World Applied Sciences Journal 16 (8): 1137-1141, 2012
ISSN 1818-4952

Particle Swarm Optimization Algorithm Based Methodology for Breaking the Chi-Square Attack in Steganalytic Systems

M. Hamghalam and S. Mirzakuchaki

Department of Electrical Engineering, Iran University of Science and Technology,
Narmak, Tehran 16846-13114, Iran

Abstract: Steganalytic techniques are used to detect whether an image contains a hidden message. By analyzing different image features between stego-images (the image within which information is hidden) and cover-images (the Image in which information is to be hidden), a steganalytic system is able to detect stego-images. In this paper, we present a new method in LSB embedding by avoiding the change of statistic features. After embedding data in the first LSB bits, we apply PSO (Particle Swarm Optimization) by adjusting the second LSB bits of a stego-image while creating the desired statistic features to generate the modified stego-images that can break the inspection of chi-square attack. Experimental results show that our algorithm can not only pass the detection of chi-square test, but also leave our hidden message unchanged on the first LSB bit and enhance the peak signal-to-noise ratio of stego-images. By using PSO algorithm, speed of convergence also is improved.

Key words: Steganalytic systems · Stego-images · Chi-square attack · PSO algorithm

INTRODUCTION

Steganography is the art of hiding a secret message within a cover-medium (image) in such a way that others (warden) can not discern the presence of the hidden message. It differs from cryptography which encodes messages, so that nobody can read it without the specific key. The famous steganographic methods include covert channel, invisible ink, microdot and spread-spectrum communication [1, 2]. Steganography is an old subject; however, computer technology provides a new aspect of applications by hiding messages in the media, such as audio and images.

Steganalytic is the term for any techniques which are used by warden to detect wheatear an image contains a hidden message. Warden is free to inspect all the messages with two options, passive or active. The passive way is to inspect the message in order to determine whether it contains a hidden message and then to perform a proper action. On the other hand, the active way is always to alter messages though Warden may not perceive any trace of a hidden message. Note that, in this paper, we focus on the passive warden.

The essential requirement in the embedding data is that the stego-media should be indistinguishable to the degree that it does not raise suspicion. In other words, the hidden information introduces only slight modification to the cover-media. Most passive warden distinguishes between stego and cover images by analyzing their statistic features.

Generally, the steganalytic systems are categorized into two classes: spatial-domain steganalytic system (SDSS) and frequency-domain steganalytic system (FDSS). The SDSS [3, 4] is adopted for checking the lossless compressed images by analyzing the spatial-domain statistic features. For the lossy compression images such as JPEG, the FDSS [5, 6] is used to analyze the frequency-domain statistic features. Westfeld and Pfitzmann [3] presented two SDSSs based on visual and chi-square attacks. The visual attack uses utilizes human eyes to examine stego-images by checking their lower bit-planes. The chi-square attack or histogram attack can automatically detect the specific characteristic generated by the least-significant-bit (LSB) steganographic technique. Avcibas *et al*. [4] proposed image quality measure (IQM) which is based on a hypothesis that the

Corresponding Author: M. Hamghalam, Department of Electrical Engineering, Iran University of Science and Technology, Narmak, Tehran 16846-13114, Iran.

steganographic systems leave statistic evidences that can be exploited for detection using IQM and multivariate regression analysis. Fridrich *et al*. [5] presented a FDSS for detecting the JPEG stego-images by analyzing their discrete cosine transformation (DCT) with cropped images. The chi-square steganalytic systems [3] are used in our experiments to test the correctness of our PSO-based methodology. Since the steganalytic system analyzes certain statistic features of an image, the idea of developing a robust steganographic system is to produce the stego-image by avoiding changing the statistic features of the stego-image. In literature, several papers have presented the algorithms for steganographic and steganalytic systems. Few papers have discussed the algorithms for breaking the steganalytic systems. Yi-Ta Wu *et al*. [7] presented a Genetic algorithm base methodology for breaking steganalytic systems. Their algorithm is applied on some steganographic systems on spatial and frequency domain. They have two limitations; firstly, there is always BER (bit error rate) in their modified stego-image and also their speed of convergence is low.

In this paper, we present a new method for breaking steganalytic systems. We have modified their idea [7] in our work such that there is no bit error rate. In our method, after embedding data in the first LSB bit of the image, we manipulate the second LSB bit by the binary PSO for breaking the inspection of steganalytic systems. In fact, the binary PSO based approach is adopted to generate several stego-images by changing their second bits until one of them can break the inspection of chi-square attack. In comparison with GA method, our algorithm is converged more quickly.

