A Novel Method to Hide a Text in an Image Using a Dynamic Symmetric Key and Huffman Coding

Wa'el Ibrahim A. Almazaydeh[1], H. S. Sheshadri[2], S.K. Padma[3]
[1]*Research Scholar, PET Research Foundation, PESCE, Mandya*
[2]*PESCE, Mandya, India,* [3]*Sri Jayachamarajendra College of Engineering, Mysore,India*

Abstract- Steganography and Cryptography are two primary methods to protect data during the transmission or in its place, Cryptography is a technique used to change data from readable form to unreadable form and an intruder can know if there is a cipher text or not, Steganography is a technique used to hide data in other data and an intruder cannot know if there a hidden data or not. This paper shows three methods to hide the text in an image, the first one is the common technique (LSB), the second one is the using a dynamic symmetric key (LSB+KEY) and the third one is the new method to hide a text in an image that is an image Steganography using a dynamic symmetric key and Huffman code (LSB+KEY+HUFF), this paper uses Peak Signal to Noise Ratio (PSNR) value to compare the results among the three methods.

Keywords- *Huffman Code, Least Significant Bit (LSB), ASCII code, PSNR, zigzag scanning*

I. INTRODUCTION

In the ordinary sense of the world, the word 'security' means the state of being safe and the measures adopted to ensure safety. Security isn't a goal or an absolute because in spite of using many of the security procedures available there is no 100 percent security. Human beings have been creating and using many safety procedures since ancient times to protect their lives. In the past, only things with physical presence needed protection and security (physical security); for example: a house was used to get protection against the harshness of nature, guards were used to protect places, and weapons were used to protect human beings. Watchtowers, gates, moats, locks, and other forms of protections [3].

Encryption is a process used to transfer a secret data from sender to receiver in way that prevent an intruder to know what the encrypted file is, it changes the plaintext from readable text to unreadable text. The encryption is divided into categories, the symmetric key and asymmetric encryption, in symmetric key the same key is used to encrypt and decrypt the plaintext by the sender and the receiver, in the asymmetric key different keys is used to encrypt and decrypt the plaintext by the sender and the receiver [5].

In today's digitized world, due to tremendous increase in electronic communication technology, now it is a real and hard problem to send some sensitive data or information through a secure communication channel. This can be obtained by means of two techniques. One-Cryptography and second- Steganography.

Steganography is a method used for security to hide data in other data in way that prevent an intruder to know if there is an encrypted message or not. Hence, the main difference between the cryptography and steganography, in cryptography the plain text is converted to ciphertext from readable form to unreadable form, while as in steganography; the plaintext or the original data is hidden in other data and the observer cannot know if there is a media inside media or not [4].

A Matlab program has been created for this study. It called (Wa'el Steganography).

II. RELATED WORK

A. Previous Study

The authors Wa'el Ibrahim A. Almazaydeh, and H. S. Sheshadri presented three techniques to hide a text data in an image, the first technique is the common technique of the steganography that is the LSB, the second technique is based on using the Huffman code to decrease the size of the text data before hiding process and third technique is based on Arithmetic code to decrease the size of the text data also before the hiding process. The PSNR value is used to compare the results among the three methods, and it indicates that the new two methods is better than the common method of the steganography and the new methods also give more security to the steganography process [2].

Wa'el Ibrahim A. Almazaydeh and H.S.Shesahdri proposed a new method to hide a text in a grayscale image based on creating a key between the sender and the receiver, the key is generated according to the text data and the LSB of image pixels data [3].

Dr. Saad Abdual azize AL_ani, Bilal Sadeq Obaid Obaid showed a new technique to hide data in an image based on converting the character into six bits, by using the b-table to decrease the size of data to four bits and it used similarly to value of image pixels. The receiver gets value of location encrypted by RSA algorithm. There is no data embedded in the image data [6].

Siti Dhalila Mohd Satar, Nazirah Abd Hamid, Fatimah Ghazali, Roslinda Muda and Mustafa Mamat presented a simple method to calculate the secret message data that is wanted to be hidden in the image. This method used Connective Logical (CL) to calculate a binary number of the secret message while the most significant bit (MSB) the pixels were used as a key. MSB is a first bit of each pixel and it has a great significant value. Generally, the MSB of each pixel calculated with secret message using operator Negation, OR and XOR to get a new secret message. The new secret message will be hidden in the

LSB of image pixels. The implementation of this model will produce a low computational complexity of steganography because of the simplicity of the proposed algorithm [7].

Madhavi V.Kale , Prof. Swati Proposed s Steganography system to encrypt the secret message before hiding process using AES encryption algorithm the hide it in image, video and audio using the LSB method. But the authors didn't implement any real data or get any implementation results.

[8].

Amanjot Kaur, Dr. Bikrampal Kaur proposed a K-Modulus method for hiding a secret information in an image. According to obtained results from the PSNR the K-Modulus gives a good result of the steganography process and good results for the security process [9].

Niels Provos and Peter Honeyman showed the steganography techniques, steganography using discrete cosine transform, detecting the steganography process by statistical steganalysis, discussing the using the information hiding in the secret communication and watermarking. [10].

