Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 876946, 6 pages
doi:10.1155/2010/876946

Research Article

Improved Adaptive LSB Steganography Based on Chaos and Genetic Algorithm

Lifang Yu, Yao Zhao, Rongrong Ni (EURASIP Member), and Ting Li

Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Yao Zhao, yzhao@bjtu.edu.cn

Received 17 November 2009; Accepted 19 May 2010

Academic Editor: Yingzi Du

We propose a novel steganographic method in JPEG images with high performance. Firstly, we propose improved adaptive LSB steganography, which can achieve high capacity while preserving the first-order statistics. Secondly, in order to minimize visual degradation of the stego image, we shuffle bits-order of the message based on chaos whose parameters are selected by the genetic algorithm. Shuffling message's bits-order provides us with a new way to improve the performance of steganography. Experimental results show that our method outperforms classical steganographic methods in image quality, while preserving characteristics of histogram and providing high capacity.

1. Introduction

Steganography is the science of hiding messages in a medium called *carrier* or *cover* object in such a way that existence of the message is concealed. The cover object could be a digital still image, an audio file, or a video file. The hidden message called *payload* could be a plain text, an audio file, a video file, or an image [1, 2].

Steganographic methods can be classified into *spatial domain embedding* and *frequency domain embedding*. Least Significant Bit (LSB) replacing is the most widely used steganographic method in spatial domain, which replaces the cover image's LSBs with message bits directly. Although it has several disadvantages such as vulnerable to attacks, LSB steganography is a popular method because of its low computational complexity and high embedding capacity. In frequency domain, popular steganographic methods mostly base on Discrete Cosine Transformation (DCT). After performing DCT on each 8×8 block and quantizing the DCT coefficients, message bits are embedded into the quantized DCT (qDCT) coefficients. Recently, many steganographic schemes using LSB and its improved versions on qDCT have been invented, which offer reasonably high embedding capacity while attempting to preserve the marginal statistics of the cover image, such as J-Steg [3], F5 [4], and OutGuess [5]. It is well known that J-Steg is detectable using the χ^2 attack [6, 7] since it is based on simply flipping LSBs. F5 employs matrix encoding to decrease the change for one payload, but its shrinkage at 0s makes it detectable. OutGuess embeds message bits into a part of coefficients and uses the other part to compensate artifacts on the histogram, so it preserves characteristics of histogram. But its embedding efficiency and capacity are low because of compensation.

Our contributions are in two folds. First, we present improved adaptive LSB steganography that can embed messages adaptively and thus can satisfy various requirements (high capacity, high security, high image quality, etc.). Second, our method minimizes degradation of the stego image through finding the best mapping between the secret message and the cover image based on chaos and the genetic algorithm (GA).

The rest of the paper is organized as follows. Section 2 introduces general principles of chaos and GA. Section 3 illustrates our proposed method in detail, which includes the improved adaptive LSB steganography, a method to shuffle message bits based on the logistic map and GA, the embedding procedure and the extraction procedure. Experimental results are shown in Section 4, where we demonstrate that our method has good stego image quality, high security-preserving characteristics of histogram, and high capacity. Finally, conclusions are addressed in Section 5.

2. Preliminary

2.1. Chaos and Its Application in Information Hiding. The chaos phenomenon is a deterministic and analogously stochastic process appearing in a nonlinear dynamical system [8, 9]. Because of its extreme sensitivity to initial conditions and the outspreading of orbits over the entire space, it has been used in information hiding to increase security [10, 11].

Logistic map is one of the simplest chaotic maps, described by

$$x_{n+1} = \mu x_n(1 - x_n), \tag{1}$$

where $0 \le \mu \le 4$, $x_n \in (0, 1)$.

Researches on chaotic dynamical systems show that the logistic map stands in chaotic state when $3.5699456 < \mu \le 4$. That is, the sequence $\{x_n, n = 0, 1, 2, \ldots\}$ generated by the logistic map is nonperiodic and nonconvergent. All the sequences generated by the logistic map are very sensitive to initial conditions, in the sense that two logistic sequences generated from different initial conditions are uncorrelated statistically. The logistic map was used to generate a sequence as the watermark [11] or to encrypt the embedded position [10, 11] in former works. In our algorithm to be described below, we use the logistic map to shuffle bits-order of the message.

