**Journal of Science and Technology 13(June 2015): 82-88**

# DIGITAL WATERMARKING TECHNIQUE FOR SECURING INFORMATION USING COLOR SELECTOR AND PARITY CHECKER

**M.D. Hossain[1*], T. Sultana[2], J. Alam[1] and A.H.F. Salehin[1]**

[1]Department of Computer Science and Information Technology; [2]Department of Telecommunication and Electronic Engineering, Hajee Mohammad Danesh Science and Technology University, Dinajpur 5200, Bangladesh

## ABSTRACT

The development of internet and digital technologies has reduced the security of digital contents in recent years. Security techniques that are based on cryptography or steganography, only provide assurances for data confidentiality, authenticity and integrity during data transmission through a public channel such as transmission through an open network. However, such security techniques do not provide protection against unauthorized copying or transmitting of illegal materials. That's why digital watermarking technique is needed for providing copyright and authentication of digital content. It can provide an efficient security protection and embed information in digital signal. In this paper, a new technique is developed to mark digital media (image, audio or video) using watermarking. A true image or color image pixels has three color components which are Red, Green and Blue. Each color component has different values. In the proposed method red component is used as a selector whether the message bit will be embedded in green component or blue component. This method uses filtering to insert message bits. As the message bits are not inserted in to the fixed position that's why the proposed method is much secure and more difficult to attack. Parity checker and Least Significant Bit are used to insert the message bits for securing the message.

**Key words:** Color selector, parity checker, pixel, steganography, watermarking

## INTRODUCTION

Communication of information by hiding in and retrieving from any digital media is known as information hiding. The digital media can be an image, an audio, a video or simply a plain text file. It is a general term for encompassing many sub disciplines. However, generally it encompasses three disciplines. These are cryptography, watermarking and steganography (Gupta 2012). Digital technologies present new threats for Intellectual properties and contents. People have easy access to information and information can be modified easily. So methods that prevent unauthorized access to copyrighted digital contents are required in wide ranges of applications. Watermarking (Hsu and Wu 1999; Nikolaidis and Pitas 1996) is an embedded image or pattern in paper. People can view it by transmitted or reflected light. It is used as security features of banknotes, passports, postage stamps and other documents. Digital watermarking allows users to embed special pattern or some data into digital contents without changing its perceptual quality (Robert and Shanmugapriya 2009). It can be used to verify the authenticity or integrity of the carrier signal or to show the identity of its owners. It is prominently used for tracing copyright infringements and for banknote authentication (Saini and Shrivastava 2014). Like traditional watermarks, digital watermarks are only perceptible under certain conditions, i.e. after using some algorithm. Media watermarking research is a very active area and digital image watermarking became an interesting protection measure and got the attention of many researchers since the early 1990s (Verma and Tiwari 2014). The main objective of this paper is to mark a digital media through which the security and copy right is ensured by merging owner identification.

## MATERIALS AND METHODS

In this paper, digital watermarking technique is used to mark digital media (image, audio or video) to provide security of data where the identity of an owner is merged with the media at the transmission side and this owner identification is used at the receiver side to recognize the authentication of data. In this system, filtering is used to insert message bits. Parity checker and Least Significant Bit are also used to insert the message bits for securing the message.

*Corresponding author: Md. Delowar Hossain, Department of Computer Science and Information Technology, Hajee Mohammad Danesh Science and Technology University, Dinajpur 5200, Bangladesh, Cell Phone: +8801712262634, E-mail: delowar.cit@gmail.com*

So, it is more efficient and secure technique for information hiding.