Yun-Te Lin, Chung-Ming Wang, Wei-Sung Chen, Fang-Pang Lin, and Woei Lin proposed a new technique to hide data for HDR images encoded by the OpenEXR format. The presented method hides data in the 10-bit mantissa field of each pixel, while the 1-bit sign and 5-bit exponent fields are kept intact. They recommend an optimal base allowing secret messages to be hidden with the least pixel distortion. An aggressive bit encoding and decomposition scheme is introduced herein, which offers the benefit for hiding an extra bit in a pixel group without incurring pixel distortion [12].

B. Steganography

Steganography technique consists of the following elements as shown in the following Figure 1:

- Secret Message: characters on the original form that you want to hide it in the original image or in the cover media.
- Stegokey: the key that used in the Steganography process to hide the secret message in the original image.
- Cover Media: the medium that is used to conceal the secret message inside it such as: text, image, video, audio, etc.
- Encoding Algorithm: the method that is used in Steganography process for hiding the secret message in the cover media to create a stego media.
- Stego Media: the medium after hiding the secret message in the cover media by using the Steganography method, Stegokey and encoding algorithm.
- Decoding Algorithm: the method that is used to in Steganography process for extraction the secret message from Stego media.

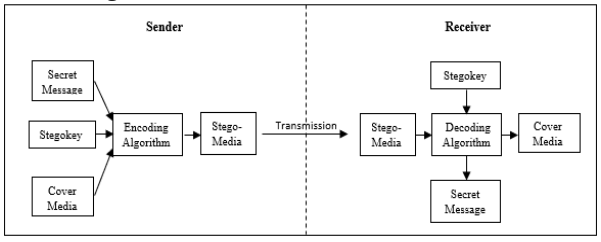

Fig.1: Steganography technique

a. ASCII Code

American Standard Code for Information Interchange (ASCII) is the most common format for characters in the computer systems. In an ASCII code, each alphabetic, numeric, or special character is represented with 7 bits binary number (a string of seven 0s or 1s). For example, the ASCII Code for (R, r, Y, * and @) are (82, 114, 89, 42 and 64) respectively. In this paper, the ASCII code is used to convert each character of the secret message to the corresponding value in ASCII code.

b. Least Significant Bit (LSB)

The most common technique used for Steganography is the Least Significant Bit (LSB). In steganography, LSB method substitutes each bit of the binary text bit in LSB bit of each pixel in the original image. For example: if we have 8 bytes of image data and we want to hide the number 148 which is represented in ASCII code as 10010100 [3]. Figure 2 shows how the LSB technique for the last example.

We shall Hide 148 which is represented as 10010100 in ASCII code by using one bit substitute:

Byte 1	**Byte 2**	**Byte 3**	**Byte 4**
1000010**0**	1000011**0**	1000100**1**	1000110**1**
1	0	0	1
1000010**1**	1000011**0**	1000100**0**	1000110**1**

Byte 5	**Byte 6**	**Byte 7**	**Byte 8**
0111100**1**	0110010**1**	0100101**0**	0010011**0**
0	1	0	0
0111100**0**	0110010**1**	0100101**0**	0010011**0**

Fig.2: Least Significant Bit (LSB) Technique.

c. Zigzag Scanning

Zigzag scanning is used to choose the image pixels to increase the security, because the scanning image is not row by row but it is according to the figure 3. The image pixels that will be used to embed the secret message bits inside it are chosen through a method of zigzag scanning.

1010111**1**	1011010**1**	1100111**0**	1010111**1**	1001001**1**
0011100**1**	0111011**1**	1010101**1**	1010101**1**	1111111**1**
1110000**0**	1110011**1**	1110011**1**	1110011**1**	1110011**1**
1110011**0**	1010100**0**	1100001**1**	1111011**1**	1110011**0**
0011001**1**	0010010**0**	1111011**1**	0101010**1**	0110101**1**
1101001	0010110	1111110	0011100	0011101

1	**1**	**1**	**0**	**1**

Fig.3: Zigzag Scanning

d. Huffman Coding

It was developed by David A. Huffman while he was a Ph.D. student at MIT, and published in the 1952 paper "A Method for the Construction of Minimum-Redundancy Codes". Huffman code is an algorithm to compression based on the frequency of occurrence of a symbol in the file that is being compressed. For example to compression the message (ABEACADABEA) we need to construct a Huffman tree which is the button-up approach according of the following steps [1]:

- Count the frequency of each character in the message as a list as shown in the table 1.
- Sort the list by frequency and make the two lowest elements into leaves, creating a parent node with a frequency that is the sum of the two lower element's frequencies.
- The two elements are removed from the list and the new parent node is inserted into the list by frequency. So now the list, sorted by frequency.
- You then repeat the loop, combining the two lowest elements.
- You repeat until there is only one element left in the list.

Table 1: the frequencies and probabilities of the text (ABEACADABEA)

Symbol	frequency	Probability
A	5	5 / 11 = 0.45
B	2	2 / 11 = 0.18
C	1	1 / 11 = 0.09
D	1	1 / 11 = 0.09
E	2	2 / 11 = 0.18

To generate a Huffman code you traverse the tree to the value you want, outputting a 0 every time you take a left hand branch and a 1 every time you take a right hand branch [1]. Figure 4 illustrate the Huffman tree for the text (ABEACADABEA).

