RESEARCH ARTICLE OPEN ACCESS

Enhancement the Efficiency of Data Hiding Using Data Compression and Dividing Data

Ola Haydar [1], Kinda Aboukassem [2]
M.Sc., Department of Computer and Control Engineering
University of Tishreen
Syria

ABSTRACT

Steganography is the art and science of invisible communication by hiding secret information into other sources of information like text, image, audio, video etc. There are a lot of steganography techniques proposed to hide data like LSB, DCT, pixel-value differencing etc. This paper improves information security through providing three levels of security: compression, encryption and steganography. The secret image is compressed by Deflate then by Huffman(Deflate Based Huffman. Then information header is encrypted by XOR gate and then both of the secret image and the header are embedded into 2 cover images using more than one hidden algorithm. So there is an extra safe. thus, reduces the chance of the hidden message being detected. MSE (Mean Square Error) and PSNR (Peak Signal to Noise Ratio) are two common quality measurements to measure the difference between the cover-image and the stego-image. Results showed that the proposed method gives better results than other algorithms with PSNR about 61% and MSE about 0.05 .

***Keywords* :—** Deflate, Huffman, LSB, MSE, PSNR.

I. INTRODUCTION

Steganography is the art and science of hiding the one information into other sources of information like text, video, audio, image etc. so that it is not visible to unintended users. It is derived from Greek words Steganos (covered or secret) and Graphie (writing) literally means "covered writing". Its ancient origins can be traced back to 440 Be. Since Roman era, usually carried out by military to send secret messages. Messages sent by tattooing it on the slave's scalp that previously shaved, after the hair grows, the slaves then sent to allies. To read the messages, the allies shaved the slave's head. Today, the media carried steganography include (image, video, audio, and text).

At first we should note the basic difference between encryption and Steganography. Encryption Converts understandable data into obscure data, thus hiding the secret data content, while Steganography hides secret data (without change) within other data , thus hiding the existence of secret data.

To improve the security of information system, Steganography and encryption can be combined to form a stronger algorithm.

Steganography algorithms can be divided into two main categories: spatial domain algorithms and transform or frequency algorithms. In the spatial domain, the stego image is obtained by replacing the bits of the secret message directly with the bits of the cover image like LSB technique. LSB is one of the most common algorithms in image steganography witch belong to spatial domain where the hidden message is embedded into cover media by replacing bits with the least significant bits of the related media. On the other hand, in frequency domain, using some mathematical functions like Discrete Cosine Transform and Discrete Wavelet Transform, hidden bits are inserted into the coefficients calculated from the pixel values of the cover image[10].

II. PROPOSED METHOD AND DESIGN

The proposed image steganography method is composed of embedding stage and extraction stage. In the embedding stage, the secret image is compressed and encoded and then resultant stream is embedded into two cover images.

In the extraction stage the secret image will be comprehend within the stego image.

A. Embedding Stage

RGB image (secret image) will be hidden in two RGB images (2 cover images), starting hiding at Hidden Key which is a position that sender choose it at the begin of the programme.

Fig. 1 depicts the proposed method's framework and process flow diagram for embedding stage.

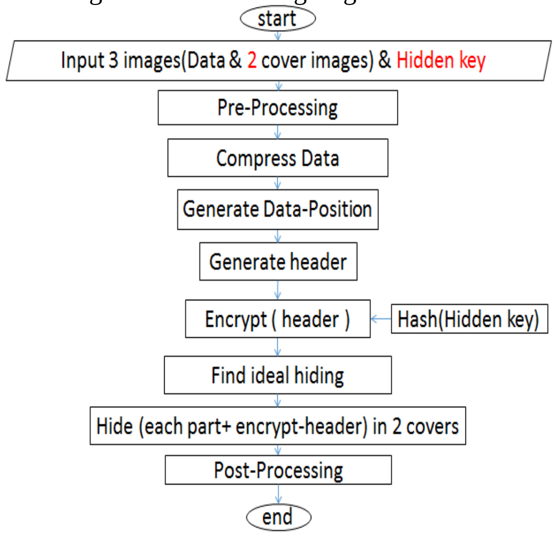

Fig. 1 flow diagram for proposed embedding stage

*1) **Pre-processing:*** The secret image and the two cover images are converted into stream of bits. To ensure that the inserted hidden key is convenience with the size of the two cover images, modulo operation is occur between the original hidden key and the size of each cover image, the output is: hidden key1 &hidden key2.

