Image Hiding by Using Genetic Algorithm and LSB Substitution

Masoumeh Khodaei[1] and Karim Faez[2]

[1] Islamic Azad University of Qazvin, Iran
m.khodaei@qiau.ac.ir
[2] AmirKabir University of Technology, Iran
kfaez@aut.ac.ir

Abstract. In this paper, we propose a new image hiding method by using LSB substitution for improving stego-image quality. In this method, we try to transform the secret image into meaningless picture by using a bijective mapping function so that the difference of embedded secret image bits and LSB bits of host image pixels is of minimum possible value. This operation results in the encryption of the secret image. Thus, if grabbers detect existence of the secret image into stego-image, they won't be able to recognize the secret image exactly. As a result, the security of our method will be increased. We use genetic algorithm for setting parameters of bijective mapping function to obtain the best condition in distribution of the pixels. We compare the our proposed method with LSB substitution method and Chang et al.'s method. The experimental results show that our proposed method has enhanced both the quality and the security of stego-image by using LSB substitution method. In addition, our method results are approximately close to the results of Chang et al.'s method.

Keywords: Cryptography, Genetic Algorithm, Image Hiding, Steganography.

1 Introduction

With expanding network and internet communications, steganography and cryptography methods have been increasingly important to raising the security of message sending between sender and receiver. Cryptography is used to protect important data against illicit access, while a steganographic method is used for hiding the secret message in the host data. The main goal of steganographic methods is to hide data in the host image with an acceptable decrease of image quality so that the quality of image containing the secret data is acceptable and the distortion of images would be imperceptible to the viewers. Secret data and host data can be text, image, video or audio. The image which the secret data will be embedded in it is called host image and the image with secret data embedded within is called stego-image and the encrypted image is called crypto-image.

In some papers, many techniques about data hiding have been proposed [1-4]. One of the most famous techniques is based on manipulating the least-significant-bits (*LSB*) of host image pixels by replacing the *LSB* bits of host image pixels with the

A. Elmoataz et al. (Eds.): ICISP 2010, LNCS 6134, pp. 404–411, 2010.

secret message bits. In 2001, Wang et al. [5] proposed a genetic algorithm based schema and *LSB* substitution to hide the secret image in the host image. In 2007, a method was proposed by Li and Wang [6] which was based on Particle Swarm Optimization algorithm (*PSO*) to enhance the quality of stego-image. In 2008, Chang et al. [7] presented a reversible steganographic method which used a genetic algorithm to find an approximate optimal common bitmap to embed the secret data in the image.

In this paper, we use an image as secret data. We try to encrypt the secret image by using genetic algorithm so that the difference between *LSB* bits of the host image pixels and the bits of the secret image pixels is of minimum value and also, the security of our proposed method will be increased by applying the secret image encryption.

This paper is organized as follows: Related works are introduced in Section 2. The proposed scheme is presented in section 3. Experimental results are illustrated in section 4, and the conclusions are reported and discussed in section 5.

2 Related Works

2.1 Image Hiding by LSB Substitution

We describe the concept of image hiding by *LSB* substitution which is presented by Wang et al [5]. Suppose we want to hide an image as secret image into an image as host image and resultant image called stego-image. All images are 8-bit gray-scale images. Size of host image H is k times the secret image S. The simplest method is to hide the bits of S in the least- significant-bits (*LSBs*) of H. This work can be done in three steps. In the first step, the method decomposes the pixel values of secret image S into several k-bit units and considers each k-bits unit as a single k-bits pixel. The resultant image is called S'. In the second step, k *LSB* bits are extracted from each pixel of host image H which is called R. Finally, it replaces S' with R in order to obtain the embedded image of H'.