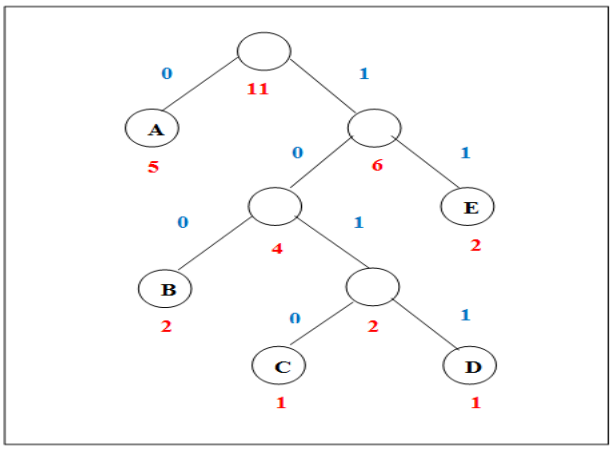

Fig.4: Huffman tree to (ABEACADABEA) [1]

According the above Huffman tree we obtain the following code word in table 2.

Table 2: The code word of the text (ABEACADABEA) using Huffman tree

Symbol	Code word
A	0
B	100
C	1010
D	1011
E	11

After completion Huffman tree to the text (ABEACADABEA) we obtain (23 bit) code word:

01001101010010110100110

While the message in ASCII code is represented as 77 bits (11 characters × 7 bits), so Huffman code saves more than 25% in the size of the message [1].

e. PSNR

Peak Signal to Noise Ratio (PSNR) (equation 2) is measured on a logarithmic scale. It depends on the mean squared error (MSE) between an original image and stego image, relative to $(2^n - 1)^2$ (the square of the highest-possible signal value in the image, where n is the number of bits per image sample) [11].

$$MSE = \frac{\sum_{M,N} [A(m,n) - B(m,n)]^2}{M \times N} \quad (1)$$

$$PSNR_{db} = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (2)$$

"PSNR can be calculated easily and quickly and is therefore a very popular quality measure, widely used to

compare the 'quality' of compressed and decompressed video images" [11].

When the PSNR value is high, the change in the image resolution is low, when the PSNR value is low the change in the image resolution is high and when the PSNR value equal to infinity that means no any bit change between the original image and the stego image, thus, no any change on the image resolution.

C. Methodology

This paper shows three method to hide text in an image, the first one is the common method for Steganography that is (LSB), the second one is image Steganography using LSB and a dynamic Symmetric key (LSB+KEY) and the third one is the new method that is image Steganography using a dynamic symmetric key and Huffman coding (LSB+KEY+HUFF). Figure 5 shows the methodology of this study.

The Huffman coding has been used here for two reasons: the first reason is to reduce the size of the secret message and thus reduce the number of pixels that will need to hide the secret message in it and thus get better results for the Steganography process, and the second reason is to give the Steganography process more security.

The results of the two methods have been compared using the (PSNR) value between the original image and the stego image.