For implementing the proposed watermarking technique algorithm, a cover image should be used and every pixel of that image should be considered. In the embedding process, this technique uses the ASCII value of alphabets and converts it into binary value, for example, “C” will be embedded in the cover image as “0100011”. First, it embeds the size of the embedding message using the required pixels of the cover image. After that it uses the resting pixels for embedding the secret message (binary representation of ASCII value of each character). Every pixel has three color components and these are Red (R), Green (G) and Blue (B). Each color component has different values. Red component is used as a selector whether the message bit will be embedded in green component or blue component. After selecting the right color component to hide the message bit the even parity and odd parity concepts are used. It is already known that even parity means the bits stream contains even number of 1’s and odd parity means the bits stream contains odd number of 1’s. Figure 1 shows the overall operation of the proposed method.

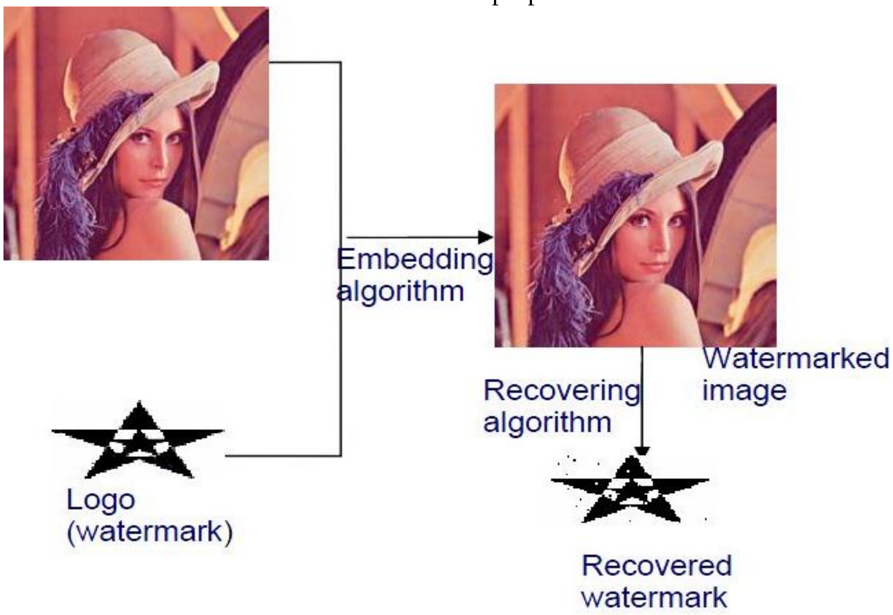


**Figure 1.** The overall operation of the proposed method

**a) Embedding Process:** Embedding is the process where a secret data i.e. either an image or an audio or video is embedded into a host data which may an image or an audio or video, using a key. This process includes the color components of every pixel which are Red (R), Green (G) and Blue (B). First bits stream of red component is checked whether it is even or odd. If it is even parity then green component is selected otherwise in case of odd parity blue component is selected. After selecting the right color component again parity of bits stream of selecting color component is checked. If the parity of a stream is even and the corresponding message bit is “1”, the bits of the color components will be unchanged, and if the message bit is “0”, it will change a single LSB of those color components from which the stream is made, to form the stream odd and vice versa. In this way we can change the LSB of any color component. Table 1 shows the actions for embedding message bits. If an embedded message is “1010”, Table 2 describes the changes in pixel value after embedding.

**Table 1**. Actions to embed message bits

| Pixel | Color Component | Component Value | Parity | Select G or B component | Message Bit | Change of LSB | Resulting Color Value |
|---|---|---|---|---|---|---|---|
| 1st | R | 1000101**1** | Even | R has even parity, so Insert message bit to G component | 1 | G is even parity.<br>No change at LSB | 0000111**1** |
| | G | 1000110**1** | Even | | | | 1000110**1** |
| | B | 1000110**0** | | | | | 1000110**0** |
| 2nd | R | 0000101**1** | Odd | R has odd parity, so Insert message bit to B component | 0 | B is even parity.<br>Make it odd parity. | 0000101**1** |
| | G | 1001110**0** | | | | | 1001110**0** |
| | B | 0100111**0** | Even | | | | 0100111**1** |
| 3rd | R | 0110101**0** | Even | R has even parity, so Insert message bit to G component | 1 | G is odd parity.<br>Make it even parity. | 0110101**0** |
| | G | 1001010**0** | Odd | | | | 1001010**1** |
| | B | 1011000**0** | | | | | 1011000**0** |
| 4th | R | 1100101**1** | Odd | R has odd parity, so Insert message bit to B component | 0 | B is odd parity.<br>No change at LSB | 1100101**1** |
| | G | 1011110**1** | | | | | 1011110**1** |
| | B | 0100101**0** | Even | | | | 0100101**0** |

