Data Hiding by LSB Substitution using Gene Expression Programming

Marghny H. Mohamed
Associate Professor
Dep. of Computer Science, Faculty of Computers and Information Systems, Assuit University, Egypt

Hussein I. Abul-Kasim
MSc Candidate
Dep. of Computer Science, Faculty of Science, South Valley University, Egypt

ABSTRACT

Steganography is defined as the art and science of writing hidden in such a way that no one apart from the sender and the intended recipient even knows that there is a hidden message. One of the most important techniques of Steganography is the Least Significant Bit (LSB) which embeds the secret message in the host image. It is based on replacing the LSBs of the host-image with the secret message bits giving a stego-image. The proposed scheme consists mainly of two phases: In the first phase, we propose a hybrid data hiding scheme incorporates LSB technique with a key permutation method. While in the second phase, we proposed a new scheme for finding the optimal key-permutation by using gene expression programming (GEP). Where, GEP is a powerful evolutionary algorithm for data analysis and combines the advantages of both genetic algorithms (GA) and genetic programming (GP).

General Terms

Information Security, Data Hiding Techniques, Algorithms.

Keywords

Steganography, Data hiding, Key-permutation, LSB substitution, Gene Expression Programming.

1. INTRODUCTION

The recent interest of data hiding is fueled by the increased amount of communication through the internet to transmit a large amount of information. Some of them may be secret information which is candidate to unauthorized access. In order to keep the unauthorized users away, variety of techniques has been proposed for providing a secure transmission of information. Data encryption and data hiding techniques have become popular and complement each other. Whereas data encryption transforms data into seemingly meaningless bits called ciphertext through cipher algorithms, this en-able only the user that has a key to decrypt the secret data from the cipher texts to the plain texts. For any unauthorized user who does not have a key, the ciphertext will look like nothing but streams of meaningless codes. Although data encryption is a good way to prevent unauthorized user from accessing secret data, it still has some weaknesses. The appearance of ciphertexts would give un-authorized user an impulse to recover them. Data hiding techniques embeds the important data inside multimedia data such as images, videos or audio. Digital images are considered good cover carriers because of their insensitivity to human visual systems. Watermarking and steganography are two major kinds of information hiding technology.

Watermarking is used to embed a distinguishable symbol, e.g., a signature or a trademark, into host signals to authorize the ownership of the signals, The Steganography is used to hide information inside information, thus hiding the existence of the communicated information [1].

The word steganography is of Greek origin which means “covered or hidden writing” [2]. The general purpose of steganography differs from cryptography, which is intended to make a message unreadable by a third party but does not hide the existence of the secret communication. Some of authors categorize steganography as a form of cryptography although steganography is separate and distinct from cryptography where hidden communications are a form of secret writing.

Many of techniques of data hiding have been proposed [3-6, 9,17, 22] the Least-Significant-Bit (LSB) technique is one of the most widely used scheme for image Steganography, based on manipulation the least significant bit (LSB) plans. This technique replaces some LSB of the cover-image with the secret data.

Wang et al. [15] proposed a method to embed secret messages in the moderately significant bit of the cover-image. A genetic algorithm is developed to find an optimal substitution matrix for the embedding process. They also proposed to improve the image quality of the stego-image using a local pixel adjustment process (LPAP). Wang et al. [16] also proposed a novel method to embed data inside the host image. This method based on simple LSB substitution data hiding. They also developed the optimal k rightmost LSB substitution method to solve the problem when k is large. Chan et al. [7] proposed a method by applying an optimal pixel adjustment process to the stego-image obtained by the simple LSB substitution method. The proposed method improves the image quality and computational efficiency. Chang et al. [18] proposed a method of finding the optimal LSB in image hiding by dynamic programming strategy. The proposed method finds the optimal LSB substitution that Wang [16] found of approximate OLSB as well as reduces the computation time. Ching and Shu [14] proposed a method to find the optimal LSB substitution. The proposed method improves the qualities of the stego-images using ant colony optimization algorithm.

In this work, we proposed a method to embed secret messages inside the host image based on LSB substitution. The method depending on permutation and gene expression programming, whereas, to make the important data out of reach except the authorized users and obtain better embedding results, a key-permutation method with an optimal LSB substitution method is presented. The general idea of key-permutation method stated as: A random key is generated and distributed to the communication parties. Then the data is mapped with the help of the key at the sending end before embedding process, an opposite operation is then performed at the receiving end to reveal the secret data. Using the proposed method gene expression programming, the key is optimized to select the best embedding results for a set of all possible keys (key space).

The rest of this paper is structured as follow: In section 2, the concept of image hiding by LSB substitution is represented. In section 3, we describe an overview of gene expression programming. In section 4, the key-permutation method is described. In section 5, the proposed algorithm introduced, and then the optimal substitution of the LSB by using key permutation method is demonstrated. Experimental results with a brief discussion are clarified in section 6. Finally conclusions are presented in section 7.