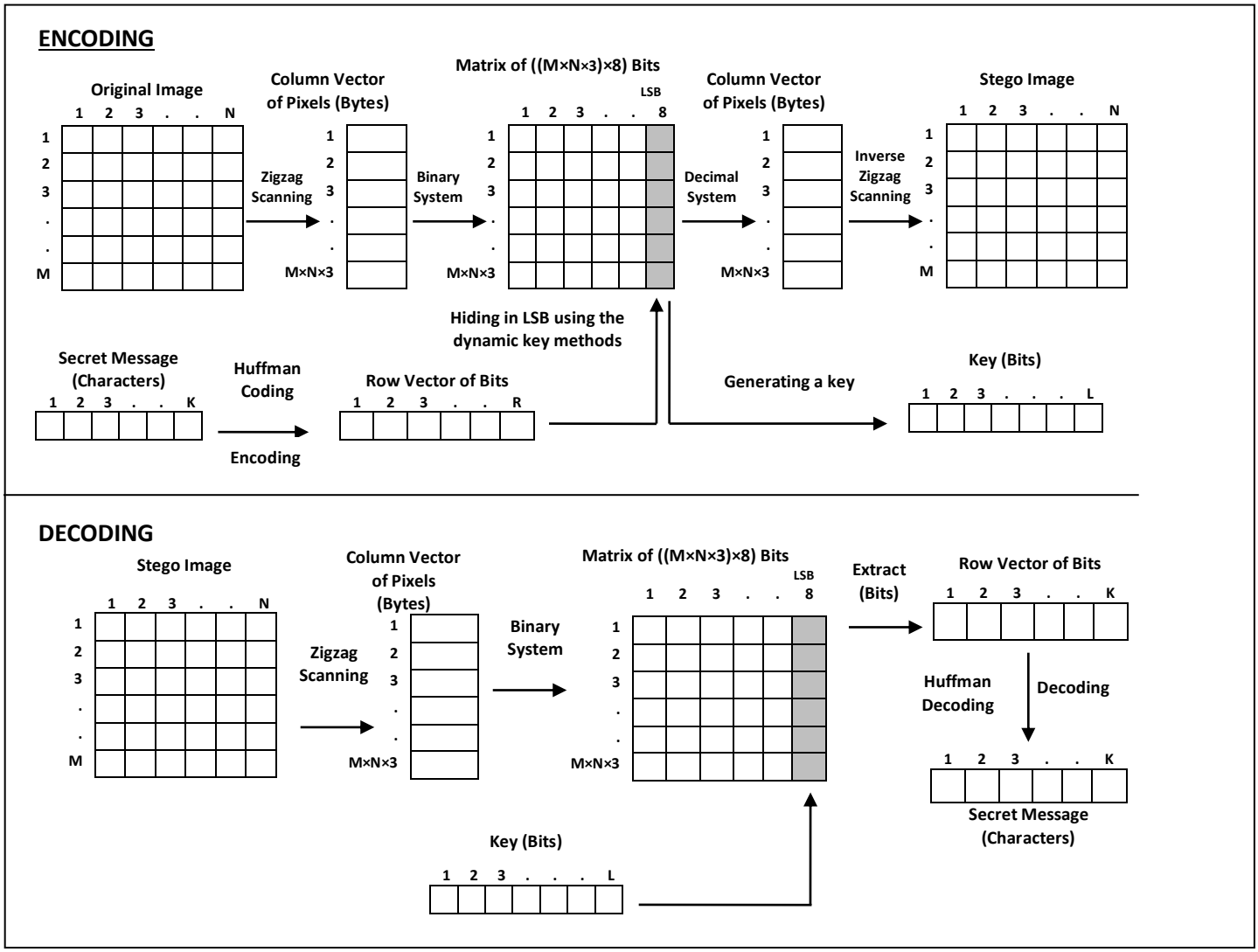

Fig.5: The Encoding and Decoding of the (LSB+KEY) method.

a. Image Steganography LSB

The common technique used to hide a text in an image is the Least Significant Bit (LSB), it based on converting the text to binary values using the ASCII code then hiding it in the LSB of the image pixels bits.

The size of the secret message (text) that is allowed to be hidden by this method is computed as:

$$S1 = (M \times N \times 3) - 27 \quad (3)$$

Where (S1) is the size of the secret message, M is the number of rows in the image, N is the number of columns in the image, (3) is the number of planes, and (27) is represented as: the first seven bits from (1 to 7) are reserved to the Steganography type may be [1, 2, 3, …, 127], for example, when the Steganography type equals 2 means that Steganography process is another Steganography process and etc. The bits from (8 to 27) are the length of the secret message.

b. Image Steganography using a Dynamic Symmetric Key Algorithm

This method is created and published by (Wa'el Ibrahim A. Almazaydeh) [3], it is based on creating a key between the sender and the receiver. Figure 6 shows the encoding and decoding of the algorithm. This algorithm was used for grayscale images. This paper develops the algorithm to colored images and uses the Huffman code to decrease the size of the secret message.

This method hides a secret message in a colored image. It converts the image pixels to binary values using zigzag scanning (as shown in figure 4) with size equal to (M×N×3×8) bits, where (M) is the number of rows in the image, (N) is the number of columns in the image, (3) refers to the number of planes (red, green, and blue), and (8) is the number of bits (each pixel in the plane is represented by 8 bits) [3].

Then, gets the two least significant bit of each pixel value according to the position, where the (LSB) position equal to 0 and the bit before LSB position (LSB-1) equal to 1. In a parallel process, the secret message is converted to a row of binary values with size equal (1×K), where K is the number of bits in the secret message. After this, each bit of the secret message is compared with the two bits of the (LSB). Figure 5 shows the encoding and decoding that has been done.

There are three instances of matching process of the image Steganography using a dynamic symmetric key [3]:

1) If the bit of the secret message matches the position 0 of the (LSB), the key will be 0.
2) If the bit of the secret message matches the position 1 of the (LSB), the key will be 1.
3) If the bit of the secret message doesn't match the first and the second position of the (LSB); we will change the position 0 of the (LSB) to the value of that bit of the secret message and the key will be 0.

At the end of this process, we get a row vector that is the key. The key refers to the position of the secret message in the stego image. This key, is a shared secret key, between the sender and the receiver; and without this key, the receiver will not be able to get the secret message. The key is considered the base of this method [3].

The size of the data (secret message) that can be hidden into the image by using this method can be calculated by using the following formula:

$$S2 = (M \times N \times 3) - 27 \quad (4)$$

Where (S2) is the size of the secret message, M is the number of rows in the image, N is the number of columns in the image, (3) is the number of planes, and (27) is represented as: the first seven bits from (1 to 7) are reserved to the Steganography type may be [1, 2, 3, …, 127], for example, when the Steganography type equals 2 means that Steganography process is another Steganography process and etc. The bits from (8 to 27) are the length of the secret message.

