Image Steganography to Hide Unlimited Secret Text Size

Wa'el Ibrahim A. Almazaydeh

Aqaba College University, Al-Balqa Applied University, Aqaba, Jordan

Summary
This paper shows the hiding process of unlimited secret text size in an image using three methods: the first method is the traditional method in steganography that based on the concealing the binary value of the text using the least significant bits method, the second method is a new method to hide the data in an image based on Exclusive OR process and the third one is a new method for hiding the binary data of the text into an image (that may be grayscale or RGB images) using Exclusive and Huffman Coding.
The new methods shows the hiding process of unlimited text size (data) in an image. Peak Signal to Noise Ratio (PSNR) is applied in the research to simulate the results.

Keywords:
Steganography, Huffman Code, Zigzag Scanning, Symmetric key.

1. Introduction

Today, due to the threats facing the process of transferring data from one place to another, the need to send data in secure ways has arisen. There are two primary process give the transmitted data more security during the tram the transmission process through the networks or where it is stored. The first method is encryption, this method is concerned with changing the state of the data from one form to another unreadable form. When the encrypted data is intercepted, the attacker will not be able to know the original text because it is unreadable. The second method is steganography, steganography hides the secret text data in other data. So, When the attacker intercepts this data, he will not know if it contains secret data or not.

Steganography is the process of hiding media (such as image, text, audio in another media) in another media without any doubts if there is any hiding data inside the cover media.

Steganography is an act of hiding media within other media in order to hide the existence of the secret text data in the cover media [4]. The art of concealment was used in the past, for example, the pharaohs when they used hieroglyphs symbols to express the lifestyle prevailing in that period, and also the Romans used secret ink to write between the lines of the original message.

Steganography is a good option for the security of confidential communications if it is applied correctly.

Manuscript received April 5, 2022
Manuscript revised April 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.4.11

This study is based on developing a Matlab program to simulate the results of the methods.

2. Related Work

2.1 Previous Study

Wa'el Ibrahim A. Al-Mazaydeh presented two method of image steganography to conceal a secret text values inside the image, he presented the traditional process of the steganography that is LSB, and a new method that was LSB+HUFF, the results were tested using the PSNR, the PSNR proved the new method is better than the traditional method [1].

The authors Almazaydeh, and Sheshadri presented three techniques of hiding secret text data inside an image of the image steganography to hide a text in an image, the three techniques are: LSB that was the traditional method, LSB +HUFF and LSB+ARITH. PSNR was used to test the results among the three method and it proved the LSB+HUFF and LSB+ARITH are better than the traditional method [2].

The authors Prof. Sheshadri and Almazaydeh presented two methods of hiding secret text inside an image: the LSB method to hide the secret text, and a new method that was called LSB+KEY method. The results were tested using the PSNR to show that the LSB+KEY is better than the traditional method (LSB). In this research, the authors present the concepts of dynamic symmetric key [3].

Nandhini Subramanian, Omar Elharrouss, Somaya Al-Maadeed, and Ahmed Bouridane reviewed many used method of image steganography in the current time, they showed the datasets, details of evaluation matrices, the Challenges, some gaps and they showed some futuristic visions [13].

Ghazanfar Farooq Siddiqui, Muhammad Zafar Iqbal, Khalid Saleem, Zafar Saeed, Adeel Ahmed, Ibrahim A. Hameed, and Muhammad Fahad Khan presented a new IRD algorithm of image steganography to conceal the variable-sized patient secret data in MRI host images [16].

Yun-Te Lin, Chung-Ming Wang, Wei-Sung Chen, Fang-Pang Lin, and Woei Lin have showed a new algorithm of image steganography to hide data in HDR images that encoded by the OpenEXR format [5].

Dharmesh Mistry, Richa Desai, and Megh Jagad showed that Steganography will not replace the cryptography and the steganography integrates and supports the cryptography. They showed if the message is encrypted before the steganography process for the same message, this will give two layers of protection, thus reduces the probability of detecting the hiding message [6].

NIELS PROVOS AND PETER HONEYMAN showed the basics of the steganography processes, watermarking, and they presented the detecting of the steganography process using statistical steganalysis[7].

Amanjot Kaur, Dr. Bikrampal Kaur presented a new method of steganography of the colored images, the method is depended on k-Modulus. According to the obtained results using the PSNR, they conclude that the steganography using K-Modulus gives better security to the steganography process [9].

Sandeep Panghal, Sachin Kumar, Naveen Kumar showed that the LSB technique is a good technique of the steganography process. Steganography using the LSB togother with the AES method will give a good and more security model of the steganography process [10].

Dr. Saad Abdual azize AL_ani, Bilal Sadeq Obaid Obaid have presented a new technique of image steganography to hide data. The technique depend on b-table that used to compress the size of the characters. To extract the hidden data, the receiver gets the value of location that encrypted by RSA algorithm [11].

