Genetic Algorithm Based Imperceptible Spatial Domain Image Steganography Technique with High Payload Capacity

Pratik D. Shah, Rajankumar S. Bichkar

***Abstract:** Data security is a very important factor in any form of digital communication. Steganography can be used to enhance the security of digital communication. There are various methods to perform steganography on digital images, but very few deal with increasing imperceptibility and data embedding capacity together. In this paper, a high data embedding capacity spatial domain image steganography scheme is proposed which is highly imperceptible. In the proposed technique steganography is modeled as a search and optimization problem and Genetic algorithm (GA) is used to solve this problem to find the near-optimal solution. Optimal pixel adjustment procedure (OPAP) is further used to improve the quality of stego-image. Experimental results exhibited that the proposed technique provides an improvement in imperceptibility of stego-image at high data embedding rate when compared to several other popular steganography techniques. The average PSNR value of various stego-images at two bit per pixel data embedding rate was 46.39.*

***Index Terms:** Genetic algorithm (GA); Image steganography; Information hiding; Spatial domain; Steganalysis*

I. INTRODUCTION

The technique in which secret data is embedded in digital media, to hide the existence of confidential communication is called as Steganography [1]. Image steganography is very popular since images have an enormous amount of redundancy, which can be effectively used to hide secret data in it, without being noticed [2]. The secret data is embedded in cover image and the resultant image obtained after inserting the secret data is called stego-image [3]. The main aim of image steganography is to conceal the existence of secret communication; it is done by reducing the difference between stego-image and the cover image. Mostly, four parameters are used to evaluated the performance of image steganography they are robustness, imperceptibility, payload capacity and security [1]. Imperceptibility represents similarity of stego-image with the cover image. Payload capacity is also known as embedding capacity or data hiding capacity. It is used to describe the amount of secret data which can be hidden inside the cover image. Robustness represents the ability of a steganography technique to resist any image manipulation attacks. Security is the parameter which exposits the capability of steganography technique to resist the steganalysis attacks [2]. Steganalysis is a process striving to detect the presence of hidden data [4].

Image steganography methods are mostly classified as spatial domain techniques and frequency domain or transform domain techniques [5]. In spatial domain techniques, image pixel intensities are directly manipulated to hide the secret message [6]. Frequency domain techniques don’t operate on pixel intensities directly; they first convert the image into the frequency domain using various transforms like Discrete cosine transform, Discrete Fourier transform, Discrete Wavelet transform, etc. [5]. Then the secret message is embedded by modifying the coefficients in the corresponding transform domain [5]. Spatial domain algorithms usually have a better visual quality of stego-images; however, their performance against statistical steganalysis attacks is poor [7]. Spatial domain techniques might have good data embedding capacity, but it has an adverse effect on the visual quality of stego-image. Transform domain methods are good against statistical steganalysis attacks as the secret data is spread across the entire image through frequency domain coefficients. However, these techniques have small payload capacity and imperceptibility is lower [3].Significant amount of work has been carried out in steganography; both in spatial as well as transform domain, but limited work is presented on the use of evolutionary computation in image steganography. Wang et al. [8] proposed a low capacity steganography technique to resist RS steganalysis. This technique is developed for spatial domain; the secret data is hidden in 1st LSB of the cover image and the 2nd LSB is modified so that it can undo or reduce the changes incurred due to secret data embedding process.In this technique genetic algorithm is used to find the possible values of 2nd LSB so that the stego-image image can resist the RS steganalysis attack. Bedi et al. [5] presented a Particle Swarm Optimization (PSO) based spatial domain image hiding scheme. This technique uses PSO to find the best pixel locations in the cover image to hide the secret data. The focus of this technique was on improving both, quality and robustness of the stego-image. Kanan and Nazeri [7] proposed a spatial domain image steganography technique. They have modeled steganography as a search and optimization problem in which GA was used to find various possibilities to embedded secret data in cover image. These possibilities include exploring different starting point, different embedding order, different pixel combinations, etc. to embedded secret data in cover image. Maheswari and Hemanth [9] presented a GA and PSO based transform domain image steganography scheme.

Revised Manuscript Received on 30 January 2019.
* Correspondence Author

Pratik D. Shah*,Research Scholor Dept. of E&TC, G. H. Raisoni College of Engineering and Management, Assistant Professor Dr. D. Y. Patil School of Engineering, SavitribaiPhule Pune University, Pune 412207, India.

