We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

3

RABS: Rule-Based Adaptive Batch Steganography

Hedieh Sajedi
Department of Computer Science, Tehran University, Iran

1. Introduction

Steganography is used mostly when the fact of communicating needs to be kept covert. This is carried out by embedding the secret data in apparently innocuous covers. Typical covers are image, video, and audio files (Munuera, 2007). For steganographic systems, the major requirement is that the stego object should be perceptually and statistically indistinguishable to the degree that it does not raise suspicion. In other words, the hidden information should impose slight or undetectable modification to the cover objects (Wu & Shih, 2006).

The steganography methods that embed secret data in Discrete Cosine Transform (DCT) of images are well known because of the vast usage of JPEG images. Steganography methods like F5 (Westfeld,2001), Model-based (MB) (Sallee, 2003), Perturbed Quantization (PQ) (Fridrich et al. 2004), and YASS (Solanki et al. 2007) manipulate some DCT coefficients for embedding the secret data. Some other methods have been proposed which embed secret information in other transform domains, such as Contourlet-based steganography method (Sajedi & Jamzad, 2008) that hides secret data in contourlet coefficients of a cover image.

Stego version of images may have different grades of visual and statistical undetectability due to their different contents. It is shown in (Wu & Shih, 2006) that, when the size of hidden data gets larger than a threshold then it becomes easier for a steganalysis algorithm to detect the presence of the hidden data. Consequently, a steganography method can employ a technique to embed in a cover image until it does not attract the attention of steganalysis methods and makes steganalyzers to misclassify the observed stego image as a cover (clean) image. In this way, we can find an upper bound for embedding rate such that if the size of hidden data is less than that upper bound, the stego image is safe and it can not be detected by steganalysis methods.

So far, the performance of a steganography method is evaluated in its average steganography capacity. However, there is no guarantee that a specific stego image could not be detected reliably by the steganalyzers. In this chapter, we propose an approach to estimate the steganography capacity of images. Then, the steganographer can embed securely a secret data with the size smaller than or equal to the steganography capacity of the image.

In investigating the problem from a theoretical model, in order to obtain the actual results, the statistical models of images should be simplified to an extent that would cause the results less relevant. Instead of using statistical models, we extract the signature of cover images as a practical model for images and evaluate the security of stego images with respect to this model.

The usual concept of steganography assumes that the steganographer embeds the secret data in only one cover image and passes it on to the recipient by means of a communication channel. Since the steganography capacity of an image is limited (Fridrich et al. ,2007), to hide a large secret data securely, we have to split the secret data to some parts and embed each part in a cover image. Consequently, the large secret data is hidden in a set of cover images and the resulted set of stego images is passed through the communication channel. This technique is called Batch steganography, which is first proposed by Ker (Ker, 2006). He states in (Ker, 2008) that in batch steganography, the best choice for the steganographer is to spread the secret data equally between cover images. This statement is due to the assumptions considered to simplify the theoretical presentation of batch steganography.

In (Sajedi& Jamzad, 2009) to describe the problem in batch steganography, we called the Ker's definition of batch steganography as Static Batch Steganography (SBS). SBS divides the secret data into some parts of equal size and embeds each part in a cover image separately (Ker, 2006; Ker, 2008). Since in SBS the steganographer embeds in cover images blindly without considering the differences between various cover images, two problems may be occurred. First, the steganographer may embed in an image less than its steganography capacity. In this case, the steganographer has not used the steganography capacity of the image efficiently. The second problem occurs when the steganographer hides secret data in an image while the size of secret data is larger than the steganography capacity of image. Consequently, the security of the resulted stego image is reduced. In fact, SBS assumes identical steganography capacity for each cover image independent of its content. In SBS, a small part of secret data is embedded in each image so as the security of the produced stego image is not threatened.

Previously, we proposed an adaptive batch steganography called ABS in (Sajedi& Jamzad, 2009). In this method, an ensemble of steganalyzers determines the steganography capacity of images and then the secret data is divided based on the steganography capacity of cover images. In (Sajedi& Jamzad, 2009), the estimation time of steganography capacity is rather high because the feature extraction part of steganalyzers, which are used in the ensemble steganalyzer, is done completely. As we know, the feature extraction part of each steganalyzer is the most time-consuming part of it. Consequently, in this chapter we tend to reduce the estimation time of steganography capacity in adaptive batch steganography.

