A Developed Text Compression Method for Image Steganography

[1]A. Elhawil , [2]Saad A. Talha , [3]M. Almezoghi and [4]K. AbdiEljwad
[1,3,4]Computer Engineering Department,
Faculty of Engineering, University of Tripoli,
Tripoli, Libya
Email: A.elhawil@uot.edu.ly

[2]College of Electronic Technology – Tripoli
Tripoli – Libya
Email:saseya@ltt.net

***Abstract*— In steganography the message or text to be hidden in an image, text, video or audio is first compressed to reduce its size; this is necessary because small size messages can be sent faster via networks. In this paper, we propose a new method for improving the text compression. The suggested method concerns the compression of English text by representation principle. Any word to be embedded using steganography is replaced by equivalent three letters. This requires fast access data structure such as hash table to store and retrieve the data. The method is tested on image steganography using Least Significant Bits. It also compared with *Lempel-Ziv-Welch* (LZW) compression method. It has proven that the method provides both security and high compression ratio. The results are very motivating and show the efficiency of the proposed method.**

***Index Terms*— Steganography, LSB, Embedding, Extraction.**

I. INTRODUCTION

Hiding a secret information has been used for a long time. In the Greek historian Herodotus writes of a nobleman, Histaeus, who needed to communicate with his son-in-law in Greece. He shaved the head of one of his most trusted slaves and tattooed the message onto the slave's scalp. When the slave's hair grew back the slave was dispatched with the hidden message. In the Second World War the Microdot technique was developed by the Germans. Information, especially photographs, was reduced in size until it was the size of a typed period. Extremely difficult to detect, a normal cover message was sent over an insecure channel with one of the periods on the paper containing hidden information [1]. Today steganography is mostly used on computer communications with digital data being the carriers and networks being the high speed delivery channels. There are many types of steganography methods: text, image and audio.

The performance of digital steganography is based on compression of the embedded data file size. Many methods and algorithms have been investigated. In general there are two basic strategies of compression, the first concerns compressing the transfer media itself such as image as presented in [2] and [3], text [4] and [5] or audio [6]. The second strategy is based on compressing the text to be embedded. One of the most popular data compression methods is Lempel-Ziv-Welch (*LZW*) created by Lempel and Ziv in 1978 (LZ78) and was further refined by Welch in 1984. LZW starts out with a dictionary of 256 characters (in the case of 8 bits) and uses those as the "standard" character set. It then reads the eight bits at a time such as 't', 'r', etc. Then encodes the data as a number that represents its index in the dictionary. Every time it comes across a new substring such as "tr", it adds it to the dictionary; every time it comes across a substring it has already passed, it just reads in as a new character and concatenates it with the current string to get a new substring. The next time LZW revisits a substring, it will be encoded using a single number. Usually a maximum number of entries (say, 4096) is defined for the dictionary, so that the process does not run away with memory. Thus, the codes which are taking place of the substrings in this example are 12 bits long (2^12 = 4096). It is necessary for the codes to be longer in bits than the characters (12 vs. 8 bits), but since many frequently occurring substrings will be replaced by a single code, in the long haul, compression is achieved [7]. However, the aim of this paper is to propose a new method to compress text that will be embedded. The method provides both security and high compression ratio.

II. COMPRESSION METHOD

This section explains the procedure that has been investigated to compress the text. Many algorithms have been used for compressing text. There are two basic strategies that are applied in the design of these algorithms. The first strategy is a statistical method that takes into account the frequencies of symbols. Huffman

coding method is one example. It replaces the occurrence of each character in the text by a new symbol called "codeword". The second strategy is based on scanning the text, replaces some already read segments by just a pointer to their first occurrences. This strategy often provides better compression ratio [8]. Lempel-Ziv-Welch (LZW) algorithm [9] is a very common compression technique. The repeated characters are coded as indexes into a dynamically built dictionary. That means each text needs a dictionary.