Jagan Raj Jayapandiyan, C. Kavitha, and K. Sakthivel presented a new technique of Image steganography called (eLSB) that is an enhancement technique of LSB to hide a secret text inside an image. According to the obtained results using PSNR, MSE and RMSE, They proved that the new technique (eLSB) gives better results than the traditional technique (LSB) [14].

Huaibo Sun, Hong Luo, and Yan Sun presented a method to hide the text in an image based on the NLP and FPE. NLP and FPE is used to encrypt the secret text before the steganography process, They achieved the goal of protecting the security of information from both the appearance and the internal aspects with their technology, because they can encrypt common plaintext text into cipher text that still has a plaintext format [8].

Kiswara Agung Santoso, Ahmad Kamsyakawuni and Abduh Riski presented the hiding process of the text in an image using max-plus algebra. The proposed method give them the possibility to hide number of character as much as the pixels number of the image, this possibility allow to hide more characters in an image more than the LSB method [12].

2.2 Steganography

Steganography process is elaborated in the figure 1, where:

- Secret Text: the desired characters that the steganography process will hide it in the cover media.
- Cover Media: the media that is used in steganography process to hide the data in it, cover media can be (image, video, audio, etc).
- Stego key: the key that is created from the steganography process it is based on the secret text in the cover media.
- Encoding Algorithm: the used method to hide the secret text in the cover media in the steganography process using a stego key.
- Stego Media: the resultant media after hiding the secret text in the cover media in the steganography process.
- Decoding Algorithm: the used method to extract the secret text fron the stego media in the steganography process using a stego key.

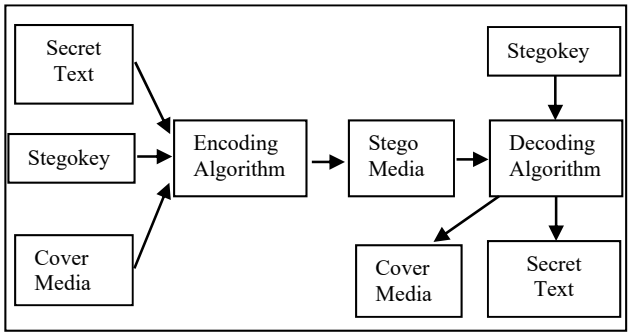

Fig. 1 Steganography process.

2.3 ASCII Code

American Standard Code for Information Interchange (ASCII Code) is a technique to represent the character (A to Z, a to z, 1 to 10 and many special characters like $, +, -, etc.) to decimal number. Each character is represented in the ASCII code by 7 bits. For example, the Ascii Code of the characters (W, w, B, &, +) are (87, 119, 66, 38, 43) respectively. This paper uses the Ascii code to convert the format of the secret text to the corresponding value of the Ascii code for each character.

2.4 Least Significant Bit (LSB)

LSB is the bit closest to the right an holds the lowest value in a block of multi bits in the binary system. For example, the LSB of the block (00110111) is closest bit to the right that is 1, the influence of the LSB to the previous block will be very least; However, the block (00110111) is equal to 55 in the decimal number, while as if the block is changed to (00110110) then the block will equal to 54 in the decimal number, that means the affecting of the LSB on the block is least.

The LSB used in steganography to hide the secret text inside the LSB of the image pixel values. So, the change of the resolution of the pixels will be increase or decrease by 1 value or stay at the same value.

Each pixel in the grayscale image needs one byte to represent the pixel resolution, and in RGB (Red, Green and Blue) images needs three bites to represent its pixel resolution.

For example: To hide the character "W" that has the ASCII code of "1010111", the process needs 7 bytes as shown in the figure 2.

The hiding of the character "W" needs 7 bytes:

Byte 1	**Byte 2**	**Byte 3**	**Byte 4**
1010010**0** 1	1010011**0** 0	1000100**1** 1	1000110**1** 0
1010010**1**	1010011**0**	1000100**1**	1000110**0**

Byte 5	**Byte 6**	**Byte 7**
0111100**1** 1	0110010**1** 1	0110101**0** 1
0111100**1**	0110010**1**	0110101**1**

Fig. 2 Hiding Process of the Character "W".

2.5 Zigzag Scanning

Zigzag scanning is a good way to traverse through the matrices in secure way, it is used in this paper to select the desire pixels to hide the binary secret text date in it. Zigzag scanning (as shown in the figure 3) is used also to increase the security in the steganography process.

