Improving the Non-filtering Steganographic Algorithm using LSB Verification Method

Md. Olioul Islam
Dept. of Telecommunication and Electronic Engineering,
Hajee Mohammad Danesh Science & Technology University,
Dinajpur-5200, Bangladesh
rahi_044@yahoo.com

Md. Sazzad Hossain, Md. Sayed Hassan Siddique,
Durjoy Kumar Saha
Dept. of Computer Science and Engineering,
Begum Rokeya University, Rangpur,
Rangpur -5400, Bangladesh
sazzad.brur@gmail.com, sayedbrur@gmail.com,
durjoysaha39@gmail.com

Abstract— **Steganography is the art or practice of concealing secrete information into a cover media like text, protocol, file etc. In our proposed algorithm, a new Steganographic technique is developed to hide large amount of data in Bitmap image using the concept of LSB verification method. This method uses first three color components to detect the actual position of the secret bit into the fourth color for embedding and extracting data. This method uses adjacent three colors of image pixels to define the embedding position of message bit into the forth color.**

*Keywords—**Image Steganography; Bitmap Image; LSB Verification Method; Bit Position; Binary Word.***

I. Introduction

Computer has become an important tool for electronic communications, information gathering, recreational activities and many more [1]. Now a day, as the uses of the internet communication is increasing, the scope of hacking information is increasing too [2]. To protect the important data from unauthorized agent, a technique is widely used namely Steganography. It is a technique of hiding data inside other media (image, text, sound, video etc) so that the hidden message will not be detectable by enemy or any other person in normal sense [3]. From the Greek historian Herodotus in his history of the novel Histaeus first use the term Steganography. The Greek origin word Steganography is classified into two parts: Steganos which means "secret or covered" (where you want to hide the secret messages) and the graphic which means "writing" (text) that means "hidden writing" [4]. Cryptography is another term in security factor. Cryptography & Steganography both are used for message security but there security level is different. In the term of Cryptography a person can see the message but he doesn't understand what it is, on the other hand in Steganography the total message will be invisible into a media. Attackers doesn't have any idea what media contain the message and which algorithm use to embed/extract it. For this above reason Steganography is more reliable than Cryptography. Figure-1 shows the main categories of file formats that can be used for Steganography [5]. Almost all digital file formats can be used for Steganography. But among those of all media, digital images are the most widespread cover files used for Steganography, namely Image Steganography due to the insensitivity of the human visual system (HVS). Human Visual System (HVS) does not have a super high sensitivity to the color. That mean it is not capable to identify a very small difference in color, and this characteristic has been taken as an opportunity in Image Steganography. Furthermore, digital images can easily be used as cover files without any suspicion because of their presence everywhere on the Internet [6].

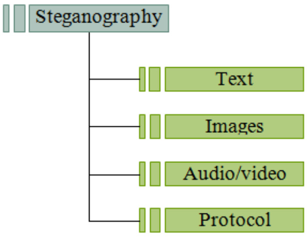

Fig. 1. Categories of Steganography.

Each image hiding system consists of an embedding process and an extraction process. An innocuous-looking original image is used as the cover-image to conceal the secret data. The secret data are embedded into the cover-image by modifying the cover-image to form a stego-image [2].

Image steganography systems have two fundamental characteristics, which must be investigated, in order to evaluate the efficiency of such systems. These are the security and capacity of the steganography system [7]. Image steganography systems can be considered secure if it is impossible for attackers to detect the presence of a hidden message in the stego image by using any accessible means. Moreover, a steganography system is perfectly secure if the statistics of the cover image and the stego image are identical [6]. That means, a Steganography technique should underestimate the following things:

- How much data the stego image can contain.
- How to minimize the change in image color.

- How to remain the file sizes same.

Because of Cyber-crime is increased day by day, image steganography can be more reliable system to pass secret information. Today, Image Steganography technique is being using comprehensively by defense, International intelligence agencies and International community to insure secure data transmission. The commonly used terms while working with information hiding techniques are given in TABLE-I [6].