In this paper, we propose a new compression method. The main features of this method are both the simplicity and effectiveness. The basic idea of the proposed compression method is as following: First, the English language has 109,645 words that almost cover all the used English words. The small and capital letters of English language are 52 letters. If each English word is coded using only 3 letters (3 letters representation), the total codes that can be obtained are $(52)^3$, which is equivalent to 140,608 words. Note that, this number is more than the English words that exist. The proposed method replaces each English word using only 3 letters. In this manner, the whole text size will be compressed. Table 1 shows how English words can be represented using only 3-letters.

TABLE 1. SAMPLE OF 3-LETTER ENGLISH WORD REPLACEMENT

English words	3-letters representation
assembler	abc
assemblers	bbc
assembles	cbc
assemblies	dbc
assembling	ebc
assembly	fbc
assemblyman	gbc

All English words and their three letters representation are stored in two hash tables: the first hash table is called compression hash table which contains the English words as keys and their 3 letters representations as values. The second one is called extraction hash table which contains the 3 letters representations as keys and the English words as values.

In order to compress the text, we just replace the English words by their equivalent three letters' representation that are stored in the compression hash table. If a word does not exist in the English 109,645 words, we consider it a noun or symbol. In this case, the word is copied as it is, and a symbol '|' will be added at the beginning and end of the word, for example the word "Ali" is converted as |Ali|.

However, based on this method each word in the text can be compressed or replaced by three letters. It is easy to expect the size of the compressed text which is about 3 times the number of words in the text. Assume the input string is “This is a sample string”. It consists of 5 words, 23 letters (including the spaces). The compressed text will be “fcKxPsLaatbFmwI” as follows:

This	Is	a	sample	string
|	|	|	|	|
fcK	xPs	Laa	tbF	mwI

In this example, the compressed text has 3 * 5 = 15 letters. We do not need to include the spaces between letters. In this example, the compression ratio is about 34.78 %.

III. IMAGE STEGANOGRAPHY

We chose image steganography to test our method. To hide a message inside an image without changing its visible properties, slight variations in its colors will be indistinguishable from the original image by a human being. The most popular technique used in image steganography is the least-significant bit or LSB method. The hiding steps are illustrated in Fig. 1. It starts with compressing the message then decoding it. The message can be encrypted using a key to protect the secret message. In the last step the binary message is embedding into the image using LSB method. The obtained image is called as stegano-embed image. The stegano-embed image now can be sent by the website or e-mail. On the receiving side, the extraction process is performed. In this process, we extract the binary data from the image. Then the extraction hash table is used to decompress the message. The stages of the extraction process are shown in Fig. 2.

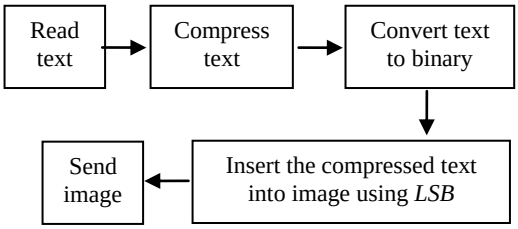

Figure 1. Text hiding steps

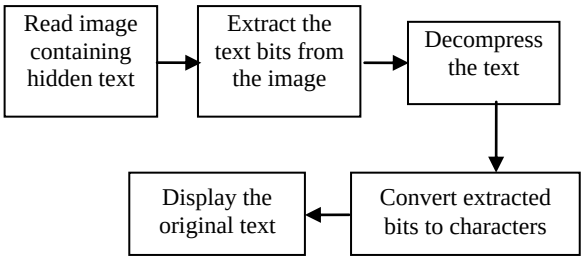

Figure 2. Text extraction steps

IV. LSB IMAGE STEGANOGRAPHY

Least Significant Bit (LSB) has been considered as the simplest and common used steganography technique. In this method, the bits of a message are embedded in the least significant bit of the image. Modulating the least

significant bit does not result in human-perceptible difference because the amplitude of the change is small [10]. Because this method uses bits of each pixel in the image, it is necessary to use a lossless compression format, otherwise the hidden information will get lost in the transformations of a lossy compression algorithm. When using a 24-bit color image, a bit of each of the red, green and blue color components can be used, so a total of 3 bits can be stored in each pixel [11].