In section 2 we introduce LSB Embedding and then describe chi-square test or the histogram attack. In section 3, our PSO-based breaking algorithm is discussed. Experimental Results are outlined in Section 4. Paper is concluded in section 5.

LSB Embedding and the Histogram Attack: Digitally embedding a message in a cover-Image usually involves two steps; first, identify the redundant bits of a cover-Image and deciding which redundant bits to use and then modifying them. Generally redundant bits are likely to be the least-significant bit (s) of each data word value of the cover-image.

The LSB embedding leaves characteristic artifacts in the histogram of pixel values. Therefore, we can use this point to build a feature vector for steganalytic system. One method which uses this point, called the histogram attack [3].

For LSB embedding, even pixel values are either left unmodified or increased by 1, while odd pixel values are either left unmodified or decreased. Thus, the grayscale values (2i, 2i + 1) form a pair of values (PoV) that are exchanged into each other during embedding. This asymmetry in the embedding function can be used in the following manner.

Let T_c [j]; j=0,1,…,255 denote the intensity histogram of the cover image and T_s be the corresponding histogram of the stego image after embedding qn bits, where n is the total number of pixels and $0 \le q \le 1$. Equivalently, we can say that we are embedding q bits per pixel (bpp) or that q is the relative message length.

According to Ingemar J. Cox *et al*. [8], T_s can be calculated as a function of T_s and q. Thus, we can write:

$$E\{T_s[2i]\} = (1 - \frac{q}{2})T_c[2i] + (\frac{q}{2})T_c[2i+1] \quad (1)$$

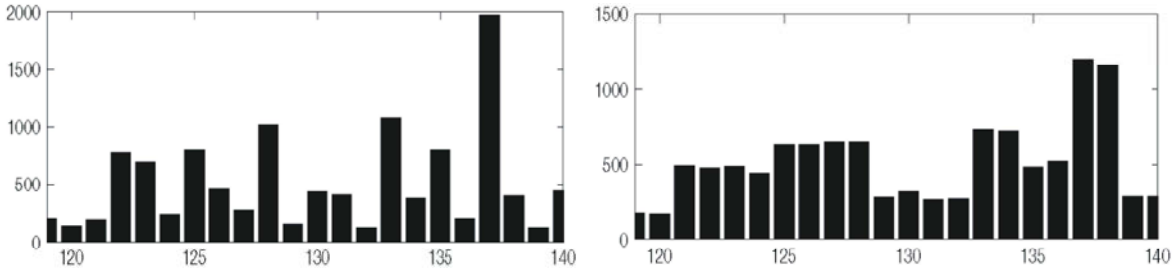
$$E\{T_s[2i+1]\} = (\frac{q}{2})T_c[2i] + (1 - \frac{q}{2})T_c[2i+1] \quad (2)$$

Note that, for a fully embedded image, (q =1), E{Ts [2i]} = E {Ts [2i + 1]}. In other words, the histogram bins 2i and 2i +1 will have approximately the same values, which will cause very obvious step artifacts in the histogram. The theoretically expected value for Ts[2i] is therefore. Thus, it is possible to detect LSB embedding for q = 1 by testing whether Ts [2i] = Ts [2i + 1]. Here, we apply Pearson's chi-squared test [9] to determine whether Ts[2i] =. The chi-square test starts by calculating the chi-square test statistics:

$$S = \sum_{i=1}^{k} \frac{(Ts[2i] - Ts[2i])^2}{Ts[2i]} \quad (3)$$

With (k-1) =127 degrees of freedom. Assuming that even grayscale values indeed follow the probability mass function, T_s [2i], the test statistic, S, follows the chi-square distribution with k - 1 degress of freedom. Intuitively, a small value of S indicates that the data follows the expected distribution and we can conclude that the image contains a message embedded using LSB embedding. On the other hand, large values of the test statistic imply that no message is embedded. The statistical significance of S is measured using the so-called p-value, which is the probability that a chi-square distributed random variable with k-1 degrees of freedom would attain a value larger than or equal to S:

$$p(S) = \frac{1}{2^{\frac{k-1}{2}}.G(\frac{k-1}{2})} \int_{s}^{\infty} e^{-\frac{x}{2}} ... x^{\frac{k-1}{2}} dx \quad (4)$$



(a) (b)

Fig. 1(a): A portion of a cover image histogram. (b) The same portion of the histogram of an image fully embedded with LSB embedding

If the image does not contain a hidden message, S is large and p(S) ≈ 0. Fig. 1 (a) shows a portion of histogram of cover-image. And Fig. 1.b shows changes in histogram after fully embedding data in LSB bits.