2. OVERVIEW OF DATA HIDING BY SIMPLE LSB SUBSTITUTION

Initially, the operations of data hiding by simple LSB substitution method is described as follows: Let C be the original 8-bit grayscale cover-image of $M_c \times N_c$ pixels, represented as

$$C=\{x_{ij}\,|\,0\le i\le M_c, 0\le j\le N_c, x_{ij}\in\{0,1,2,\ldots,255\}\}, \quad (1)$$

and M is the n-bit secret message which can be represented by

$$M=\{m_i\,|\,0\le i\le n, m_i\in\{0,1\}\}. \quad (2)$$

Suppose that the n-bit secret message M is to be embedded into the k-rightmost LSBs of the cover-image C. Firstly, the secret message M is rearranged to form a k-bit virtual image M', which can be represented as

$$m'=\{m'_i\,|\,0\le i\le n', \in\{0,1,\ldots,2^k-1\}\}, \quad (3)$$

where, $n' = M_c \times N_c$. The mapping between the n-bit secret message M= $\{m_i\}$ and the embedded message $M' = \{m'_i\}$ can be defined as follows:

$$m'_i=\sum_{j=0}^{k-1} m_i\times k + j\times 2^{k-1-j}. \quad (4)$$

Secondly, a sub-set of n' pixels $\{x_1, x_2,., x_n\}$ is chosen from the cover-image C in agreed upon sequence. The embedding process is completed by replacing the k LSBs of xi by m'i. Mathematically, the pixel value xi of the chosen pixel for storing the k-bit message m'_i is modified to form the stego-pixel x'_i as follows:

$$X'_i = x_i - x_i \bmod 2^k + m'_i. \quad (5)$$

The extraction process, gives the stego-image S, the embedded messages can be extracted without referring to the original cover-image. Using the same sequence as in the embedding process, the set of pixels $\{x'_1, x'_2,.., x'_n,\}$ storing the secret message bits are selected from the stego-image. The k-rightmost LSBs of the selected pixels are extracted and lined up to reconstruct the bits of the secret message. Mathematically, the embedded message bits m'i can be recovered by

$$m'_i = x'_i \bmod 2^k. \quad (6)$$

Moreover, the quality of the stego-image produced by simple LSB substitution may be not acceptable. This means that the method degrades the image quality and probably attracts unauthorized attention. To solve these problems, a key permutation technique is integrated with an optimal LSB substitution method to improve the security of the model and quality of the stego-image.

3. KEY-PERMUTATION METHOD

In key-permutation method [22], the cover image C, and the secret message M are defined and rearranged to form block-bits (*blk*) getting C'' and M'' respectively.

Where

$$C''=\{c''_i\,|\,0\le i\le 2^{blk}-1\,|\,c''_i\in\{0,1,2,\ldots,2^{blk}-1\}\}, \quad (7)$$

$$M''=\{m''_i\,|\,0\le i\le 2^{blk}-1\,|\,m''_i\in\{0,1,2,\ldots,2^{blk}-1\}\}. \quad (8)$$

Mathematically, the ciphering process will be obtained by performing bitwise XOR operating $\oplus$ to each block of the C'' with M'' as follows:

$$cipher_i = c''_i \oplus m''_i,\ 1\le i\le length\,of\,(\text{M})\text{ in } blk(M_{blk}),$$

then

$$cipher=\{cipher_i\,|\,1\le i\le length\ of\ M''in\ blk\,|$$

$$cipher_i\in\{\{0,1,2,\ldots,2^{blk}-1\}, \text{ where}$$

$$cipher_i = c''_i \oplus m''_i \quad (9)$$

3.1 Key generation

All possible permutations of the blk-bit key is generated, as

$$\left(key_{blk}=\{e_1,e_2,e_3,\ldots,e_{2^{blk}}\}\right), \quad (10)$$

where e_i is the *ith* element of the key, *i* is the index of the *ith* element in the key, where each key is of size blk.

Before the sender embeds the secret data into the k-LSBs of the cover-image C'', the method utilizes a sequential search in order to locate and return the positions of all elements in the key sequence representing the binary of the ciphered secret data plain-text characters as follows:

$$Position_i = locate(cipher_i, key_{blk}), \text{ where}$$

$$position=\{position_i\,|\,1\le i\le length\ (M'')\,|$$

$$position,\in\{0,1,2,\ldots,2^{blk}-1\}\}. \quad (11)$$

Finally, the embedding process is completed by replacing the k-LSBs of C'' by the positions getting the stego-image S.