2.2. Genetic Algorithm. The genetic algorithm (GA), introduced by Holland [12] in his seminal work, is commonly used as an adaptive approach that provides a randomized, parallel, and global search. It bases on the mechanics of natural selection and genetics to find the exact or approximate solution for a given optimization problem.

GA is implemented as a computer simulation in which a population of abstract representations of candidate solutions to an optimization problem evolves toward better solutions. The evolution usually starts with some randomly selected genes as the first generation. All genes in a generation form a *population*. Each individual in the population is called *chromosome*, which corresponds to a solution in the optimization problem domain. An objective, called *fitness function*, is used to evaluate the quality of each chromosome. A new generation is recombined to find the best solution by using three operators: *selection*, *crossover*, and *mutation* [13]. The process is repeated until a predefined condition is satisfied.

Once we have the genetic representation and the fitness function defined, pseudocode algorithm of GA is illustrated as follows.

(1) Generate initial population.

(2) Evaluate the fitness of each individual in the population.

(3) Select best-ranking individuals to reproduce.

(4) Breed a new generation through crossover and mutation (genetic operations) and give birth to offspring.

(5) Evaluate the individual fitness of the offspring.

(6) Replace the worst ranked part of population with offspring.

(7) Repeat (3) to (6) until termination condition is satisfied.

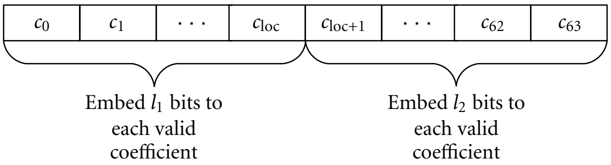
c_0
c_1
$\cdots$
c_{loc}
$c_{\text{loc}+1}$
$\cdots$
c_{62}
c_{63}
Embed l_1 bits to each valid coefficient
Embed l_2 bits to each valid coefficient

FIGURE 1: Division of 64 coefficients in a 8 × 8 block.

3. Our Proposed Method

3.1. Improved Adaptive LSB (IA-LSB) Steganography. The classical LSB steganography replaces cover images' LSBs with messages' bits directly. This embedding strategy leads to dissymmetry. When the LSB of a coefficient in the cover image equals to its corresponding message bit, no change is made. Otherwise, this coefficient is changed from $2n$ to $2n+1$ or from $2n + 1$ to $2n$—changes from $2n$ to $2n - 1$ or from $2n + 1$ to $2n + 2$ never happen. This dissymmetry is utilized by steganalysis, known as χ^2 attack [6, 7].

In order to avoid dissymmetry, improved adaptive LSB (IA-LSB) steganography is proposed. First, the number of bits to be embedded in a certain coefficient is adaptive. With proper parameters, we can get high capacity while preserving high security. Second, less modification rule (LMR) is used to minimize modification.

3.1.1. Adaptively Decide Bits to be Embedded in Each Coefficient. Let $C = \langle c_0, c_1, \ldots, c_{63} \rangle$ denote the sequence of quantized DCT coefficients in a certain 8×8 JPEG block of the cover image. loc divides 64 coefficients into two parts. In the first part, l_1 bits are embedded into each valid coefficient, and in the second part, l_2 bits are embedded (shown in Figure 1). We can adjust l_1, l_2, and loc to get high performance according to the content of the cover image.

3.1.2. Less Modification Rule (LMR). Suppose c_i is assigned to hold l ($l \in \{l_1, l_2\}$) bits. Denote c_i's corresponding l message bits as m_i ($m_i \in \{0, 1, \ldots, 2^l - 1\}$) decimally, and denote its corresponding coefficient in the stego image as s_i. Let $\text{LSB}_l(x)$ be the decimal expression of the least significant l bits of x. That is, $\text{LSB}_l(x) = x \mod 2^l$.

Let $s_i' = c_i + m_i - \text{LSB}_l(c_i)$, $s_i'' = c_i - (2^l - (m_i - \text{LSB}_l(c_i)))$ be two candidates for s_i. Because $\text{LSB}_l(s_i') = \text{LSB}_l(s_i'')$, s_i' and s_i'' hold the same message bits. In classical LSB steganography, $s_i = s_i'$. In our method, s_i' or s_i'' is chosen according to less modification rule formulated as follows:

$$s_i = \begin{cases} s_i' & \text{if } |s_i' - c_i| < |s_i'' - c_i|, \\ s_i'' & \text{if } |s_i' - c_i| > |s_i'' - c_i|, \\ s_i' \text{ or } s_i'', \text{randomly}, & \text{if } |s_i' - c_i| = |s_i'' - c_i|. \end{cases} \tag{2}$$

In this rule, we always choose the change that introduces less modification. For example, if $l = 2$, $m_i = 3$, and $c_i = 8$,

Table 1: PSNR of gray images embedded by IA-LSB with and without shuffling message bits, simply denoted as "with" and "without".

