International Journal of
Intelligent Systems and Applications in Engineering
IJISAE
ISSN:2147-6799 www.atscience.org/IJISAE
Advanced Technology and Science
Original Research Paper

Lossless Text Compression Technique with LSB Technique to Hide Secret Message inside an Image (CLSB)

Hanan Ali Al Zaloutee[1], Hanan Ettaher Dagez*[2]

Accepted 4th August 2015 DOI: 10.18201/ijisae.61154

Abstract: This paper presents CLSB algorithm to improve and increase the security ofhiding message inside an image by using Least Significant Bit (LSB) method. This research attempts to improve the way has been introduced in [1], where they use digital images to hide Secret Text files. This method is working by distributing data (BMP image format) randomly without use any table to store the path of hiding data. CLSB propose new method to reduce size of dictionary index. Therefore, the main objective of this research is to develop a new method to increase the security by adding a secret key to decode and improve the quality by using LZW method to reduce the size of text file before hiding. Results proved and has also presented in this paper.

Keywords: *Security system, Cryptography, Watermarking, Steganography,LSB, LZW.*

1. Introduction

The need for private and personal communication become the main research concern in now days .This can only happen with privacy in digital communication where confidential information is being shared between two entity using computer communication. To provide confidential and ensure secure communication, there are two main techniques cryptography and information hiding, as shown in Figure.1[2].
Cryptography is the science of converting elements values of secret message from any format (text, Image or sounds.... etc.) into a different values [3]. Information hiding is imbedding information into the cover carrier, e.g. image, or video is imperceptible to the human beings [4].

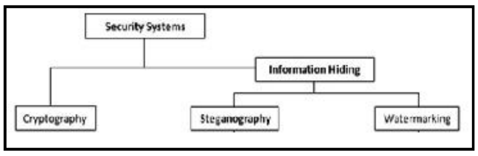

Figure.1 Secrecy system types

Cryptography differs of information hiding in the sense that where cryptography focuses on keeping the contents of a message secret, information hiding focuses on keeping the existence of a message secret[5].
As shown in Figure.1 for information hiding there are two techniques Steganography and watermarking. Steganography use to hide huge data and digital watermarking used to confirm the authenticity of data [6]. Nowadays the most popular technique for hiding data in image is steganography. This is because it can hide big data comparative to another media.

[2]*Faculty of Information Technology at university of Tripoli/Libya*
** Corresponding Author:Email: hanandageez@yahoo.com*
Note: This paper has been presented at the International Conference on Advanced Technology&Sciences (ICAT'15) held in Antalya (Turkey), August 4-7, 2015

There are three methods in image Steganography used for hiding data, Least Significant Bit (LSB), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) [7]. LSB is hiding data in spatial domain, where choose a subset of cover elements and substitute the least significant bits of each element by the message bits. This method considered the easiest way to detect hide message, and lowest robustness against image manipulation, and it is the best of payload capacity [7]. There are many researches improved this method by encrypting text file before hiding but the encryption algorithms grows huge text file size, which reduce the chance of hiding characters and increase the noise in image. Therefore, this paper attempts to improve this method to increase the number of hiding characters by use random pixels, and using a key to ensure security. The Steganography bitmap image type has been used and LZW (Lemple_Zive_Welch)algorithm used to hide text message randomly. Where this algorithm characterized by the ratio compression, time entropy, and compression size [7][8].

2. The Research Scheme

This research algorithm designed to work according to four main mechanisms as follows:

2.1. Compression method:

Before hiding process the need of compressing secret message text file become the first step for the following reasons:

1) To reduce the size of the secret message text file.
2) To reduce the execution time.
3) To initialize the encryption process by changing the secret message contains.
4) Increase the capacity of hiding characters.
5) To reduce the square root error in cover image.

As it is known in LZW method the dictionary contains 256 characters and depends on character’s ASCII code. Thus, the index dictionary starts from 256 and above.As a result, we need more than one byte for every additional index dictionary or iteration. Therefore, CLSB propose new method to reduce size of dictionary index as in the following manner:

1) The dictionary designed as one dimensional array contains all the characters{ ' ','A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J','K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z','1','2','3','4','5','6','7','8','9', ',', '?','!','.',';','(',')','[',']','$','*','@','^','&','%','\n','\r','"','=','>','<'} and the index for every character is their position. As a result the index dictionary starts from 56 rather than 256.
2) All small letters in the text file will be change into capital letters.