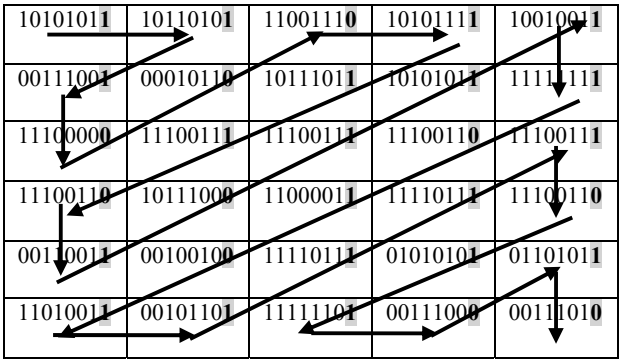

Fig. 3 Zigzag Scanning.

2.6 Huffman Code

Huffman Code is very useful to compress the text to decrease the size of the alphabets. It is used in this paper for two reasons: the first one is decrease the size of the key and the second one is to give the steganography process more security.

Huffman code is based on the frequency of occurrence of the characters in the secret text. For example, to compress the secret text (ABEACADABEA) the following steps must be followed [1]:

- Finding the count of the frequency of each character in the secret text and put it in a list, table 1 shows that.
- Sorting the list according to frequency in the secret text.
- Putting the lowest element frequency in the tree leaves, then creating a parent node to be the sum of the two lower element's frequencies. As in figure 4.
- Removing the two elements from the list and the new parent node is inserted into the list by frequency. So now the list is sorted by frequency.
- Repeat the three previous steps to the all characters in the secret text.
- Assigning the 0 value to the left edge and the 1 value to the right edge.

Table 1: The Frequency and Code of the Text (ABEACADABEA)

Symbol	Frequency	Code	Size
A	5	0	5 * 1 = 5
B	2	100	2 * 3 = 6
C	1	1010	1 * 4 = 4
D	1	1011	1 * 4 = 4
E	2	11	2 * 2 = 4
5*8 =48 bits			18 bits

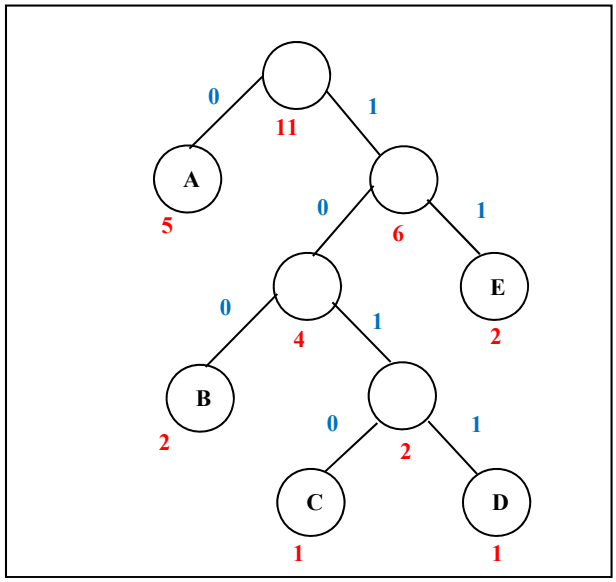

Fig. 4 Huffman tree of the secret text (ABEACADABEA).

According to the previous Huffman Tree, the Huffman code of the secret text (ABEACADABEA) will be 010011010100101101001l0.

2.7 Dynamic Symmetric Key

Symmetric key is the key that is used for Steganography algorithm to hide the secret text characters in the original image (cover media) and the same key is used to extract the secret text characters from the stego media. The symmetric key must be secret and shared key between the sender and the receiver in the Steganography process.

The word dynamic used because the key is not fixed and is changed according to the image data and the secret text data. The Symmetric key can be only logical value (0 or 1), and it is not determined by the sender or the receiver; it creates itself according to the image data and the secret text data. Later, figure 6 shows the process to generate the dynamic symmetric key in this paper.

2.8 Exclusive OR

Exclusive or (XOR) is a logical operator working on two input binary digits (0 or 1) that has the one output if either of its inputs is one, and will be zero output if the two inputs are 0 or 1. Table 2 shows the truth table of (XOR).

Table 2: The truth table of the XOR logical operator

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

2.9 PSNR

Peak Signal to Noise Ratio is a good method to measure the amount between the original image and the stego image, (equation 1) is measured on a logarithmic scale. It depends on the mean squared error (MSE) between an original image and stego image, relative to (2n - 1)2 (the square of the highest-possible signal value in the image, where n is the number of bits of the image sample) [15].

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

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

In the equation 1, A is a matrix of the values of the original image (cover media), B is a matrix values of the image after the steganography process (stego image), m is the number of rows of the original or the stego image and n is the number of columns in the original or the stego image, "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" [15].

If the result of the PSNR is high then the change of the resolution of an image is low, and if the PSNR value is low then the change of the resolution of an image will be high.

3. Methodology

This paper presents three methods to hide text in an image: the first one is to hide a secret text in an image using the traditional method that is Least Significant Bit (this

steganography process is called LSB), the second one is to hide a secret text in an image based on a dynamic symmetric key (this steganography process is called LSB+XOR) and the third on to hide a secret text in an image based on a dynamic symmetric key by using Huffman code (this steganography process is called LSB+HOR+HUFF). The results of the methods have been computed and compared using (PSNR).