TABLE I. INFORMATION HIDING TERMINOLOGY

Term	Description
Embedded	Something to be hidden in something else
Stego	Something that has an embedded message in it
Cover	An input which is the original form of the Stego
Embedding	The process of hiding the secret message
Extracting	Getting the embedded message out of the Steganographic message

II. STEGANOGRAPHIC TECHNIQUES IN SPATIAL DOMAIN

The steganographic algorithms operating in the spatial domain as the method for selecting the pixels can be classified into three categories: non-filtering algorithms, randomized algorithms and filtering algorithms [8].

A. Non-filtering Algorithm

The non-filtering steganographic algorithm is the most popular and the most vulnerable steganographic technique based on LSB. The embedding process is done as a sequential substitution of each LSB of the image pixel for each bit of the message [3]. For its simplicity, this method can camouflage a great volume of information [8]. This method also generates an unbalanced distribution of the changed pixels, because the message is embedded at the first pixels of the image, leaving unchanged the remaining pixels [9].

B. Randomized Algorithm

This technique was born as a solution for the problems of the previous method. Each of the sender and the receiver has a password denominated stego-key which is generated through a pseudo-random number generator [8]. This creates a sequence which is used as the index to have access to the image pixel. The message bit is embedded in the pixel of the cover image as the index given by the pseudo-random number generator [8]. The two main features of this technique are: a) use of password to have access to the message and b) well-spread message bits over the image which is difficult to detect compare to the previous one [3].

C. Filtering Algorithm

The filtering algorithm filters the cover image by using a default filter and hides information in those areas that get a better rate [3]. The filter is applied to the most significant bits of every pixel, leaving the less significant to hide information [9]. The filter gives the guarantee of a greater difficulty of detecting the presence of hidden messages [3]. The retrieval of information is ensured because the bits used for filtering are not changed, implying that the reapply the filter will select the same bits in the process of concealment. It is the most efficient method to hide information [9].

In this paper, a new Steganography technique is being developed to hide large data in image using *Non-filtering Algorithm.*

III. PROPOSED TECHNIQUE

Every pixel has three color components, Red (R), Green (G) and Blue (B). If we get the color components of the adjacent pixels, we can write a color series like the TABLE II is showing.

TABLE II. COLORS IN ADJACENT PIXELS

Pixel No	Color No	Color
1	C_{n-3}	R
	C_{n-2}	G
	C_{n-1}	B
2	C_n	R
	C_{n+1}	G
	C_{n+2}	B
3	C_{n+3}	R
	C_{n+4}	G
	C_{n+5}	B
So on		

We propose a Steganography method where adjacent three colors will define a bit position in the next color, which will be used to hide as a secrete message bit. Here, 32 colors that mean almost 11 pixels will be used to hide a single character due to the binary representation of ASCII value of a character (0 to 255 i.e. 8 bits). For the above table, we can summarize the color numbers as shown in figure-2.

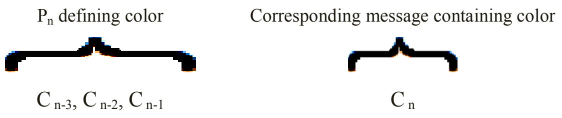
P_n defining color
Corresponding message containing color
C_{n-3}, C_{n-2}, C_{n-1}
C_n

Fig. 2. Colors in use.

In order to hide a message bit into the color information C_n, firstly it is needed to define the bit position P_n. For doing that, the LSBs of C_{n-3}, C_{n-2} and C_{n-1} are used. Three LSBs can define a bit position 0 to 7. The P_n position bit of the C_n color holds the corresponding message bit M_n. The proposed method does not change the P_n bit of C_n, rather it uses the LSB of the color C_n to verify the bit of P_n if it matches with M_n or not. If it matches, the LSB of C_n will be 1, 0 otherwise. Figure-3 shows the flow chart depicting the overall operation.

