Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 8, No. 3, August 2017), Pages: 105-122
www.jacr.iausari.ac.ir

Presenting an Imperceptible Steganographic Algorithm through Genetic Algorithm and Mix Column Transform

Mahsa Amini Kaleibar, Saeid Taghavi Afshord✉
Department of Computer Engineering, Shabestar Branch, Islamic Azad University, Shabestar, Iran
mahsaamini@iaushab.ac.ir; taghavi@iaushab.ac.ir

Received: 2016/08/31; Accepted: 2017/01/12

Abstract

Due to growth and development of data communications, the need for fast and secure transmission of information is very important. In order to address this problem, especially over the internet, some of the security systems such as cryptography and steganography can be used. Steganography is a way for secure and confidential communication. In this paper, an algorithm for color image steganography through mix column transform and the genetic algorithm is presented, which is a distinct type of the transform. The proposed method divides the image into blocks and the mix column transform of each block is obtained. Then, the genetic algorithm is applied to determine the best permutation for inserting the secret message. By using the genetic Algorithm, bits of the message are embedded in the least significant bits of the image. Experimental results show that, not only the visual quality of the stego image is improved, but also the embedding time of the image and the capacity are increased.

Keywords: *Steganography, Mix Column Transform, Irreducible Polynomial Mathematics, Cryptography, Genetic Algorithm*

1. Introduction

Nowadays, the need for fast and secure communication of data is absolutely essential. Steganography techniques are used for secure data transmission. Among them, unlike other security approaches, the steganography techniques ignore the existence of hidden data [1]. The first recorded uses of steganography in fifth century were described by Herodotus, a Greek historian [2]. Steganography is originally a Greek word, which derives from the words steganos, meaning "covered concealed, or protected" and graptos, meaning "writing" (covered writing) [3]. Steganography is a subset of information hiding science, and one of the important point about it, is the invisibility of the information within the image [4].

It is noteworthy to mention that the images are actually the multidimensional array, and the numerical value of each element is a number between 0 up to 255, which depicts the color of each pixel within the image [5]. The cover agent can be used in steganography for hiding data such as digital images, video files, audio files, text files and etc. Among the cover media, digital images are considered as a kind of suitable cover media in steganography, due to their frequent use of the Internet, limited human visual perception of the changes, and tolerance of many changes. Choosing the image format has a great

impact on the steganography system [6]. Compressed images and JPEG as general formats have been extensively utilized.

Three factors are usually investigated in steganography. They are preserved according to the applications, presuppositions and the other conditions as needed. These factors are imperceptibility, capacity and security. The combination of the cryptography and steganography provides a higher level of security [7]. In this paper, an association of these two methods are used in which cryptography is based on the mix column transform and steganography is based on the genetic algorithm.

The steganographic techniques are divided into two categories. They are spatial domain and frequency domain [2]. In the first method, the secret message is hidden in the least significant bits of each image, which is known as the Least Significant Bit (LSB) method. In the second method, the image is converted into the other domain, and then the hidden message is embedded within it. The most of works on steganography, that have been done by the spatial domain techniques, have shown low embedding capacity or have created distortion after embedding, or sometimes they have had low security. The proposed method is one of the frequency domain methods and the inserted message in the image is a text. This is the main difference of the proposed method in comparison with the other methods. Some of the steganography techniques are introduced in the following parts.

Several studies have been performed on the image steganography. In RGB (Red, Green, and Blue) images, each pixel is represented by 3 bytes indicating the intensities of red, green, and blue channels in that pixel. Gutub et al. present a technique based on RGB intensity values of the pixel, where one of the channels is considered as indicator channel. They used one or both of the remaining two channels to conceal data bits. The last two bits of the indicator channel are telling whether the data bits are hidden in the other two channels or not [8]. Plenty of space has been lost in this method due to the index storage, and the image quality will be evidently affected because of recording of these data. The main idea of utilizing the mix column transform in steganography traces back to an article published in 2013 [9]. It represents a reversible steganography method, which is based on dividing the image into blocks with various sizes. Then, some of blocks are selected for embedding the secret message regarding to the secure key. The proposed transform is applied on certain blocks through taking out three least significant bits of each value and by hiding the hidden message within matrix (block). The reported experimental results have been proved the ability of this method in balancing three critical properties: capacity, security and imperceptibility [9]. However, the capacity is limited to 542920 bits only. In [10] present a method for data hiding within transform domain of the color images. Proposed method is based on dividing an image into blocks, then applying the mix column transform based on irreducible polynomial mathematics on specified blocks and hiding the secret message within blocks. The results have proven high capacity and security alongside maintaining reasonable level of imperceptibility. However, the capacity is limited to only 609129 bits.

The genetic algorithm (GA) is recently used to improve the operation of the information hiding systems [11]. The capacity or imperceptibility can be increased by applying GA in steganography [12, 14]. An example of using GA in steganography occurred in [13], where 64 quantized Discrete Cosine Transform (DCT) coefficient is embedded in 8×8 block of JPEG image as 64 genes clusters in a chromosome. Steganography is implemented in a set of chromosomes with coefficient values of 0 and 1. Then, the obtained image is evaluated by the fitness function Mean Absolute Difference (MAD).

