IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 3, Ver. II (May.-June. 2017), PP 75-79
www.iosrjournals.org

Watermarking System Using LSB

Hewa Majeed Zangana
Department of Computer Science / College of Computer Science and IT / Nawroz University / Kurdistan Region of Iraq

Abstract: *In the current paper we present a system of watermarking due to nowadays the data security is one of the major challenges and watermarking is one of the techniques where a digital pattern or signal has been inserted into a digital photo or image for security reasons. Usually, watermarking is used for protecting the copyright like owner authentication and the security of the card id. In the current system we used the technique of least significant bit (LSB) in watermarking. The results show us that the system is working well and the security of the image is really good.*
Keywords: *LSB, watermarking*

I. Introduction

The system proposed in a program, developed in Matlab environment, which provides the ability to insert a watermark in an image. Wikipedia gives the following definition of watermarking: “inclusion of information within a multimedia file, or any other kind, which can subsequently be detected or extracted"; basically it is a technique used in various fields (from the protection of content by entering digital signatures or copyright to data encryption brands), used even by terrorism (stenographic watermarking). The starting point was to find a technique to insert a text string within an image chosen by the user of the program; this string must however not be visible to the human eye, which should not perceive changes in appearance or colors in the image. Subsequently, the implementation of this technique has been refined, allowing the user to also use a numeric key to encrypt the file. Finally, since this technique is implemented by working at low level (operating on individual bits of data and images), it tried to repeat the operation of watermarking, hiding this time, instead of a string of characters, an image (size smaller than the image "container").

II. Proposed System

The general technique is very simple to describe. Suppose you have a fact to want to insert into the image without worrying for the moment about what this data represents. We denote with W then the binary representation of this figure. The image in which you want to insert instead can be seen as one or more arrays of pixels, where each element of these arrays is represented on 8 bits.
The technique consists in modifying the least significant bit (LSB) of each element of the array, putting in the place of each one, a bit W.
First Sample:

W=10011010

Assume for simplicity that the image is represented by a single 4x4 pixel matrix (one color component):

11000111	00010100	11111111	01001010
01000100	10111001	00101000	11010010
10101010	01011011	00101011	11011001
11100011	11111111	11010010	00101010

Once inserted the watermark the image will then become:

W=10011010

1100011**0**	0001010**1**	11111111	01001010
0100010**1**	1011100**0**	00101000	11010010
1010101**0**	0101101**0**	00101011	11011001
1110001**1**	1111111**1**	11010010	00101010

As can be noted, the watermark has been inserted starting from its least significant bit and advancing per column in the array that represents the image. We note that some elements of the matrix had already its least significant bit set to the desired value, this means that the modified number of array elements are at the most equal to the number of bits of W (8 in this example). In addition, the modification carried out corresponds to the modification of the elements of a single color component, by varying its representation to the next or previous level, which for the human eye is virtually indistinguishable.

A B

Fig.1: A: Original Image. B: Image contains the string "How Are You"

1.1 String Entry

The inclusion of a character string has specifically some issues more than in the general case. Furthermore, in the more general case, if the image in which to hide the data is in RGB format, you might have to work on a three matrices structure (the three color components). In reality, the fact of working with one or more matrices is not a big deal. Specifically Matlab, once the image is accessed; it is placed into a single vector that contains sequentially the columns of a single image:

```
im=img( : );      %Amount the image into a single column vector
```

A more significant problem instead is how we can understand the extraction phase where it terminates the string. To solve this problem it was decided to ensure proper operation of this application with all and only the characters of the ASCII table STANDARD. The advantage lies in the fact that the standard ASCII characters may be represented with 8 bits, but the significant bits to the recognition of the character are actually 7. In this way it was possible to use a sort of "plug character" represented on 8 bits but that It does not belong to the standard characters. For this purpose, the character 255 is selected or 8 bits all setup to 1.

Shown with Nc the number of the string components characters, we then insert (Nc + 1) * 8 bits, which means that to contain this string an image of (Nc + 1) * 8 pixels is required in the case of images to a color component or (Nc + 1) * 8/3 pixels in the case of RGB images.

The bit insertion is performed in a linear fashion and for columns, so that it will be simple in phase dewater marking extract all the characters, up to the cap character, since the decoding is carried out in the same insertion order.

1.2 String With Key Number

The method used so far is quite efficient, but in case you wish to encrypt sensitive data, it is very exposed to attacks if you know the algorithm used.

One solution to this problem has been to insert the bits of the watermark is not in a linear fashion, but in the pseudo random mode! In practice this is to place the bits in a "scattered" within the matrix that represents the image. Initially, you might think of using the rand command of matlab to generate the index in the array. Staring at the seed in an arbitrary way you could then continue insertion, while for extracting the user must necessarily know the seed (key) used! However, use rand presents many problems, as it may provide two or more times the same index for access to the matrix, which would entail the loss of data and the malfunction of the entire application.

An alternative to rand then is a much more useful command for our purpose, always based on the rand command: the randperm feature! randperm provides a permutation of the values ranging from 1 to the value passed to it as input, chosen so pseudo random thanks to the rand command.