3.2 Confusion and diffusion

Confusion and diffusion are two properties of the operation of secure cipher which were suggested by Claude Shannon [20, 21]. Confusion is used to hide the relationship between the ciphertext and the key to frustrate the adversary who uses ciphertext statistics to find the key, and can be achieved by means of substitution techniques. Diffusion is used to hide the relationship between the ciphertext and the plaintext to frustrate the adversary who uses ciphertext to find the plaintext, which can be achieved by permutation techniques.

3.3 Secret message recovery

At the receiving end, we must follow in order the following steps:

- Position extraction: The positions data are extracted from the k-LSBs of the stego-image S by
  Position = extract (stego-image, k) (12)
- Cipher data retrieval: here the ciphered data will be obtained from the key by according to its position
  cipher = key_{blk} (position) (13)
- Deciphering: M" will be obtained by xoring the ciphered data with the C", as follows
  M"= cipher C" (14)
- Original secret message reconstruction: now M will be reconstructed by rearranging the M" from blk-bit to its original form
  M= map (M") (15)

4. GENE EXPRESSION PROGRAMMING (GEP)

GEP is a powerful evolutionary algorithm incorporates both the simple linear chromosomes of fixed length similar to the ones used in genetic algorithms (GAs) and the ramified structures of different sizes and shapes similar to the parse trees of genetic programming (GP) [9-12, 23]. The main difference among the three algorithms resides in the nature of the individuals: In GAs the individuals are linear strings of fixed length (chromosomes), in GP the individuals are non-linear entities of different sizes and shapes (parse trees), in GEP the individuals are encoded as linear strings of fixed length (the genome or chromosomes) which are afterwards expressed as non-linear entities of different sizes and shapes (i.e., simple diagram representations or expression trees).

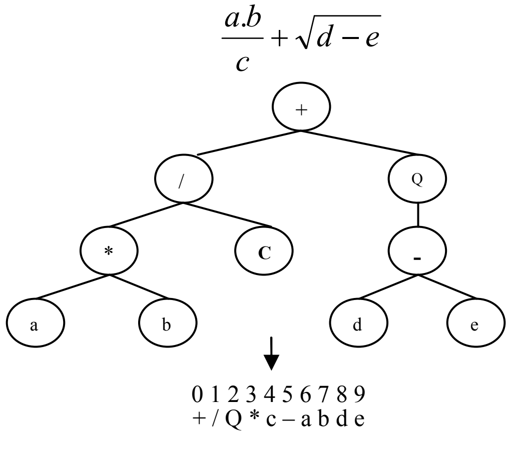

Figure 1: An example of expression trees and Karva language [12]

The main players in gene expression programming are only two: the chromosomes and the expression trees (ETs). The expression of the genetic information's encoded in the chromosome. Else, the process of information decoding is called translation and this process implies a kind of code and a set of rules. The genetic code of GEP can be represented in a simple way: a one-to-one relationship between the symbols of the chromosomes and the function and terminals they represent in the trees. The rules determine the spatial organization of functions and terminals in the ETs and the type of interaction between sub-ETs in multigenic systems. Therefore, there are two languages in GEP: the language of the genes and the languages of the expression trees (figure 1), However, thanks to the simple rules that determine the structure of ETs and their interactions, it is possible to infer immediately the phenotype given the sequence of the genotype, and vice versa, this bilingual and unequivocal system is called Karva language.

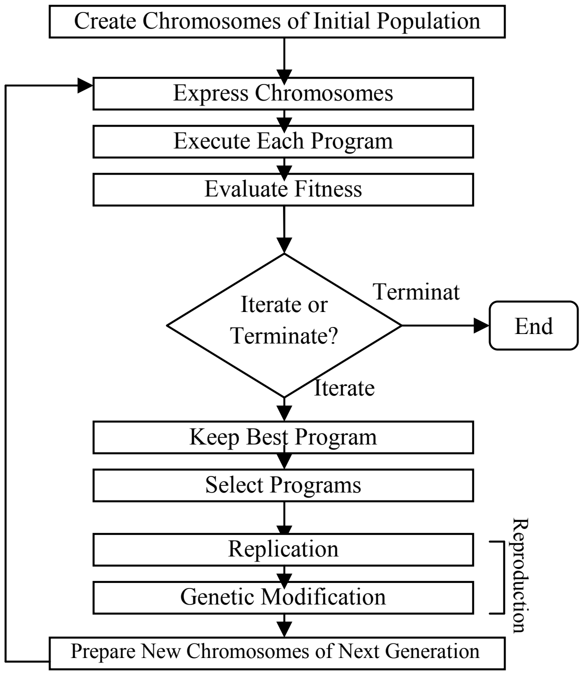

Figure 2: The flowchart of a GEP [12]

In gene expression programming algorithm, the process begins with creating the random generation of the chromosomes of a certain number of individuals (the initial population). Then these chromosomes are expressed and the fitness of each individual is evaluated against a set of fitness cases (also called selection environment which, in fact, is the input to a problem). The individuals are then selected according to their fitness (performance) to reproduce with modification, leaving progeny with new traits. These new individuals are, in their turn, subjected to the same developmental process: expression of the genomes, confrontation of the selection environment, selection, and reproduction with modification. The process is repeated for a certain number of generations or until a good solution has been selected (see figure 2).