hidden key1=mod(hidden key, size of cover1)

hidden key2=mod(hidden key, size of cover2)

*2) **Compress Secret Image :*** At first the secret image is compressed by Deflate and the output of Deflate is compressed by Huffman. Deflate is two-stage lossless data compression algorithm that uses the combination of LZ77 and Huffman coding. This will take advantage of both the algorithms. It is a popular compression method that was originally used in the well- known Zip and Gzip software and has since been adopted by many applications[6]. The output of this phase is compressed data(compressed secret image and compressed header information).

*3) **Generate Data-Position:*** The secret data is separated into two parts, a part contains bits with odd indexes and the other part contains bits with even indexes(odd part and even part). In general, the odd part will be hidden in the first cover image and the even part will be hidden in the second cover image. We will form a new array (Data-Position). Bits of odd part are compared with bits of the first cover image(excepted LSB bit), if the comparison result is equal then the value '1' is stored in odd positions of Data-Position and if comparison result is different then the value '0' is stored. this process occurs in conjunction with comparison the bits of even part with bits of the second cover image "at the same way". In this way we have the secret data in a different form (Data-Position). The process of generation is shown in Fig. 2.

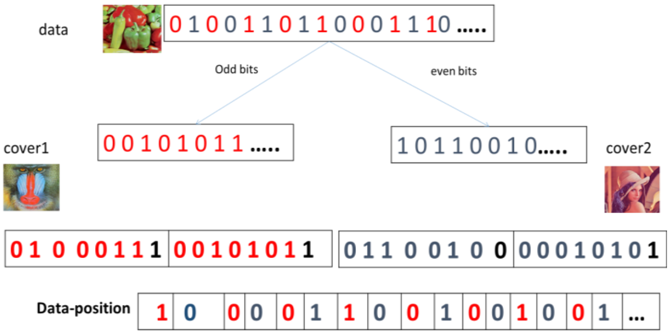

Fig. 2 Generation of Data-Position

*4) **Generate Header:*** The header will be generated, it contains the important information that make the extraction process possible. It contains:

- Header E ={kind bit, order bit, length(Header E1), Header E1 }
- Header O={kind bit, order bit, length(Header O1), Header O1 }

Where:

- kind bit: determines that the cover image contains the odd bits or the even bits.
- order bit: determines that the stego-image is the first image or the second.
- Header E1={height bits, depth bits, even bits, odd bit, Huffman even bits}
- Header O1={ Huffman odd bits}

*5) **Hash Function:*** Hash function (MD5) is applied on the hidden key

K1=Hash(hidden key1) & k2=Hash(hidden key2).

*6) **Encryption Header***: To increase the security of system, header E and header O are encrypted by XOR gate with k1, k2 hidden key as encryption keys, so encryption the header(header e1, header o1) only is more efficiently because it takes less time than encryption the whole secret data.

*7) **Find Ideal Hiding:*** There are two probabilities for hiding; the odd part will be hidden in the first cover image and the even part will be hidden in the second cover image, MSE and PSNR are calculated for each cover image. Then the opposite process, the even part will be hidden in the first cover image and the odd part will be hidden in the second cover image, MSE and PSNR are calculated for each cover image. At last we choose the convenient way which causes the least MSE, therefore the greatest PSNR. The process is shown in Fig. 3.