The experimental results demonstrate that the method is working properly, considering an efficient solution, and it proves that steganography is the least damaging system. Ghasemi et al. have proposed an imperceptible steganography method based on the genetic algorithm [14]. In this method, data are embedded within 8×8 blocks of cover image by using mapping function and based on the genetic algorithm. Here after, Optimal Pixel Adjustment Process (OPAP) is applied after the message embedding. This method has appropriate embedding capacity, quality, and imperceptibility. The main problems with this method are the high computational cost and execution time [14]. The paper is organized as follows. In section 2, we discuss preliminary and basic concepts. The proposed method is presented in section 3. The results of simulation and performance analysis are represented in section 4. Finally, Conclusion remarks are given in section 5.

2. Basic Concepts

This section discusses on irreducible polynomial mathematics, mix column transform and genetic algorithm.

2.1 Irreducible Polynomial Mathematics

The forward Mix Column Transformation (also, called Mix Columns) operates on each column individually. Each byte of a column is mapped into a new value that is a function of all four bytes in that column. The results of the Mix Column operation are calculated using $GF\left(2^8\right)$ [1] operations. Each element of $GF\left(2^8\right)$ is a polynomial of degree 7 with coefficients in $GF\left(2\right)$. Thus, the coefficients of each term of the polynomial can take the values 0 or 1. Considering that there are 8 terms in an element of $GF\left(2^8\right)$, an element can be represented by bit string of length 8, where each bit represents a coefficient. The Least Significant Bit is used to represent the constant of the polynomial, and going from right to left, represents the coefficient of x^i by the bit b_i, where b_i , is i bits to the left of the least significant bit [15]. If b is assumed one byte, which consists of $b_0b_1b_2b_3b_4b_5b_6b_7$, it is considered by a polynomial with coefficients (0, 1):

$$b_0x^0 + b_1x^1 + b_2x^2 + b_3x^3 + b_4x^4 + b_5x^5 + b_6x^6 + b_7x^7 \quad (1)$$

For example, the bit string (10101011) represents $(x^7 + x^5 + x^3 + x + 1)$. For convenience, a term x_i is found in the expression if the corresponding coefficient is 1; and a term x_i is omitted from the expression if the coefficient is 0 [15].

An example of mix column transform is explained in the appendix A.

2.2 Genetic Algorithm

In this method, the GA is applied to determine the blocks scanning manner in the image and starting blocks in the scanning, in such a way to obtain an image with the highest imperceptibility. For each image, 16 scanning directions can be considered. For instance, the scanning for an image of 4×4 size is drawn in figure 1:

[1] Gallois Field

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 **Code**: 0000	4 3 2 1 8 7 6 5 12 11 10 9 16 15 14 13 **Code**: 0001	4 3 2 1 5 6 7 8 12 11 10 9 13 14 15 16 **Code**: 0010	1 2 3 4 8 7 6 5 9 10 11 12 16 15 14 13 **Code**: 0011
1 5 9 13 2 6 10 14 3 7 11 15 4 8 12 16 **Code**:0100	1 8 9 16 2 7 10 15 3 6 11 14 4 5 12 13 **Code**: 0101	1 8 9 13 2 7 10 14 3 6 11 15 4 5 12 16 **Code**: 0110	1 8 9 16 2 7 10 15 3 6 11 14 4 5 12 13 **Code**: 0111
16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 **Code**: 1000	13 14 15 16 12 11 10 9 5 6 7 8 4 3 2 1 **Code**: 1001	1 5 9 13 2 6 10 14 3 7 11 15 4 8 12 16 **Code**: 0110	1 8 9 16 2 7 10 15 3 6 11 14 4 5 12 13 **Code**: 0111
16 12 8 4 15 11 7 3 14 10 6 2 13 9 5 1 **Code**: 1110	13 12 5 4 14 11 6 3 15 10 7 2 16 9 8 1 **Code**: 1101	4 8 12 16 3 7 11 15 2 6 10 14 1 5 9 13 **Code**: 1110	4 5 12 13 3 6 11 14 2 7 10 15 1 8 9 16 **Code**: 1111

Fig.1 types of block scanning methods within the image

On the other hand, the starting point can be displaced for each of the above 16 scanning [16]. For instance, for the first explained scanning, 4 different starting points can be showed as the figure 2.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 **Starting row: 0** **Starting column: 0**	13 14 15 16 1 2 3 4 5 6 7 8 9 10 11 12 **Starting row: 1** **Starting column: 0**	11 12 13 14 15 16 1 2 3 4 5 6 7 8 9 10 **Starting row: 1** **Starting column: 2**	3 4 5 6 7 8 9 10 11 12 13 14 15 16 1 2 **Starting row: 3** **Starting column: 2**

Fig.2 Four different starting points

Determining the scanning direction and starting point of the secret message can be performed through GA. According to the image size, which is 256×256 and since the image is divided into 4×4 blocks, so that the same $2^6 \times 2^6$ considers 4 bytes for scanning direction, 6 bytes for the column starting point and 6 bytes for the row starting point in this chromosome; consequently the length of chromosome would be 16 bytes. Therefore, a chromosome is designed in the following manner. Also, according to the scanning direction and the starting point of the column and row, the scanning function is selected. For an instance, figure 3 depicted a chromosome along with the scanning manner of the image.

Scanning direction	Column starting point	Row starting point
0100	000001	000010

3	15	11	7
4	16	12	8
5	1	13	9
6	2	14	10

Fig.3 Chromosomes with a starting point and the scanning manner

3. The Proposed Method

The proposed method aims to enhance the speed of the algorithm implementation, the acceptable imperceptibility of the image, and increase level of security by using the mix column transform and GA. The steganography methods are generally formed of two parts of embedding a secret message in the media and extracting the secret message from the media.

3.1 Insertion Algorithm

The embedding algorithm of secret message in the media is as follows. The block diagram of inserting algorithm is shown in figure 4, as well.