5. PROPOSED ALGORITHM

A model developed using the key-Permutation method with the proposed gene expression programming algorithm to select the key that is optimal from all possible keys in such a way that minimum effect could be noticed in the stego-image after embedding the data by the selected key. The GEP algorithm distinguishes with its high accuracy and its fast performance, due to its structure.

5.1 Encoding

Data representation is an essential process for implementing GEP according to the nature of the problem. In order to design the chromosomes, a GEP technique called Multigene Families (MGFs) are used which are very useful for finding solutions to combinatorial problems as different items can be organized into MGFs. These MGFs consist of clusters of related genes encoding, and each gene has the length g=1 and exclusively composed of one terminal *t*=1. This kind of genes is obtained when the head length *h* is zero. Where, the terminal *t* evaluated by the equation

$$t = (n-1)h + 1, \tag{16}$$

and *n* denotes the largest arity of the functions used in the gene's head.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
7	1	12	5	13	3	14	0	8	9	15	11	2	4	6	10

Figure 3: A GEP chromosome composed of one MGF with sixteen genes each gene represents an element in the key

Hence, all the genes of the key-permutation can be encoded in a MGF, whereas the expression consists of the spatial organization of all the elements and the elements of MGF must all be presented and cannot be represented more than once [10]. In our study, a chromosome G in GEP composed of one MGF, consisting of 2^k genes is described by a key-permutation as follows

$$G = g_0 g_1 \ldots g_{2^k-1}, \tag{17}$$

where g_0 represents the first element of the key, g_1 represents the second element of the key, and so on.

Consider, for instance k=4, then the chromosome length is 2^4 =16 and they can be represented as shown in Figure 3.

5.2 Creation of the initial population

In the first step of the proposed algorithm, many individual solutions are randomly generated to form an initial population of a certain size. These initial individuals are the first set of candidate solutions to the problem at hand and the population size depends on the nature of the problem.

5.3 Fitness function

The fitness function evolutes the quality of the represented solution, it is considered the most fundamental component of the gene expression programming algorithm. Where, it directs the evolution toward the desired objective. An individual's fitness value should represent how good of a solution to the given problem that it represents. The fitness function is defined in this work as the mean square error MSE. It takes the differences between the original cover image and the optimized stego-image. For our purpose the optimal (most fit) key solution is the one used to embed secret message in the cover image produces a highest capacity with minimum distortion compared with other keys. The measurement of maximum capacity and minimum distortion is evaluated by the maximum PSNR, which means minimum MSE for each key embedding, so that our goal is to select a solution with maximum PSNR values on our problem consideration. The PSNR is estimated in decibel (dB), defined as:

$$PSNR = 10 * \log_{10}\left(\frac{255*255}{MSE}\right). \tag{18}$$

Where, MSE is the mean square error, which is defined as:

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij} - y_{ij}\right)^2 . \tag{19}$$

Where, x_{ij} denotes the original pixel value, and y_{ij} denotes the processed pixel value, and m and n denote the width and height of the image respectively.

5.4 Roulette wheel selection

In this selection, individuals are selected according to their fitness by roulette wheel sampling. The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness. Where, a random number is generated and the individual whose segment spans the random number is selected. The process is repeated until the desired number of individuals has been obtained. This technique is analogous to a roulette wheel with each slice proportional in size to the fitness [10, 13].

5.5 Elitism selection

Elitism is a technique which guarantees that the fittest Chromosome of the population is cloned into the next generation without being altered by genetic operator(s), preserving the best material from one generation to another. Moreover, it allows the use of several modification operators at relatively high rates without the risk of causing a mass extinction [10].

5.6 Reproduction

In reproduction process, the second generation population of solutions is generated from those selected through genetic operator(s) by roulette-wheel sampling coupled with elitism, for each new solution to be produced a (parent) solution is selected according to roulette wheel selection to produce a (child) solution using the inversion operator, the new solution typically shares of the characteristics of its parents. Then the process continues until a new population of solutions of a certain size is generated. The processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Indeed, the average fitness will have increased by this procedure for the population. Although inversion is the only combinatorial-specific genetic operator in this work, it is possible to use other operators such as restricted permutation, gene deletion/insertion, generalized permutation, and/or sequence deletion/insertion [10].