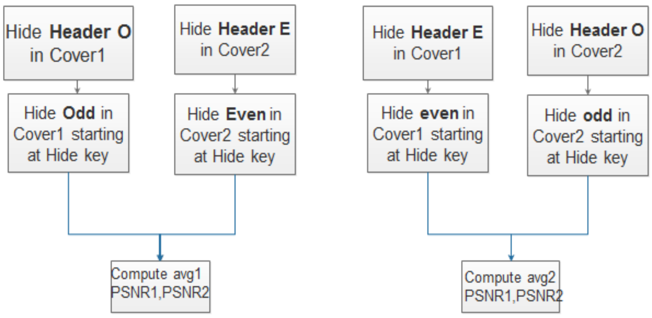
Hide Header O in Cover1
Hide Header E in Cover2
Hide Header E in Cover1
Hide Header O in Cover2
Hide Odd in Cover1 starting at Hide key
Hide Even in Cover2 starting at Hide key
Hide even in Cover1 starting at Hide key
Hide odd in Cover2 starting at Hide key
Compute avg1 PSNR1,PSNR2
Compute avg2 PSNR1,PSNR2

Fig. 3 Find ideal hiding

*8) **Hiding Data:*** after finding the ideal algorithm for hiding, secret data will be hidden using LSB algorithm. Hiding is shown in Fig. 4.

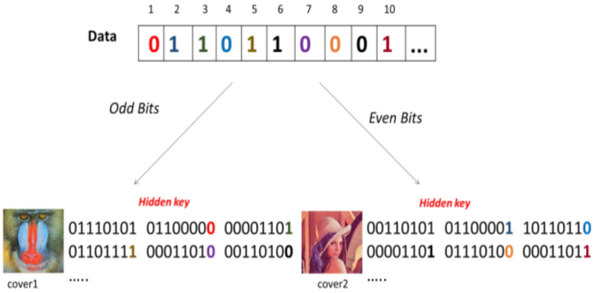

Fig. 4 *Hiding data*

B. *Extraction stage*

The extraction stage is the reverse of the embedding stage as shown in Fig. 5.

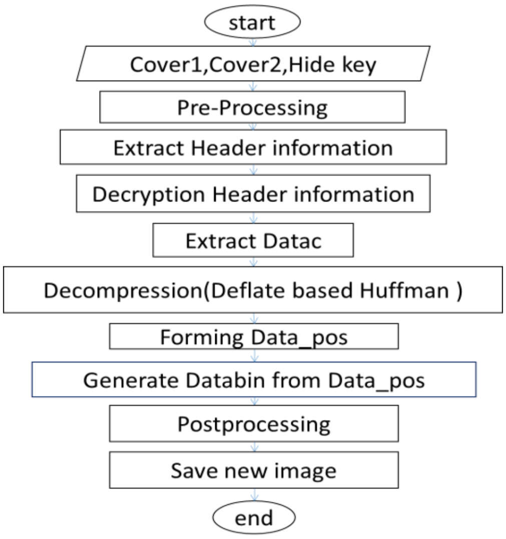
start
Cover1,Cover2,Hide key
Pre-Processing
Extract Header information
Decryption Header information
Extract Datac
Decompression(Deflate based Huffman)
Forming Data_pos
Generate Databin from Data_pos
Postprocessing
Save new image
end

Fig. 5 flow diagram for proposed extraction stage

III. EXPERIMENTAL RESULTS AND ANALYSIS

Some experiments are carried out to prove the efficiency of the proposed method where simulation is done on Matlab, set of RGB image of size 512 × 512 is used as the cover image to hide an image of size 158 × 158 to form the stego-image. With the experimental study, we noticed that the visual differences between the original cover images and stego image with other techniques is hardly detected with naked eyes, as shown in Fig. 6 and Fig. 7. Furthermore, from the comparison of Fig. 8 and Fig. 9, we can see that the histogram of them are basically the same.