**Table 2.** The changes in pixel value after embedding.

| Case | Parity | Message Bit | Action | Resulting Parity |
|---|---|---|---|---|
| 1 | Even | 1 | No | Even |
| 2 | Even | 0 | Toggle a bit | Odd |
| 3 | Odd | 1 | Toggle a bit | Even |
| 4 | Odd | 0 | No | Odd |

The following algorithm shows the embedding procedure of inserting message bit 0 or 1. It uses the LSB technique and parity checker.

i. Get cover image.
ii. Check the parity of bits stream of the red color component.
iii. If the parity of the red component is even then select green component to hide message bit otherwise select blue component.
iv. Again check the parity of bits stream of the selected color component (Even or Odd).
v. Get message bit (0 or 1).
vi. If parity is even and message bit is “1”, do nothing.
vii. If parity is even and message bit is “0”, toggle the value of a LSB of the color components from which the stream was built.
viii. If parity is odd and message bit is “1”, toggle the value of a LSB of the color components from which the stream was built.
ix. If parity is odd and message bit is “0”, do nothing.
x. If embedding message bit exists, go to step (ii) else go to step (xi).
xi. End.

**b) Extracting Process:** To extract the embedded message, first bits stream of red component is checked whether it is even or odd. If it is even parity then green component is selected otherwise in case of odd parity blue component is selected. Again parity of bits stream of the selecting color component is checked. If the parity of a stream is even, message bit is “1” and if the parity of a stream is odd, the message bit is “0”. Table 3 shows the actions for extracting message bits.

**Table 3.** Actions to extract message bits

| **Pixel** | **Color Component** | **Component Value** | **Parity** | **Select G or B component** | **Parity of G or B component** | **Message Bit** |
|---|---|---|---|---|---|---|
| 1st | R | 10001011 | Even | R is even parity. Go to G component | G is even parity. | 1 |
| | G | 10001101 | Even | | | |
| | B | 10001100 | | | | |
| 2nd | R | 00001011 | Odd | R is odd parity. Go to B component | B is odd parity. | 0 |
| | G | 10011100 | | | | |
| | B | 01001111 | Odd | | | |
| 3rd | R | 01101010 | Even | R is even parity. Go to G component | G is even parity. | 1 |
| | G | 10010101 | Even | | | |
| | B | 10110000 | | | | |
| 4th | R | 11001011 | Odd | R is odd parity. Go to B component | B is odd parity. | 0 |
| | G | 10111101 | | | | |
| | B | 01001010 | Odd | | | |

The following algorithm describes how the secret message can be extracted from the cover image.

i. Get cover image.
ii. Check the parity of bits stream of the red color component.
iii. If parity of the red component is even then go to the green component to extract message bit otherwise select blue.
iv. Again check the parity of bits stream of the selected color component (Even or Odd)
v. If the parity is even, store "1" as the message bit, else store "0" as the message bit.
vi. Do the same from step (ii) to step (v) until the entire message bits are not retrieved.
vii. End.

## RESULTS AND DISCUSSION

For implementing the proposed algorithm, C# .Net is used. A 24-bit bitmap image, named lena.bmp and 50 Characters message (You have to dream before your dream can come true.) was used for experiment which is shown in Figure 2. From the figure it can be observed that after implementing the algorithm the same output result is found. Histogram analysis which is shown in Figure 3 is also carried out to prove the efficiency of the proposed algorithm. From the histogram analysis it can be clearly seen that the output of the program was remarkably similar to the original image. The text is hidden in the image which is shown in Table 4.