Some additional works can be done to increase the security and the quality of stego-image H'. For this work, a bijective (i.e., one-to-one and onto) mapping function is presented to transform each pixel location in S' into a new random location. First, the pixel locations in S' are numbered sequentially from 0 to s-1, where s is number of pixels in S'. Next, the original location pixel x is transformed to a new location $f(x)$ by using the following bijective mapping function:

$$f(x) = (k_1 + k_2 \times x) \ mod \ \ s \tag{1}$$

and

$$gcd(k_2, s) = 1$$

Where k_1 and k_2 are two constants that are taken as keys, and $gcd(.)$ is the greatest common divisor. After location transforming, we get a meaningless image S'' and call it crypto-image. Then, we replace S'' into R and get stego-image H'. By using this method, the security of image hiding is increased because secret image is encrypted

and grabbers cannot analyze the embedded data in the stego-image. The following section will describe our method to obtain optimal crypto-image *S″*.

3 Proposed Method

In this section, we present the method for setting optimal value of k_1 and k_2 in eq. (1) so that the difference between *k*-bits crypto-image and *k LSB* bits of the host image is slight. This act results in increasing the quality of stego-image. We use genetic algorithm to find optimal value of k_1 and k_2 which is introduced in next sub-section.

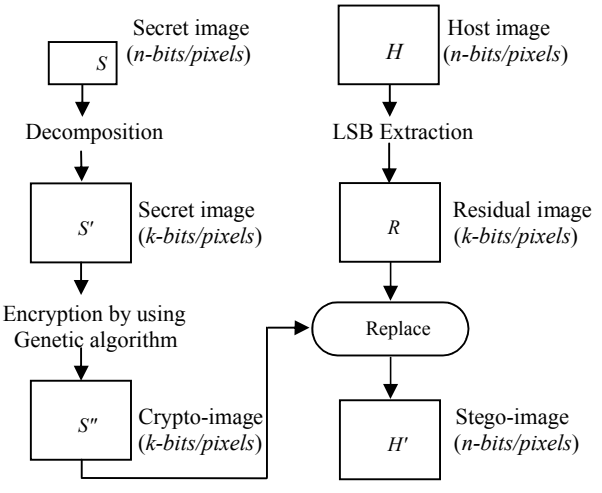


Fig. 1. The flowchart of embedding by proposed method

3.1 Secret Image Encryption by Using Genetic Algorithm

In our method, we want to find optimal value of k_1 and k_2 in eq. (1) by using genetic algorithm to encrypt the secret image so as to reduce the rate of changes resulted from embedding the secret image bits in the host image to enhance the quality of stego-image. Fig.1 illustrates the flowchart of our proposed method. According to Fig.1, we first decompose the pixel values of secret image *S* into several *k*-bit units and call it *S′*. Then, we extract *k LSB* bits of each pixel in the host image *H* and denote this image *R*. Now, we must encrypt *k*-bits secret image *S′*. To do this, we number the pixel locations in *S′* from 0 to *s*-1 sequentially, where *s* is number of pixels in *S′*. Next, the original location pixel *x* is transformed to a new location *f(x)* by using a bijective mapping function as previously mentioned in eq. (1).

Here, we describe genetic algorithm steps to find optimal value of k_1 and k_2 in eq. (1). The initial step in genetic algorithm is to code the solution space of problem as chromosomes such that a chromosome is an individual in a *GA*. There are many genes in a chromosome. We present the following step to find optimal value of k_1 and k_2 by genetic algorithm:

Step 1: Initial population Generation: Generate N chromosomes as initial population randomly. Suppose a chromosome has two genes. Define a chromosome G as follows:

$$G = g_1 g_2 \tag{2}$$

Where g_1 represents the k_1 value and g_2 represents the k_2 value.

Step 2: Calculation of fitness value: Set the value of g_1 and g_2 of each chromosome in eq. (1) to get new locations of k-bit units in S' and call it S''. Then, calculate fitness value of each chromosome to judge the goodness of chromosomes by

$$Fitness = \frac{1}{\sum_{i=1}^{m} (R_i - S_i'')^2} \tag{3}$$

Where m is the image size of R and S''**.**

Step 3: Parents selection: Select two chromosomes among the population as parents by proportional to fitness value. i.e., the chromosomes with more fitness value would have more chance of selection.

