IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 11, November 2015.
www.ijiset.com
ISSN 2348 – 7968

A New Method to Information Hiding by Using Colors

Orooba Ismaeel Ibraheem Al-Farraji

Msc. In Computer Science/Al-Technology University-Baghdad

Abstract:

This method is new to steganography by hiding a small picture (hiding message) in a large picture (cover), both two pictures convergent colors through each compared pixel in small picture with the pixels in the big picture in the case of equal values of the pixel's color with any pixel's color in big picture storage location of the pixel in the big picture.

Keyword : Information Hiding , Steganography, Colors, Digital Image, Watermarking

1.Introduction :

Information hiding techniques have recently become important in a number of application areas. Digital audio, video, and pictures are increasingly furnished with distinguishing but imperceptible marks, which may contain a hidden copyright notice or serial number or even help to prevent unauthorized copying directly (2).

Information Hiding is considered very important part of our lives. There exist many techniques for securing the information (1).

2. TECHNIQUES FOR INFORMATION HIDING

There are three major data hiding techniques popular: watermarking, cryptography and steganography (1).

A watermark is a recognizable image or pattern that is impressed onto paper, which provides evidence of its authenticity . Watermark appears as various shades of lightness/darkness when viewed in transmitted light. Watermarks are often seen as security features to banknotes, passports, postage stamps and other security papers. Digital watermarking is an extension of this concept in the digital world . A watermarking system's primary goal is to ensure robustness, i.e, it should

ISSN 2348 – 7968

be impossible to remove the watermark without tampering the original data (4) .

Digital watermarking technology as an important branch of information hiding technology research field, since it has been paid attention to by many domestic and foreign experts and scholars and business groups, and gradually become a research hotspot in the field of information security (5)

Cryptography is about communication in the presence of an adversary. It encompasses many problems (encryption, authentication, key distribution to name a few). The field of modern cryptography provides a theoretical foundation based on which we may understand what exactly these problems are, how to evaluate protocols that purport to solve them, and how to build protocols in whose security we can have confidence. We introduce the basic issues by discussing the problem of encryption

Modern cryptography abandons the assumption that the Adversary has available infinite computing resources, and assumes instead that the adversary's computation is resource bounded in some reasonable way. In particular, in these notes we will assume that the adversary is a probabilistic algorithm who runs in polynomial time. Similarly, the encryption and decryption algorithms designed are probabilistic and run in polynomial time.(7)

Steganography is the art and science of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message. While commonly thought of as messages hidden in pictures it is not limited to just pictures, although this is one the common uses, but messages can be embedded in any number of digital media types. It can even be embedded into sound files.(6)

3- Color Images

A color image is made up of pixels each of which holds three numbers corresponding to the red, green, and blue levels of the image at a particular location. Red, green, and blue (sometimes referred to as RGB) are the primary colors for mixing light—these so-called additive primary

ISSN 2348 – 7968

colors are different from the subtractive primary colors used for mixing paints (cyan, magenta, and yellow). Any color can be created by mixing the correct amounts of red, green, and blue light. Assuming 256 levels for each primary, each color pixel can be stored in three bytes (24 bits) of memory. This corresponds to roughly 16.7 million different possible colors. Note that for images of the same size, a black and white version will use three times less memory than a color version (8).

3-1. Least Significant Bit

Least Significant Bit Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image [10]. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue colour Text Images Audio/ video Protocol Transform Domain Image Domain JPEG LSB in BMP LSB in GIF Patchwork Spread Spectrum components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data (8). 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 00001100) (11010010 10101100 01100011) Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size [8]. Since there are 256 possible intensities of each primary colour, changing the LSB of a pixel results in small changes in the intensity of the colours. These changes cannot be perceived by the human eye - thus the message is successfully hidden.

ISSN 2348 – 7968

3-2.LSB in BMP

When embedding a message in a “raw” image, that has not been changed with compression, such as a BMP, there exists a trade-off between the invisibility of the message and the amount of information that can be embedded. A BMP is capable of hiding quite a large message, but the fact that more bits are altered results in a larger possibility that the altered bits can be seen with the human eye (9) .