The compression part in CLSB works as LZW method. For example if the text file contains "ABCABBC" the result of LZW compression will be "65,66,67,256,257". However, the result using the CLSB compression part is "1,2,3,54,55", which, obviously noticed the difference in the size index.

2.2. Protection and Combination Framework :

To verify the authorized person for the secret message and to increase the security in the CLSB algorithm the secret public key will be length of 16 characters. After the compression process the secret key will be combined with the first sixteen numbers of compression text file results. Two cases will be produced based on this combination:

- First case: if the size of the secret key less than the compression text file length then the combination will take first character of the secret key and translate it into ASCII code with first number of compression text file and then hide them into two pixels inside the image. This will be continued until all the characters of the secret key complete. Then combine every two numbers of the compression secret text file together.
- Second case: if the size of secret key more than the length of compression text file then the combination will be the same and continues with the secret key.

The following explain how the combination technique process is implemented in CLSB algorithm based on previous cases.
The combination in CLSB will be between two numbers (Figure. 2) represent the number in one byte.

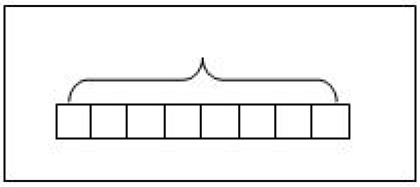

Figure 2. Represent one number (1byte)

The combination starts by taking two numbers to create new two numbers. The first new number created based on two bits (7th,6th) of first number and three bits (7th,6th,5th) of the second number and three bits (5th,4th,3th) of the first number.
The second new number is created based on two bits (4th,3th) of second number and three bits (2th,1th,0th) of the first number and three bits (2th,1th,0th) of the second number.
This way to ensure that every part of the numbers hiding in one pixel and change the result of the compression before hiding process and consider the finalize part of the encryption technique in CLSB. Which, will increases the security of CLSB algorithm.
(Figure.3) Presents example of how the combination process will be implemented. Assume that the numbers has chosen are 67 and 68 the combination will be based on the following equation:

To create new first number the (7th,6th) bits of 67 will use (67 AND 192), to take (7th,6th,5th) bits of 68 will use (68 AND 224) and shift right them two bits. To take (5th,4th,3th) bits of 67 will (67 AND 56) and shift right them 3bits.
New num1= (67 AND 192) OR (SFTR (68 AND 224), 2) OR (SHFR(67 AND 56) ,3))=80.
$192=(11000000)_2$, $224=(11100000)_2$, $56(00111000)_2$.
Using shift right (SFTR) and (OR) operations to put all bits in its position in for the new first number.
To create new second number the (4th,3th) bits of 68 will use (68 AND 24) and shift left them three bits. To take (2th,1th,0th) bits of 67 will use (67 AND 7) and shift left them three bits and to take (2th,1th,0th) bits of 68 will use (68 AND 7).
New num2= (SHFTL (68 AND 24),3) OR (SHFTL (67 AND 7),3) OR (68 AND 7)=12.
$24=(00011000)_2$,$7=(00000111)_2$.

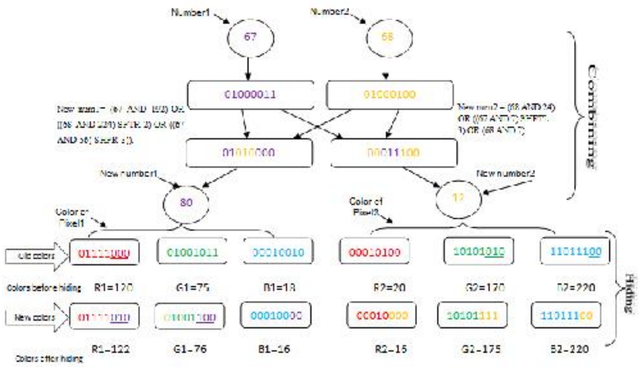

Figure 3. Explain combining and hiding of proposal

2.3. Hiding :

The hiding process will be as the following steps:

1) The hiding operation starts when combining every two numbers.
2) Randomly take two pixels beside each other, then hide them. Figure. 3 Shows example of hiding process.