We propose Rule-based Adaptive Batch Steganography (RABS) in which we estimate the steganography capacity of each image according to the signature of clean images. Therefore, instead of embedding a constant amount of data in each image as Ker proposed, we embed unequal amounts of secret data in each image reliant to its steganography capacity. RABS is an intelligent solution to the problem of hiding a large secret data and it guaranties the security of stego images against the existing steganalyzers.

On the other hand, since the communication channel is limited, we prefer to have the lowest number of images in the stego image set to occupy the communication channel lesser. RABS attempts to decrease the number of required cover images by applying an intelligent cover selection approach. In this approach, to hide a large secret data we split the payload and embed each part in an image with steganography capacity of equal or higher than the secret data part.

Generally, in batch steganography by considering the steganography capacity of images, the steganographer can embed securely every large-size secret data in a set of images, which are selected to have a sufficient total steganography capacity. In RABS, for embedding a secret data with a certain size, the database is checked and a set of proper cover images is suggested for embedding. This strategy can be used with any existing steganography methods.

In this chapter, steganography capacity of images is estimated by extracting and using the signature of clean images. Our approach first analyzes an image database to discover the signature of clean images. By the signature, we mean the effective features of clean images and their relative values. This signature is a set of fuzzy if-then rules that represents similarity between clean images. A secure stego image is the one that after data hiding stimulates the generated fuzzy rules significantly and follows the signature of clean images. The process of generating the signature of clean images is done by an Evolutionary Algorithm (EA). EA have been used as rule induction and optimization tools in design of fuzzy rule-based systems (Cordon et al. 2009, Hu et al. 2003).

We considered steganography capacity measure in applying MB, PQ, and YASS steganography techniques and validated it using an image database. The results illustrate that embedding in cover images based on their steganography capacity reduces the detection accuracy of state-of-the-art steganalysis methods considerably compared to the traditional usage of the steganography methods. In addition, we used RABS for hiding some large secret data in cover image sets and compared the results with ABS and SBS.

The rest of this chapter is organized as follows. In Section 2, we describe the related works. RABS method is presented in Section 3. Steganography capacity estimation based on signature of clean images is described in Section 4. We explain the experimental results in Section 5. Finally, we conclude our work in Section 6.

2. Related works

2.1 Batch steganography

Batch steganography is hiding the secret data into multiple cover objects. It seems that considering a fixed secret data length, the embedding in many cover objects will reduce the embedding rate for each image and thus make the detection harder. Moreover, a recent paper by Fridrich (Fridrich et al. ,2007) highlights the fact that whatever the steganography algorithm, it remains highly detectable when the embedding rate is above a threshold (0.05 bits per non-zero DCT coefficients). The use of small embedding rates provided by batch steganography seems attractive in this sense. Ker in (Ker, 2006) takes the main following assumptions about the batch steganography process:

- The number of cover objects is fixed.
- All cover objects have the same steganography capacity.
- The data to embed is of fixed length.
- The number of cover objects is known to the steganalyzer.

Additionally, Ker in (Ker, 2008) states that the best choice of the steganographer is to spread payload equally between covers. As discussed in (Ker, 2006), some of these assumptions are taken in order to establish a proper theoretical framework for the unexplored subject of batch steganography. However, these assumptions may not be totally applicable to a practical case. For example, if the number of cover images is low and the total capacity of cover images is not enough to conceal the secret data, then each resulted stego image is overloaded, and its security would be low. If the number of cover images is high, with spreading the secret data equally between cover images, then the security of stego images is high but the communication channel will be occupied a lot.

2.2 Steganography capacity

A number of ways to compute the steganography capacity have been proposed previously (Chandramouli, & Memon , 2003- Sajedi & Jamzad,2009a). A definition of steganography capacity is presented in (Chandramouli & Memon,2003) from a steganalysis perspective. This work argues that as the main goal of steganography is hidden communications, steganography capacity is dependent on the type of steganalysis detector employed to break the embedding algorithm. It defines γ-security so that in presence of a steganalysis detector D, a steganography algorithm is said to be perfectly secure if $\gamma D = 0$.