Rajankumar S. Bichkar,Principal, VidyaPratishthan'sKamalnayan Bajaj Institute of Engineering and Technology, Baramati, SavitribaiPhule Pune University, Pune, India.

For image transformation Contourlet transform and Fresnelet transform are used and GA and PSO are used to find best the coefficients to embed the secret data.

In this paper, we propose a GA based spatial domain image steganography technique with high data embedding capacity and imperceptibility. In the proposed technique, two bit of secret data is embedded in each pixel of cover image, resulting in a 25% data embedding capacity. The proposed technique is a modified version of the technique developed by Kanan and Nazeri [7]. In the proposed technique we embed the modified secret data in LSB's of the cover image using LSB replacement steganography, but the data is not inserted sequentially. Linear Congruential Generator (LCG) is used to find the sequence in which the binary stream of secret data is to be embedded in cover image and GA is used to modify the parameters of LCG. The secret data is modified before embedding it in the cover image, the modification is done in two ways. First the data polarity is modified based on value of the chromosome and finally the arrangement of two secret bits are also altered based on the value of chromosome. After embedding secret data in cover image OPAP is used to further improve the quality of stego-image. For experimentation, standard test images are chosen and the performance of the proposed technique is compared with several other popular steganography techniques. PSNR and MSE parameters are used to evaluate the performance of the proposed technique with other techniques for same combinations of cover image and secret data image.The main contribution of this paper is to use genetic algorithm for modifying the parameters of LCG to generate best sequences to hide secret data in cover image. We also propose an algorithm to modify the LCG so that is will generate perfect, non-repetitive sequence. The idea behind using random sequence for choosing the order in which secret data is hidden is intuitive. Since a particular order of data insertion in LSB's may result in more matching between LSB of cover image and secret data bits, the resultant stego-image will be more imperceptible, as the corresponding changes in stego-image will be less. The value of cover image pixels and secret data is binary i.e. either 1 or 0, so there is a possibility of a sequence which may provide cent percent match between both. It is very difficult because there will be more than a billion permutations for that sequence. Hence we employ genetic algorithm to do the tedious and almost impossible work of finding the sequence which may give us least possible changes in cover image. This will help to increase the data embedding capacity without compromising on the imperceptibility of the stego-image. The rest of the paper is organized as follows: Section 2 describes the proposed technique in detail along with genetic algorithm, LCG and OPAP. Section 3 presents the experimental results and discussion; finally paper is concluded in section 4

II. PROPOSED TECHNIQUE

As discusses in the previous section, the main aim of this technique is to find the best places in the cover image to embed modified secret data. This is done by finding a sequence which can generate maximum similarity between data to be embedded and LSB's of the cover image. For generation of sequences we use LCG, but for an image of size 256 × 256, we have 65536 factorial combinations i.e. 5.16×10^{287193}. Exploring the possibility of these many combinations for embedding data is impossible hence we employ genetic algorithm to do this impossible task for us and find a near optimal solution. Genetic algorithm is used to control the parameters of LCG. To understand the proposed technique knowledge of a few important preliminary concepts is required; these concepts will be discussed in the following sub-sections. LCG is discussed in section II.A. LCG has few drawbacks as it cannot always generate perfect non-repetitive sequences hence we propose a new modified LCG algorithm in section II.B. In section II. C. the chromosome structure is explained. Optimal pixel adjustment procedure is explained in section II.D. In section II.E.the process of embedding secret data in a cover image is explained and section II.F.explains the process of extracting the secret data from the stego-image. The section II.G.describes various GA parameters.

A. Linear Congruential Generator

LCG algorithm is used to generate a pseudo-random sequence of $(X_1, X_2 ..., X_M)$ over an interval $[0, M\text{-}1]$ as shown in equation 1. To generate a pseudo-random sequence of M numbers, LCG requires a seed value, a multiplying factor and an offset value [10].

$$X_{n+1}= (a \times X_n+ c) \bmod M \tag{1}$$

Where $X_{n+1}$is the next integer value of the pseudo-random sequence, X_n is the present integer value or seed value in the initial state, a is the constant multiplying factor, c is an offset value and M is the sequence length [11]. The assumptions for the equation 1 to function properly is that the value of a, c, $X_0<M$ and $M>0$.