Step 4: Performing crossover operator: Suppose two chromosomes $G_1=p_1p_2$ and $G_2=q_1q_2$ are chosen as parents. Perform crossover operator on G_1 and G_2 with probability of P_c and produced two offspring G'_1 and G'_2 by using fixed-point crossover according to Fig.2. Divide G_1 and G_2 into two parts and exchange two parts of them to get $G'_1=q_2p_2$ *and* $G'_2=q_1p_1$. Now, check the validation of two Chromosomes G'_1 and G'_2. The chromosome will be valid which contains $gcd(p_1,s)=1$ or $gcd(p_2,s)=1$. If an offspring isn't valid, transmit the fitted parent in new population.

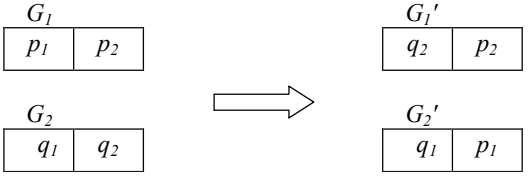


Fig. 2. Crossover operation on chromosomes G_1 and G_2

Step 5: Performing mutation operator: Perform mutation operator on Chromosome $G'=q_1q_2$ with probability of P_m. to do this, generate two random numbers in ranges [0, 10000] and add them to the values of q_1 and q_2 of Chromosome G' to get G''. Then, check the validation of Chromosome. The chromosome will be valid that the greatest common divisor of q_2 and s is 1. If G'' isn't valid, transmit G' in new population.

Step 6: Repeat steps (3-5) until N new individuals are generated.

Step 7: Survivors selection: Select all of produced offspring as new population.

Step 8: Repeat steps (2-6) until it reaches the termination condition. It can be either by reaching the predefined number of generations or reaching the given fitness value.

After terminating *GA*, we choose a chromosome that has greatest fitness value between current populations. The selected chromosome represents the optimal value of k_1 and k_2.

3.2 Embedding Secret Image into Host Image

After finding the optimal value of k_1 and k_2, we set this values in eq. (1) to get new locations of *k*-bits units in *S′* and obtain image *S″*. Then, we replace image *S″* into image *R* and get stego-image *H′*. By using this method, the stego-image quality and the stego-image security is increased. During stego-image sending, we should send the values of k_1 and k_2 with stego-image for receiver to decrypt the crypto-image and get the original secret image.

4 Experimental Results and Analysis

In this section, we present the experimental results of the proposed method. We use four 8-bits gray-level images Lena, Baboon, Jet and Elaine that are shown in Fig.3 as host images in our experiments. Moreover, three images Boat, House and Tank with size 128×128 are used as secret images that are shown in Fig.4. The size of all host images are 256×256. We utilize the peak signal-to-noise ratio (*PSNR*) to evaluate the quality of stego-images. Great *PSNR* value shows that quality of stego-image is high. Theoretically, the *PSNR* value between *H* and *H′* can be calculate by

$$PSNR = 10 \times Log\left(\frac{(255)^2}{MSE}\right) \tag{4}$$

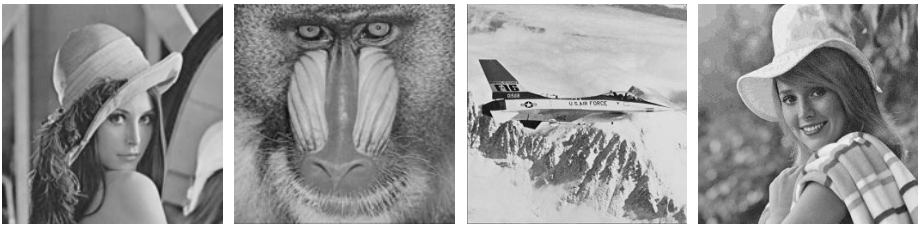
Where *MSE* is defined by

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(H_i - H'_i)^2 \tag{5}$$

Here, *m* is the size of image *H* and *H′*. In all experiments, number of *k*-bits for embedding into host image pixels is equal to 4 and the size of population *N* is 30. Also, we assume that the probability value P_c of crossover operation is 0.8 and the probability value P_m of mutation operation is 0.2.