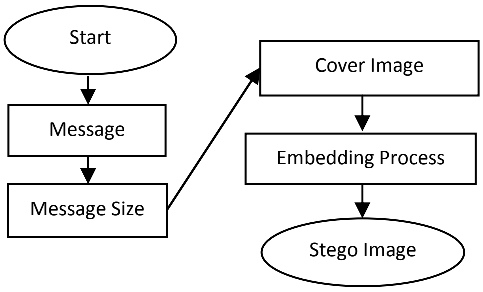

Fig. 3. Flow chart of overall operation.

A. Embedding Process

Firstly, the LSBs of C_{n-3}, C_{n-2} and C_{n-1} color components are collected. Using the collected LSBs, a binary word of three bits is structured. Then it is converted into decimal position P_n. If the position P_n becomes 0 that means it coincides with LSB position, the LSB is changed by the corresponding message bit M_n. Otherwise the bit of P_n is taken to match with the corresponding message bit M_n. If those match, LSB of C_n is changed by 1, 0 otherwise. By this way, this method changes the LSB of the next 4 components and so on till the embedding message exists. The actions are shown in the TABLE-III.

B. Embedding Algorithm

i. Get cover image
ii. Collect 3 LSBs from the color components(R, G, and B) of adjacent pixels
iii. Form a binary word of 3 bits
iv. Convert it to decimal position P_n
v. Check P_n is 0 or not
vi. If yes, set LSB of the next color component by the message bit
vii. If not, compare the message bit M_n to the bit of position P_n of color C_n
viii. If it coincides, set the LSB of the color C_n by 1, 0 otherwise
ix. If embedding message exists, go to step (ii), else go to step (ix)
x. END

TABLE III. ACTIONS TO EMBED MESSAGE BITS

Case	P_n	Bit at P_n position of the color C_n	Message Bit M_n	Modified LSB of C_n
1	Not 0	1	1	1
2	Not 0	0	1	0
3	Not 0	1	0	0
4	Not 0	0	0	1
5	0	No matter what it is	1	1
6	0	No matter what it is	0	0

Figure-4 shows the flow chart depicting the insertion of message bit.

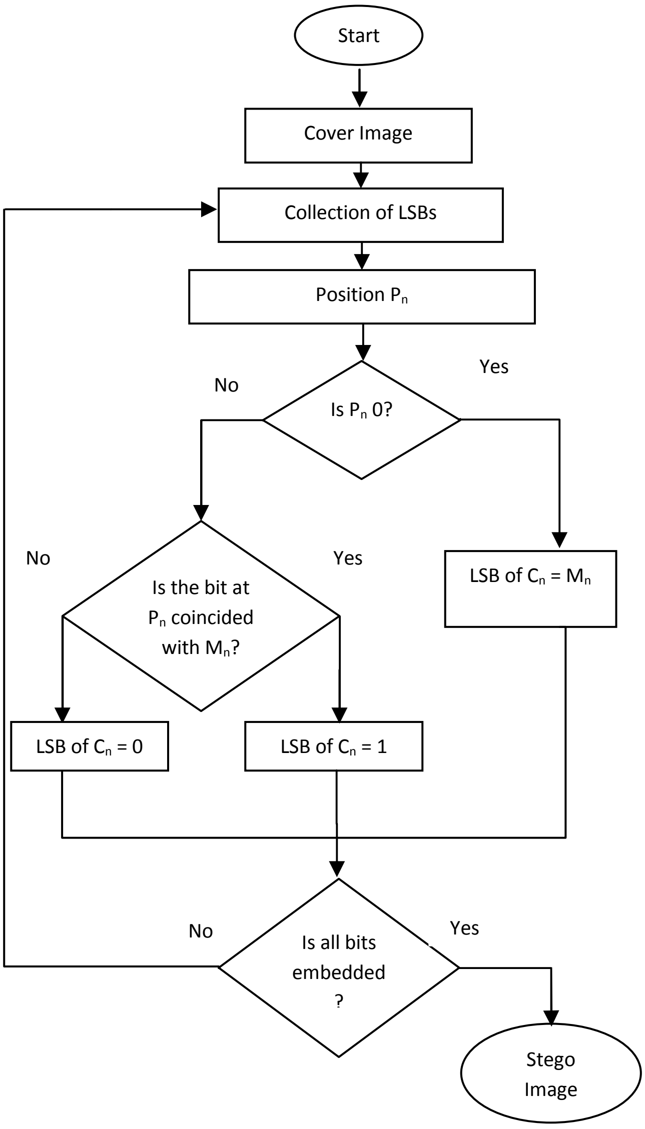

Fig. 4. Flow chart of embedding operation.