The work in (Cachin, 1998) defines a steganography method to be ε-secure ($\varepsilon \geq 0$) if the relative entropy between the cover and the stego probability distributions (Pc and Ps, respectively) is at most ε, i.e.,

$$D(P_c \| P_s) = \int P_c \log \frac{P_c}{P_s} \leq \varepsilon \tag{1}$$

A stego technique is said to be perfectly secure if $\varepsilon = 0$. This definition assumes that the cover and stego images are independent identically distributed (i.i.d.) random variables. This assumption is not true for many real life cover signals (Chandramouli & Memon,2003). One approach to rectify this is to put the constraint that the relative entropy computed using the nth order joint probability distributions must be less than εn. One can then force a steganography technique to preserve the n order distribution. But, it may then be possible to use (n+1) order statistics for steganalysis. The research in (Moulin & Mihcak, 2002) provides an estimate of steganography capacity of images, based on a parallel Gaussian model.

Ker in (Ker, 2007) defines batch steganography capacity and theoretically proves that the size of secret data can safely increase no faster than square root of the number of cover images.

In (Sajedi & Jamzad,2009a), an ensemble system that uses different steganalyzer units, considers the steganography capacity by determining the security limits for embedding in cover images. In this system, each steganalyzer unit is formed by a combination of multiple

steganalyzers from the same type. Each steganalyzer in a steganalyzer unit is trained to detect stego images with a certain payload. The upper bound of embedding rate for an image is determined based on the confidence of all the steganalyzers about the image. In fact, considering steganography capacity, the steganographer can minimize the risk of detection by selecting from a database a proper cover image that is secure for a certain payload. To calculate the steganography capacity of an image, the embedding rate is increased steadily until the security of the produced stego image is threatened by the ensemble steganalyzer. The time (t) required for secure embedding (Sajedi & Jamzad,2009a) is shown by equation (2):

$$t = t_s + t_{sce}$$

$$t_{sce} = k \times (t_s + \sum_{i=1}^{I} t_{su}(i)) \; ; \; t_{su}(i) = \sum_{j=1}^{k} t_{sz}^{i}(j) \tag{2}$$

where t_{sce} is the time of steganography capacity estimation, t_s is the embedding time of employed steganography method, k is the number of iterations of incremental embedding algorithm and $t_{su}(i)$ is the time of the ith steganalyzer unit. I is the number of steganalysis units that are used in the ensemble steganalyzer and $t_{sz}^{i}(j)$ is the time required for the jth steganalyzer that is trained for detection of stego images with payload of multiple j in the ith steganalyzer unit. Although the time complexity of secure embedding (Sajedi & Jamzad,2009a) is more than traditional embedding, but since it provides more secure stego images, its time complexity can be acceptable. Due to differences in contents of various images, the total time of incremental embedding may differ. Considering the fact that the main goal of steganography is to embed the secret data securely and if any of the steganalyzers gets suspicious, then the purpose of steganography is broken, therefore it is worth to spend further time to make stego images more secure.

In this chapter, the evolutionary rule induction algorithm, which is proposed in (Sajedi & Jamzad,2009b) is employed. The generated fuzzy rule base is used to form the signature of clean images and afterward to estimate the steganography capacity of images.

2.3 RABS: Rule-based Adaptive Batch Steganography

Considering steganography capacity of an image, we can embed in the image a portion of secret data that its size is less or equal to the steganography capacity of the image. The remaining unconcealed portion of secret data can be hidden in some other cover images. Our approach aims to improve the undetectability of stego images while utilizing the cover images perfectly. Additionally, RABS tries to hide a large secret data in a cover image set very quickly. Differing from the static batch steganography (Cachin, 1998; Cordon et al. 2009), a more practical approach is used in this chapter which its details and procedure is described in the following. Our assumptions are as follows:

1. The steganographer has the option to select cover images from an image database.
2. The size of secret data can be variable.
3. The number of cover images in a cover image set can be variable.

4. The receiver knows the strategy of steganographer for breaking the payload into some parts and the order of stego images.

The steganographer can select cover images randomly from the database or based on their steganography capacities to minimize the detection rate and the number of images in the stego image set. In RABS, at first the steganography capacity of cover images is estimated using 'Signature of Clean Images' (which is described in the next section) and then the embedding algorithm is activated.

The following steps demonstrate the embedding algorithm of RABS approach. Let CI denotes the steganography capacity of image I and EDS denotes the size of embedding secret data. The inputs of the algorithm are SDS, IDB, and CIS that respectively denotes secret data size, image database, and signature of clean images. The output of the algorithm is SIS, which denotes the stego image set.

Step 1. Select cover image *I* from *IDB* and set *EDS=0*.

Step 2. Hide portions of secret data incrementally until the image deviates from *CIS*. The size of data that is embedded in the image shows the steganography capacity C_I of image I . Set $EDS = EDS + C_I$.