Assuming that the colours in the first pixel are:
Red1= $((120)_{10}(01111000)_2)$, Green1= $((75)_{10}(01001011)_2)$ and Blue1= $((18)_{10}\ (00010010)_2)$.
In the second pixel are:
Red2= $(20)_{10}$ $(00010100)_2)$, (Green2= $(170)_{10}$ $(10101010)_2)$ and (Blue2= $(220)_{10}$ $(11011100)_2)$.
AND, OR and Shift Right operations are used to hide numbers in Least Significant Bit (LSB). The three bits(7th,6th,5th) of the chosen number hide them into three bits (2th,1th,0th) of the red colour, three bits(4th,3th,2th) of the chosen number hide them into three bits (2th,1th,0th) of the green colour and two bits(1th,0th) of the chosen number hide them into two bits(1th,0th) of the blue colour because it is more sensitivity to the eye.
To implement the hiding process the following equation example how to hide the number (80) as resulted in above combination example into pixel1 by using values of its colours in above:
R1(RED)=(120 AND 248) OR (SHFTR(80 AND 224),5)
$248=(11111000)_2$, $224=(11100000)_2$
G1(Green)=(75 AND 248) OR (SHFTR(80 AND 28), 2)
$28=(11100)_2$
B1(Blue)=(18 AND 252) OR (80 AND 3)
$252=(11111100)_2$, $3=(00000011)_2$.
The hiding process for the second number (12) as resulted in above combination example into pixel2 with the same above steps.
The results of the hiding process are:

The new colours of pixel1 are:
Red1=$(122)_{10}$=$(01111010)_2$,Green1=$(76)_{10}$=$(01001100)_2$ and Blue1=$(16)_{10}$ =$(00010000)_2$.
The new colours of pixel2 are:
Red2=$(16)_{10}$=$(00010000)_2$,$Green_2$=$(175)_{10}$=$(10101111)_2$and Blue2=$(220)_{10}$(11011100)2.

2.4. The distance calculation:

To determine the next two pixels the mechanism will be as the following:

1) After the hiding process into two pixels, the number of 0_{th} bit of the red colour, 1_{th} bit of the green colour and 2_{th} of the blue colour from the second pixel to create value (d).
2) Create key number to prevent that the value (d) result of the first step is not zero. This is key (k) created from first character of the secret key which the value of the first four bits not zeros.
3) Once the key number is identified then the following formula will be calculated:

The next position for the next hiding is (y) =y+d+k

2.5. Extracting:

- Extract hiding: start by reading two pixels from image same mechanism as in hiding process and then read tree colours R,G, and B and find the two hidden numbers Hiding number1 (Hnum1) and Hiding number2 (Hnum2) based on the following equation:

Hnum1 = ((SHFTL(G1 AND 7), 2) + (SHFTL (R1 AND 7) , 5) + ((B1 AND 3))).
7=$(00000111)_2$, 3=$(00000011)_2$.
Hnum2= ((SHFTL (G2 AND 7) , 2) + (SHFTL (R2 AND 7) , 5) + ((B2 AND 3))).
For example the colours of pixel1 are : (R1=$(122)_{10}$=$(01111010)_2$,G1=$(76)_{10}$=$(01001100)_2$ and B1=$(16)_{10}$ =$(00010000)_2$)
Hnum1 = ((SHFTL(76 AND 7), 2) + (SHFTL (122 AND 7) , 5) + ((16 AND 3)))=80
As above in pixel2 to get Hnum2=12.

Extract combining:

After extract hiding will use operation (And, Or, Shift Left and Shift Right) either to extract the combining and find first and second number, as following:

Num1 = (Hnum1 AND 192) OR (SHFTL (Hnum1 AND 7) , 3) OR (SHFTR(Hnum2 AND 56) , 3)

Num1=67.