5.7 Inversion Operator

The inversion operator is the most efficient combinatorial-specific genetic operators, causing populations to evolve with great efficiency even if used as the only source of genetic modification will produce better results than when combined with the others operators. The inversion operator randomly selects the chromosome, the multigene family to be modified, the inversion points in the MGF, then inverts the sequence between the two selected points. Where, each chromosome can only be modified once by this operator.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
7	1	12	5	13	3	14	0	8	9	15	11	2	4	6	10

A

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
7	1	12	5	13	3	14	2	11	15	9	8	0	4	6	10

B

Figure 4: a) Genes 7 and 2 are randomly chosen in the chromosome of the MGF as the inversion points. b) Then the sequence between the selected points is inverted to form a child chromosome

Moreover, inversion is not directly applied to all the selected chromosomes but it is applied to a selected number of chromosomes according to a predefined inversion probability percentage. In addition, the whole MGF can be inverted when the first and the last genes of a multigene family are chosen as inversion point or it allows small adjustment when two close genes are chosen [11].

6. RESULTS AND DISCUSSIONS

Three experiments were carried out to evaluate the effectiveness of our proposed method in case of k=1, 2 and 4-LSBs insertion, and the optimal GEP parameters are listed as follow:

- Maximum generation = 100, as shown in figure 7.
- Population size = 200, as shown in figure 6.
- Inversion rate = 0.3, as shown in figure 5.
- No. of MGFs = 1.
- No. of genes per MGF = 2,4,16, for k=1-LSB, 2-LSB, and 4-LSBs insertion respectively.

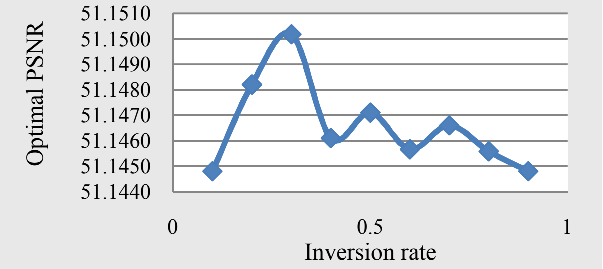

Figure 5: Optimal PSNR vs. inversion rate

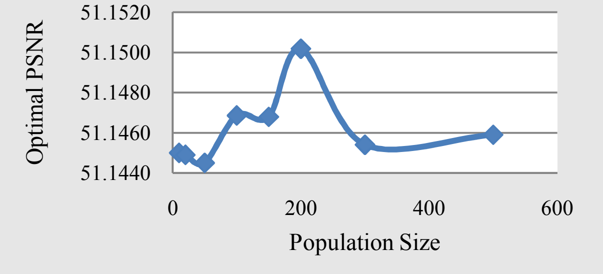

Figure 6: Optimal PSNR vs. population size

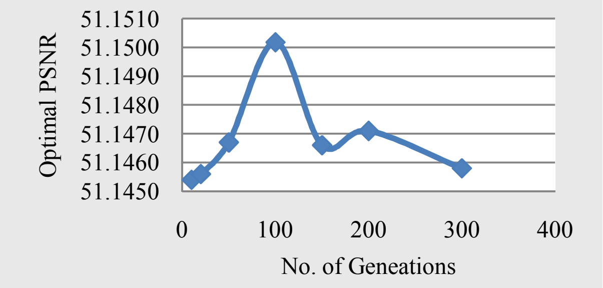

Figure 7: Optimal PSNR vs. No. of generations

6.1 Experiment 1

In the first experiment, the method is applied on two standard 8-bits per pixel, gray scale cover images, “baboon” and “lena”, each has the size 512×512 pixels and the secret messages are deferent size of a gray-scale image “tiffany” as shown in figure 8. These images of sizes 512×256 pixels for 4-LSB insertion, 256×256 pixels for 2-LSB insertion and 256×128 pixels for 1-LSB insertion.

The results of embedding the secret images into the cover images are listed in table 1. The PSNR is used in this work to evaluate the image quality as described in equation 18.

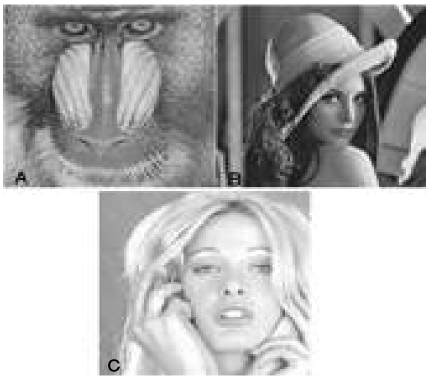

Figure 8: Cover images (a) Baboon, (b) Lena and the secret image (c) Tiffany

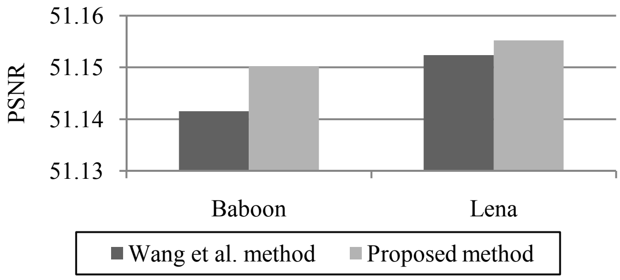

Figure 9: The results of optimal embedding Secret image to the 1-LSB of the cover images