1. Dividing the cover image into blocks, each block of size (4×4);
2. Applying the mix column transform on each block and building a converted image;
3. Determining the scanning direction and the starting point for embedding the secret message through GA (the best permutation);
4. Embedding a secret message in the least significant bit (zero bit plane) of the converted image;
5. Embedding the scanning direction and the starting point in the least significant bit of the converted image;
6. Applying the inverse mix column transform on each converted block and building a stego image.

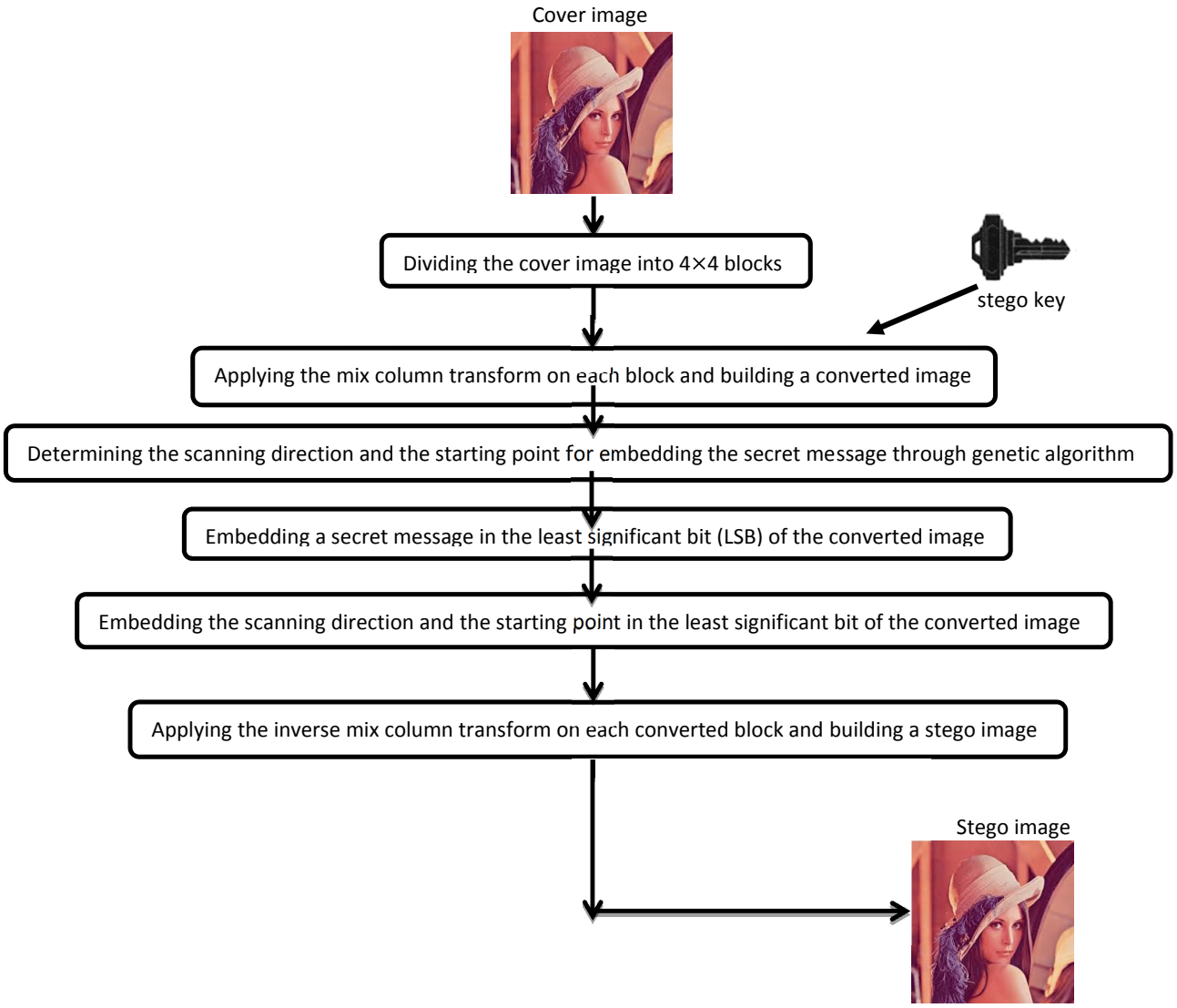

Fig.4 The block diagram of the embedding algorithm

3.2 Extraction Algorithm

The proposed method is a blind algorithm. So, there is no need for the cover image during the extraction process. The block diagram of the proposed secret data extraction is indicated in figure 6.

1. Dividing the stego image into blocks with the same size, which is defined during embedding;
2. Applying the mix column transform on the image and building the converted image;
3. Extracting the least significant bit planes (zero bit plane);
4. Extracting the scanning direction and the starting point of each obtained bit plane in previous stage;

5. Extracting the secret message bits from the bit planes on the basis of the obtained scanning direction and starting point;
6. Reconstructing a secret message from the extracted bits.