Num2 = (SHFTL(Hnum1 AND 56), 2) OR (SHFTR (Hnum2 AND 192) , 3) OR (Hnum2 AND 7).

Num2=68.

3. The Proposed Algorithms

Algorithm at the sender side:

```
1) Begin
2) Input: Cover_Image, Secret_Message, Secret_Key
3) Set end symbol in Secret_Message
4)     If (the Cover_Image is small) exit
5) Compress Secret_Message
6) Calculate K from Secret_Key.
7) Get first pixel in image Calculate new distance
8)    While (Compress Secret_Message) is not complete
      do
9)    Read colours in two pixels (R1G1B1,R2G2B2)
10) Read two numbers (if the Secret_Key is not complete ) take one character and convert it into ASCII code and one number of compress Secret_Key) else two numbers in Secret_Message.
11) Combine two numbers
12) Chang colours
13) Calculate new distance
14) end while
15) end
```

Algorithm at the receiver side:

```
1) Begin
2) Input: stego_Image, Secret_Key
3) Calculate K from Secret_Key.
4) Get first pixel in image and Calculate new distance
5)   While there are secret message
6)  Get two pixels near together in image
7)      Read colors in two pixels ( R1G1B1,R2G2B2)
8)      Extract two numbers
9)      if (complete sixteen characters)
10)          cheek secret_key
11)           if (secret_key is false) go to 14
12 )       Calculate new distance
13) end while
14) Decompress result
15) end
```

4. Simulation and Results

The simulation of this algorithm was applied using visual studio C# 2013. Four text files in different size and one bitmap image format type. Hide all text files in same image to get four stego_image and using the improved LSB (CLSB) and the normal LSB as shown in (Figure. 4). Therefore, no difference can be recognized at the first sight. However, the results of using CLSB method have increased the number of hiding characters. Table I and (Figure. 6&7) presents the results of comparing between the CLSB and LSB of how many number of characters each method can hide in one image and shows the hiding and the extracting time and the square root of error erms, which calculated using the following formula [1].

$$e_{RMS} = \sqrt{\frac{1}{N^2}\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}[\bar{I}(r,c) - I(r,c)]^2}$$

e_{rms}: is the square root error.
N: is size image by pixel.
$I^{'}$(r, c): Are elements of Image after hiding.
I (r, c): Are elements of Image before hiding

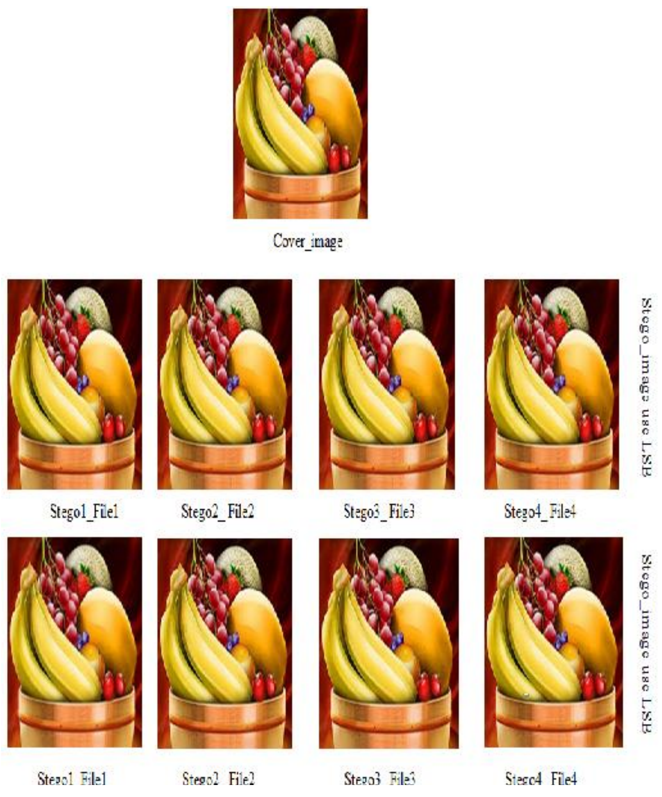

Figure 4. Sego_files in LSB and new LSB

Table 1. Comparative of LSB and CLSB

Text file size	***LSB***				***CLSB***			
	NC	***HT***	***ET***	E_{rms}	***NC***	***HT***	***ET***	E_{rms}
File1:300B	300B	9	76	0.29	300B	5	1	0.27
File2:9KB	8448KB	1384	54130	1.58	9KB	137	29	1.39
File3:18KB	8448KB	2638	50549	1.59	18KB	181	78	1.90
File4:36KB	8448KB	6046	50776	1.58	26.41KB	430	149	2.30
NC	Number of hiding characters.							
HT	Hiding time by millisecond.							
ET	Extracting time millisecond							
E_{rms}	Square root error							