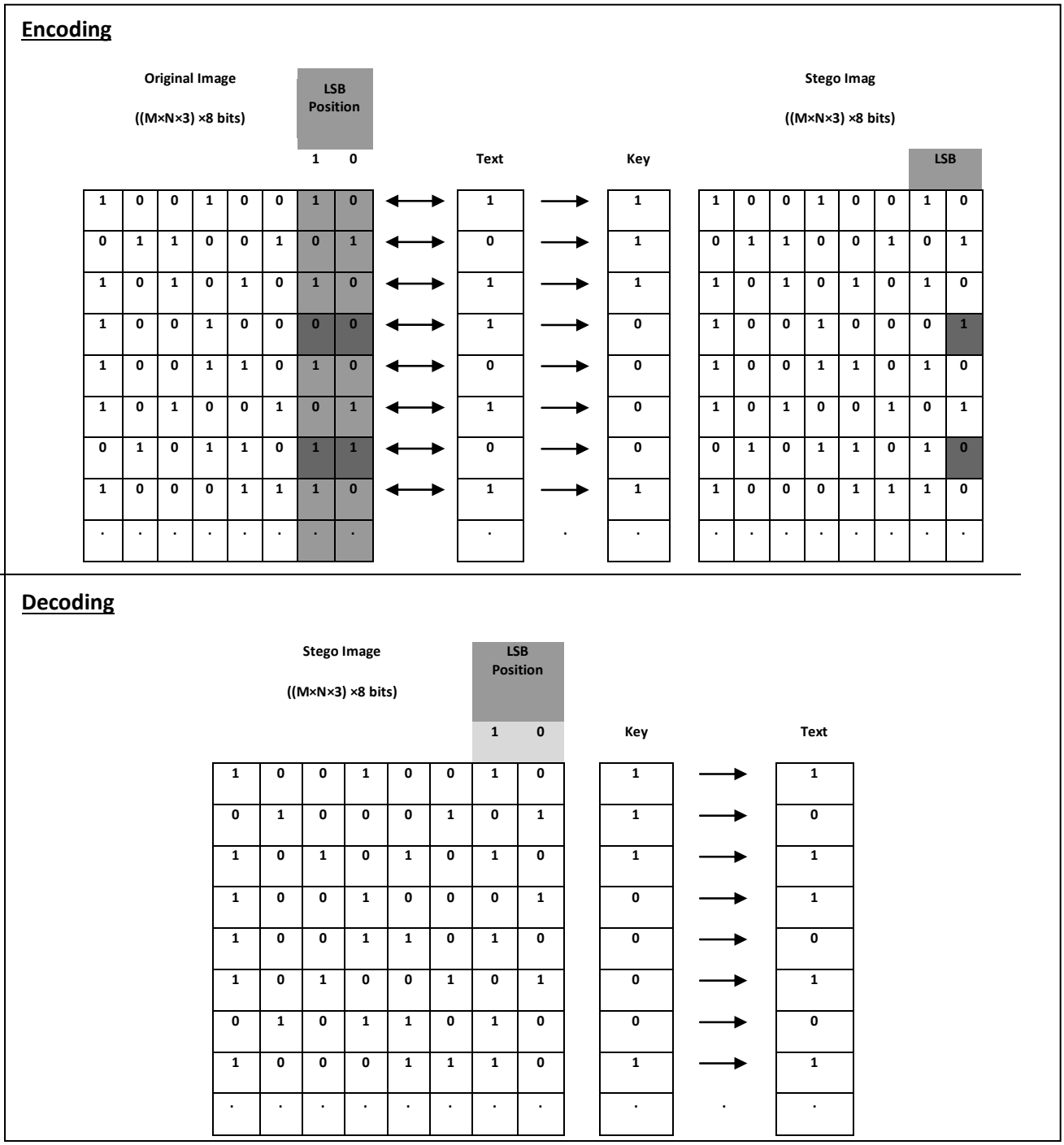

Fig.6: The Encoding and Decoding of the Dynamic Symmetric Key Algorithm.

c. Image Steganography using a dynamic symmetric key and Huffman coding

Image Steganography using LSB and dynamic symmetric key using Huffman code algorithm is an enhancing method for the image Steganography using a dynamic symmetric key algorithm.

Huffman coding is the most common method to compress the text; it used here to reduce the size of the secret message. Results obtained from this method are better than the image Steganography using LSB and a dynamic symmetric key.

The size of the secret message after Huffman coding process that allowed to be hidden by using this method (LSB+KEY+HUFF) is computed as the following formula:

$$S3 = (M \times N \times 3) - 47 \qquad (5)$$

Where (S3) is the size of the secret message, (M) is number of rows in the image, (N) is the number of columns in the

image, (3) is the number of planes in the colored image, (47) is the number of reserved bits for this algorithm; where the bits from 1 to 7 are of the Steganography type. The bits from 8 to 47 are the length of two variables (A and B) that it is needed for Huffman code compression process.

D. Experimental Results

a. The Implementation

A MATLAB program has been developed to implement the algorithms. The program is called (Wa'el Steganography) relative to the name of the first author of this paper.

The images that have been used for implementation the algorithms are color images (RGB images) of the type (JPEG, PNG, and PMB).

III. ENCODING

Petra colored image has been used with size (845×709×3) pixels of type of (JPEG). The secret message that has used is shown in the figure 7. The number of characters of the secret message are (2136) and the number of bits are (14952) bits.

The following steps explain the encoding process of the (LSB+KEY) algorithm and (LSB+KEY+HUFF) algorithm using the (Wa'el Steganography) program:

- Pressing on the push button (open image) to choose the image Petra image from the bath storage.
- Pressing on the push button (open text) to choose the secret message that is saved as txt file shown in the figure 7. The text also will appear in the text on the program window.
- Opening the Steganography method list to choose the Steganography method (LSB+KEY or LSB+KEY+HUFF).
- After that, a dialog window will appear to ask the user to save the key. The key that has been generated for the secret message and the Petra image using the (LSB+KEY) method is shown in the figure 9.
- Then, the stego media will appear in the stego image area.
- Pressing on the push button (Save Stego Image) to save the stego image on the disk.