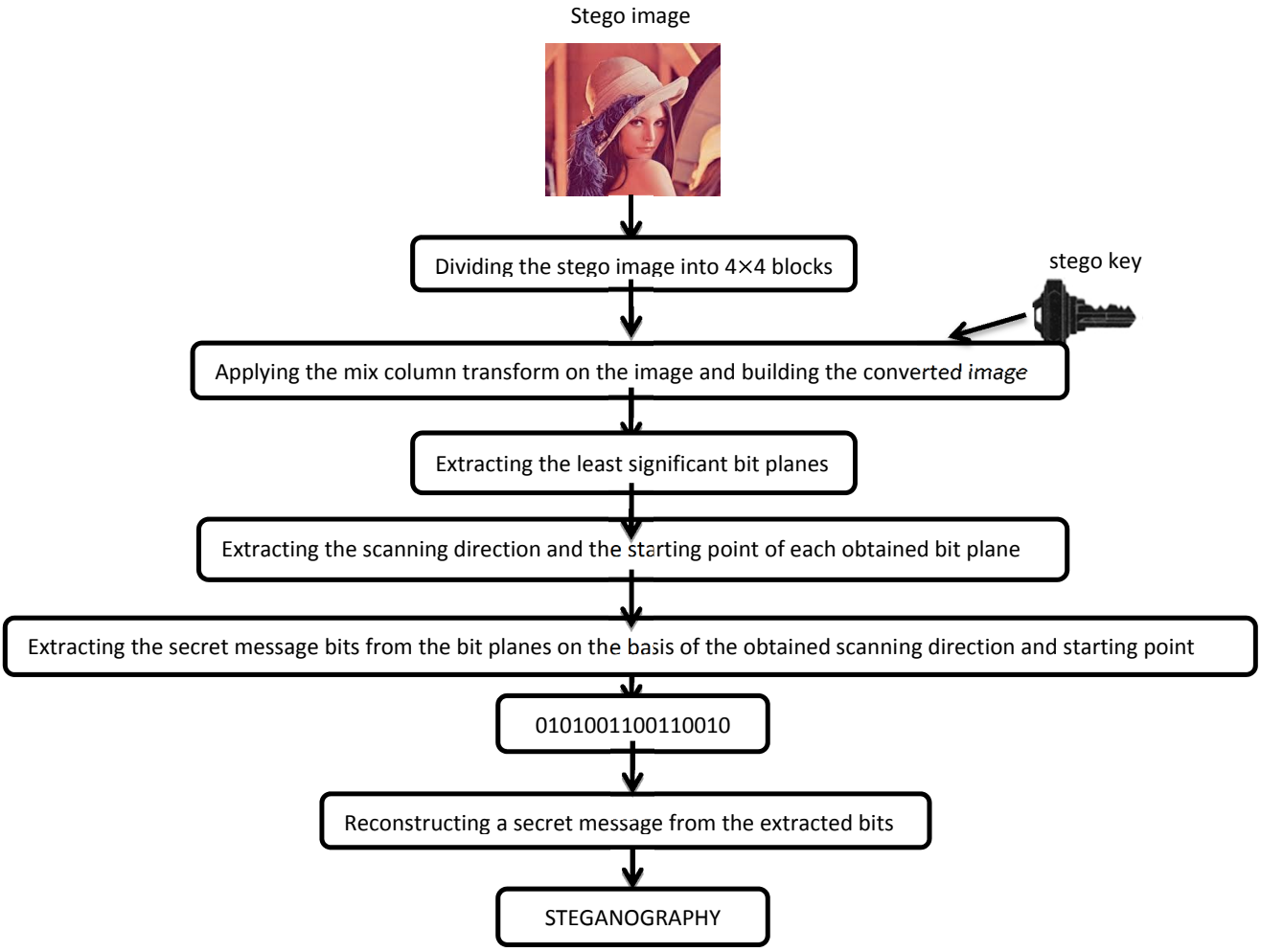

Fig.5 The block diagram of the extraction algorithm

4. The Results of Experiments

The proposed method is implemented on the 8-bit color images with size of 256×256, and JPEG formats (Barbara, penguins, koala, Tulips, Tree, duck [17, 18]) as shown in the figure 6. The secret information to be embedded in these images is shown in figure 7, as well. Implementation is done with 2.53 GHZ Core i3 processor, 4GB memory and Win 7 operating system in MATLAB Software.