Step 3. Add the stego image to *SIS.*

If $SDS > EDS$ then

Select a new cover image from *IDB* to embed the remaining part of the secret data. Go to step 2.

Otherwise,

SIS is the output.

Figure 1 shows the block diagram of steganography capacity estimation. In RABS, embedding is continuing in different cover images until the secret data become concealed completely. Steganographer can select cover images in random or based on a cover selection criterion. To reduce the number of images in stego image set, the steganographer can employ a cover selection strategy and choose cover images with higher steganography capacities. The image that is the output of the incremental embedding routine is a secure stego image because it conceals a part of secret data that its size is not more than the steganography capacity of the image.

3. Steganography capacity estimation

3.1 Signature extraction of clean images

In this chapter, we utilize an iterative evolutionary fuzzy algorithm for estimation of steganography capacity of images. In this utilization, the algorithm extracts a fuzzy rule base for obtaining the signature of clean images. Consequently, to have a secure covert communication, we can embed in an image until it deviates from the clean images signature. It should be noted that the embedding procedure could be carried on by any steganography method.

It is generally believed that a blind steganalyzer trained on sufficiently many diverse steganography algorithms will become universal in the sense that it will generalize to previously unseen (novel) steganography methods. While this is a partially correct statement if the embedding mechanism of the novel method resembles some of the methods on which the classifier was trained (Pevny & Fridrich, 2008), it demonstrated that if the classifier is presented with stego images produced by a completely different embedding mechanism, it may fail to detect the images as stego even for an easily detectable method. Motivated by this observation, (Pevny & Fridrich, 2008) explored two approaches for construction of universal steganalyzers—one-class and one-against-all classifiers. One-against-all classifiers may fail on previously unseen stego algorithms. One-class methods are less likely to fail to detect unknown stego algorithms. Considering the above discussion, to have more generalization capability and to cover unseen steganography methods, we model clean images instead of stego images according to the method shown in Figure 1.

The process of extracting steganographic features from an image is a mapping $f : C \mapsto \mathbb{R}^d$ from the space of all the covers, C, to a d-dimensional feature space. In steganalysis, learning methods are used to find a distinguishing statistics $S : \mathbb{R}^d \mapsto \mathbb{R}$, on which a threshold is set to classify images to the classes of cover and stego (Pevny et al., 2009).

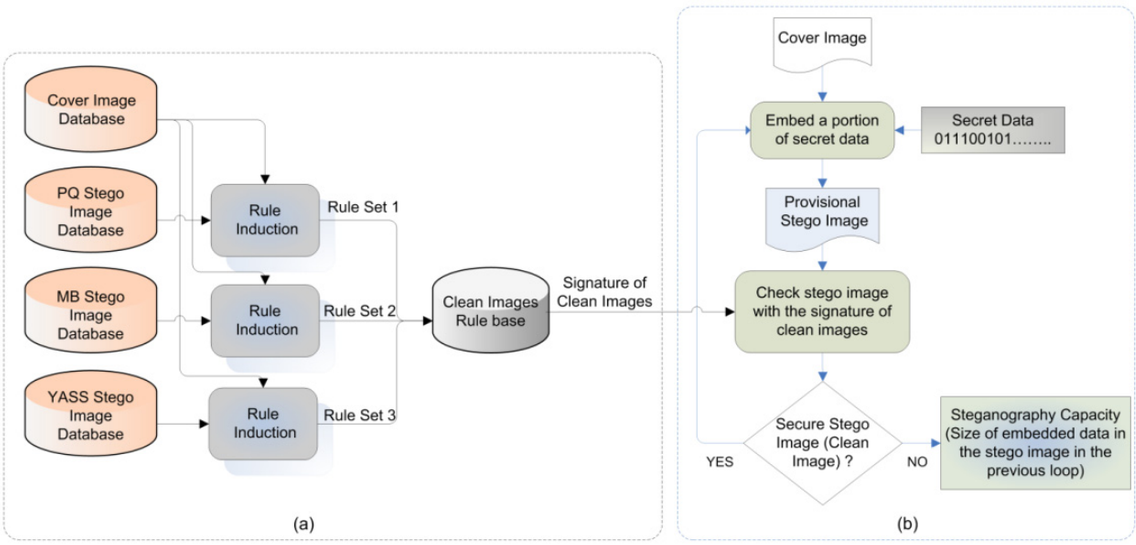


Fig. 1. The block diagram of steganography capacity estimation, (a) Extracting signature of clean images by evolutionary fuzzy rule induction, (b) Steganography capacity estimation by incremental embedding.