As the selection of a, c, M and X_0values affects the length of the random sequence, care is required to select all these parameters. For example for values a=5, c=1, M=8 and X_0=5 the resultant sequence obtained will be 2, 3, 0, 1, 6, 7, 4, 5 which consist of all possible values in the range 0-7, after it the same sequence repeats itself. For another example assume a=2, c=3, M=7 and X_0=1 the resultant sequence obtained is 5, 6, 1 after it the numbers repeats themself. From the above example, it is clear that to obtain all the values from 0 to $M\text{-}1$ of M length sequence considerable precaution is required for selecting the LCG parameters. To avoid this problem, we have proposed a modified LCG, which will generate perfect non-repetitive sequences. The modified LCG algorithm is discussed in the next sub-section

B. Modified LCG

In this approach, the equation used to generate pseudo-random sequence is the same as the one used for LCG. However the result obtained after each iteration is processed to avoid repetition of values. To generate the sequence, any positive integer value can be selected for M and the values of a, c, X_0 can be picked from 0 to $M\text{-}1$. In this method if a value is repeated then next value is used and a flag is maintained to avoid repetition. The algorithm for this approach is given below.

```
Step 1: Declare & initialize values of $M, a, c, X_0$ and $Seq[M]$
Step 2: Declare & initialize array $flag[M] \leftarrow 0$
Step 3: $flag[X_0]=1$, $Seq[1]=X_0$
Step 4: for $i=2$ to $M$
```

```
        Generate sequence using equation
        $X_i= (a \times X_{i-1}+ c) \bmod M$
        if(flag[$X_i$]!=1)
     flag[$X_i$]=1
              Seq[i]=$X_i$
        else
              k= $X_i$+1
              forj=1 to M
   if (k<=M)
        if (flag[k]!=1)
        flag[k]=1;
        Seq[i]=k;
         break;
        else
        k=k+1;
              else
       k=1;
```

C. Chromosome Structure

The chromosome structure of proposed technique consists of three genes as shown in Figure 1.

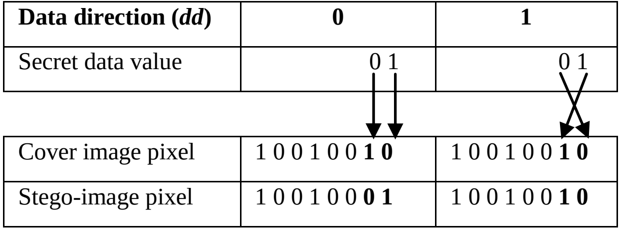
X_o	a	C	dd	dp

Fig. 1: Chromosome structure

Each gene in the proposed chromosome is of different lengths as illustrated in Figure 1. The first gene is comprised of X_o, *a* and*c* each consists of 18 bits and is used to control the sequence generated by LCG. The second gene i.e. *dd* is of 1 bit length and it is used to decide the direction of secret data embedding. The third gene i.e. *dp* is of 2 bits in length and it is very crucial as it decides the polarity of secret data to be hidden in the cover image pixel. The effect of *dd*and *dp* is explained in detail with an example in Figure 2 and Figure 3

i. Data direction (*dd*)

The proposed technique is developed to embed two bit of secret data in each pixel of cover image. Hence there are two possible combinations to embed secret data in LSB's. The data direction (*dd*) gene is used to control this parameter. Consider '01' as the two bits of secret data to be embedded in cover image pixel whose value is 146 (10010010), the two possibilities are as shown in figure 2.



Data direction (*dd*)	**0**	**1**
Secret data value	0 1	0 1
Cover image pixel	1 0 0 1 0 0 **1 0**	1 0 0 1 0 0 **1 0**
Stego-image pixel	1 0 0 1 0 0 **0 1**	1 0 0 1 0 0 **1 0**

Fig. 2: Effect of *dd* on secret data embedding procedure for constant value of *dp*=11

ii. Data polarity (*dp*)

In the proposed technique we modify the secret data bits before embedding them in cover image pixels, the modification is done based on the value of data polarity gene. The data polarity gene is of two bits. If the value of *dp* is 11, then no change is done to the secret data. If the value of *dp* is 00, then both the secret data bits are complemented. Similarly, if the value of *dp* is 01 or 10, then one bit of secret data is complimented and the other is kept unchanged. Now consider '11' as the two bits of secret data to be embedded in cover image pixel valued 146 (10010010), the possibilities due different values of *dp* are as shown in Figure 3.