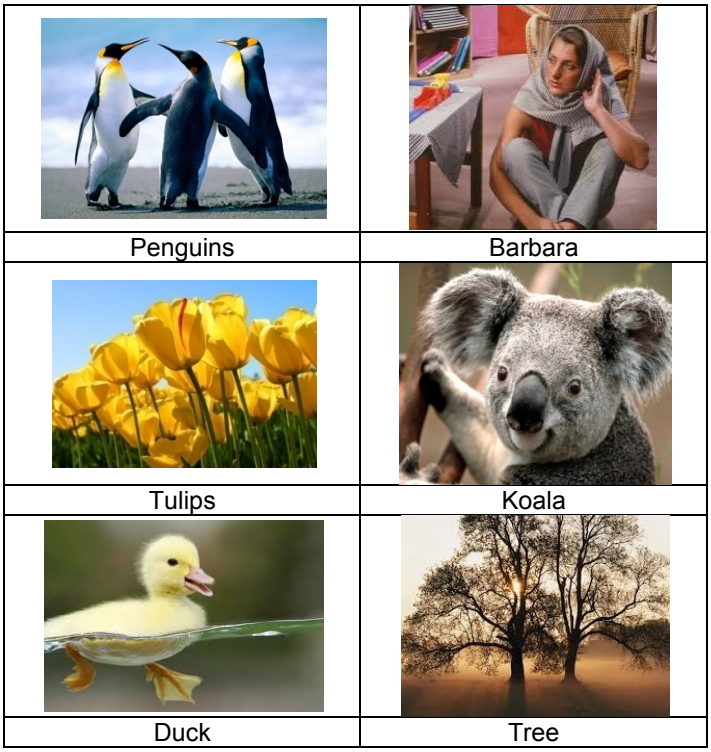

Fig.6 The selected images for steganography

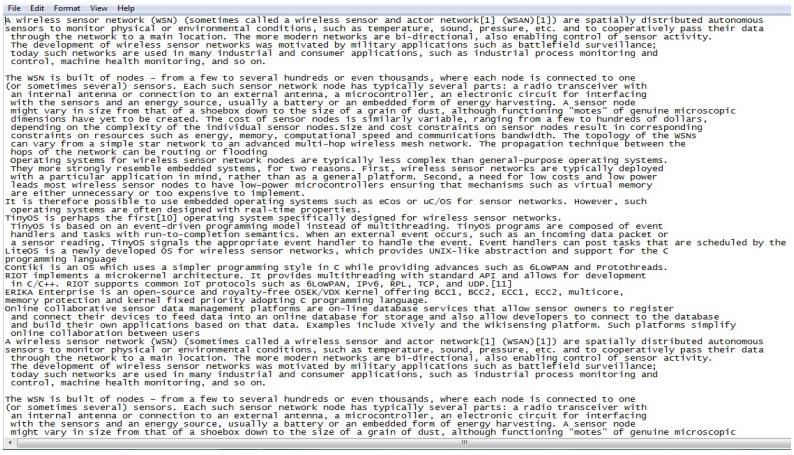

Fig.7 The secret message

The quality of the image is primarily tested by using peak-signal-to-noise ratio. The mean squared error between the cover image and the stego image should be obtained. It is needed to get the PSNR, which is calculated under the equation (2):

$$MSE = \frac{1}{hw}\sum_{x=1}^{h}\sum_{y=1}^{w}(x_{ij} - y_{ij})^2 \qquad (2)$$

In equation (2), w and h parameters show the width and length of the image, respectively. In addition, x_{ij} and y_{ij} stand for the values of pixel [i,j] in the cover image, and the steganographic image [15]. MSE_{avg} can be obtained by the equation (3):

$$MSE_{avg} = \frac{MSE_R + MSE_G + MSE_B}{3} \quad (3)$$

Where, MSE_R , MSE_G and MSE_B are mean squared error in the channels red, green and blue [15].

PSNR measures the peak signal-to-noise-ratio, which is estimated as dB. The fitness function calculates the fitness of each chromosome. The peak signal-to-noise-ratio is used to compare the cover image with stego image. The fitness function is defined in terms of the peak signal to noise ratio according to the equation (4):

$$PSNR = 10 \times \log_{10}\left(\frac{C_{MAX}^2}{MSE}\right) \quad (4)$$

Here, C_{MAX} shows the maximum value in the image and the maximum pixel value from the original image. It is 255 for an 8-bit image. The value of PSNR below 30 indicates the low quality. The PSNR value of Stego image with high quality should be 40dB.

Another measure for understanding image quality is Mean Structural Similarity (SSIM), which is designed to improve on traditional methods such as peak signal-to-noise ratio and mean squared error (MSE). SSIM index takes values in [0, 1]. We calculate it by using the default parameters and based on the code available at [19]. MSSIM is SSIMs averages of red, green and blue channels.

In a group of images with the size of 256×256 which equals (256×256×3) =196608 pixels in figure 8, a message with the length of 13152 characters is embedded in three channels: red, green, and blue. Furthermore, a message with the length of 56615 characters is embedded in three channels (red, green and blue) in another group of the images with the size of 512×512 which equals (512×512×3) =786432 Pixels. The obtained results are shown in tables 1, 2, 3 and 4; and the hidden data embedded in the images are shown in the figure 8, as well.

As it is seen from tables 1 and 2, the proposed method improves visual quality of stego image. So, the message with the same size enhances the image quality and can be concluded in the same quality. The proposed method has more capacity for embedding message. It is noticed that the embedding time is proportionally increased with the image capacity.

Table.1 Steganography results on three color channels for (256×256) images using GA

Color image with size of 256 x 256	Message size (Characters)	Embedding capacity (Bits)	measure PSNR (dB)	MSSIM index	Embedding time (second)
Barbara	13152	105216	37.8019	0.9624	75.1268
Penguins	13152	105216	37.4770	0.9310	96.3344
Koala	13152	105216	37.8524	0.9734	87.0643
Tulips	13152	105216	37.7306	0.9610	102.0415
Tree	13152	105216	37.7440	0.9724	123.8353
Duck	13152	105216	37.9462	0.9320	154.7252

Table.2 Steganography results on red, green, blue channels for (256×256) images without GA

Color image with size of 256 x 256	Message size (Characters)	Embedding capacity (Bits)	measure PSNR (dB)	MSSIM index	Embedding time (second)
Barbara	13152	105216	31.8324	0.8586	42.1768
Penguins	13152	105216	31.6970	0.8013	41.4830
Koala	13152	105216	31.8681	0.8984	41.3590
Tulips	13152	105216	31.3800	0.7937	42.3933
Tree	13152	105216	31.8183	0.9004	41.6750
Duck	13152	105216	31.8562	0.7598	40.7511

Table.3 Steganography results on three color channels for (512×512) images using GA

Color image with size of 512 x 512	Message size (Characters)	Embedding capacity (Bits)	measure PSNR (dB)	MSSIM index	Embedding time (second)
Barbara	56615	452920	37.8763	0.9464	169.7541
Penguins	56615	452920	37.4566	0.9167	157.6727
Koala	56615	452920	37.8470	0.9629	148.3373
Tulips	56615	452920	37.7121	0.9464	164.2869
Tree	56615	452920	37.7380	0.9643	150.5609
Duck	56615	452920	37.9569	0.9262	147.4934