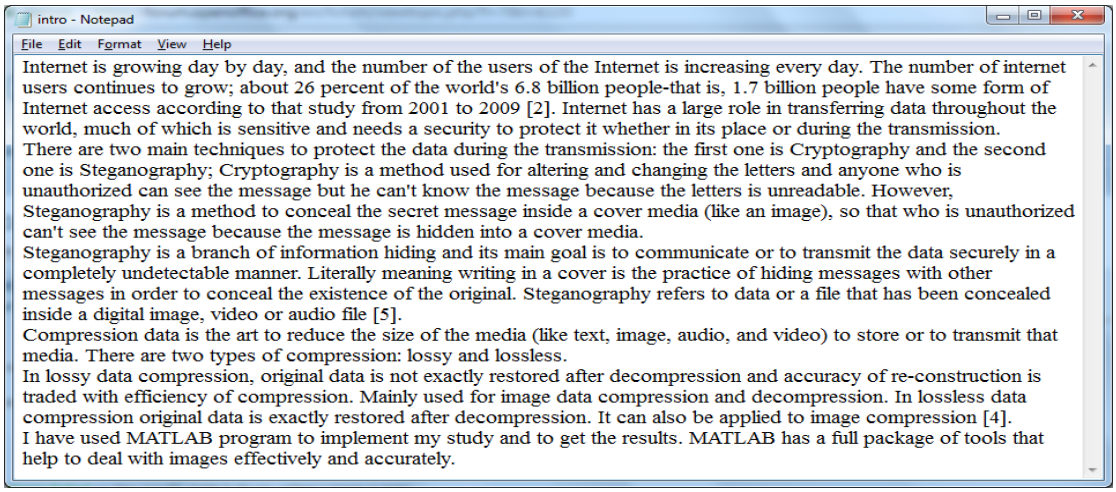

Fig.7: The secret message.

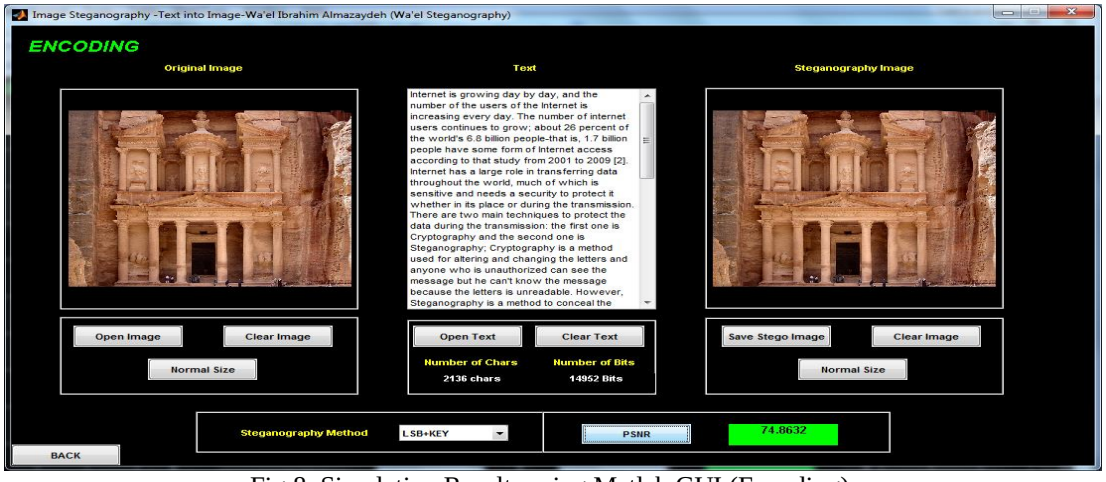

Fig.8: Simulation Results using Matlab GUI (Encoding).

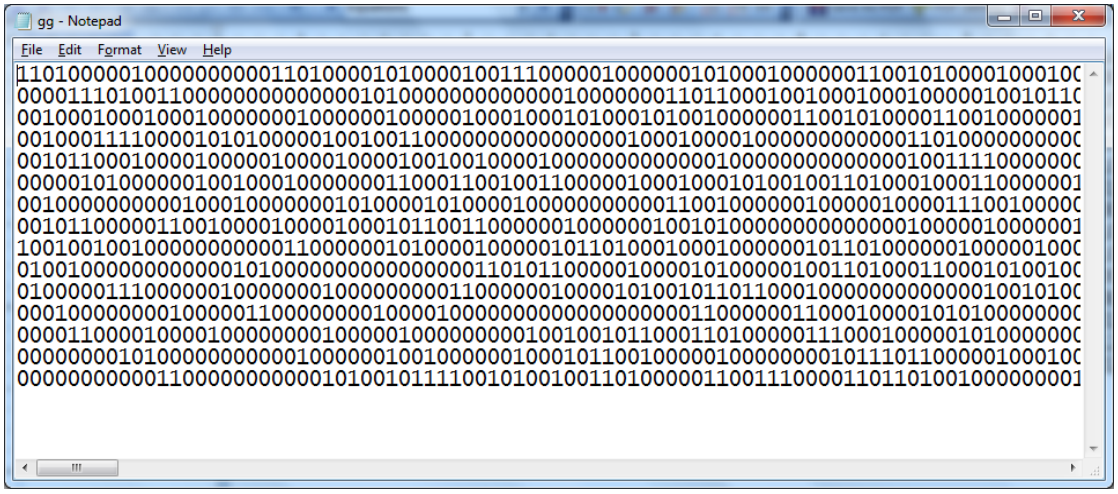

Fig.9: The key.