In our current problem we seek a function $\varphi : \mathbb{R}^d \mapsto R^{n1}$ revealing the signature of clean images (i.e. significant features with their values for cover images), where R demonstrates fuzzy rules and $n \ll d$.

The signature of clean images $CIS = \left\{ R_j = \{(x_i, y_i) \middle| x_i \in \mathbb{R}^d, y_i \in L^n, i \in \{1,...,l\}\} \middle| j \in \{1,...,J\} \right\}$ is a set of j fuzzy rules with length of i and L includes linguistic fuzzy values. Following subsections describe the details of this approach.

3.2 Feature extraction

We use the feature vector $X = \left\{ x_i \middle| x_i \in \mathbb{R}^d, i \in \{1,...,d\} \right\}$, which is produced by appending the features of four efficient and well-known steganalyzers. 636 features are computed according to the features of Pevny-Fridrich (Pevný & Fridrich, 2007), Chen et. al (Chen et. al, 2006), Lyu-Farid (Lyu & Farid, 2002), and Li et. al (Li et. Al, 2008) steganalysis methods. In the following, we briefly review the features used by these steganalyzers.

1. Pevny and Fridrich (Pevný & Fridrich, 2007) extract 274 features by merging 193 extended DCT features with 81 averaged calibrated Markov features. However, many of the 274 features may be highly correlated to each other. In this method, Markov features model intra block DCT dependencies and DCT features model inter block relations. In the rest of this chapter, we refer to this steganalysis method as 274-dim steganalyzer.
2. In (Chen et. al, 2006), Chen et. al proposed a steganalysis method that employs a 324-dimensional feature vector for analysis. It is based on statistical moments derived from both image 2-D array and JPEG 2-D array. This steganalyzer considers both the first order and the second order histograms. Consequently, the moments of 2-D characteristic functions are also used for steganalysis. In the following, this method is referred to as 324-dim steganalyzer.
3. Wavelet-based steganalysis method (Lyu & Farid, 2002), presented by Lyu and Farid, builds a model for clean images by using higher order statistics, and then shows the deviation of stego images from the constructed model. Quadratic Mirror Filters are used to decompose the image into wavelet domain, after which higher order statistics such as mean, variance, skewness, and kurtosis are calculated for each subband. The higher order statistics are calculated from wavelet coefficients of each high-frequency subband to form one group of features. Another group of features is similarly formulated from the prediction errors of wavelet coefficients of each high-frequency subband. We called this method WBS steganalyzer.
4. Yet Another Steganographic Scheme (YASS) (Solanki et al. 2007) is designed to be a secure JPEG steganographic algorithm. Attacking YASS is proposed in Li et. al (Li et. Al, 2008). The success of YASS is attributed to its innovation in embedding, i.e., hiding data in embedding cover blocks whose locations are randomized. However, the locations of the embedding host blocks are not randomized enough. Some locations in an image are possible to hold an entire embedding cover block and some locations are definitely not. Additionally, YASS employs a Quantization Index Modulation embedding strategy in order to enhance the robustness of the embedded data, which on the other hand introduces extra zero coefficients into the embedding cover blocks during data hiding. Consequently, statistical features extracted from locations, which are possible to hold embedding cover blocks are different from those locations, which are impossible to hold embedding cover blocks. Here we called this method YASS-steganalyzer.

Table 1 depicts the types of all the 636 features.

Feature Group	Number of Features	Feature Type
274	11	Global Histogram
	66	5 AC Histograms
	99	11 Dual Histograms
	1	Variation
	2	Blockiness
	25	Co-occurrence Matrix
	81	Markov Features
324	39	Histogram of Spatial Representation and Discrete Wavelet Transform (DWT) Representation.
	39	Histogram of Prediction Error and DWT of Error
	39	Histogram of JPEG Representation and its DWT.
	78	Horizontal 2-D Histogram of JPEG Representation and its DWT 2-D Histogram
	78	Vertical 2-D Histogram of JPEG Representation and its DWT 2-D Histogram
	78	Diagonal 2-D Histogram of JPEG Representation and its DWT.
	39	Histogram obtained from Prediction Error of JPEG representation and its DWT
24	24	Higher order statistics of each Wavelet subband.
14	14	A group of frequencies of zero rounded re-quantized DCT coefficients.