Table.4 Steganography results on red, green and blue channels for (512×512) images without GA

Color image with size of 512 x 512	Message size (Characters)	Embedding capacity (Bits)	measure PSNR (dB)	MSSIM index	Embedding time (second)
Barbara	56615	452920	31.8477	0.8012	165.1426
Penguins	56615	452920	31.6947	0.7668	165.5910
Koala	56615	452920	31.8314	0.8612	167.5302
Tulips	56615	452920	31.3756	0.7417	165.2855
Tree	56615	452920	31.7965	0.8762	171.2344
Duck	56615	452920	31.9205	0.7463	148.4263

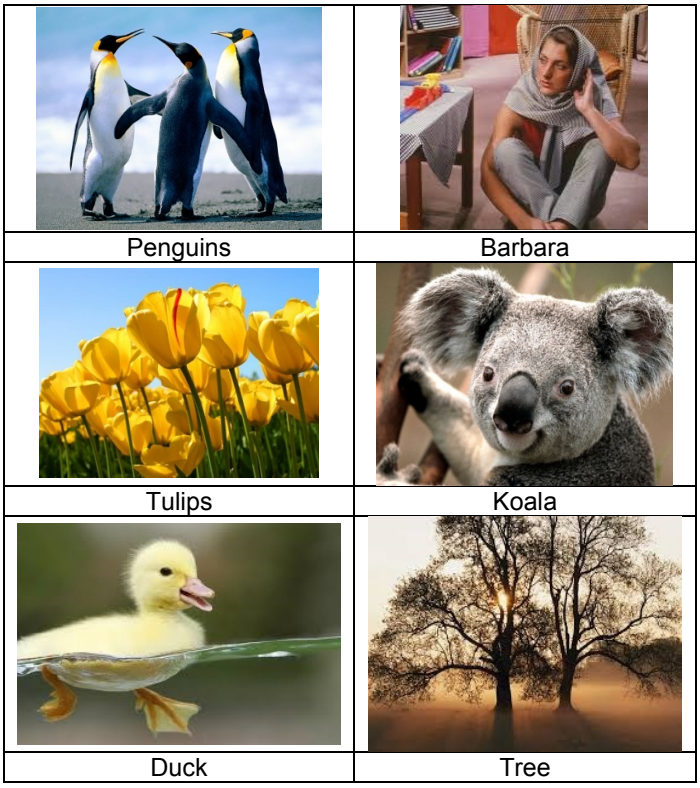

Fig.8 The images obtained by steganography in 3 color channels

The results obtained from the proposed algorithm are compared with the obtained results of the other algorithms in terms of the capacity, PSNR and MSSIM values. They are used to evaluate the algorithm. The obtained results using the proposed method on the images "Lena", "Barbara" in compare to the methods of [14] and [15] are shown in Tables 5.

Table.5 Comparisons of the results with the other alternative schemes

The steganographic schemes		Cover image	PSNR of the Stego image (dB)	MSSIM index	PSNR (dB) of our scheme with same capacity as of the compared scheme
1	Reference [14]	Lena.jpg (512×512)	35.24	0.9395	37.82 (capacity 131072 Bits)
2	Reference [15]	Lena.jpeg (512×512)	32.87	0.9399	37.86 (capacity 609129 Bits)
3	the proposed method	Barbara.jpg (512×512)	37.88	0.9466	37.88 (capacity 800000 Bits)

The proposed method is better than the other similar works in terms of capacity without adding much distortion to the stego image, since the obtained PSNR value is 37.88, and

the MSSIM value is 0.9466. They are relatively high by referring to the increased quality of the images. For example, in row 2 of reference [15], the maximum achieved capacity is 609129 bits with PSNR 32.87 dB, while the proposed method has PSNR 37.86 dB. On the other hand, when comparing the proposed scheme with its alternative method, our method is better for purposes of capacity and imperceptibility since our method has been achieved to higher PSNR and MSSIM using the same embedding capacity and same cover image.

5. Conclusion and Future Works

This paper presents a method for steganography in the transform domain of color images. It provides two levels of security, one in the cryptography and the other in steganography. The genetic algorithm is applied to find the best space for embedding data. Furthermore, the difference between the cover image and stego image can be minimized by the genetic algorithm in order to obtain an acceptable PSNR. Then, the parallel genetic algorithm by calculating fitness is used to reduce the rather high running time of this method. The use of mix column transform enhances the security of the proposed method since the security is depend on a different type of transform, which is on the basis of Irreducible Polynomial Mathematics. Furthermore, the proposed method is a blind algorithm which enhances the security.

As a future work, another evolutionary algorithm with mix column transform can be used, and the other color channels can be evaluated to embed the messages.

References

[1] Maiti, C., Baksi, D., Zamider, I., Gorai, P., & Kisku, D. R. (2011). Data hiding in images using some efficient steganography techniques. In *Signal Processing, Image Processing and Pattern Recognition* (pp. 195-203). Springer Berlin Heidelberg

[2] Cheddad, A., Condell, J., Condell, K., and Kevitt, P.M., (2010). "Digital image steganography: Survey and analysis of current methods". Signal Processing, 90, pp. 727-752.

[3] Petitcolas, F. A., Anderson, R. J., & Kuhn, M. G. (1999). Information hiding-a survey. *Proceedings of the IEEE*, *87*(7), 1062-1078.