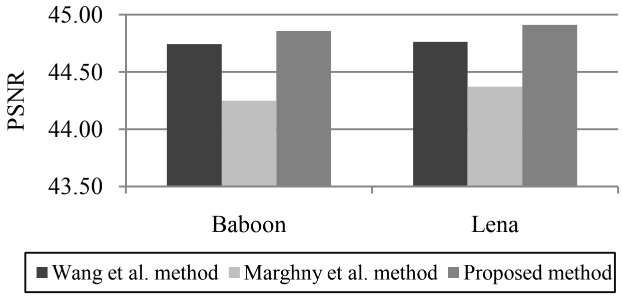

Figure 10: The results of optimal embedding Secret image to the 2-LSB of the cover images

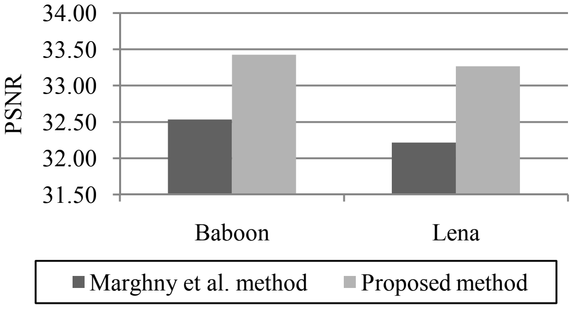

Figure 11: The results of optimal embedding secret image to the 4-LSB of the cover images

6.2 Discussion of experiment 1

The experiment compares the embedding results obtained by Wang et al. method [16], Marghny et al. method [22], and the proposed method, when k=1,2 and 4-LSBs insertion as shown in table 1 and figures 9,10,11. The results demonstrated that the quality of the stego-image is improved by using the GEP approach.

6.3 Experiment 2

In the second experiment, the effect of increasing the number of the keys on the ciphered messages according to our proposed method is evaluated and listed in table 3, the following data set is used in this experiment.

Figure 12: Cover images

Cover images: figure 12.

- Baboon 131×131 pixels – grayscale (17161 bits)
- Lena 131×131 pixels – grayscale (17161 bits)
- Barbara 131×131 pixels – grayscale (17161 bits)
- Pepper 131×131 pixels – grayscale (17161 bits)

Secret messages: random data of the following:

- File Secret1 of size 65× 33 bytes (17160 bits) for the 1-LSB insertion.
- File Secret2 of size 66× 65 bytes (17160 bits) for 2-LSB insertion.
- File Secret4 of size 131× 65 bytes (17030bits) for 4-LSB insertion.

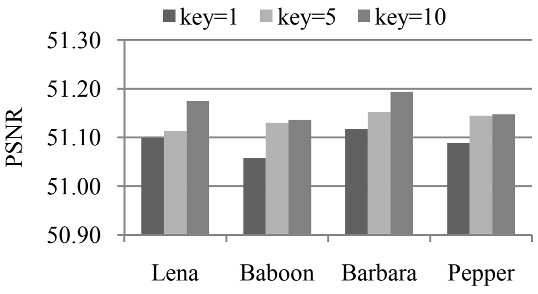

Figure 13: The results of optimal embedding *Secret1* to the 1-LSB of the cover images with different key numbers

6.4 Discussion of experiment 2

Experiment results of table 3 and figure 13 demonstrates that the quality of the stego-images increases with increasing of number of key-permutations for key=1, 5 and 10 respectively, and different LSBs insertions (k=1, 2 and 4), this means that the quality of the stego-image improved by applying the optimal key-permutation using GEP.

The column labeled Cipher in table 2 is the PSNR of the encrypted secret data before applying the key-permutation method.

6.5 Experiment 3

Table 2 and figure 12 shows the effect of increasing the number of key permutations of the proposed method on the computation time, to conduct this experiment we used 8-bit per pixels gray scale cover image "lena" with the size 131× 131 pixels. The secret messages are the same random data files used in experiment 2.