Data Polarity	00	01	10	11
Secret data value	11	11	11	11
Cover image pixel	100100**10**	100100**10**	100100**10**	100100**10**
Stego-image pixel	100100**00**	100100**01**	100100**10**	100100**11**

Fig. 3: Effect of *dp* on secret data embedding procedure for constant value of *dd*=0

D. Optimal Pixel Adjustment Process

OPAP is applied to stego-image obtained after data embedding. OPAP is used to reduce the difference between the pixel values of cover image and stego-image. Chan and Cheng [12] developed the idea of OPAP to improve the quality of stego-image after LSB steganography. The elementary concept of OPAP is as follows:

- Let p_i, p_i' *and* p_i'' be the corresponding pixel values of the i^{th} pixel in the original cover image *C*, the stego-image *C'* obtained by LSB replacement steganography and *C''* the modified stego-image obtained after the OPAP.
- Let $\partial_i= p_i' - p_i$ be the embedding error between p_i and p_i' obtained after embedding *k* bits of secret message per pixel.
- Therefore $-2^k< \partial_i< 2^k$
- The value of ∂_i can be further divided into three intervals:
  Interval 1: $2^{k-1}< \partial_i< 2^k$,
  Interval 2: $-2^{k-1} \le \partial_i \le 2^{k-1}$,
  Interval 3: $-2^k< \partial_i< -2^{k-1}$
- Depending upon these three intervals OPAP modifies p_i' to p_i''.
  Case 1: ($2^{k-1}< \partial_i< 2^k$): If $p_i' \ge 2^k$, then $p_i''= p_i' - 2^k$; otherwise $p_i''= p_i'$
  Case 2: ($-2^{k-1} \le \partial_i \le 2^{k-1}$): $p_i''= p_i'$
  Case 3: ($-2^k< \partial_i< -2^{k-1}$): If $p_i' < 256 - 2^k$, then $p_i''= p_i' + 2^k$; otherwise $p_i''= p_i'$
  From the above discussion it is clear that OPAP modifies stego-image only if the embedding error ∂_i is greater than 2^{k-1} or less than -2^{k-1}. The proposed technique is developed for 2 bit per pixel data embedding hence OPAP will modify only those pixels whose ∂_i is 3 or -3.

E. Embedding the Secret Data

In this subsection, the process of embedding the secret data in cover image is discussed. The data insertion is done by LSB replacement technique. Two bits of secret data are embedded in each pixel of the cover image. The secret data is converted into two arrays which are one dimensional. One array has 4 MSB's of all secret data image pixels while another array has 4 LSB's. One bit of data from each array is inserted in the cover image using LSB replacement steganography. The order in which these secret data bits are inserted is dependent on the sequence generated by LCG.

The parameters which affect the sequence generated by LCG are obtained from GA chromosome. After obtaining the sequence the data insertion is carried as per the sequence. While performing data insertion the secret data bits are first modified as per the value of *dd* and *dp* gene of GA chromosome. The algorithm for embedding secret data in cover image is given below.**Input**: Cover image $C = \{c_1, c_2,\ldots,c_{m\times n}\}$, Secret message image
$M = \{m_1, m_2,\ldots, m_{(m/2)\times(n/2)}\}$
Output: Stego-image $S = \{s_1, s_2,\ldots,s_{m\times n}\}$

1. Create two, one dimensional bit arrays *sm1* and *sm2* of length $len=m\times n$.
2. Insert four MSB's of all pixels from *M* in consecutive locations of *sm1* and similarly insert four LSB's of all pixels from *M*in consecutive locations of *sm2*.
   $sm1 \leftarrow (M)_{8\text{-}5}$
   $sm2 \leftarrow (M)_{4\text{-}1}$
3. Initialize population size *p* and number of iterations *iter*.
4. Initialize *p* chromosomes randomly. A chromosome consists of x_0, *a*, *c*, *dp*and *dd*. Where x_o, *a* and *c* are parameters for sequence generation through LCG, *dp is* polarity of secret data and *dd* is direction of secret data.
5. Generate *p* sequences *(seq)* of length *len* using values of X_o, *a* and *c* obtained from chromosomes. Use the modified LCG algorithm for generation of sequence.
6. Embed two bit of secret data in LSB's of cover image based on the values of *dp*, *dd* and sequence *(seq)* obtained from the particular chromosome.
   For each chromosome in the population initialize $S \leftarrow C$

```
count ← 1
for i=1 to m
  for j=1 to n
  ifdd==0
$S7(i,j) = \overline{dp1}\cdot\overline{sm1(seq(count))} + dp1\cdot sm1(seq(count))$
$S8(i,j) = \overline{dp2}\cdot\overline{sm2(seq(count))} + dp2\cdot sm2(seq(count))$
  else
$S7(i,j) = \overline{dp2}\cdot\overline{sm2(seq(count))} + dp2\cdot sm2(seq(count))$
$S8(i,j) = \overline{dp1}\cdot\overline{sm1(seq(count))} + dp1\cdot sm1(seq(count))$
count = count + 1;
```