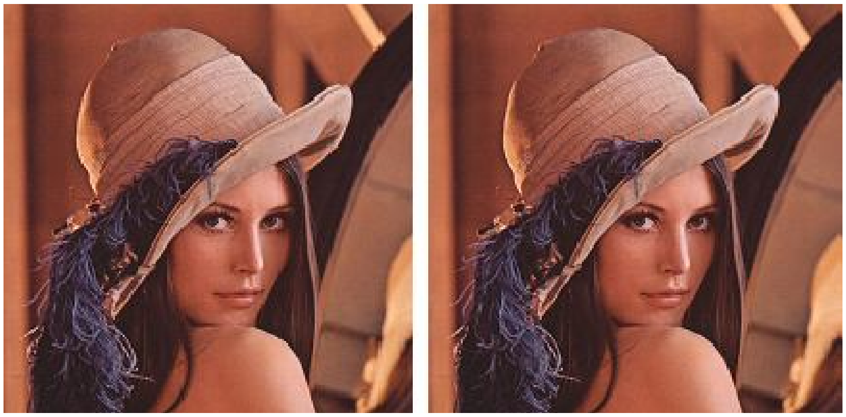
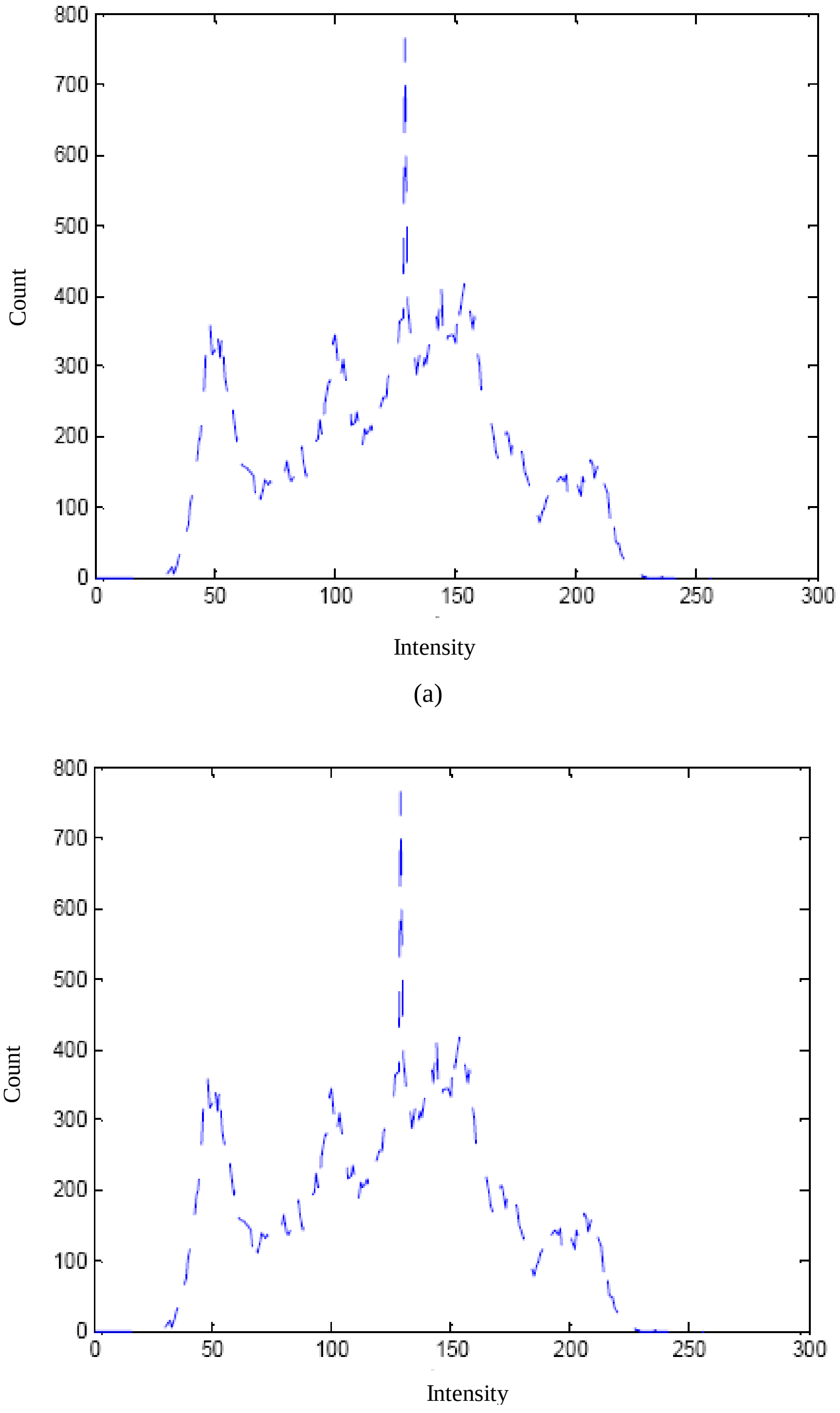