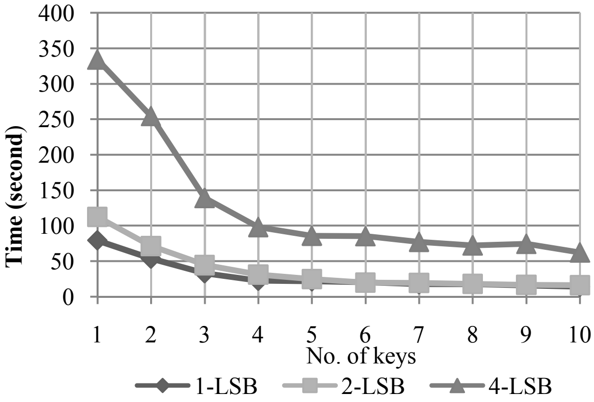

Figure 14: The average computation time over 10-runs in case of 1-LSB, 2-LSB, and 4-LSB insertion

6.6 Discussion of experiment 3

The experiment indicated that the computation time decreases by the number of keys increase until reaches some points. This is due to reduce the search space of GEP process, which will get fast result compared to huge search space. This means that, by increasing the number of key permutations the computation time decreases and the system immunity will increase against attacks.

7. CONCLUSION

Steganography is a process of hiding a secret message within an image in such a way that no one even knows the presence of the hidden message. One of the simplest methods is the least significant bit (LSB) substitution method that embeds a secret image in the least significant bits into the pixels of a host image.

In this paper, we have proposed a new scheme for solving the problem of hiding important data in the rightmost k LSBs of the cover-image when k is large. The proposed method is a hybrid scheme of key permutation method and the gene expression programming algorithm which combines the advantages of both genetic algorithms and genetic programming. Our experimental results have demonstrated that the proposed method improves the image quality and provides large message capacity and low computation time as well as increase in the system security.

Table 1: The results of embedding the secret images into the cover images (Baboon and Lena)

		Wang et al. method		Marghny et al. method		Proposed method	
Cover images	K	Simple- LSB	Optimal-LSB	Simple- LSB	Optimal-LSB	Simple- LSB	Optimal-LSB
Baboon	1	51.1415	51.1415	51.1380	51.1723	51.1407	51.1502
	2	44.0205	44.7440	44.0526	44.2475	44.0999	44.8577
	3	37.8642	38.7295	-	-	-	-
	4	31.3307	-	31.4595	32.5326	31.5813	33.4263
Lena	1	51.1299	51.1524	51.1471	51.1681	51.1395	51.1552
	2	44.0216	44.7638	44.0656	44.3714	44.0998	44.9118
	3	37.8626	38.7242	-	-	-	-
	4	31.2818	-	31.4258	32.2161	31.3268	33.2682

Table 2: The average computation time over 10-runs for different keys number 1, 2…10

Keys No.	Time/ PSNR	1-LSB	2-LSB	4-LSB
Key=1	Time (Second)	79.2603	92.6522	334.1168
	PSNR	51.0991	44.4291	32.0871
Key=2	Time (Second)	53.6722	56.533	254.6023
	PSNR	51.1045	44.4305	33.1409
Key=3	Time (Second)	32.8771	32.6586	139.2492
	PSNR	51.1121	44.4362	33.1604
Key=4	Time (Second)	22.5442	26.3473	98.3881
	PSNR	51.1129	44.4588	33.1638
Key=5	Time (Second)	21.8545	24.9661	86.157
	PSNR	51.1132	44.4696	33.1742
Key=6	Time (Second)	20.3083	19.9567	85.4114
	PSNR	51.1494	44.4803	33.2208
Key=7	Time (Second)	17.5523	19.4501	77.499
	PSNR	51.1523	44.4997	33.2268
Key=8	Time (Second)	17.4862	17.9843	74.6362
	PSNR	51.1641	44.5002	33.2686
Key=9	Time (Second)	15.8616	16.7294	73.1551
	PSNR	51.1678	44.5093	33.2944
Key=10	Time (Second)	13.9923	16.3988	62.5323
	PSNR	51.1742	44.5267	33.2992

Table 3: The results of optimal embedding when k=1,5and 10

Cover images	K	LSB	Cipher	Optimal GEP Key=1	Optimal GEP Key=5	Optimal GEP Key=10
Lena	1	51.1348	51.0282	51.0991	51.1132	51.1742
	2	44.1643	43.9321	44.4291	44.4696	44.5267
	4	32.1935	31.1285	32.0871	33.1742	33.2992
Baboon	1	51.1180	51.0581	51.0581	51.1301	51.1361
	2	44.1670	43.9103	44.4918	44.5819	44.7106
	4	32.2757	31.1376	32.9711	33.1108	33.4683
Barbara	1	51.1299	51.1174	51.1174	51.1522	51.1937
	2	44.1649	43.9379	44.3862	44.4336	44.4901
	4	32.2813	31.3588	32.2966	33.2019	33.2756
Pepper	1	51.1395	50.9953	51.0881	51.1446	51.1475
	2	44.1658	43.8887	44.1920	44.5133	44.5642
	4	32.2966	31.2432	31.9349	33.1992	33.3786

8. REFERENCES

[1] Fabien, Petitcolas, Anderson Ross, and Kuhn Markus. "Information Hiding - A Survey." Proceedings of the IEEE, special issue on protection of multimedia content 87, no. 7 (July 1999): 1062-1078.