PSNR (db)	Average embedding capacity (bpc)					
	0.46		0.624		0.731	
	with	without	with	without	with	without
Lena	39.93	39.72	38.521	38.181	37.376	37.221
Baboon	33.59	33.41	33.058	32.821	32.381	32.23
Milkdrop	44.32	44.21	40.187	39.92	39.274	38.934
Plane	38.73	38.44	37.586	37.281	36.727	36.506

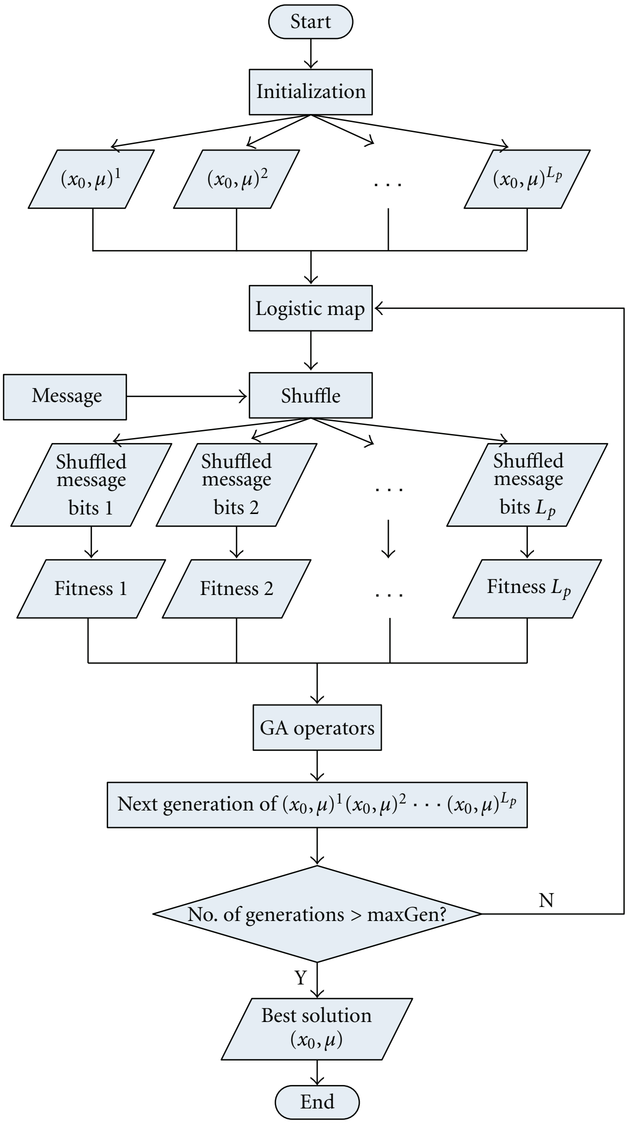

Figure 2: Process of using GA to find the best pair input for logistic map.

then LSB$_l(c_i) = 0$, $s_i' = c_i + 3 = 11$, $s_i'' = c_i - 1 = 7$. LSB$_l(s_i')$ = LSB$_l(s_i'')$, but the absolute value of change from c_i to s_i' is 3 while to s_i'' is 1, so choose s_i'' as s_i. Take another example, $l = 2$, $m_i = 3$ and $c_i = 10$, then LSB$_l(c_i) = 2$, $s_i' = c_i + 1 = 11$, $s_i'' = c_i - 3 = 7$. In this case, choose s_i', which is closer to c_i, as s_i.

Table 2: PSNR of color images embedded by IA-LSB at 0.45 bpc.

PSNR (db)	with	without
Lena	35.134	35.011
Baboon	28.62	28.556
Milkdrop	39.384	39.252
Plane	34.518	34.391

3.2. Shuffle Message Bits Based on Chaos and Genetic Algorithm. Shuffling message bits changes the way of modifying the cover image during embedding thus influences image quality and security of the stego image. By finding a proper way to shuffle, we can improve the image quality or security or both. In this paper, we use the logistic map for shuffling and use GA to find proper parameters for the logistic map.