V. RESULTS AND DISCUSSION

The suggested method has been implemented using Microsoft Visual studio C# 2010. Fig. 3 shows the developed user interface. It allows the user to enter both the secret message and the image. The message can be written directly in a text box or loaded from a file. By pressing the button "Hide Message in Image", the system will compress the message and hide it into the image. The new image is automatically saved in the same folder as shown in Fig. 4. The second command in the user interface is used to extract an embedded message from an image as illustrated in Fig. 5. Once the user chooses the image and presses the button "Extract Message", the extracted message will be displayed in the text box.

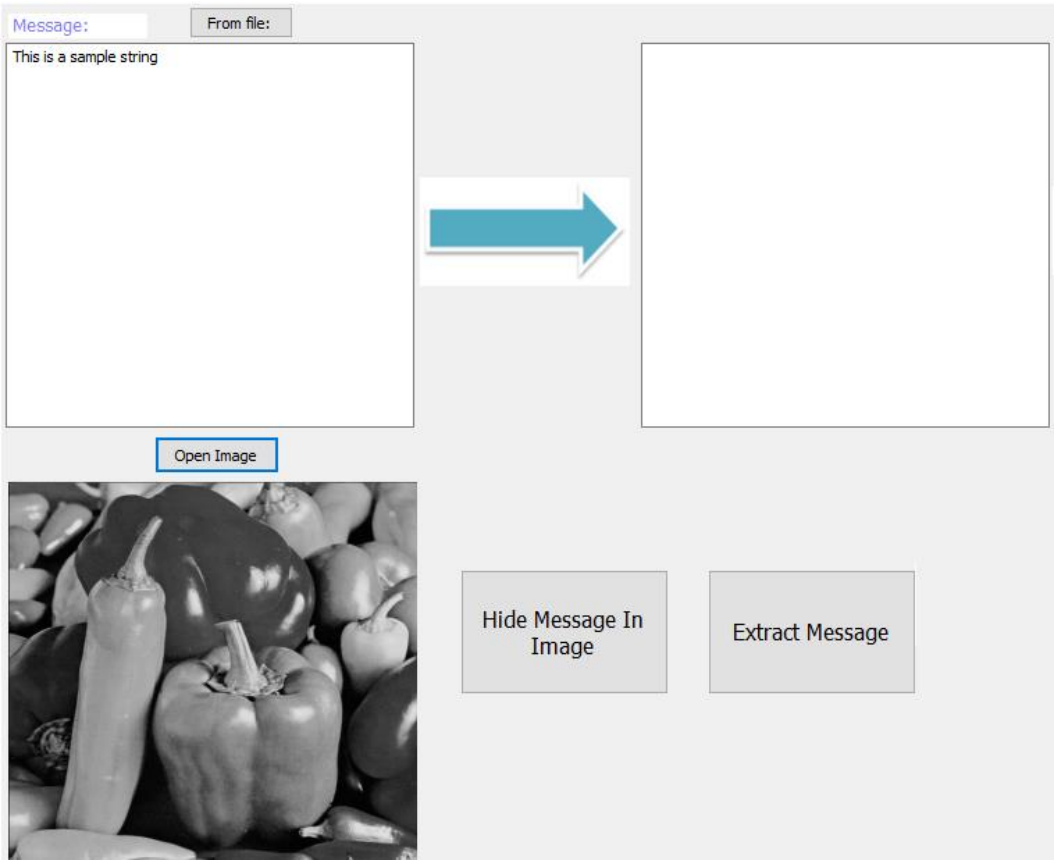

Figure 3. User interface

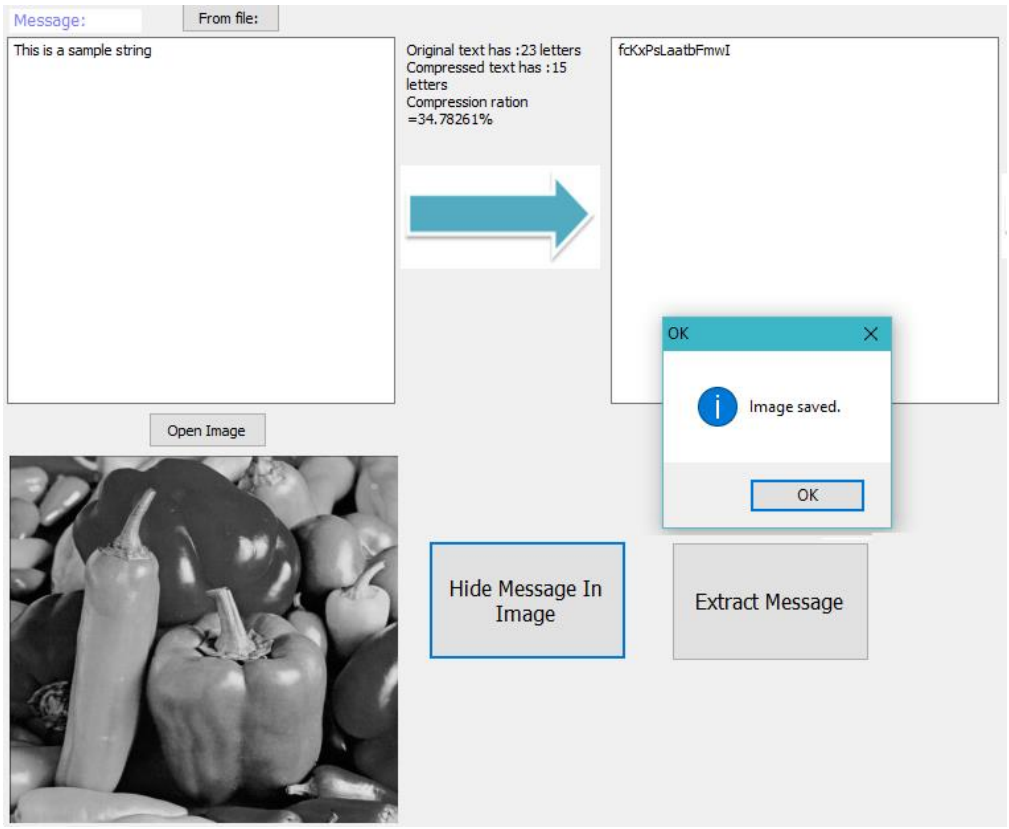

Figure 4. The compressed message is hidden successfully in the image

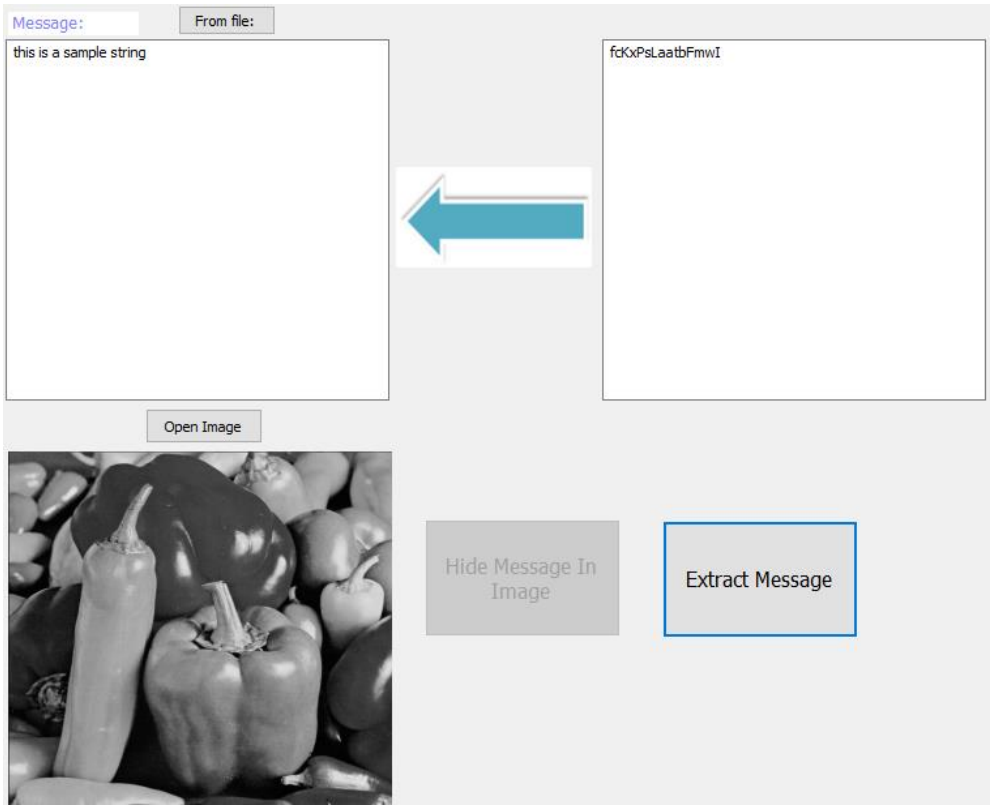

Figure 5. The hidden message is extracted from the image.