7. Apply OPAP to stego-image obtained after step 6.
8. Calculate fitness of each population using fitness function.
9. Apply genetic algorithm operators: reproduction, mutation and crossover to generate new population.
10. Repeat step 4 to step 8 *iter*number of times.
11. Hide the chromosome with the best fitness value as the secret key in predefined pixel locations

F. Extracting the Secret Data

The process of extracting the secret data from stego-image starts with first getting GA chromosome from the predefined locations of stego-image. After getting chromosome, next step is generating sequence used to embed the secret data, it can be done using LCG and parameters required for it can be obtained from GA chromosome. After obtaining sequence each pixel is visited as per sequence generated by LCG and two bits of secret data are extracted from each pixel. The data extracted is modified as per the values of *dd* and *dp* gene of GA chromosome and stored in two separate arrays. These arrays are finally used to reconstruct the secret data image. The algorithm for extracting secret data from stego-image is given below.
Input: Stego-image $S = \{s_1, s_2,\ldots,s_{m\times n}\}$
Output: Secret message image $M = \{m_1, m_2,\ldots, m_{(m/2)\times(n/2)}\}$

1. Extract secret key from predefined pixels locations of stego-image.
2. Obtain the value of X_0, *a* and*c* from the secret key.
3. Generate a sequence of *len* numbers using modified LCG algorithm.
4. Initialize two, one dimensional bit arrays of length *len*as *se1*and *se2* to extract and store secret data image.
5. Extract two bits of data from LSB's of each pixel and store them in *se1* and *se2* depending upon the value of *seq*, *dp* and *dd.*
6. Concatenate four bits of *se1* and *se2* to form one pixel of the secret image. Continue this process till *m/2* × *n/2* pixels are obtained.

The extraction of secret data from the stego-image is extremely difficult for eavesdroppers. To extract the secret data, first the secret key needs to be mined and further, the process of using secret key to generate sequence and extract data based on *dd* and *dp* is necessary. Hence data extraction can only be done by person who has prior knowledge of the embedding technique.

G. Genetic Algorithm Parameters

Genetic algorithm is a metaheuristic technique which is used to reach optimal and near optimal solutions [11]. Genetic algorithm is based on Darwin's theory of evolution and it simulates the evolution of biological being. In the proposed technique the initial value of chromosomes or population used for GA is generated randomly. The size of population is set to fifty. This population will compete with each other for their survival and seeding next generation. Each individual in the population will be evaluated based on their fitness. The fitness function used for the proposed technique is the PSNR value of stego-image. Reproduction, mutation and crossover operators are used to obtain the population for next generation. A batch of five chromosomes is selected randomly and tournament selection competition is used to pick two chromosomes to seed next generation. This step is repeated till all chromosomes are evaluated. The process of finding new solutions is explored till 500 iterations.

III. RESULTS AND DISCUSSIONS

For experimentation, standard test images were chosen and the performance of the proposed technique was compared with several other popular steganography techniques. The performance of various techniques was evaluated using PSNR parameter for the same combinations of cover and secret data images. The resolution of cover image is 512×512 and that of secret data image is 256×256. Both the images are grey scale images. Software tool used to implement the proposed technique was Matlab 8.1.

Equation 2 was used to find the MSE of stego-images, in equation 2 M and N represents number of rows and columns in the image respectively. X_{ij} and Y_{ij} are pixel values of ij^{th}location of original image and stego-image respectively. Equation 3 was used to find the PSNR value of stego-images.

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{ij} - Y_{ij})^2 \qquad (2)$$

$$PSNR = 10 \cdot log_{10}\frac{(255)^2}{MSE} \qquad (3)$$

In Figure 4 all the test images used for experimentation are displayed.

Lena, Jet, Pepper, Sailboat and Baboon are the images used as cover image and Figure 4(f) a test pattern is used as secret data image.