Figure 5 shows the base methodology of the three algorithms for encoding and decoding process.

In encoding process (sender), the original image (grayscale or colored image) is converted to the column vector of pixels values using zigzag scanning, the column vector is converted to binary data using ASCII code. In parallel procedure, the secret text is converted to binary data (digits) using ASCII code, after that, the binary data of the secret text is put in the LSB bits of the binary data of the image.

Then, by applying the inverse zigzag scanning the stego image will be created, and it will be ready for sending to the receiver.

When the receiver get the stego image the decoding process will start, the stego image is converted to the column vector of pixels value using zigzag scanning, the column vector is converted to binary date using ASCII code. The secret text is extracted from the LSB bits of the binary data according to each of the three steganography processes.

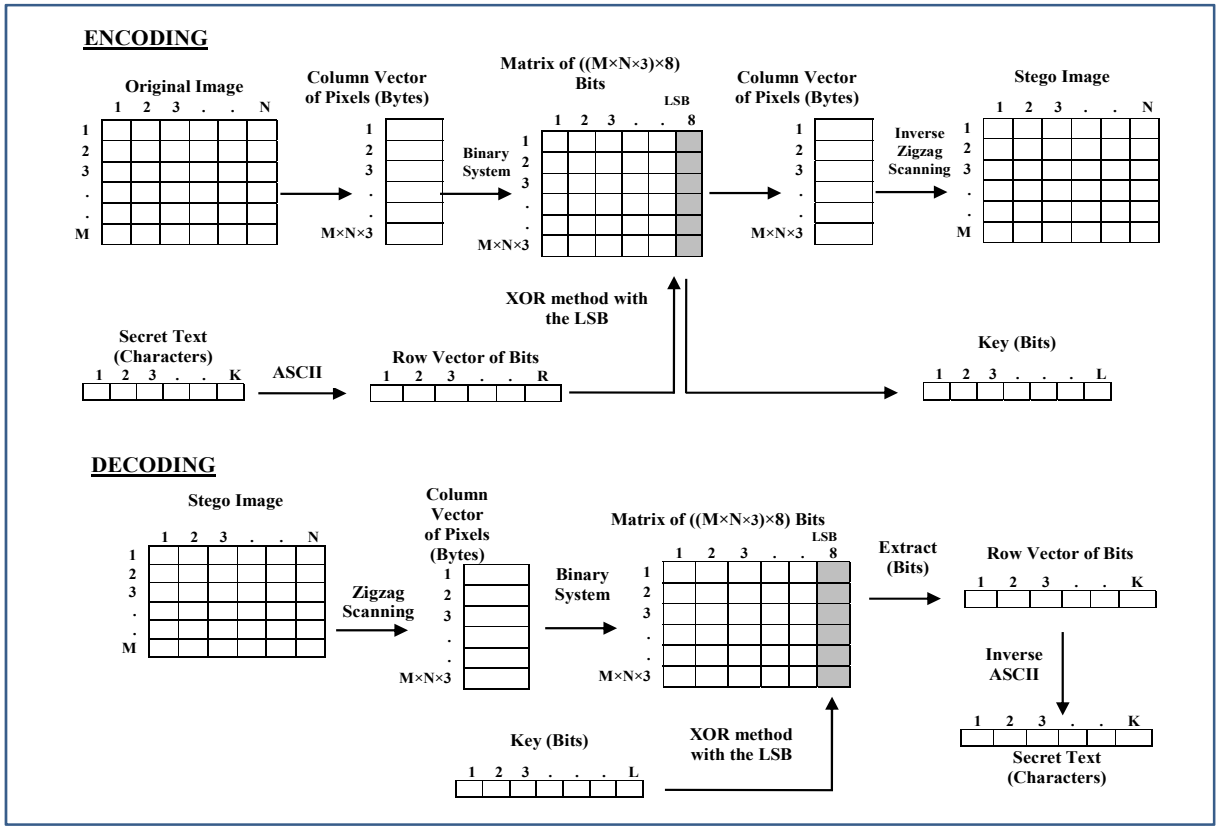

Fig.5 Encoding and decoding process of the steganography process of this research.

3.1 Image Steganography using LSB

This is the traditional technique for hiding process, it is based on hiding the secret text data in the LSB of the image; where the first 7 bits of the LSB is the steganography type and the 20 bits after the first seven bits is the length of the bits that is to be hidden in the image.

The size of secret text that can be hidden in the original image using this method:

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

Where S1 is the maximum size of the secret message bits that allowed to be hidden, M refers to the number of rows in the original image, N refers to the number of columns in the original image and "27" is: the first 7 bits from 1 to 7 are reserved bits to the Steganography process type that may be [1, 2, 3, …, 127]. For example, when the Steganography type equals 1 that means Steganography process is LSB, when the Steganography type equals 2 that means Steganography process is another Steganography process and etc., the bits from 8 to 27 refers to the length of the actual secret text.