In order to verify the efficiency of our compression method, the results of the suggested method are compared with *LZW* algorithm. *LZW* algorithm is implemented using Matlab 2008. The function *norm2lzw* is used. If we consider the text "This is a sample string", the output compressed text of *LZW* algorithm is shown in Table 2. It is clear that, the compression occurs only for two substrings "is" and " s" because they are repeated patterns . The rest of the letters are returned as they are. The compression ratio of this string is only 4.9 %. Obviously *LZW* algorithm is extremely effective when there are repeated patterns in the data.

TABLE 2 RESULT OF *LZW* COMPRESSION METHOD

#	Character input	Code output
1	T	84
2	h	104
3	i	105
4	s	115
5		32
6	is	258
7		32
8	a	97
9		32
10	s	115
11	a	97
12	m	109
13	p	112
14	l	108
15	E	101
16	' ' s	264
17	t	116
18	r	114
19	i	105
20	n	110
21	g	103

We have chosen some arbitrary messages of different sizes. These messages are compressed using the proposed method as well as LZW algorithm. From Table 3 it can be the compression ratio is about 38% for long text. LZW algorithm gives a compression ratio of 53 %. This means that the message contains many repeated patterns. Fig. 6 shows another example of a text consists of 1840 letters. After the compression operation, it becomes 1005 letters. The compression ratio is about 45%. On the other hand, the compression ratio of *LZW* algorithm for the same text is 47.8 %.

TABLE 3 COMPARISON BETWEEN THE PROPOSED METHOD AND LZW COMPRESSION METHOD

Original text letters	Proposed *method*		*LZW* method	
	Compressed text letters	Compression ratio (%)	Compressed text letters	Compression ratio (%)
513	296	42.3	315	38.6
915	584	36.2	519	43.2
2780	1733	37.7	1303	53.1

Finally, the compressed message is embedded in the Basic Metabolic Panel (BMP) image format. The test is done on varies black and white and colored images. An example is shown in Fig. 7. Note that the difference between the original and stegano-embed is indistinguishable. There is no any apparent loss of quality.

VI. CONCLUSION

In summary, the proposed method of text compression is effective. In addition, the message hiding and recovery are effective compared to LZW method. The method may be modified to support repeated patterns and other language rather than English.

Moreover, the compression method can be also used in voice or video steganography process instead of images.

