publication by V. Venets, a member of the Editorial Board