3.2 Image Steganography using LSB+XOR

This method is working as shown in figure 5, where the encoding process (in the sender) is working as following:

- Converting the original image to binary data for each pixels in manner of zigzag scanning of the matrix values of the original image.
- Converting the secret text characters to vector of binary data using the Ascii code.
- Applying the XOR logical operator between the LSB of the binary image data and the binary data of the secret text as in figure 6.
- The result of the XOR in the previous step will get a vector of bits (this is called the stego key). This stego key is symmetric and shared between the sender and the receiver whose do not know the actual value of the key. This key is called dynamic symmetric key, it is called dynamic because it based on the LSB of the image values and the secret text values, and called symmetric because the same key(shared) is used in the sender and the receiver.
- The stego image and the stego key will send to the receiver

The decoding process (in the receiver) is working as following:

- Converting the original image to binary data for each pixels in manner of zigzag scanning of the matrix values of the original image.
- Applying the XOR process between the stego image and the stego key as in figure 6.
- The result of the XOR will give the secret text binary data
- Using Inverse Ascii code to convert the binary data of the secret text the original secret text characters.

The size of the secret text bits that can be hidden using this technique equals:

$$S2 = Infinity \quad (4)$$

Where S2 is the size of the secret text, and Infinity because if the process is arrived to the last bit in the image and the text is not complete the hiding process then it can go back from the first pixel of the image and continue by the same method.