[4] Cheddad, A., Condell, J., Curran, K., & McKevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. *Signal processing*, *90*(3), 727-752.

[5] Gonzalez, R. C., Woods, R. E., & Eddins, S. L. U. (2004). Digital Image Processing Using MATLAB: Pearson Prentice Hall. *Upper Saddle River, New Jersey*.

[6] Sajasi, S., & Moghadam, A. M. E. (2015). An adaptive image steganographic scheme based on Noise Visibility Function and an optimal chaotic based encryption method. *Applied Soft Computing*, *30*, 375-389.

[7] Chen, M., Zhang, R., Niu, X., & Yang, Y. (2006, December). Analysis of current steganography tools: classifications & features. In *2006 International Conference on Intelligent Information Hiding and Multimedia* (pp. 384-387). IEEE.

[8] Joshi, S. V., Bokil, A. A., Jain, N. A., & Koshti, D. (2012). Image steganography combination of spatial and frequency domain. *International Journal of Computer Applications*, *53*(5).

[9] Abduallah, W. M., Rahma, A. M. S., & Pathan, A. S. K. (2013, September). Reversible Data Hiding Scheme Based on 3-Least Significant Bits and Mix Column Transform. In *International Conference on Security and Privacy in Communication Systems* (pp. 405-417). Springer International Publishing.

[10] Yu, L., Zhao, Y., Ni, R., & Zhu, Z. (2009). PM1 steganography in JPEG images using genetic algorithm. *Soft Computing*, *13*(4), 393-400.

[11] Raja, K. B., Kumar, K., Kumar, S., Lakshmi, M. S., Preeti, H., Venugopal, K. R., & Patnaik, L. M. (2007, December). Genetic algorithm based steganography using wavelets. In *International Conference on Information Systems Security* (pp. 51-63). Springer Berlin Heidelberg.

[12] Ji, R., Yao, H., Liu, S., & Wang, L. (2006, December). Genetic algorithm based optimal block mapping method for LSB substitution. In *2006 International Conference on Intelligent Information Hiding and Multimedia* (pp. 215-218). IEEE.

[13] Fard, A. M., Akbarzadeh-T, M. R., Varasteh-A, F., & Varasteh-A, F. (2006, April). A new genetic algorithm approach for secure JPEG steganography. In *2006 IEEE International Conference on Engineering of Intelligent Systems* (pp. 1-6). IEEE.

[14] Ghasemi, E., & Shanbehzadeh, J. (2010, December). An imperceptible steganographic method based on Genetic Algorithm. In *Telecommunications (IST), 2010 5th International Symposium on* (pp. 836-839). IEEE.

[15] Abduallah, W. M., Rahma, A. M. S., & Pathan, A. S. K. (2014). Mix column transform based on irreducible polynomial mathematics for color image steganography: A novel approach. *Computers & Electrical Engineering*, *40*(4), 1390-1404.

[16] Kanan, H. R., & Nazeri, B. (2014). A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm. *Expert Systems with Applications*, *41*(14), 6123-6130.

[17] Mustafa, W., Monem, A., Rahma, S., & Khan, A. S. (2013). A New Data Hiding Technique Based on Irreducible Polynomials.

[18] Joshi, S. V., Bokil, A. A., Jain, N. A., & Koshti, D. (2012). Image steganography combination of spatial and frequency domain. *International Journal of Computer Applications*, *53*(5).

[19] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2003). The SSIM index for image quality assessment. *MATLAB implementation available online from: http://www. cns. nyu. edu/lcv/ssim*, *23*(66), 6.

Appendix A

How to calculate Mix Column Transform

A block matrix is assumed from a cover image with size (4×4) in order to apply a mix column transform; and the other matrix, called transform matrix, which is generated at random and considered with the same size:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 03 & 02 \end{bmatrix} \times \begin{bmatrix} BF & CD & BF & 8A \\ BF & AF & C0 & AD \\ 6E & 73 & 9D & A3 \\ 85 & 82 & 89 & 86 \end{bmatrix}$$

Transformed Matrix Block Matrix

The values of both matrices should be converted into the polynomial equations in order to multiply each rows of the transform matrix with each column of the original values of the block matrix as shown below:

$$\begin{bmatrix} x & x+1 & 1 & 1 \\ 1 & x & x+1 & 1 \\ 1 & 1 & x & x+1 \\ x+1 & 1 & 1 & x \end{bmatrix}$$

$$\begin{bmatrix} x^7+x^5+x^4+x^3+x^2+x+1 & x^7+x^6+x^3+x^2+1 & x^7+x^5+x^4+x^3+x^2+x+1 & x^7+x^3+x \\ x^7+x^5+x^4+x^3+x^2+x+1 & x^7+x^5+x^3+x^2+x+1 & x^7+x^6 & x^7+x^5+x^3+x^2+1 \\ x^6+x^5+x^3+x^2+x & x^6+x^5+x^4+x+1 & x^7+x^4+x^3+x^2+1 & x^7+x^5+x+1 \\ x^7+x^2+1 & x^7+x & x^7+x^3+1 & x^7+x^2+x \end{bmatrix}$$

It is clear that the largest element, appeared in this example is x^7, because the results of the Mix Columns operation are calculated using $GF(2^8)$ operations, where each element of $GF(2^8)$ is a polynomial of degree 7 with coefficients in $GF(2)$.