If an embedded message is "11000", TABLE-IV describes the changes in pixel value for embedding.

TABLE IV. CHANGES IN PIXEL VALUE AFTER EMBEDDING

Pixel	Color Component	Component Value	Color Number	Binary Word = Position P_n	Bit at position P_n	Message Bit M_n	Modified Color Value C_n
1st	R	00001111	C_1	110_2 = 6_{10}	1	1	00001111
	G	00000101	C_2				00000101
	B	10000000	C_3				10000000
2nd	R	0**1**001010	C_4				0100101**1**
	G	10011100	C_5	010_2 = 2_{10}	0	1	10011100
	B	01001111	C_6				01001111
3rd	R	11010000	C_7				11010000
	G	10001**0**10	C_8				1000101**0**
	B	01001110	C_9	001_2 = 1_{10}	1	0	01001110
4th	R	01001010	C_{10}				01001010
	G	10001101	C_{11}				10001101
	B	101100**1**1	C_{12}				1011001**0**
5th	R	10001010	C_{13}	011_2 = 3_{10}	0	0	10001010
	G	10111101	C_{14}				10111101
	B	01001111	C_{15}				01001111
6th	R	0000**0**110	C_{16}				0000011**1**
	G	11101010	C_{17}	000_2 = 0_{10}	1	0	11101010
	B	01001000	C_{18}				01001000
7th	R	11001100	C_{19}				11001100
	G	0000010**1**	C_{20}				0000010**0**

C. *Extracting Process*

Firstly, the LSBs of C_{n-3}, C_{n-2} and C_{n-1} color components are collected. Using the collected LSBs, a binary word of three bits is structured. Then it is converted into decimal position P_n. If the position P_n becomes 0, collect the LSB of C_n as message bit M_n. Otherwise check the LSB of color C_n. If it is 1, collect the P_n position's bit of C_n, otherwise collect the toggling bit of P_n position as message bit M_n. In this way this method extracts the next embedded bit from the latter 4 components and so on until the entire message bits retrieve. The actions are shown in TABLE-V.

TABLE V. ACTIONS TO EXTRACT MESSAGE BITS

Case	P_n	LSB of the color C_n	Bit at P_n position of the color C_n	Extracting Bit M_n
1	Not 0	1	1	1
2	Not 0	0	1	0
3	Not 0	1	0	0
4	Not 0	0	0	1
5	0	No matter what it is	1	1
6	0	No matter what it is	0	0

D. *Extracting Algorithm*

i. Get cover image

ii. Collect 3 LSBs from the color components (R, G, and B) of adjacent pixels

iii. Form a binary word of 3 bits

iv. Convert it into decimal position P_n

v. Check P_n is 0 or not

vi. If yes, extract the LSB of the next color component C_n as message bit M_n

vii. If not, check the LSB of the next color component C_n

viii. If the LSB is 1, extract the P_n position's bit of C_n, otherwise extract the toggling bit of same position as message bit M_n

ix. If entire message bits do not retrieve, go to step (ii) else step (x)