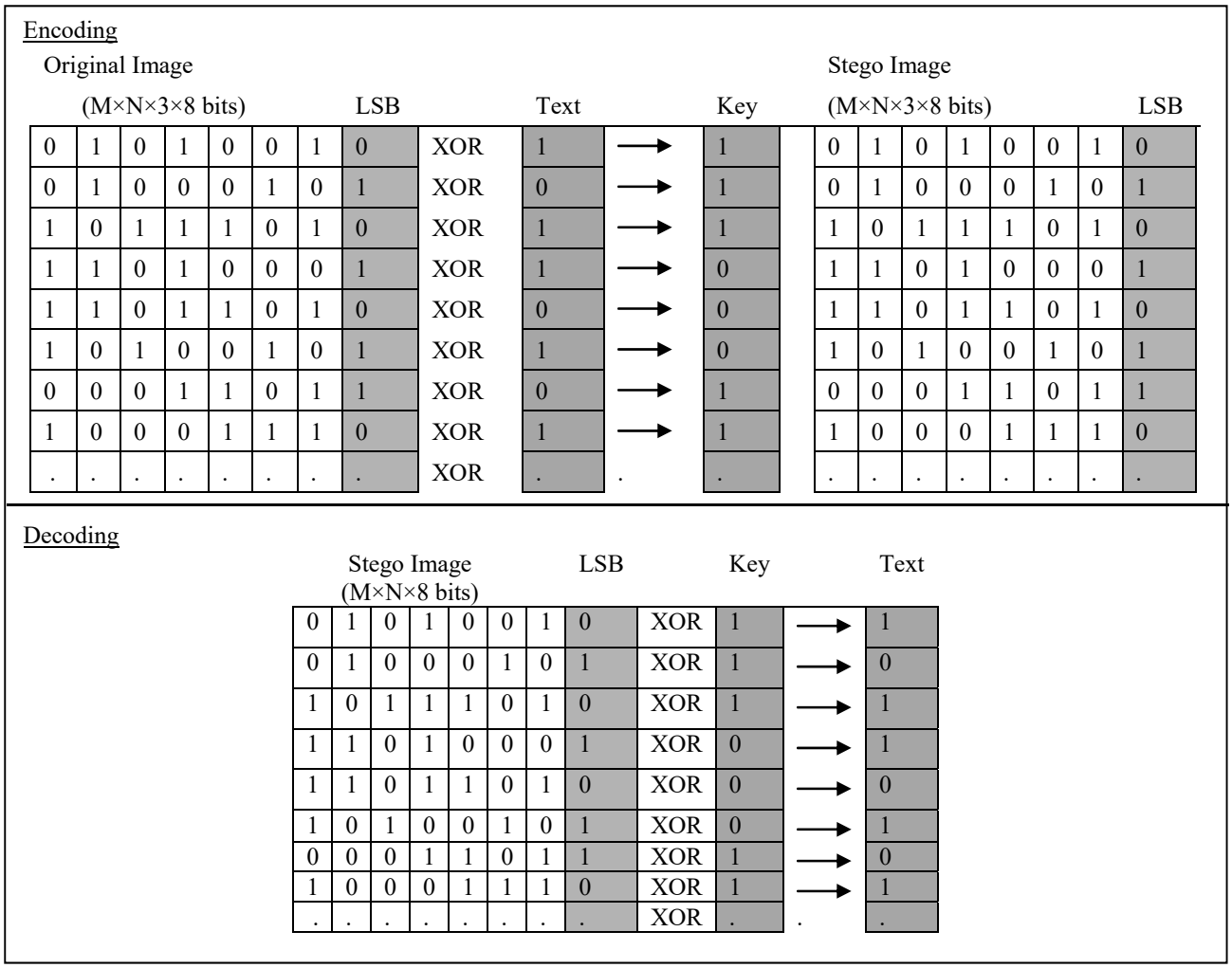

Fig. 6 The encoding and decoding process of the LSB+XOR Method.

3.3 Image Steganography using LSB+XOR+HUFF

This method follows the same structure of the LSB+XOR method, but this method applies the Huffman code on the secret text; to reduce the size of the stegokey and to add more security to the steganography process.

This method adds 40 bits extra to the stegokey in the start of the stegokey that are necessary to for encoding and decoding of the Huffman code process.

4. Experimental Results

A Matlab program (called Wa'el Steganography) has been developed to implement the algorithms; it is illustrated in the figure 9 and 11 for the encoding and decoding process alternatively. The images that have been applied for implementation phase are colored images (RGB images) of the type (PNG, JPEG and PMB).

4.1 The Implementation

Aqaba city image has been implemented of size (980×655×3) pixels with type is (JPEG) that shown in figure 7. The secret text that has been applied is illustrated in the figure 8.

The count number of characters of the secret text equal to (2107) characters and the number of bits equals to (14749) bits.

Fig. 7 Aqaba City Image.

4.2 Encoding

The encoding process of this paper includes the following steps:

- Select an image to hide a secret text in it (here, the image is Aqaba image as shown in the figure 7).
- Select a secret text (for example as shown in figure 8).
- Select the steganography method that is either LSB, LSB+XOR or XOR+HUFF.
- Then, the steganography method will start its job to concealing the secret text in the image.

After complete the steganography method, the stego media will appear in the right hand of the program that is must be saved by the user, if a steganography method created a key the program will order to save the stego key, the stego key of the (LSB+XOR) after this process is appeared as shown in the figure 10.

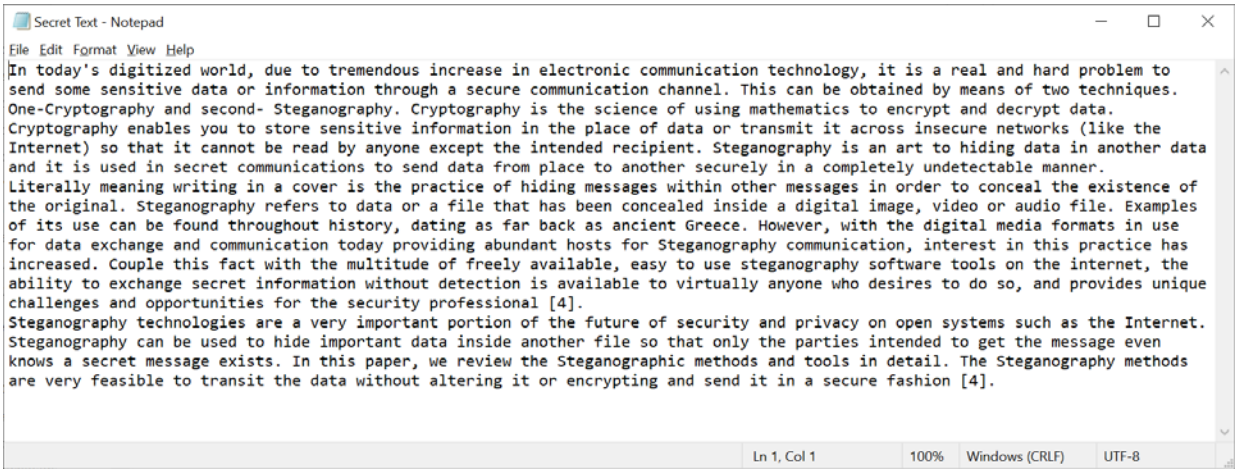

Fig. 8 The secret text.