Denote the message with length L as $M = \{m_0, m_1, \ldots, m_{L-1}\}$. The process of using the logistic map to shuffle is stated as follows.

(1) Given a pair of input (x_0, μ), the logistic map will generate a sequence $\{x_n, n = 0, 1, 2, \ldots\}$. Wipe off the first k (e.g. 1000) elements of the sequence, and use the consecutive L different elements to form a vector $Y = \{y_0, y_1, \ldots, y_{L-1}\} = \{x_k, x_{k+1}, \ldots, x_{k+L-1}\}$.

(2) Sort the elements of Y in descending order. The suffixes of the sorted elements form a sequence $I = \{i_0, i_1, \ldots, i_{L-1}\}$.

(3) Shuffle message bits according to I. That is, the message bit with suffix i_r in M is put to position r.

Here comes an example of using the logistic map to shuffle message bits. Let $M = \{0, 1, 1, 1, 0, 1\}$, $Y = \{0.1, 0.6, 0.4, 0.2, 0.8, 0.7\}$, then $I = \{4, 5, 1, 2, 3, 0\}$, and shuffled message sequence is $\{0, 1, 1, 1, 1, 0\}$.

From the shuffling process mentioned above, we can see that the pair of parameters (x_0, μ) decides the order of shuffled message bits. In order to improve the performance of the shuffling method, GA is used to select a proper pair of (x_0, μ). In our scheme, we choose to improve quality of the stego image in the sense of PSNR and select PSNR as GA's fitness function:

$$\text{fitness} = \text{PSNR} = -10 \cdot \log_{10}\left\{\frac{1}{255^2 MN}\sum_{m=1}^{M}\sum_{n=1}^{N}[d(m,n)]^2\right\}, \tag{3}$$

where M and N are number of rows and columns of the cover image, respectively; $d(m, n)$ is the difference between

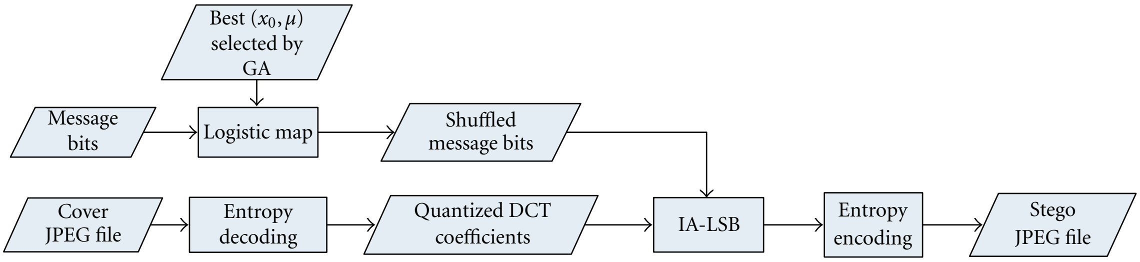

FIGURE 3: Embedding procedure of our proposed method.

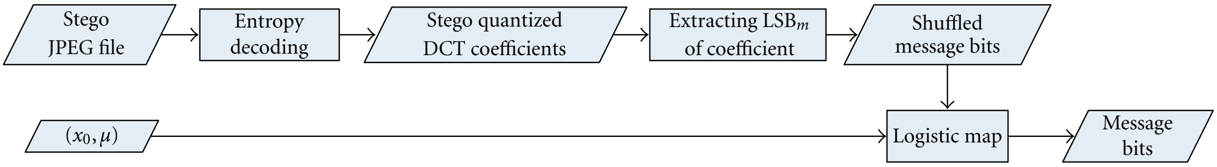

FIGURE 4: Extracting procedure of our proposed method.

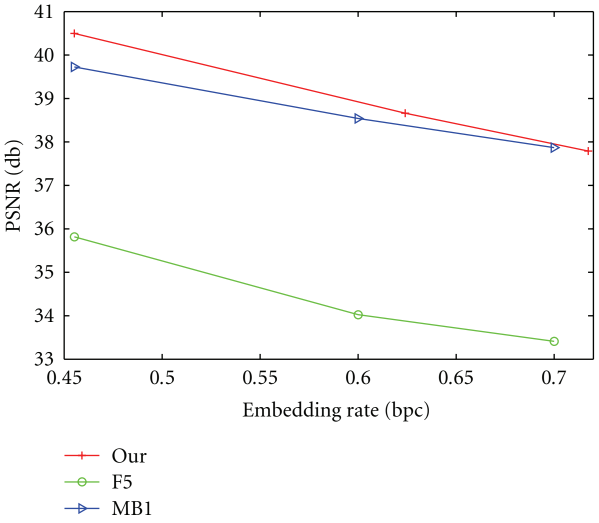

FIGURE 5: PSNR of our method, F5, and MB1.