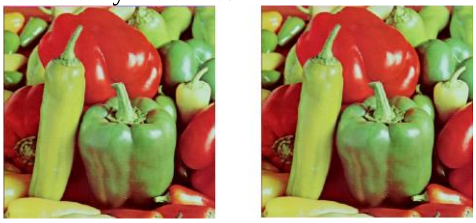

Fig. 6 the cover1 image after embedding with the original cover1

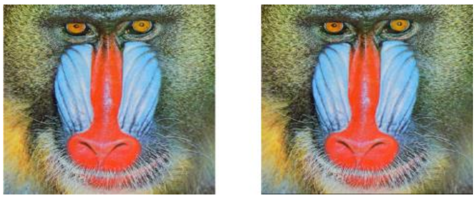

Fig. 7 the cover2 image after embedding with the original cover2

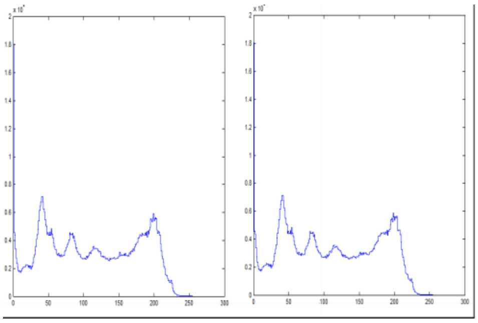

Fig. 8 the cover1 histogram after embedding with the original cover1 histogram

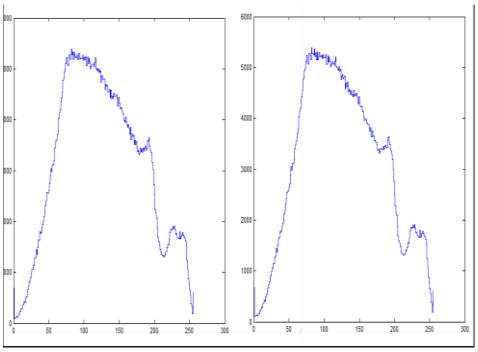

Fig. 9 the cover2 histogram after embedding with the original cover2 histogram

MSE (Mean Square Error) and PSNR (Peak Signal to Noise Ratio) are two common quality measurements to measure the difference between the cover image and the stego-image.

MSE is the averaged pixel-by-pixel squared difference between the cover-image and the stego-image.

$$\text{MSE}= \frac{1}{M.N}\sum_{i=1}^{M}\sum_{j=1}^{N}[C(i,j) - S(i,j)]^{2}$$

where, M and N are the rows and columns of the cover image respectively, and C(i, j) and S(i, j) means the pixel value at position (i, j) in the cover-image and the corresponding stego-image, respectively.

The PSNR is expressed in dB's and can be calculated using MSE as:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}$$

Where, P is the peak signal value of the cover- image, and P=max (C(i, j),S(i, j)).

We have calculated MSE and PSNR and compare it with the simple LSB algorithm and random LSB[2].

Following Table 1 and Table 2 present the comparison of these three techniques with respect to MSE and PSNR respectively[2]. Fig. 10 and Fig. 11 present the above comparison of techniques graphically using a bar diagram with respect to MSE and PSNR respectively[2].

TABLE I

COMPARISON OF MSE VALUES IN COVER IMAGE AND STEGO IMAGE

Cover Image	Secret Image	Simple LSB	Random LSB	Proposed Work
512*512	Camera-man	MSE	MSE	MSE
Lena	158*158	0.3807	0.3806	0.0509
Baboon	158*158	0.3804	0.3800	0.0506
Pepper	158*158	0.3808	0.3800	0.0508

TABLE III

COMPARISON OF PSNR VALUES IN COVER IMAGE AND STEGO IMAGE

Cover Image	Secret Image	Simple LSB	Random LSB	Proposed Work
512*512	Camera-man	PSNR	PSNR	PSNR
Lena	158*158	51.9777	51.9788	61.054
Baboon	158*158	51.4320	51.3986	61.082
Pepper	158*158	51.3902	51.3986	61.066