As mentioned above, Embedding data in the first LSB bits make changes in histogram which will be detected by chi-square test by steganographic system. So, warden can discern the presence of the hidden message in the stego image. In order to maintain statistics features, we utilize PSO. In next section, we present our PSO-based breaking algorithm.

The Particle Swarm Optimization: The PSO algorithm is introduced by Kennedy and Eberhart [10-12] that simulates the social behaviors of bird flocking or fish schooling and the methods by which they find roosting places, foods sources or other suitable habitat.

In general, the PSO starts with some randomly selected particles in a swarm. Every particle in the swarm corresponding to a solution in the problem domain. An objective, called fitness function, is used to evaluate the quality of each particle. Each individual within the swarm is represented by a vector in multidimensional search space $X_i = (x_{il}, x_{i2},...x_{id})$ which d is number of dimensions. This vector has also one assigned vector which determines the next movement of the particle and is called the velocity vector. This vector denoted by $V_i = (v_{il}, v_{i2},...v_{id})$ and selected randomly at the beginning of process.

The PSO algorithm also determines how to update the velocity of a particle. Each particle updates its velocity based on current velocity and the best position it has explored so far (P_{ibest} ($p_{i1}, p_{i2},...,p_{id}$)); and also based on the global best position (P_{gbest} ($p_{g1}, p_{g2},...,p_{gd}$)) explored by swarm. So the position of the particle and its velocity is being updated using following equations:

$$V_i(t+1) = w.\ v_i(t) + c_1\varphi_1(p_i - x_i(t)) + c_2\varphi_2(p_g - x_i(t)) \quad (5)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (6)$$

Where c_1 and c_2 are positive constants and φ_1 and φ_2 are two random variables with uniform distribution between 0 and 1. In this equation, w is the inertia weight which shows the effect of previous velocity vector on the new vector. According to [13] in Binary space the particle swarm formula remained unchanged, except that now P_{id} and x_{id} are integers in {0, 1} and v_{id}, since it is a probability, must be constrained to the interval [0.0, 1.0]. A logistic transformation $S(v_{id})$ can be used to accomplish this last modification. The resulting change in position then is defined by the following rule:

$$if\ (rand\ () < S(v_{id}))\ \text{then}\ x_{id} = 1;\ else\ x_{id} = 0 \quad (7)$$

Where the function $S(v)$ is a sigmoid limiting transformation and rand() is a quasirandom number selected from a uniform distribution in [0.0, 1.0].

This algorithm is repeated until a predefined condition is satisfied or a predefined number of iterations are reached. The predefined condition in this paper is the situation when we can correctly modify statistic features of the stego image.

In order to apply the PSO for manipulating statistic feature of the stego-image, we use the particles consisting of 64 dimensions (d=64). Figure 2 give an example of these particles. The particles is used to adjust the pixel values of a stego image to modify histogram of image, so the chi-square attack cannot detect presence of the message in the stego image and at the same time the embedded message can be extracted correctly. Since we embed message in the first LSB bit and change the second.

2	0	0	0	2	0	0	2
2	0	2	0	0	2	0	0
0	2	0	0	2	2	2	0
0	2	0	2	2	0	0	0
0	0	0	0	2	0	2	0
2	0	2	0	0	2	2	0
0	2	0	0	2	0	0	0
0	0	2	2	0	0	2	0

Fig. 2: A typical particle in our algorithm

LSB bit to modify statistic features, the hidden message always is unchanged after optimization in our algorithm. In comparison with Yi-Ta Wu et al, we only define a fitness function to evaluate the statistic features. On the other hand, in our algorithm there is no BER.

Let MS and S, respectively, denote the modified stego and the stego images of size 8*8. We modified the stego-images by adding the particles as

$$S = \{Ms_i + S_i + x_i, \text{ where } 0 \le i \le 63\} \quad (8)$$

$$X_i = \{x_0, x_1, ..., x_{63}\} \quad (9)$$

Fitness Function: The fitness function evaluates the difference S, chi-square test, between the cover-image and the stego-image in order to maintain the statistic features. In fact, Fitness Function evaluates the statistic features of the stego-image and compares them with those of the cover-image such that the differences should be as small as possible;

$$Object = |S_C - S_{MS}| \quad (10)$$

Where, S denotes chi-square test of cover image. Our PSO based algorithm is presented below:

Step 1: Embed data in the first LSB bit of cover image and create stego image.

Step 2: Divide the stego-image into a set of stego-images of size 8*8.

Step 3: For each 8*8 stego-image, we modify the second LSB bits based on the PSO to ensure that histogram of stego and cover image are as identical as possible. (Minimizing fitness function).

Step 4: Combine all the 8*8 stego-images together to form a complete stego-image.