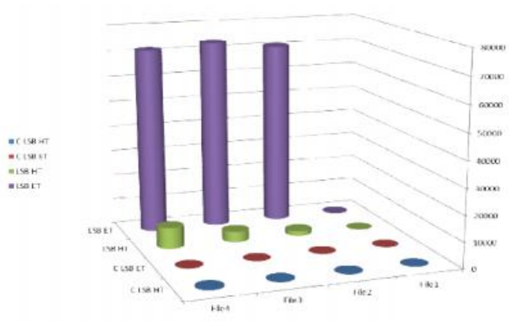

Figure 6. Comparative of hiding and extracting time in LSB and CLSB

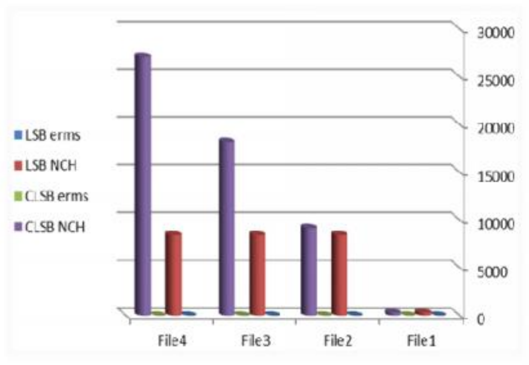

Figure 7. Comparative of Erms in LSB and CLSB

5. Conclusion

This paper presents new method to improve LSB technique to increase security by increase the number of hiding characters and reduce the time when extract the file. The CLSB algorithm has been presented, successful results have also provided and the following fundamentals have concluded:

1) Using LZW method and hide each two numbers beside each other as in CLSB can increase the number of characters compare to LSB.
2) The compression and combining mechanism can works like encryption without huge data.
3) Hiding time and extracting time can be reduced using combining and LZW mechanism.
4) e_{rms} is reduced by using LZW technique .
5) Using image file contains more details can make noise, which difficult to recognize the hidden message.

References

[1] Maan Abdul Khaliq and Aamir Sohail (2010). New Method for Using Digital Images to Hide Secret Text files", Academic scientific journals Iraqi, volume 23 Issue : 6 pages:44-55.

[2] Piyush Goel (2008). Data Hiding in Digital Images: A Steganographic Paradigm. Master's thesis in Computer Science and Engineering, Indian Institute of Technology Kharagpur.

[3] Neha harm, Mr.J.S.Bhatia and Neena Gupta (2014) . Aa Crypto – Stegotechnique Based Secure Data Transmission System, [online] Available: http://www.researchgate.net/publication, May 31.

[4] Stuti Goel, Arun Rana & Manpreet Kau (2013). A Review of Comparison Techniques of Image Steganography. Global Journal of Computer Science and Technology Graphics & Vision, Volume 13 Issue 4 Versions 1.0.

[5] Ravi kumar, Kavita Choudhary, Nishant Dubey (2013). An Introduction of Image Steganography Techniques and Comparison", International Journal of Electronics and Computer Science Engineering, ISSN 2277-1956/V1N3-1000-1005.

[6] Shawn D. Dickman (2007). An Overview of Steganography", James Madison University Infosec Tech report, Department of Computer Science.

[7] Haroon Altarawneh , Mohammad (2011). Altarawneh"Data Compression Techniques on Text Files: A Comparison Study. International Journal of Computer Applications (0975 – 8887) Volume 26– No.5.

[8] R. Kaur and M. Goyal (2013). A Survey on the different text data compression techniques. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 2.