[2] Simmons, Gustavus. "The Prisoner's problem and the subliminal channel in Advances in Cryptology." Proc.crypto , 1983: 55-67.

[3] Fridrich, Jessica, Miroslav Goljan, and Rui Du. "Detecting LSB Steganography in Color and Gray-Scale Images." IEEE Multimedia Special Issue on Security, October- November 2001: 22–28.

[4] Avcıbas, Ismail, Nasir Memon, and Bülent Sankur. "Steganalysis Using Image Quality Metrics." IEEE TRANSACTIONS ON IMAGE PROCESSING 12, no. 2 (FEBRUARY 2003): 221.

[5] Westfeld, Andreas, and Andreas Pfitzmann. "Attacks on Steganographic Systems - Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and Stools - and Some Lessons Learned." Lecture Notes in Computer Science, 1768, 2000: 61-75.

[6] Johnson, Neil, and Sushil Jajodia. "Exploring Steganography: Seeing the Unseen." IEEE Computer Mag. (George Mason Univ.), February 1998.

[7] Chan, Chi-Kwong, and Cheng. "Hiding data in images by simple LSB substitution." Pattern Recognition 37, no. 3 (2004): 469–474.

[8] Katzenbeisser, Stefan, and Fabien Petitcolas. "Information Hiding Techniques for Steganography and Digital Watermarking." Artech house, Inc., 2000.

[9] Ferreira, Cândida. "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems." Complex Systems 13, no. 2 (2001): 87-129.

[10] Ferreira, Cândida. Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence. 2nd Edn. Berlin Heidelberg: Springer-Verlag, 2006.

[11] Ferreira, Cândida. "Combinatorial Optimization by Gene Expression Programming: Inversion Revisited." In J. M. Santos and A. Zapico, eds., Proceedings of the Argentine Symposium on Artificial Intelligence, 2002: 160-174.

[12] Ferreira, Cândida. "Function Finding and the Creation of Numerical Constants in Gene Expression Programming." 7th Online World Conference on Soft Computing in Industrial Applications, 2002.

[13] Pohlheim, Hartmut. http://www.geatbx.com/docu/algindex-02.html (accessed May 15, 2011).

[14] Hsu, Ching-Sheng, and Shu-Fen Tu. "Finding optimal LSB substitution using Ant Colony Optimization Algorithm." Second International Conference on Communication Software and Networks. IEEE, 2010.

[15] Wang, Ran-Zan, Chi-Fang Lin, and Ja-Chen Lin. "Hiding data in images by optimal moderately significant-bit replacement." IEEE Electron. Lett. 36, no. 25 (2000): 2069–2070.

[16] Wang, Ran-Zan, Chi-Fang Lin, and Ja-Chen Lin. "Image hiding by optimal LSB substitution and genetic algorithm." Pattern Recognition, no. 34 (2001): 671-683.

[17] Li, Xiaoxia, and Jianjun Wang. "A steganographic method based upon JPEG and particle swarm optimization algorithm." Information Sciences (Elsevier Inc.), 2007: 3099–3109.

[18] Chang, Chin-Chen, Ju-Yuan Hsiaob, and Chi-Shiang Chan. "Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy." Pattern Recognition Society 36 (2003): 1583 – 1595.

[19] Wu, Ming-Ni, Min-Hui Lin, and Chin-Chen Chang. "A LSB Substitution Oriented Image Hiding Strategy Using Genetic Algorithms." AWCC 2004, LNCS 3309 (Springer-Verlag Berlin Heidelberg), 2004: 219–229.

[20] Forouzan, B.A. "Cryptography and Network Security." Tata. McGraw Hill, 2007.

[21] Chandrasekaran, Jeyamala, B. Subramanyan1, and Raman Selvanayagam. "A Chaos Based Approach for Improving Non Linearity in S Box Design of Symmetric Key Cryptosystems." CCSIT 2011, Part II, CCIS 132 (Springer-Verlag Berlin Heidelberg), 2011: 516–522.

[22] Mohamed, Marghny, Fadwa Al-Afari, and Mohamed Bamatraf. "Data Hiding by LSB Substitution Using Genetic Optimal Key-Permutation." International Arab Journal of e-Technology. 2, no. 1 (2011).

[23] Ferreira, Cândida. "Gene Expression Programming in Problem Solving." WSC6 tutorial, 2001.

[24] Khodaei, Masoumeh, and Karim Faez. "Image Hiding by Using Genetic Algorithm and LSB Substitution." ICISP 2010, LNCS 6134 (Springer-Verlag Berlin Heidelberg), 2010: 404–411.