4- The Proposed Scheme

The new proposed system adding the small picture in the big picture by using the pixels that equal in value colors

5- The Proposed System Operation

The Add Small picture of new system has many operations:

1- Input the picture to be hidden that can be done by open new file and entered directly.
2- Open –Small Image and convert it to Array
3- Open –Image and split the body to blocks, we split the body in order to obtain small number of position to be easy in hide.
4- Find the position where to hide the pixel of the Picture in the blocks of image and store the block numbers and the number of position in file to be use later.
5- Find the position where to hide the pixel of the Picture in the blocks of image and store the block numbers and the number of position in file to be use later.
6- Substitute the pixel in the position get from the search program, in fact the substitution it merely locate the position that hide in it.
7- Hide position will hide the block numbers where the character was hidden.
8- Combine the blocks into a one file and then combine the header and the body file into a file to perform the Image (bmp file).

The new proposed system has a key which is used to extract the embedding small image from image. We use a key, as number's position we began hide in.

ISSN 2348 – 7968

5.1 – The Algorithm of The Proposed System

The following steps describe the algorithm:

Algorthim 1: Processing the big image (cover) to hiding information

Input : Image

Output : Array of binary code

Step1- Open Image (the bmp-file) Operation

This operation will open the bmp file and save header in a file and save the palette value of body in another file.

Step2- Split the body of the image file operation(each part 300X 300).

This operation will split the body image in equal blocks (300 x 300) to use these blocks in hide text, we split the body in order to obtain small number of position to be easy in hide.

Step3- save the value of pixel palette in array

Step4-End

Algorthim2: Search about pixles that possible hide in them

Input :Array of binary code of image , Array of binary code of small image

Output :positions of pixel that equal to Pixel of small image

---

Step1-Find the position operation this operation is done by read a block from block image file

Step2-Test if block is suitable to hide a pixel from small image or not. The testing is done by compare the value of the palette with the value of each pixel in small image.

ISSN 2348 – 7968

Step3-If value of the palette equal with a pixel from small image, this mean we find the position that we hide the Pixel and save the block number and position in a file.

Step4-End

Algorithm 3: Hiding position of pixels that be equal the pixels in small image

Input : Array of binary code of image , Array of binary code of image

Output : positions of pixel that equal to pixel of small image

Step1-Substitute the pixel of small image operation this operation is done by read the value of pixel from small picture and locate the position where to hide the pixel in the block.

Step2-Replace the pixel in specific position and hide the position of the replaced pixel in the first row and last row of the block.

Step3-This change is unnoticeable because the number of position is small and substitute in LSB.

Step4-End

6- Extracting The Image

Extracting the text is done by using the key stored into first block position that contains the Key.

Later get the row position from the first row

Pos =(r+1) * 20- (20-c)Get the character value from the specific position.

ISSN 2348 – 7968

6.1 The Algorithms of Extracting

The following steps describe the algorithm:

Algorthim 1: Restore the big image (cover)

Input : Image

Output : Array of pixel value

step1- Open the bmp file by reading the header and getting the body of the file

Step2- Split the file into blocks (300 x 300).

Step3- by using the key get the first block that contains the first pixel of the small picture.

For I= 1 to the length of the small image do

Find the position of the embedded Pixel from the first and the last row of the block.

Get the row position from the first row.

Get the column position from the last row

Get the position by applying the following equation:

Pos =(r+1) * 20- (20-c)

Get the value of pixel from specific position.

Step4- From the current block find the position of the next block.

The number of the block is hidden in the first row and the second half of the last row.

Step5- The end

ISSN 2348 – 7968

7.Experimental Results

The results of the proposed system has been illustrated in the following

Example:

Fig. 1 shows one sample secret image , which is 200 X 200 pixels, and the cover images. Fig. 2 shows the 800 X 480 stego images . The two images close colors so easily find points for each first image (required hide) equal points in the second image (cover) where Stego image is hiding a copy of the fig1. the difference between the original images and the stego images is not visible to the human eye.