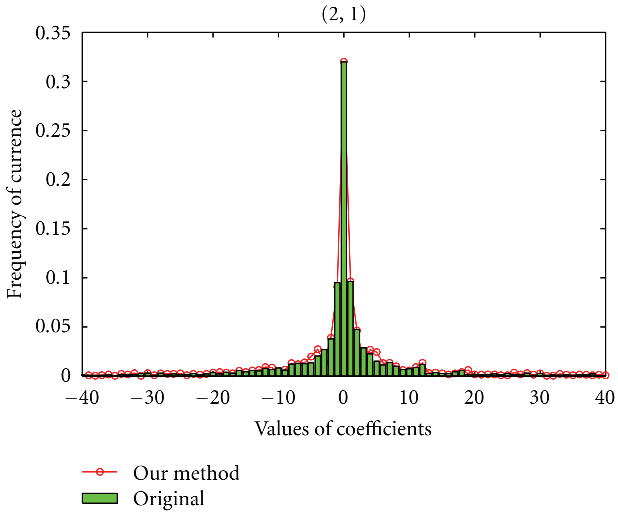

FIGURE 6: Distribution of the (2,1)th AC components.

coefficients in spatial domain at position (m, n) in the cover image and in the stego image. The process of using GA to maximize PSNR is shown in Figure 2 and stated as follows.

(1) Initialize population. Randomly generate L_p pairs of (x_0, μ), $x_0 \in (0, 1)$, $\mu \in (3.5699456, 4]$. L_p is the size of population and each (x_0, μ) is an individual.

(2) For each (x_0, μ), shuffle message bits and embed the reordered message bits into the cover image using IA-LSB steganography, then compute PSNR between the cover image and the stego image, which is the fitness function of GA. In the following operations, the individual with larger fitness function will be considered better.

(3) GA operators—selection, crossover, and mutation—are operated to generate the next generation.

(4) Repeat (2) and (3) till the number of generations equals maximum generation *maxGen* (e.g., 100).

(5) Put out the best pair of (x_0, μ) selected by GA.

3.3. Embedding Procedure. A coefficient c_i is *valid*, if $c_i \neq 0$ and it is not a DC coefficient. The whole embedding procedure is depicted in Figure 3. Firstly, the message bits are shuffled by the logistic map whose input pair (x_0, μ) is selected by GA. Secondly, the cover JPEG file is decoded, obtaining quantized DCT coefficients. Thirdly, the shuffled message bits are embedded into the valid quantized DCT coefficients using IA-LSB steganography. Finally, stego quantized DCT coefficients are encoded to the stego JPEG file.

It needs to be taken into consideration that valid coefficients after embedding should still be valid, that is, valid coefficients should not be changed to 0. On one hand, characteristics of histogram can be preserved; on the other hand, message bits can be extracted correctly and simply. If $s_i = 0$, $s_i = s_i \pm 2^l$. To add or subtract 2^l is determined randomly.

3.4. Extracting Procedure. After receiving the stego JPEG file and (x_0, μ), we can extract the message bits as showed in Figure 4. First, the stego JPEG file is entropy decoded to obtain stego quantized DCT coefficients. Second, the shuffled message bits are extracted from LSBs of valid coefficients. Thirdly, the shuffled message bits are reordered to there natural order using logistic map with (x_0, μ) as input. Message bits are obtained.

4. Experiments

In this section, we demonstrate the performance of our proposed method and compare it with that of F5 [14], MB1 [15], and Outguess [16]. The image quality of each steganography method is expressed objectively in PSNR. Standard 256 gray-level and true color images with sizes of 256 × 256 are used as covers, such as Lena, Baboon, and Couple. The JPEG quality factor is set to 80 during compression in each method.

4.1. Image Quality. In order to demonstrate validity of shuffling message bits, we compare the PSNR of images embedded by IA-LSB steganography with and without shuffling message bits. The results of gray images are shown in Table 1. Shuffling message bits does improve the PSNR of the stego image. It can also be applied to other steganographic algorithms and provides us with a new way to improve performance of steganography. Moreover, Table 2 shows that this scheme of shuffling is not only applicable to gray images but also color images.