We use eq. (6) to calculate the similarity measure between the secret image and the crypto-image [8]:

$$r = \frac{\sum_{i=1}^{MN}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{M}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{6}$$

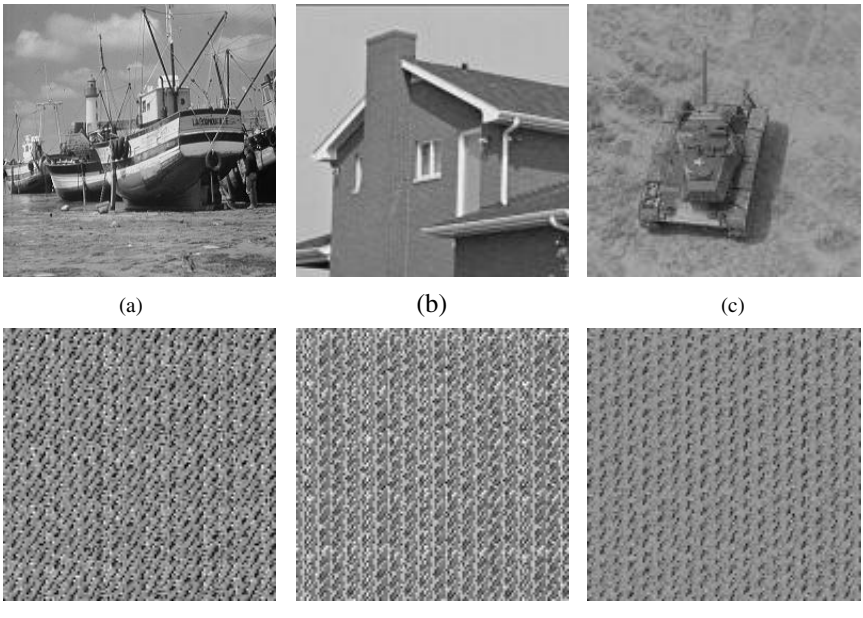


(a) (b) (c) (d)

Fig. 3. Four 256×256 host images: (a) Lena, (b) Baboon, (c) Jet and (d) Elaine

Where x_i is the value of the pixels in the secret image and y_i is value of the pixels in the crypto-image and $\bar{x}$ is the mean value of x_i and $\bar{y}$ is the mean value of y_i. Also, M is the size of secret image and N is the size of crypto-image.

The results of our proposed method are presented in Table 1, 2, 3. These tables show that *MSE* and *PSNR* values given by the proposed method is better than the simple *LSB* method and is approximately close to the values that are derived from Chang et al.'s method [2].



(a) (b) (c)

(d) (e) (f)

Fig. 4. Three 128×128 secret images: (a) Boat, (b) House and (c) Tank. Three 128×128 crypto-images obtained by the proposed method: (d) Crypto-Boat with k_1=4753 and k_2=227 and r=0.001863, (e) Crypto-House with k_1=43 and k_2=911 and r=0.003287 and (f) Crypto-Tank with k_1=27 and k_2=101 and r=0.006395.

However, this rate of increase in the stego-image quality may appear to be slight, but our proposed method causes the growth of the security due to the encryption of the secret image in image steganographic method. Therefore, if grabbers can recognize that the host image contain a secret-image, they won't understand the secret image easily unless they are aware of k_1 and k_2 values.