**Figure 2.** The left image is the input and the right image is the corresponding output of the experimental result using the proposed algorithm for hiding data.

**Table 4.** Hidden Text in Cover Image

| Cover image | Image Size | Message Size | Hidden Text |
|---|---|---|---|
| Lena.bmp | 768kB | 50 Characters or 400 Bits | You have to dream before your dream can come true. |



(a)

(b)

**Figure 3.** Histogram of input (a) and corresponding output (b) of Lena.bmp

When a large message is embedded by using our proposed algorithm, a very small number of bits will be changed. The cover image size which is used into our experiment is 320x240. This cover image can hide almost 10900 characters according to our algorithm which is of course a big message. As message bits are not inserted in to the fixed position. So our proposed method is more difficult to attack. Moreover, it is not necessary to work with LSBs of the color components. The proposed technique can be applied using any bit position which makes more difficult to retrieve the message by Stegoanalyst. In addition, the histogram is also showing very negligible changes.

## CONCLUSION

Now a day's technologies are developing rapidly and digital contents like image, audio, video etc. are being used widely through internet but they are not secure. Furthermore access of information has become much easier. So there is a serious problem of copyright and ownership. That is why we need a prevention method that will prevent the illegal uses of digital contents. Digital watermarking can be the solution of the ownership and copyright problem. In this paper, a new concept of watermarking is proposed. This technique has achieved the success to sterilize the secret information of an image. The goal of the proposed technique is to increase the security and to make it difficult for the unauthorized person to determine the presence of a secret message.

## REFERENCES

Gupta P. 2012. Cryptography based digital image watermarking algorithm to increase security of watermark data. International Journal of Scientific and Engineering Research, 3(9): 1-5.

Hsu CT and Wu JL. 1999. Hidden digital watermarks in images. IEEE Transaction on Image Processing, 8(1): 58–68.

Nikolaidis N and Pitas I. 1996. Copyright protection of images using robust digital signatures. In Proceedingsof ICASSP'96, Atlanta, Georgia. 17 (2): 2168–2171.

Robert L and Shanmugapriya T. 2009. A study on digital watermarking techniques. International Journal of Recent Trends in Engineering, 1(2): 223-225.

Saini LK and Shrivastava V. 2014. A survey of digital watermarking techniques and its applications. International Journal of Computer Science Trends and Technology (IJCST), 2(3): 70-73.

Verma R and Tiwari A. 2014. Copyright protection for watermark image using LSB algorithm in colored image. Advance in Electronic and Electric Engineering, 4(5): 499-506.