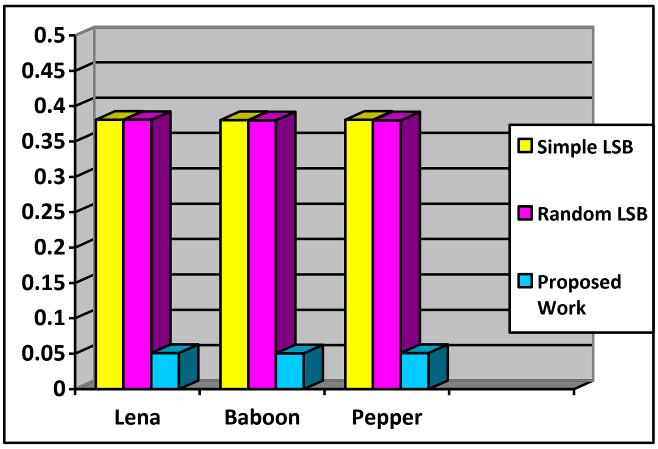

Fig. 10 Comparison of MSE values of Cover Image and Stego Image

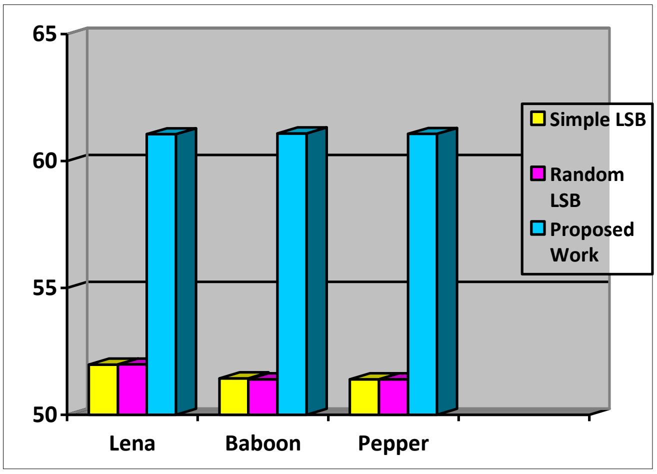

Fig. 11 Comparison of PSNR values of Cover Image and Stego Image

Table 3 and Table 4 present the values of MSE and PSNR when the size of embedded image increases. As can be seen in Fig. 12, the reduction in PSNR is very slight as compared with the increases in the size of embedded image and this suggests that the quality of the cover images remains almost constant when the size of embedded image increases. Therefore The proposed method have been shown a good performance.

TABLE IIIII
MSE VALUES OF STEGO IMAGE WITH INCREASING PAYLOAD

Amount of Data Embedded or Payload	Deflate based Huffman
	MSE
Camera man 64*64	0.00944
Camera man 128*128	0.03514
Camera man 158*158	0.0506

TABLE IVV
PSNR VALUES OF STEGO IMAGE WITH INCREASING PAYLOAD

Amount of Data Embedded or Payload	Deflate based Huffman PSNR
Camera man 64*64	68.3731
Camera man 128*128	62.6719
Camera man 158*158	61.0826

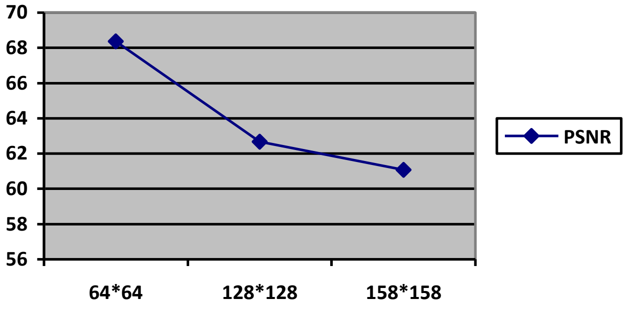

Fig. 12 PSNR values of Stego Image with increasing Payload

IV. CONCLUSION

In this paper, we try to develop a steganography scheme aim to hide the secret image into two cover images. Our system provides three levels of security: compression, encryption and steganography, and join hashing to prevent the forgery of hidden information. The resultant images appear without any noticeable degradation and results show that the proposed method gives better results than other algorithms and provides higher PSNR and lower MSE.