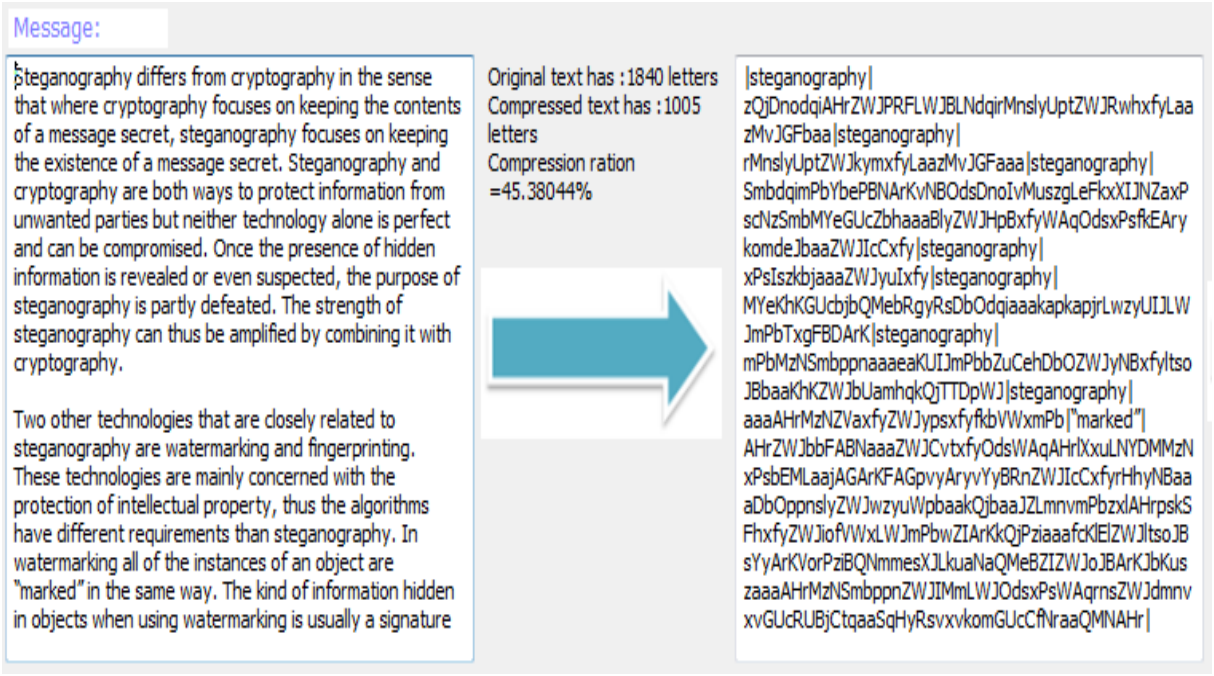

Figure 6. Example of text compression

(a) (b)

Figure 7. a) Original image b) Stegano image

REFERENCES

[1] T. Gomathi and B. Shivakumar, "A secure image encryption algorithm based on ann and rubik's cube principle," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 1, pp. 47- 54, January 2016.

[2] R. Jafari, "Increasing compression of JPEG images using steganography," presented at the IEEE International Symposium on Robotic and Sensors Environments (ROSE), 226-230 , 2011.

[3] A. Cheddad, J. Condell, K. Curran and P. Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Process*., vol. 90, pp. 727-752, 2010.

[4] M. Agarwal, "Text steganographic approaches: a comparison," *International Journal of Network Security & Its Applications (IJNSA)*, vol.5, no.1, January 2013.

[5] N. Rani , J. Chaudhary, "Text Steganography Techniques: A Review," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4 Issue 7, July 2013

[6] P. Jayaram, R. Ranganatha and H. Anupama, "Information hiding using audio steganography – a survey," *The International Journal of Multimedia & Its Applications (IJMA)*, vol.3, no.3, August 2011

[7] V. Rekha and D. Indiramma. "Multi route bandwidth optimization with enhanced compression for network backup configurations," *International Journal of Advanced Scientific and Technical Research* Issue 5 vol. 4, July-August 2015, Available: http://www.rspublication.com/ijst/index.html.

[8] H. Edelsbrunner, "LZW Data Compression," Available: http://www.cs.duke.edu/csed/curious/compression/lzw.html, Duke university, [Accessed 10 Feb 2012].

[9] M. Crochemore and T. Lecroq, "Text data compression algorithms," Algorithms and Theory of Computation Handbook, (1998), Mikhail J. Atallah editor, ch. 12, 12.1-12.23, CRC Press, Bocca Raton, FL.

[10] Nechta I. Effective steganography detection based on data compression, *Vestnik SIBSUTIS*, no.1, 50-55, 2010.

[11] M. Nosrati, R. Karimi and M. Hariri, "An introduction to steganography methods," *World Applied Programming*, vol. 1, no. 3, pp. 191-195, Aug. 2011.