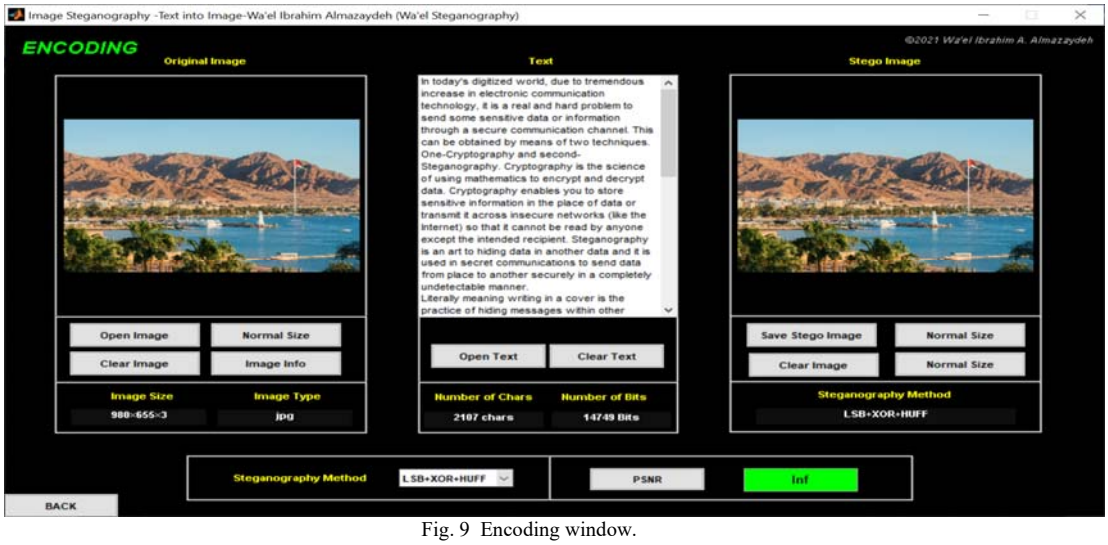

Fig. 9 Encoding window.

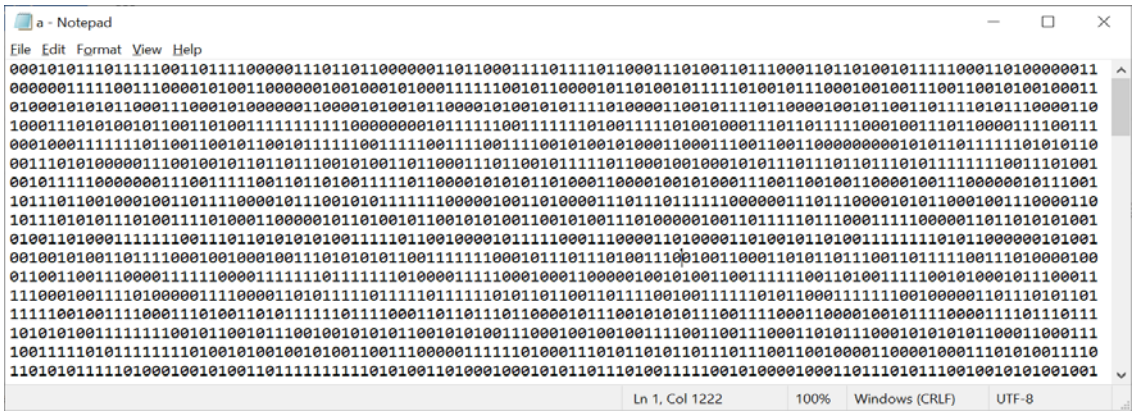

Fig. 10 The tego key.

4.3 Decoding

The decoding process (as shown in the figure 11) follows the following steps:

- Select the stego image.
- The decoding process will start, and if the steganography method needs a stego key the program will ask to select the stego key.
- After that, the secret text will be extracted, the user can save the secret text.

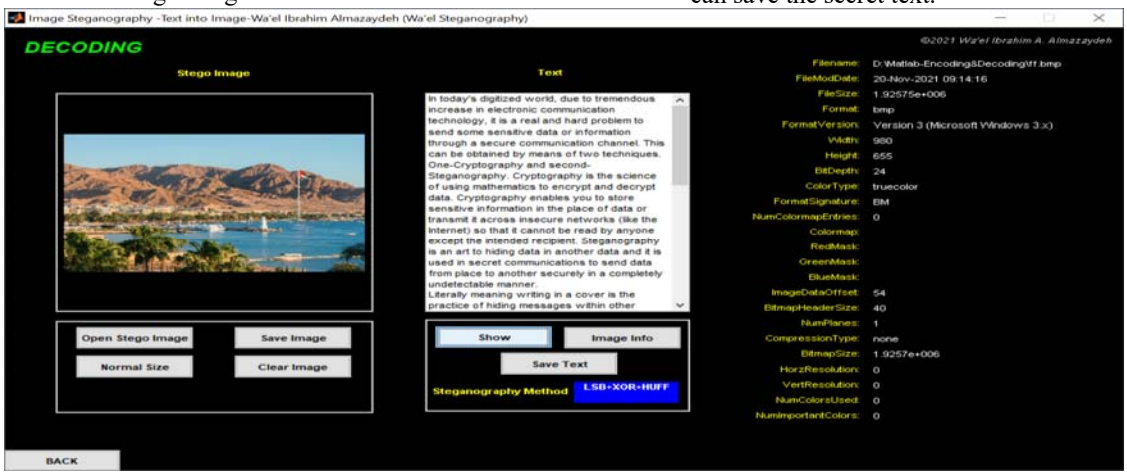

Fig. 11 decoding windo

4.3 The Results

After implementation of the steganography process of the three previous algorithms using the Aqaba city image and the secret text, Table 3 shows the results using Peak Signal to Noise Ratio (PSNR).