IV. DECODING

The following steps explain the decoding process of the (LSB+KEY) algorithm and (LSB+KEY+HUFF) algorithm using the (Wa'el Steganography) program that is shown in the figure 10 for decoding process:

- Pressing on the push button (Open Stego Image) to choose the stego image from the bath storage.
- Pressing on the push button (Show). The extraction process begins to determine which Steganography process has been performed on this image and hence the data are extracted from the image.
- Because the stego image that was created depends on the key, dialog window will appear to ask the user to choose the key from the disk storage.
- The user will choose the key fro the disk storage as a text file.
- The extracted data will appear in the text area on the program.
- The user can save the secret key by clicking on the push button (Save Text) to save it as a text file.

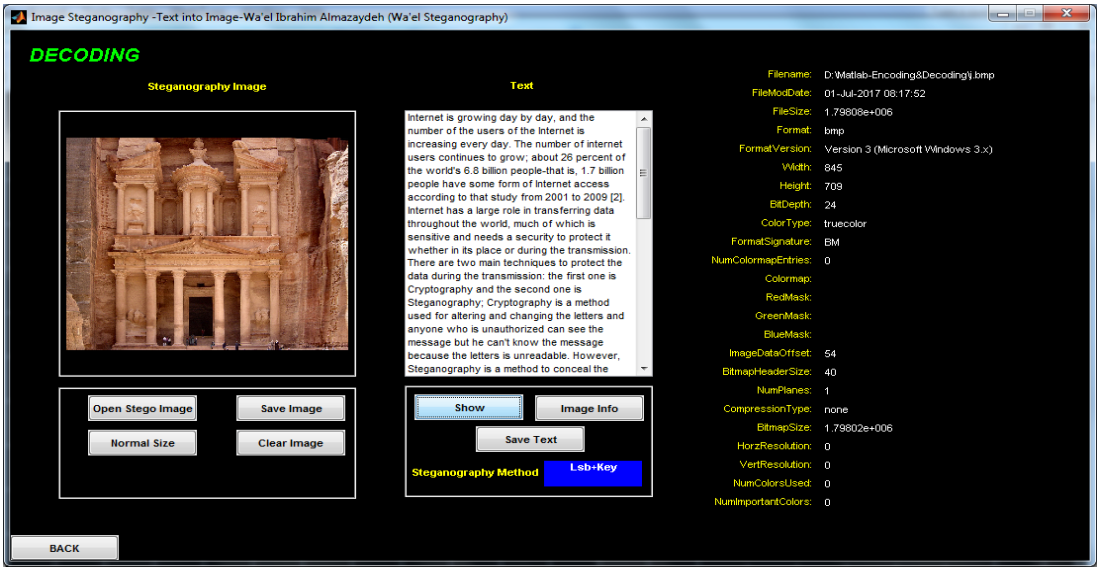

Fig.10: Simulation Results using Matlab GUI (Decoding).

The maximum size of data (secret message) that can be embedded in this image using the LSB method or the image Steganography using a dynamic symmetric key method is:

$$(1024 \times 1024 \times 3) - 27 = 3145701 \text{ bits.}$$

The maximum size of data (secret message) that can be embedded in this image using the image Steganography using a dynamic symmetric key and Huffman coding algorithm **after** coding process is:

$$(1024\,024 \times 3) - 47 = 3145701 \text{ bits.}$$

V. THE RESULTS

Figure 7 shows the secret message that has been hidden using the methods: the common technique LSB, LSB+KEY

and LSB+KEY+HUFF algorithms. Table 3 and Figure 11 shows the results obtained after applying the algorithms.

The first time of the implementation of the secret message was one copy of the secret message, the second time of the implementation of the secret message was five copies of the secret message, the third time of the implementation of the secret message was ten copies of the secret message, the fourth time of the implementation of the secret message was fifteen copies of the secret message, the fifth time of the implementation of the secret message was twenty copies of the secret message.

Table 3: PSNR values of LSB, LSB+KEYand LSB+KEY+HUFF Algorithm.

The number of copies of the secret message	The number of Characters	The number of Bits	Steganography method		
			(LSB) PSNR	(LSB+KEY) PSNR	(LSB+KEY+HUFF) PSNR
1	2136	14952	71.9532	74.8632	76.3715
5	10680	74760	64.9261	67.932	69.8661
10	21360	149520	61.9287	64.9174	66.9297
15	32040	224280	60.1774	63.1574	65.195
20	42720	299040	58.9292	61.9073	63.9323

Figure 11 shows the results (as a chart) for the same values in table 3.