Sample 1.1:
Suppose we have the same watermark and the same image of Sample 1:

W=10011010

11000111	00010100	11111111	01001010
01000100	10111001	00101000	11010010
10101010	01011011	00101011	11011001
11100011	11111111	11010010	00101010

But this time instead of inserting bits per column, we use the randperm command to calculate the index:

```
rand ('seed', key);              %Fixed the seed of the rand function
im = img (:);                    %Amount the image into a single column vector
randperm index = (length (im));  % create a vector of indices use randperm
```

Then index will contain: [6 3 16 11 7 14 8 5 15 1 2 4 13 9 10 12]

The image matrix will then be modified in this way: (**W=10011010**)

11000111	00010100	11111111	01001010
01000100	1011100**0**	00101000	1101001**0**
1010101**1**	0101101**1**	0010101**1**	1101100**1**
11100011	1111111**0**	11010010	0010101**0**

To rebuild the data so you have to necessarily know the key used, which allows you to calculate the right permutation. This method is very efficient because the permutations are possible in very large number: already for the current example 4x4 pixels of an image to a component (so small), are 16 various permutation, or 20,922,789,888,000 different permutations, E 'can then imagine how many are for an image of, for example, 800x600 pixels rgb.

1.3 Inserting Image

A little different is the situation when, instead of a string, you try to insert an image within another. The method is essentially the same but the problems are quite different to get around.The first problem is that you can not use a cap in nature, since nobody can distinguish more, whether it is a given instruction or end image. It was

Height	(n ° of rows in the matrix):	15-bit	maximum of 32.767 pixels
Width	(number of columns in the matrix):	15-bit	maximum of 32.767 pixels
Components	(n ° of matrices):	2 bits	maximum 3 matrices

thought then to place under watermarking, a sort of header of fixed size, containing details of the inserted image size. Substantially, the header must contain height and image width (in pixels) and the number of matrices that encode them. It was decided then to use unheeded 4 bytes, divided as follows:

By doing so, during decoding you can reconstruct the watermark image without problems. So the "box" image must be large enough to contain the watermark. So let us quantify this "pretty". We denote by the NW the number of components of that image (eg. 3 in the case of RGB) number of pixels in the image watermark and with CW, which needs to be stored then:

$$(NW*CW*8 + 32) \text{ bit}$$

Given that this data is stored in the least significant bits of each pixel of the image "container," then we have that this image must possess a number of pixels (multiplied by its components) equal to the number of bits needed to store the watermark.

Second Sample:
Suppose you have an RGB logo of 50x50 pixels in size. We can then calculate the occupation in bits of this logo:

$$W = (50*50*3*8 + 32) \text{ bit} = 60.032 \text{ bit}$$

As a result, an image of RGB type container that is able to contain this figure must be at least 60 032/3 pixels! If the image is square then it will have 142x142 pixels.
Indeed $142 * 142 * 3 = 60.492$.

1.4 Image With Key

As regards the use of a key on the other hand, has been applied the same method used for strings, through randperm command, with the difference that the header, although it is also inserted in such a way scattered, it remains confined in the first 32 pixels of the image container. However, using key remains very complicated back to the watermark without knowing it.

1.5 Saving Image Watermarked

Whether you choose to insert a text watermark, or if you want to insert an image, it is necessary, for later use, save the watermarked image to disk. We must therefore preserve the watermark content in that image. To do this it is necessary that the format in which the image is not affected by losses is saved (as it is for example, .jpg). For this reason, the rescue is carried out directly from the application, which allows the user to choose the file name and coding, to be chosen among bitmap, tiff and png.

A

B

Fig. 2: A: The image contains within it the logo and has been used as the following key sequence
B: Logo inserted in the image (A)

11235813

III. Conclusions

In the current paper we proposed a system of using the LSB in watermarking due to the importance of the security forced us to get more security in our systems. Even it is difficult to assume which system is better than the other in watermarking in order to use it for transferring the data in a secure mode. The software used in order to do our proposed system was the Matlab. The main goal of the proposed system is to use the LSB technique to hide information in images. In the future work of watermarking could use another technique to see some other results.

References

[1] Deepshikha Chopra, Preeti Gupta, Gaur Sanjay B.C., Anil Gupta," Lsb Based Digital Image Watermarking For Gray Scale Image", IOSR Journal of Computer Engineering (IOSRJCE)ISSN: 2278-0661, ISBN: 2278-8727 Volume 6, Issue 1 (Sep-Oct. 2012), PP 36-41

[2] Dr. Ajit ,Preeti Kalra, Sonia Dhull,"Digital Watermarking" ,International Journal of Advanced Research in Computer Science and Software Engineering, ISSN:2277 128X, Volume 3, Issue 4. April 2013

[3] Preeti Gupta, "Cryptography based digital image watermarking algorithm to increase security of watermark data", International Journal of Scientific & Engineering Research, ISSN 2229-5518 Volume 3, Issue 9 (September 2012)

[4] Manpreet Kaur, Sonika Jindal, Sunny Behal, "A Study of Digital Image Watermarking", IJREAS ,Volume 2, Issue 2 (February 2012) pp-126-136

[5] B Surekha, Dr GN Swamy, "A Spatial Domain Public Image Watermarking", International Journal of Security and Its Applications Vol. 5 No. 1, January, 2011

[6] Robert, L., and T. Shanmugapriya, "A Study on Digital Watermarking Techniques ", International Journal of Recent Trends in Engineering, vol. 1, no. 2, pp. 223-225, 2009.

[7] H.Arafat Ali, "Qualitative Spatial Image Data Hiding for Secure Data Transmission", GVIP Journal,Volume 7,Issue 2 , pages 35