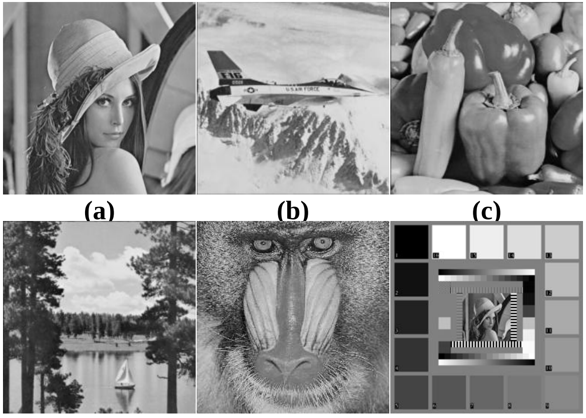


Fig. 4: Test images (a) - (e) Cover images (Lena, Jet, Pepper, Sailboat and Baboon). (f) - Secret message image (Test Pattern)

A. Imperceptibility Analysis

Imperceptibility analysis measures the changes incorporated during the process of embedding secret data in cover image. In imperceptibility analysis, we compare the cover image and the stego-image using Peak signal to noise ratio (PSNR). The test images used to compare the performance of the proposed technique with other popular steganography schemes are same. In imperceptibility analysis higher PSNR value ensures better performance of the steganography technique, as it guarantees a superior visual quality of stego-image.Table 1 displays PSNR values of stego-images obtained from the proposed technique and various other steganography techniques at same data embedding rate i.e. 2 bits per pixel. These results undoubtedly advocate the dominance of the proposed technique over other techniques. Hence we can state that the proposed scheme is exceedingly imperceptible. Figure 5 demonstrates the PSNR value of stego-images obtained from the proposed and existing spatial domain steganography technique. The comparison between cover image and stego-image obtained from the proposed technique for Lena and Jet image is shown in figure 6.

Table 1: PSNR values of the proposed technique and various other steganography methods for different stego-images

Sr.	Cover image	Lin and Tsai [13]	Yang et al. [14]	Chang et al. [15]	Wu et al. [16]	Kanan et al. [7]	Proposed technique
1	Lena	39.20	41.60	40.37	43.54	45.12	46.38
2	Jet	39.25	41.66	40.73	43.53	45.18	46.41
3	Pepper	39.17	41.56	39.30	43.56	45.13	46.38
4	Sailboat	39.18	41.51	38.86	43.55	45.10	46.39
5	Baboon	39.18	41.55	39.94	43.54	45.12	46.39

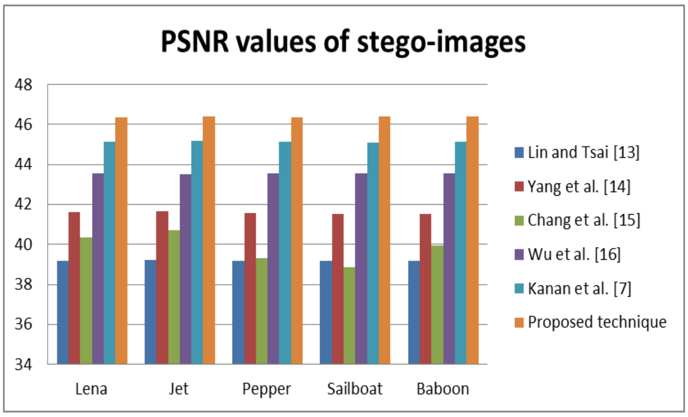


Fig. 5: PSNR values of the proposed technique and various other steganography methods for different stego-images

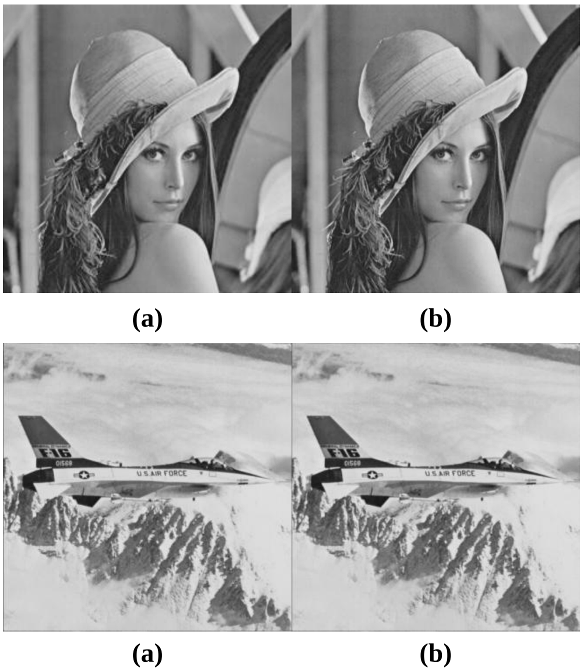


Fig. 6(a): Original image Lena and Jet. Fig. 6(b):Stego-image obtained after secret data embedding from proposed technique