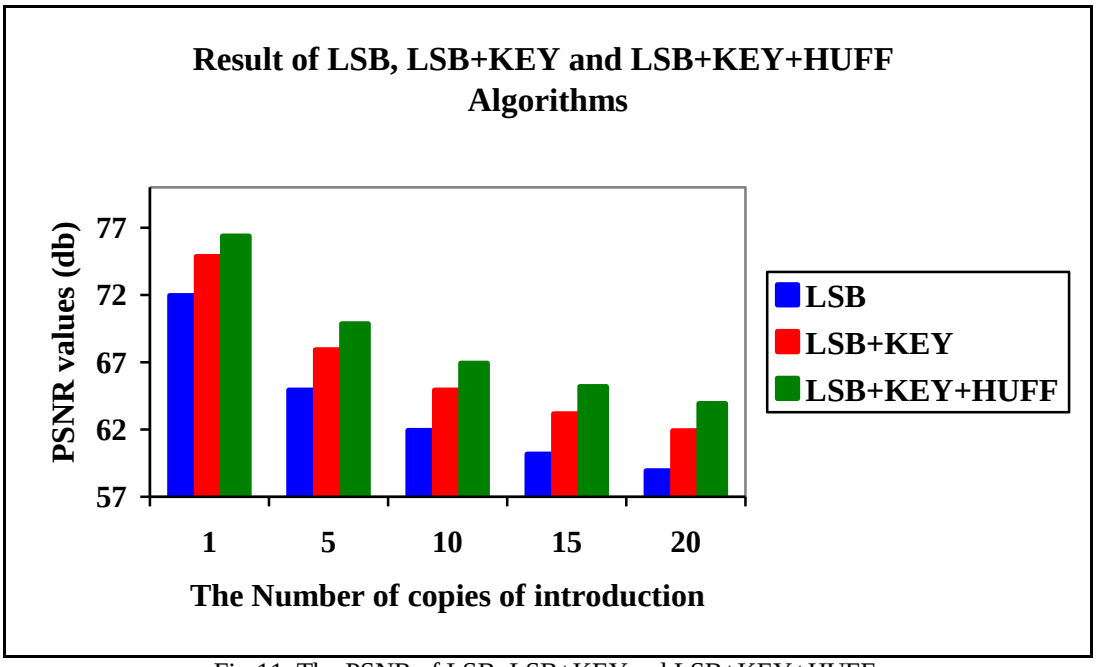

Fig.11: The PSNR of LSB, LSB+KEYand LSB+KEY+HUFF.

VI. CONCLUSION

This paper shows three methods to hide a secret (text) message in a colored image: the first one is the common technique that is the LSB (LSB), the second one is the image Steganography using a dynamic symmetric key (LSB+KEY) and the third one is the new technique that is the image Steganography using a dynamic symmetric key and Huffman coding (LSB+KEY+HUFF). The (LSB+KEY+HUFF) gives more security than the (LSB) or (LSB+HUFF) because the using of the Huffman code and the performances of the results that have been compared using the PSNR values of individual algorithms indicates that the enhancing method gives better output from the LSB and LSB+KEY with respect to the PSNR values.

VII. REFERENCES

[1]. Wa'el Ibrahim A. Al-Mazaydeh. Image Steganography using LSB and LSB+Huffman Code. International Journal of Computer Applications (0975 – 8887), Volume 99– No.5, August 2014.

[2]. Wa'el Ibrahim A. Almazaydeh, H. S. Sheshadri. Image Steganography using LSB, LSB+Huffman Code, and LSB+Arithmetic Code. International Journal of Computer Applications (0975 – 8887), olume 155 – No 11, December 2016.

[3]. Wa'el Ibrahim A. Almazaydeh, "Image Steganography using a Dynamic Symmetric Key", 2nd International Conference on

Inventive Computation Technologies (ICICT – 2017), organized by RVS Technical campus during August 24-25 2017, IEEE, Coimbatore, India.

[4]. Abhishek Koluguri, Sheikh Gouse, Dr. P. Bhaskara Reddy. Text Steganography Methods and its Tools. International Journal of Advanced Scientific and Technical Research, Issue 4 volume 2, March-April 2014.

[5]. Pramendra Kumar, Vijay Kumar Sharma. Information Security Based on Steganography & Cryptography Techniques: A Review. International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 10, October 2014.

[6]. Dr. Saad Abdual azize AL_ani, Bilal Sadeq Obaid Obaid. A Steganography Method to Embed Text in Image without Change Structure of Image. INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTER RESEARCH, Volume 3 issue 1 January 2015 Page No.824-828 ISSN :2320-7167.

[7]. Siti Dhalila Mohd Satar, Nazirah Abd Hamid, Fatimah Ghazali, Roslinda Muda and Mustafa Mamat. A New Model for Hiding Text in an Image Using Logical Connective. International Journal of Multimedia and Ubiquitous Engineering, Vol.10, No.6 (2015), pp.195-202.

[8]. Madhavi V.Kale, Prof. Swati A.Patil. Text Hiding In Multimedia By Huffman Encoding Algorithm Using Steganography. International Journal of Advance Research in Science Management and Technology, Volume 2, Issue 1, January 2016.

[9]. Amanjot Kaur, Dr. Bikrampal Kaur. Secure The Secret Information In An Image Using K-MM In Steganography. Journal of Multidisciplinary Engineering Science and Technology (JMEST), ISSN: 3159-0040, Vol. 2 Issue 8, August – 2015.

[10]. Mamta Jain, Saroj Kumar Lenka. A Review of Digital Image Steganography using LSB and LSB Array. International Journal of Applied Engineering Research, ISSN 0973-4562 Volume 11, Number 3 (2016) pp 1820-1824.

[11]. Iain E. G. Richardson. H.264 and MPEG-4 Video Compression, The Robert Gordon University, Aberdeen, UK 2003.

[12]. Sukhpreet Kaur, Vanita Rani. Designing an Efficient Image Encryption-Compression System using a New HAAR, SYMLET and COIFLET Wavelet Transform. International Journal of Computer Applications (0975 – 8887) Volume 129 – No.15, November2015.