Table 3: The results of applying the three algorithms of the Aqaba city image and the secret text.

The Steganography Algorithm	PSNR	Length of the Secret Text (bits)	The Maximum Text Size to be Hidden (bits)	Length of the Secret Key (bits)
LSB	72.2791	14,749	1,925,673	No secret Key
LSB+XOR	Infinity	14,749	Infinity	103,243
LSB+XOR+HUFF	Infinity	14,749	Infinity	68,061

4. Conclusion

This paper presents three methods of image steganography to hide a secret text in an image: LSB, LSB+XOR and LSB+XOR+HUFF. The results that have been gotten show that the two Algorithms (LSB+XOR and LSB+XOR+HUFF) are better that the (LSB) Algorithm, and the results indicate that the (LSB+XOR) and (LSB+XOR+HUFF) have the Infinity PSNR, that means no any bit change of the original image or the stego image, also the maximum text size data that allowed to be hidden is infinity, because no limit size of the secret text, but the (LSB+XOR+HUFF) is better than the (LSB+XOR) in term of the length of the secret key, so the size of the secret key using (LSB+XOR+HUFF) is less than the size of the key of the (LSB+XOR), this is due to the use of the Huffman Code.

References

[1] Wa'el Ibrahim A. Al-Mazaydeh. "Image Steganography using LSB and LSB+Huffman Code". International Journal of Computer Applications. ISSN: (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. ISSN: (0975 – 8887), Volume 155 – No 11, December 2016.

[3] Wa'el Ibrahim A. Almazaydeh and Prof. H. S. Sheshadri. "Image Steganography using a Dynamic Symmetric Key". In 2nd International Conference on Trend in Electronics and Informatics (ICOEI 2018). IEEE Conference Record: #42666, IEEE DVD ISBN: 978-1-5386-3569-8.

[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] Yun-Te Lin, Chung-Ming Wang, Wei-Sung Chen, Fang-Pang Lin, and Woei Lin. "A Novel Data Hiding Algorithm for High Dynamic Range Images". IEEE TRANSACTIONS ON MULTIMEDIA. VOL. 19, NO. 1, JANUARY 2017.

[6] Dharmesh Mistry, Richa Desai, and Megh Jagad. "Hidden Data Transmission using Image Steganography". International Journal of Computer Applications (0975 – 8887). Volume 130 – No.14, November 2015.

[7] NELS PROVOS AND PETER HONEYMAN. "Hide and Seek: An Introduction to Steganography". IEEE Computer Society. Volume: 99, Issue: 3, May-June 2003.

[8] Huaibo Sun , Hong Luo , and Yan Sun. "Data Hiding for Ensuring the Quality of the Host Image and the Security of the Message". IEEE Access, vol. 7, 2019, Digital Object Identifier 10.1109/ACCESS.2019.2907530.

[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] Sandeep Panghal, Sachin Kumar, Naveen Kumar. "Enhanced Security of Data using Image Steganography and AES Encryption Technique". International Journal of Computer Applications. (0975 – 8887) Recent Trends in Future Prospective in Engineering & Management Technology 2016.

[11] 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.

[12] Kiswara Agung Santoso, Ahmad Kamsyakawuni and Abduh Riski. "Hiding The Text Into An Image By Max-Plus Algebra". Proc. ICOMITEE 2019, October 16th-17th 2019, Jember, Indonesia.

[13] Nandhini Subramanian, Omar Elharrouss, Somaya Al-Maadeed, and Ahmed Bouridane. "Image Steganography: A Review of the Recent Advances". IEEE Access, vol. 9, 2021, Digital Object Identifier 10.1109/ACCESS.2021.3053998.

[14] JAGAN RAJ JAYAPANDIYAN, C. KAVITHA, AND K. SAKTHIVEL. "Enhanced Least Significant Bit Replacement Algorithm in Spatial Domain of Steganography Using Character Sequence Optimization". IEEE Access, Digital Object Identifier 10.1109/ACCESS.2020.3009234

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

[16] Ghazanfar Farooq Siddiqui, Muhammad Zafar Iqbal, Khalid Saleem, Zafar Saeed, Adeel Ahmed, Ibrahim A. Hameed, and Muhammad Fahad Khan. "A Dynamic Three-Bit Image Steganography Algorithm for Medical and e-Healthcare Systems". IEEE Access. DOI: 10.1109/ACCESS.2020.3028315.