$$x.(x^7+x^5+x^4+x^3+x^2+x+1) = x^8+x^6+x^5+x^4+x^3+x^2+x$$

$$(x+1).(x^7+x^5+x^4+x^3+x^2+x+1) =$$

$$x^8+x^6+x^5+x^4+x^3+x^2+x+x^7+x^5+x^4+x^3+x^2+x+ = x^8+x^7+x^6+1$$

$$1.(x^6+x^5+x^3+x^2+x) = x^6+x^5+x^3+x^2+x$$

$$1.(x^7+x^2+1) = x^7+x^2+1$$

$$x^8+x^6+x^5+x^4+x^3+x^2+x^2+x+x^8+x^7+x^6+1+x^6+x^5+x^3+x^2+x+x^7+x^2+1 = x^6+x^4+x^2$$

The matrix resulted from the mix column transform should be multiplied by the inverse matrix to obtain the original values of block matrix.

$$\begin{bmatrix} OE & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \times \begin{bmatrix} 55 & 9A & 2A & C6 \\ ED & 9F & 11 & B3 \\ 48 & 19 & DE & EB \\ 1A & 8F & 8F & 9D \end{bmatrix}$$

Inverse Matrix Block Matrix

Again, all the values of both matrices should be converted to polynomial equations.

$$\begin{bmatrix} x^3+x^2+x & x^3+x+x & x^3+x^2+1 & x^3+1 \\ x^3+1 & x^3+x^2+x & x^3+x+1 & x^3+x^2+1 \\ x^3+x^2+1 & x^3+1 & x^3+x^2+x & x^3+x+1 \\ x^3+x+1 & x^3+x^2+1 & x^3+1 & x^3+x^2+x \end{bmatrix}$$

$$\begin{bmatrix} x^6+x^4+x^2+1 & x^7+x^4+x^3+x & x^5+x^3+x & x^7+x^6+x^2+x \\ x^7+x^6+x^5+x^3+x^2+1 & x^7+x^4+x^3+x^2+x+1 & x^4+1 & x^7+x^5+x^4+x+1 \\ x^6+x^3 & x^4+x^3+1 & x^7+x^6+x^4+x^3+x^2+x & x^7+x^6+x^5+x^3+x+1 \\ x^4+x^3+x & x^7+x^3+x^2+x & x^7+x^3+x^2+1 & x^7+x^4+x^3+x^2+1 \end{bmatrix}$$

$$(x^3+x^2+x).(x^6+x^4+x^2+1) = x^9+x^7+x^5+x^3+x^8+x^6+x^4+x^2+x^7+x^5+x^3+x = x^9+x^8+x^6+x^4+x^2+x$$

$$(x^3+x+1).(x^7+x^6+x^5+x^3+x^2+1) = x^{10}+x^9+x^8+x^6+x^5+x^3+x^8+x^7+x^6+x^4+x^3+x+x^7+x^6+x^5+x^3+x^2+1 =$$

$$x^{10}+x^9+x^6+x^4+x^3+x^2+x+1$$

$$(x^3+x^2+1).(x^6+x^3) = x^9+x^6+x^8+x^5+x^6+x^3 = x^9+x^8+x^5+x^3$$

$$(x^3+1).(x^4+x^3+x) = x^7+x^6+x^4+x^4+x^3+x = x^7+x^6+x^3+x$$

$$x^9+x^8+x^6+x^4+x^2+x+x^{10}+x^9+x^6+x^4+x^3+x^2+x+1+x^9+x^8+x^5+x^3+x^7+x^6+x^3+x =$$

$$x^{10}+x^9+x^7+x^6+x^5+x^3+x+1$$

If the result of multiplication leads to a polynomial with a degree larger than 7, the resulted polynomial should be reduced through dividing it by the irreducible polynomial ($x^8+x^4+x^3+x+1$) to get the remainder. The result is $x^{10}+x^9+x^7+x^6+x^5+x^3+x+1$, which has a degree (10>7); so, it should be reduced through dividing it by ($x^8+x^4+x^3+x+1$).

$$\begin{array}{r} x^2+x \\ x^8+x^4+x^3+x+1 \overline{)\ x^{10}+x^9+x^7+x^6+x^5+x^3+x+1} \\ \underline{x^{10}+x^6+x^5+x^3+x^2} \\ x^9+x^7+x^2+x+1 \\ \underline{x^9+x^6+x^5+x^2+x} \\ x^7+x^5+x^4+1 \end{array}$$

(10110001) ◄—— $x^7+x^5+x^4+1$

Finally, the matrix obtained by the inverse mix column transform is combined again with the transformed matrix.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 03 & 02 \end{bmatrix} \times \begin{bmatrix} B1 & CD & BF & 83 \\ B6 & AF & C0 & A0 \\ 63 & 73 & 9D & A8 \\ 8E & 82 & 89 & 88 \end{bmatrix}$$

$x.\left(x^7 + x^5 + x^4 + 1\right) = x^8 + x^6 + x^5 + x$

$(x + 1).\left(x^7 + x^5 + x^4 + x^2 + x\right) = x^8 + x^6 + x^5 + x^3 + x^2 + x^7 + x^5 + x^4 + x^2 + x = x^8 + x^6 + x^5 + x$

$1.\left(x^6 + x^5 + x + 1\right) = x^6 + x^5 + x + 1$

$1.\left(x^7 + x^3 + x^2 + x\right) = x^7 + x^3 + x^2 + x$

$x^8 + x^6 + x^5 + x + x^8 + x^6 + x^5 + x + x^6 + x^5 + x + 1 + x^7 + x^3 + x^2 + 1 = x^6 + x^4 + x^2 + 1$

The result is $x^6 + x^4 + x^2 + 1$, which is equivalent to (0101 0101) = 55.