IV. CONCLUSION

In this paper an imperceptible, high payload capacity, spatial domain image steganography technique is presented.

The proposed scheme employs genetic algorithm to find the best locations and order to embed secret data in cover image. The results of proposed technique are compared with various other popular spatial domain steganography schemes. The superiority of the proposed technique is quite evident by the obtained results, both in subjective and objective analysis. The proposed method produces very small amount of change in stego-image, making it highly imperceptible and immensely challenging to identify the presence of steganography by visual inspection. The extraction of secret data from the stego-image is extremely difficult for eavesdroppers as it requires the knowledge of secret key and the process used for secret data extraction

REFERENCES

1. Cheddad A, Condell J, Curran K &McKevitt P, “Digital image steganography: Survey and analysis of current methods”, *Signal processing*, Vol. 90, No. 3, (2010), pp. 727-752.
2. Subhedar MS &Mankar VH, “Current status and key issues in image steganography: A survey”, *Computer science review*, Vol. 13, (2014), pp. 95-113.
3. Al-Dmour H & Al-Ani A, “A steganography embedding method based on edge identification and XOR coding”, *Expert systems with Applications*, Vol. 46, (2016), pp. 293-306.
4. Ker AD, “Steganalysis of LSB matching in grayscale images”, *IEEE signal processing letters*, Vol. 12, No. 6, (2005), pp. 441-444.
5. Bedi P, Bansal P &Sehgal P, “Using PSO in a spatial domain based image hiding scheme with distortion tolerance”, *Computers & Electrical Engineering*, Vol. 39, No. 2, (2013), pp. 640-654.
6. Li B, He J, Huang J & Shi, YQ, “A survey on image steganography and steganalysis”, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 2, No. 2, (2011), pp. 142-172.
7. Kanan HR &Bahram N, “A novel image steganography scheme with high embedding capacity and tunable visual image quality based on a genetic algorithm”, *Expert Systems with Applications*, Vol. 41, No. 14, (2014), pp. 6123-6130.
8. Wang S, Yang B &Niu X., “A secure steganography method based on genetic algorithm” *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 1, No. 1, (2010), pp. 28-35.
9. Maheswari SU &Hemanth DJ, “Performance enhanced image steganography systems using transforms and optimization techniques”, *Multimedia Tools and Applications*, Vol. 76, No. 1, (2017), pp. 415-436.
10. Fontaine C, “Linear congruential generator”, *Encyclopedia of Cryptography and Security, Springer, Boston, MA*, (2011), pp. 721-721.
11. Shah PD &Bichkar RS, “A Secure Spatial Domain Image Steganography Using Genetic Algorithm and Linear Congruential Generator”, *International Conference on Intelligent Computing and Applications, Advances in Intelligent Systems and Computing, Springer, Singapore*, (2018), pp. 119-129.
12. Chan C & Cheng LM, “Hiding data in images by simple LSB substitution”, *Pattern recognition*, Vol. 37, No. 3, (2004), pp. 469-474.
13. Lin C & Tsai W, “Secret image sharing with steganography and authentication”, *Journal of Systems and software*, Vol. 73, No. 3, (2004), pp. 405-414.
14. Yang C, Chen T, Yu KH & Wang C, “Improvements of image sharing with steganography and authentication”, *Journal of Systems and software*, Vol. 80, No. 7, (2007), pp.1070-1076.
15. Chang C, Hsieh Y & Lin C, “Sharing secrets in stego-images with authentication”, *Pattern Recognition*, Vol. 41, No. 10, (2008), pp. 3130-3137.
16. Wu C, Kao S & Hwang S, “A high quality image sharing with steganography and adaptive authentication scheme”, *Journal of Systems and Software*, Vol. 84, No. 12, (2011), pp. 2196-2207.