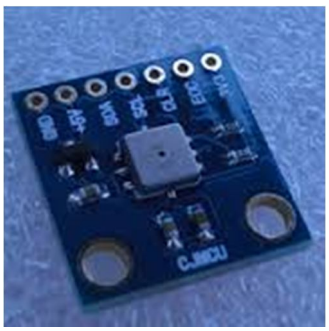

Fig. 1 : Secret Image

Fig. 2 : Cover Image (stego image)

ISSN 2348 – 7968

Fig. 3: Cover Image after hiding secret image in it.

8- Experimental Results and Performance Analysis

8-1. Use PSNR Function to Test the results

Signal to Noise Ratio (PSNR) is generally used to analyze quality of image, sound and video files in dB (decibels). PSNR calculation of two images, one original and an altered image, describes how far two images are equal. Figure 5 shows the famous formula.

MSE: Mean-Square error.

x: width of image.

y: height.

x*y: number of pixels (or quantities).

This function displays the PSNR (peak signal-to-noise ratio) between two images. The answer is in decibels (dB).

PSNR is very common in image processing. A sample use is in the comparison between an original image and a coded/decoded image. Typical quoted PSNR figures are in the range +25 to +35dB.

ISSN 2348 – 7968

The syntax for this file is PSNR(A,B), where A and B are MATLAB Intensity Images, with matrix-elements in the interval

8-2.PSNR formula.

In order to observe the image quality of secret image objectively, the PSNR (Peak Signal toNoise Ratio) value of the image is calculated using the equation 3 and if the PSNR value is greater than 35dB, the watermarked image is within acceptable degradation levels.

$PSNR = 10 \times \lg((2n-1)2 / MSE)$ (3)

Where n means the number of bits per sample value, the MSE represents mean square error between the host image and the watermarked image.

By using Matlab we input figure(2) and figure(3) to function PSNR the results equal 29.821 db and this value acceptable.

9. Conclusion

The proposed system proved to be a good system used to hide a small image in big image by compare value of palette with value of pixel if equal we hide position in another Place.

- In the proposed system don't change anything in the pixels that hide in it but change in another pixels in which the security of system is increased, in case the third person explore anything in picture he explore

- The proposed system proved to be easy to use and efficient in terms security and can people also use it to hide watermarking in case the watermarking be image.

- In future we can develop the system to include the hiding big picture.

Reference :

ISSN 2348 – 7968

1- Richa Gupta , „Information Hiding and Attacks : Review", International Journal of Computer Trends and Technology (IJCTT) – volume 10 number 1 – Apr 2014.
2- Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn, "Information Hiding—A Survey " , Proceedings of the IEEE, special issue on protection of multimedia content, 87(7):1062–1078, July 1999.
3- E. Kokabifar∗ , G.B. Loghmani∗ and A. Latif†, " Digital image watermarking using normal matrices", June 8, 2015
4- Zhu Yuefeng1, 2, Lin Li 2 , „DIGITAL IMAGE WATERMARKING ALGORITHMS BASED ON DUAL TRANSFORM DOMAIN AND SELF-RECOVERY", INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS VOL. 8, NO. 1, MARCH 2015, page 199-219 .
5- Ashdown M., Flagg M., Sukthankar R., and Rehg J.M., A Flexible Projector-Camera System for Multi-Planar Displays, Computer Vision and Pattern Recognition (CVPR), pp. II-165 - II-172, 2014.
6- Porter, Wayne , „Steganography- Hiding Information Inside of Information", http://www.spywareguide.com/articles/article_show.php?id=116&rss, at last update 2015
7- Goldwasser, Shafi, Bellare, Mihir , „Lecture Notes on Cryptography", July ,2008
8- Sachs , Jonathan, „Digital Image Basics", Copyright © 1996-1999 Digital Light & Color
9- T. Morkel 1 , J.H.P. Eloff 2 , M.S. Olivier, "AN OVERVIEW OF IMAGE STEGANOGRAPHY" ,
10- Johnson, N.F. & Jajodia, S., "Exploring Steganography: Seeing the Unseen", Computer Journal, February 1998