Fig.4 shows three encrypted secret images by using the proposed approach. Value r represents similarity measure and correlation between the secret image pixels and the crypto-image pixels. When the rate of similarity in r is less, it means that the crypto-image is encrypted suitably and indistinguishably. The obtained values of r are small. Thus, the secret images are encrypted satisfactorily and imperceptibly by using our method and this process causes to increase security of image hiding method.

Table 1. The results of embedding Boat image as secret image by proposed and LSB and Chang et al.'s [2] methods

Host Image	MSE			PSNR		
	LSB method	Chang method	Proposed method	LSB method	Chang method	Proposed method
Lena	18.74	15.62	16.92	35.40	36.19	35.84
Baboon	17.83	14.28	15.78	35.61	36.58	36.14
Jet	17.47	15.12	16.01	35.70	36.15	36.08
Elaine	18.10	15.71	16.24	35.55	36.16	36.02

Table 2. The results of embedding House image as secret image by proposed and LSB and Chang et al.'s [2] methods

Host Image	MSE			PSNR		
	LSB method	Chang method	Proposed method	LSB method	Chang method	Proposed method
Lena	18.36	16.27	17.01	35.49	36.01	35.82
Baboon	17.51	14.70	15.89	35.69	36.45	36.11
Jet	17.32	14.33	15.47	35.74	36.56	36.23
Elaine	17.55	13.93	16.09	35.68	36.69	36.06

Table 3. The results of embedding Tank image as secret image by proposed and LSB and Chang et al.'s [2] methods

Host Image	MSE			PSNR		
	LSB method	Chang method	Proposed method	LSB method	Chang method	Proposed method
Lena	17.15	15.32	16.21	35.78	36.27	36.03
Baboon	16.64	12.91	14.73	35.91	37.02	36.46
Jet	16.36	13.08	14.51	35.78	36.94	36.51
Elaine	16.55	13.67	14.48	35.93	36.77	36.53

5 Conclusions

In this paper, we proposed an image hiding method to hide the secret image into host image. In this approach, we used genetic algorithm to encrypt the secret image so that

the rate of changes resulted from embedding the secret image bits into *LSB* bits of the host image is of minimum value. Moreover, even if the grabbers get to know that the stego-image contains the secret image, they won't be able to recognize the original secret image easily unless they know the values of k_1 and k_2. Consequently, the security of image steganographic method is increased. The experimental results of our proposed method showed that the stego-image quality is better than the stego-image resultant from simple *LSB* substitution method and is approximately close to the results of Chang et al.'s method.

References

1. Chang, C.-C., Hsiao, J.-Y., Chan, C.-S.: Finding optimal least- significant-bit substitution in image hiding by dynamic programming strategy. Pattern Recognition 36, 1583–1595 (2003)
2. Chang, C.-C., Chan, C.-S., Fan, Y.-H.: Image hiding scheme with modulus function and dynamic programming strategy on partitioned pixels. Pattern Recognition 39, 1155–1167 (2006)
3. Wang, R.-Z., Tsai, Y.-D.: An image-Hiding method with high hiding capacity based on best-block matching and k-means clustering. Pattern Recognition 40, 398–409 (2006)
4. Chang, C.-C., Lu, T.-C.: Lossless Information Hiding Scheme Based on Neighboring Correlation. International Journal of Signal Processing, Image Processing and Pattern (2009)
5. Wang, R.Z., Lin, C.F., Lin, J.C.: Image hiding by optimal LSB substitution and genetic algorithm. Pattern Recognition 34, 671–683 (2001)
6. Li, X., Wang, J.: A Steganographic method based upon JPEG and Particle Swarm optimization algorithm. Pattern Recognition 177, 3099–3109 (2007)
7. Chang, C.-C., Lin, C.-Y., Fan, Y.-H.: Lossless Data hiding for color images based on block truncation coding. Pattern Recognition 41, 2347–2357 (2008)
8. AI-Taani, A.T., AL-Issa, A.M.: A Novel Steganographic Method for Gray-Level Images. International Journal of Computer and System Science and Engineering (2009)