x. END

TABLE VI. EXTRACTION OF EMBEDDED MESSAGE FROM PIXEL VALUE

Pixel	**Color Component**	**Component Value**	**Color Number**	**Binary Word = Position P_n**	**LSB of C_n**	**Bit at position P_n**	**Message Bit M_n**	**Extracting Message**
1st	R	00001111	C_1	110_2 = 6_{10}	1	1	1	11000
	G	00000101	C_2					
	B	10000000	C_3					
2nd	R	0**1**00101**1**	C_4					
	G	10011100	C_5	010_2 = 2_{10}	0	0	1	
	B	01001111	C_6					
3rd	R	11010000	C_7					
	G	10001**0**1**0**	C_8					
	B	01001110	C_9	001_2 = 1_{10}	0	1	0	
4th	R	01001010	C_{10}					
	G	10001101	C_{11}					
	B	10110**10**	C_{12}					
5th	R	10001010	C_{13}	011_2 = 3_{10}	1	0	0	
	G	10111101	C_{14}					
	B	01001111	C_{15}					
6th	R	0000**0**11**1**	C_{16}					
	G	11101010	C_{17}	000_2 = 0_{10}	0	0	0	
	B	01001000	C_{18}					
7th	R	11001100	C_{19}					
	G	0000010**0**	C_{20}					

Figure-5 shows the flow chart depicting the extracting message bit.

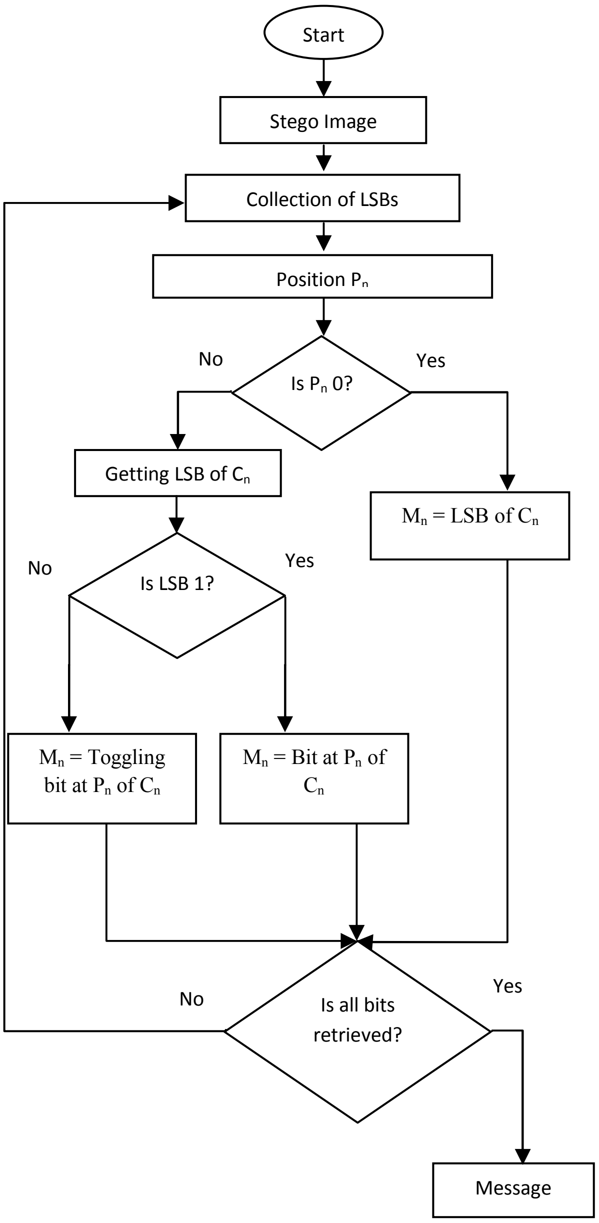

Fig. 5. Flow chart of extracting operation.

E. *Experimental Result*

The proposed technique has been implemented using C# .Net 4, running on Windows 7 operation system. It takes a very short time to execute the program using 24-bit bitmap image (Wheel.bmp) shown in Figure-6. The output of the program is remarkably similar to the original image. The text hidden in the image is given in TABLE-VI and the histogram is shown in Figure-7.

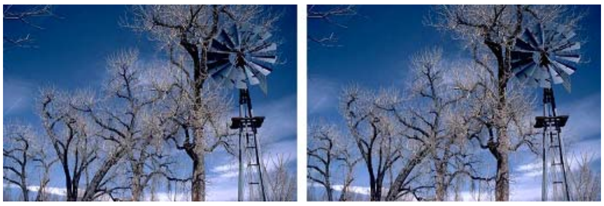

Fig. 6. The input (Left) and the corresponding output (Right) of the program use the proposed technique for hiding data.