Fig. 3: a) original image of Lena. b) Stego image of Lena (fully embedded)

Fig. 4: Modified image in our PSO algorithm

Experimental Results: In this section, we provide experimental results to show that our PSO-based steganographic system can successfully break the inspection of steganalytic systems. For testing our algorithm, we use the 512×512 grayscale image of Lena which is often-used image in image processing research. The Lena image was originally stored in the BMP format. Figure 3a shows original image of Lena and 3b shows this image after fully embedding data in the first LSB bits. Let S_c and S_s denote the chi-square test of cover and stego image, respectively. We have S_c = 3.6977e+004, S_s = 4.1152e+004.

Obviously, there are considerable differences, so the stego image can be detected easily. After applying our algorithm on the stego image, S_{MC}, denoted chi-square test of modified stego image, calculated as S-Stego (Modified) = 3.7043e+004.

Which, there is little difference between cover and modified stego image, so the steganalytic system cannot detect presence of data in the image. Our approach not only fixes the hidden data in the first LSB, but also enhances the peak signal-to-noise ratio of stego-images. The PSNR of the modified stego image is about 45.44 dB, which is suitable value in steganography. In Figure 4 the modified image has been showed. About speed of convergence, our method is converged in 9.72 min (Pentium(R) Dual-Core CPU, 2.60 GHz, 2.0 GB of RAM). In comparison with Yi-Ta Wu *et al*. which we get idea of our algorithm from them, we have three improvements,

Table 1: comparision between GA ans PSO algorithm

Algorithm	GA	PSO
Speed of convergence	11.13 min.	9.72 min.
PSNR	46.45 dB	45.44 dB.
Bit Error Rate	Depended to hidden messege (not zero).	zero.

first, we always embed the messages in the first LSB bit of images, it is not need to know about position of the data in each images. Second, since we modify the histogram by the second bit, we have no BER in our hidden messages. Third, the GA method is converged in 14.13 min. In fact, we have 4.41 min improvement at the same condition.

CONCLUSION

In this paper, we have presented a PSO-based algorithm of modifying a stego-image to break the detection of the chi-square test by artificially counterfeiting statistic features. We design a fitness function to evaluate the quality of each particle in order to adapt the stego-image that can pass through the inspection of steganalytic systems. Experimental results show that our PSO algorithm can not only successfully break the detection of the steganalytic systems, but also leave our hidden message fix. Moreover, in comparision of GA method, our PSO algorithm improve speed of convergence. Table 1 shows comparison between our method and GA method.

REFERENCES

1. Kahn, 1996. The Codebreakers, second edition, Macmillan.
2. Norman, B., 1973. Secret Warfare. Washington, DC: Acropolis Books.
3. Westfeld and A. Pfitzmann, 1999. Attacks on steganographic systems breaking the steganographic utilities EzStego, Jsteg, Steganos and S-tools and some lessons learned, in Proc. 3rd Int. Workshop on Information Hiding, Dresden, Germany, pp: 1-76.
4. Avcibas, N. Memon and B. Sankur, 2003. Steganalysis using image quality metrics, IEEE Trans. Image Process., 12(2): 221-229.
5. Fridrich, J., M. Goljan and D. Hogea, 2003. New methodology for breaking steganographic techniques for JPEGs, in Proc. EI SPIE, Santa Clara, CA, pp: 143-155.
6. Farid, 2001. Detecting Steganographic Message in Digital Images, Dept. Comput. Sci., Dartmouth College, Hanover, NH, Tech. Rep. TR2001- 412.
7. Yi-Ta Wu and Frank Y. Shih, 2006. Genetic Algorithm Based Methodology for Breaking the Steganalytic Systems, Ieee Transactions on Systems, Man and Cybernetics-part B: Cybernetics, 36(1).
8. Ingemar J. Cox, 2008. Digital Watermarking and Steganography, second edition, Morgan Kaufmann Publication.
9. Spiegel, M.R., 1961. Schaum's Outline of Theory and Problems of Statistics, third edition.McGraw-Hill, New York.
10. Kennedy, J. and R. Eberhart, 1995. Particle Swarm Optimization, IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV, pp: 1942-1948.
11. Eberhart, R. and J. Kennedy, 1995. A New Optimizer Using Particles Swarm Theory, Proc. Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service Center, Piscataway, NJ, pp: 39-43.
12. Kennedy, J. and R. Eberhart, 2001. Swarm Intelligence. Morgan Kaufmann Publishers, Inc., San Francisco, CA.
13. Kennedy, J. and R.C. Eberhart, 1997. A discrete binary version of the particle swarm algorithm, IEEE International Conference on Systems, Man and Cybernetics.