Figure 5 shows the PSNR of our method, F5, and MB1. The results are averaged on 50 gray-level images. We can see that the PSNR of our proposed method is higher than that of F5 and MB1. For the capacity of Outguess is around 0.3 bpc, it is not shown in the figure. The PSNR of Outguess is not higher than 32.86 db at 0.3 bpc (bit per nonzero AC coefficient) but of our method is higher than 37 db even at 0.72 bpc. We can conclude that our method outperforms F5, MB1, and Outguess in image quality.

4.2. Preserving Characteristics of Histogram. As a representative example, Figure 6 plots distribution of the (2,1)th quantized AC components for cover image "Lena" and its corresponding stego image with an embedding rate of 0.46 bpc. The red line illustrates the coefficients distribution of a stego image with our proposed method, and green bars illustrate that of the cover image. Figure 6 shows that our method preserves the characteristics of histogram. This is also true for the other components (e.g., (1,2)th, (2,2)th AC components) and the other testing images.

5. Conclusion

A steganographic method uses IA-LSB based on chaos and genetic algorithm is proposed. After finding the best parameters for the logistic map using GA, rearrange the secret message and embed it into the cover image using IA-LSB. Experimental results demonstrate that our algorithm achieves high embedding capacity while preserving good image quality and high security.

The important and distinctive features in the proposed method are to minimize the degradation of stego image by shuffling the secret message based on the logistic map and GA. To find better mapping between the secret message and the cover image so as to improve the steganographic performance is our future work.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (no. 60776794, no. 90604032, and no. 60702013), 973 program (no. 2006CB303104), 863 program (no. 2007AA01Z175), Beijing NSF (no. 4073038), and Specialized Research Foundation of BJTU (no. 2006XM008 and no. 2005SZ005).

References

[1] J. Silman, "Steganography and steganalysis: an overview," Tech. Rep., SANS Institute, 2001.

[2] T. Jamil, "Steganography: the art of hiding information in plain sight," *IEEE Potentials*, vol. 18, no. 1, pp. 10–12, 1999.

[3] D. Upham, 1997, http://zooid.org/~paul/crypto/jsteg/.

[4] A. Westfeld, "F5-a steganographic algorithm," in *Proceedings of the 4th International Workshop on Information Hiding*, pp. 289–302, Pittsburgh, Pa, USA, 2001.

[5] N. Provos, "Defending against statistical steganalysis," in *Proceedings of the 10th USENIX Security Symposium*, pp. 323–335, Washington, DC, USA, 2001.

[6] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," in *Proceedings of the 3rd International Workshop on Information Hiding*, 2000.

[7] N. Provos and P. Honeyman, "Detecting steganographic content on the internet," Tech. Rep., Center for Information Technology Integration, University of Michigan, 2001.

[8] Z. Liu and L. Xi, "Image information hiding encryption using chaotic sequence," in *Proceedings of the 11th International Conference on Knowledge-Based Intelligent Information and Engineering Systems and the XVII Italian Workshop on Neural Networks*, pp. 202–208, 2007.

[9] Y. Zhang, F. Zuo, Z. Zhai, and C. Xiaobin, "A new image encryption algorithm based on multiple chaos system," in *Proceedings of the International Symposium on Electronic Commerce and Security (ISECS '08)*, pp. 347–350, August 2008.

[10] R. Munir, B. Riyanto, S. Sutikno, and W. P. Agung, "Secure spread spectrum watermarking algorithm based on chaotic map for still images," in *Proceedings of the International Conference on Electrical Engineering and Informatics*, 2007.

[11] Z. Dawei, C. Guanrong, and L. Wenbo, "A chaos-based robust wavelet-domain watermarking algorithm," *Chaos, Solitons and Fractals*, vol. 22, no. 1, pp. 47–54, 2004.

[12] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, Mass, USA, 1992.

[13] Y.-T. Wu and F. Y. Shih, "Genetic algorithm based methodology for breaking the steganalytic systems," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 36, no. 1, pp. 24–31, 2006.

[14] http://os.inf.tu-dresden.de/~westfeld/publikationen/f5r11.zip.

[15] http://www.philsallee.com/mbsteg/index.html.

[16] http://www.outguess.org/download.php.