For future work, for improving this method, we must continue to study the related analysis of compression algorithms.

REFERENCES

[1] Amit JainKamaljitI Lakhtaria, "COMPARATIVE STUDY OF DICTIONARY BASED COMPRESSION ALGORITHMS ON TEXT DATA", International Journal of Computer Engineering and Applications. Issue II，May 14，Vol. VI.

[2] Rupali Bhardwaj and Vaishali Sharma, "Image Steganography Based on Complemented Message and Inverted bit LSB Substitution", Procedia Computer Science, Vol.93, 2016

[3] Dalvir KaurKamaljeetKaur, "Analysis of Lossless Data Compression Techniques", International Journal of Computational Engineering Research. Issue 4，2013，Vol. 03.

[4] GOELRANA,KAUR,M "A Review of Comparison Techniques of Image Steganography". IOSR Journal of Electrical and Electronics Engineering. Issue 1，2013，Vol. 6.

[5] Mohammed Al-lahamIbrahiemM. M. El Emary, "Comparative Study between Various Algorithms of Data Compression Techniques", IJCSNS International Journal of Computer Science and Network Security. No.4，April 2007，Vol. 7.

[6] 3hyrki AlakuijalaEvgeniiKliuchnikov, Zoltan Szabadka, and Lode Vandevenne "Comparison of Brotli, Deflate, Zopfli, LZMA, LZHAM and Bzip2 Compression Algorithms". Google Inc. 2015

[7] Pratiksha SethiV.Kapoor"A Proposed Novel Architecture for Information Hiding in Image Steganography by using Genetic Algorithm and Cryptography". International Conference on Computational Science. Issue 5，2017，Vol. 5.

[8] Prithwish DasSupriyoRay and Atanu Das, "An Efficient Embedding Technique in Image Steganography Using Lucas Sequence", Modern Education and Computer Sience MECS. No.09，8 September 2017，Vol. 09.

[9] Gaurav , "A New Method for Image Steganography Using LSB and MSB", International Journal of Recent Research Aspects, Vol. 2, Issue 4, 2015.

[10] SinghAmritpalSingh and Harpal, "An Improved LSB based Image Steganography Technique for RGB Images, IEEE,2015.

[11] Sushil Sharma and Ishpreet Singh Virk, "Image Steganography using Two's Complement", International Journal of Computer Applications, Vol.145, No.10, 2016

[12] Biswajita Datta, Upasana Mukherjee and Samir Kumar Bandyopadhyay, "LSB Layer Independent Robust Steganography using Binary Addition", Procedia Computer Science, Vol.85, 2016

[13] Orooba Ismaeel Ibraheem Al-Farraji, "NEW TECHNIQUE OF STEGANOGRAPHY BASED ON LOCATIONS OF LSB", International Journal of Information Research and Review, Vol. 04, Issue 1, 2017.

[14] Prithwish Das, Supriyo Ray and Atanu Das, "An Efficient Embedding Technique in Image Steganography Using Lucas Sequence", Vol.09, No.09, 2017.

[15] Bhoomika Parmar, Rakesh Kumar, "High PSNR Based Image Steganography", International Journal on Recent and Innovation Trends in Computing and Communication, Vol.5, Issue 10, 2017.

[16] Ashwini W, Nagraj Kyasa, "The Improved Image Steganography with Encryption Method and to Overcome the Compression Technique", International Journal of Innovative Research in Computer and Communication Engineering Vol. 5, Issue 5, 2017.

[17] Pratiksha Sethi, V. Kapoor, "A Proposed Novel Architecture for Information Hiding in Image Steganography by using Genetic Algorithm and Cryptography", International Conference on Computational Science, 2016.