TABLE VII. HIDDEN IN DIFFERENT COVER IMAGES

Bitmap	Image Size	Message Size	Hidden Text
Wheel.bmp	134 kB	39 Characters or 312 Bits	Assignment completed. Will return on 9.

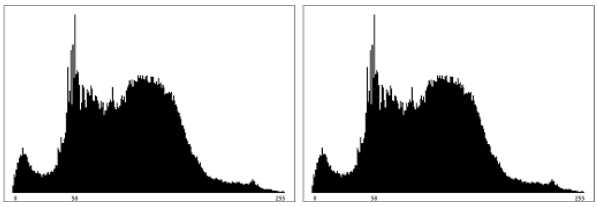

Fig. 7. The Histogram of input (Left) and the corresponding Histogram of output (Right).

It is seen that the proposed sterilization technique does not detectably distort the histogram of the input image.

IV. ADVANTAGES OF THE PROPOSED TECHNIQUE

The proposed algorithm will embed a message changing very low number of bits. This is much more efficient than other steganography technique where LSB is used as verification bit. That is why in the best case, there is no need to change any bit. Approximately, its average case always remains 50%. Using this algorithm only 4 bytes needed to hide a bit of information. That means a 320x240 image can hide a message of 7200 characters which is quite large. Additional facilities of this algorithm image quality won't change, the image size remains same after embedding it and it is immune from attack by comparing histograms as the frequency of appearance of colors in the steganographic image is very similar to that of the cover image.

V. CONCLUSION

In this project we have presented a new system for the Steganography which could be proved a highly secured method for data communication in near future. This paper proposes a new technique that improves the performance of the LSB method hiding information at the cover image and makes it difficult to the unauthorized person to determine the presence of a secret message.

ACKNOWLEDGMENT

We are mostly benefited for receiving the pre-paper on the very topic which is helpful for us to develop our idea. We are grateful to those who guided us keeping up our projected deeds. Our humble respect and thank to them, although it is impossible for us to return what we have got from them within this short time.

REFERENCES

[1] Islam M.O., Mandal A.K., Hossain M.D., "ThirdHand: A Novel and Cost-effective HCI for Physically Handicapped People", 14th International Conference on Computer and Information Technology, 2011, pp: 134-138.

[2] C.-L. Liu and S.-R. Liao, "High-Performance JPEG Steganography Using Complementary Embedding Strategy", Pattern Recognition, vol. 41, no. 9, pp. 2945-2955, 2008.

[3] Goutam Paul. ; Imon Mukherjee. ; "Image Sterilization to Prevent LSB-based Steganographic Transmission", venue in arXiv.org e-Print Archive, arXiv: 1012.5573v1 [cs.MM], Dec 27 2010, Available at http://arxiv.org/ftp/arxiv/papers/1012/1012.5573.pdf

[4] Atallah M. Al-Shatnawi. ; "A New Method in Image Steganography with Improved Image Quality", Applied Mathematical Sciences, Vol. 6, 2012, no. 79, 3907 – 3915.

[5] Islam M. O., "A High Embedding Capacity Image Steganography using Stream Builder and parity Checker", 15th International Conference on Computer and Information Technology, 2012, pp: 458-463.

[6] Saha, A.; Halder, S. & Kollya, S. "Image steganography using 24-bit bitmap images", 14th International Conference on Computer and Information Technology, 2011.

[7] H. Wang and S. Wang, "Cyber Warfare: Steganography vs. Steganalysis", Communications of The ACM, vol. 47, no. 10, pp. 76-82, 2004.

[8] S. Katzenbeisser, and F. Petitcolas, "Information hiding techniques for steganography and digital watermarking," Artech House Books, 1999.

[9] J.J. Roque and J. M. Minguet, "SLSB: Improving the Steganographic Algorithm LSB", Universidad Nacional de Educación a Distancia (Spain), Available at http://www.fing.edu.uy/inco/eventos/cibsi09/docs/Papers/CIBSI-Dia3-Sesion9(1).pdf