Table 1. Types of 636 image features

3.3 Fuzzy rule induction

We code every fuzzy if-then rule as a string and use the following symbols for denoting the six linguistic values: 1: don't care (*DC*), 2: small (*S*), 3: medium small (*MS*), 4: medium (*M*), 5: medium large (*ML*), 6: large (*L*). The fuzzy rules are as follows:

$Rule\ R_j : If\ (x_1\ \text{is}\ y_1\ \text{and}\ \ldots\ \text{and}\ x_n\ \text{is}\ y_n)\ then$ Image is clean with $CF = CF_j$.

where R_j is the label of the j^{th} fuzzy if-then rule, $\left\{x_i \middle| x_i \in \mathbb{R}^d\right\}$ are the features extracted from the observed image, $\left\{y_i \mid y_i \in L^n\right\}$ are linguistic values that represent *S*, *MS*, *M*, *ML*, *L*, and *DC*. CF_j is the certainty grade of fuzzy if-then rule R_j. Each fuzzy rule has a certainty grade that demonstrates the confidence of the rule about its antecedent part.

The membership function of each linguistic value is specified by homogeneously partitioning the domain of each feature into symmetric triangular fuzzy sets. The total

number of possible fuzzy if-then rules is 6^d (due to using six linguistic values) in case of d-dimensional feature vector. It is impossible to use all the 6^d fuzzy if-then rules in a single fuzzy rule base for large d (e.g. steganography capacity estimation based on d = 636 features). Therefore, the employed evolutionary method searches for a relatively small number of fuzzy if-then rules (e.g., J = 10 rules) with higher performance. By performance, we mean that the inducted fuzzy if-then rules should be able to show the pattern or signature of clean images with high accuracy. This signature is extracted according to the training samples of clean and stego images.

The outline of the employed evolutionary fuzzy method is presented in (Sajedi & Jamzad,2009b). We apply the following three steps to calculate the certainty grade of each fuzzy if-then rule:

Step 1: Calculate the compatibility of each training sample $X_p = (x_{p1}, x_{p2}, ..., x_{pn})$ with the fuzzy if-then rule R_j by the following product operation:

$$\mu_j(X_p) = \mu_{j1}(x_{p1}) \times \ldots \times \mu_{jl}(x_{pl}), \quad \begin{matrix} P = 1,2,\ldots,M \\ l \le n \end{matrix} \tag{3}$$

where $\mu_{ji}(x_{pi})$ is the membership function of i^{th} feature of p^{th} sample and M denotes the total number of samples.

Step 2: For clean and stego images, calculate the relative sum of the compatibility grades of training samples with fuzzy if-then rule R_j :

$$\beta_{Clean}(R_j) = \frac{\sum_{X_p \in Clean} \mu_j(X_p)}{N_{Clean}} \tag{4}$$

$$\beta_{Stego}(R_j) = \frac{\sum_{X_p \in Stego} \mu_j(X_p)}{N_{Stego}} \tag{5}$$

where $\beta_{Clean}(R_j)$ and $\beta_{Stego}(R_j)$ are the relative sum of the compatibility grades of training samples that represent clean and stego images, respectively. Note that N_{Clean} and N_{Stego} represent the number of clean and stego images that are being used as training samples.

Step 3: The grade of certainty CF_j is determined as follows:

$$CF_j = \frac{\left(\beta_{Clean}(R_j) - \beta_{Stego}(R_j)\right)}{\left(\beta_{Clean}(R_j) + \beta_{Stego}(R_j)\right)} \tag{6}$$

The employed evolutionary fuzzy algorithm learns fuzzy if-then rules by optimizing one fuzzy rule in each iteration of the algorithm. At first, all the training samples have the same weight and each individual in the algorithm is initialized by the feature vector of an image. In each iteration of the algorithm, the rule with highest fitness is considered as the output of

the iteration. Then the learning mechanism reduces the weight of those training samples that are learned correctly. Samples with higher weight are more significant in the training process. Therefore, the next rule induction cycle, searches for fuzzy rules that cover the training samples, which are uncovered by the rules obtained in previous iterations. In brief, the fuzzy rules that cover the training samples more than other rules are included in the final rule base.

Reducing the weight of training samples helps to aggregate different disciplines between features of training samples to form a perfect fuzzy rule base. In the above learning framework, we have used a fitness function in evolutionary process. It is computed according to equations (7) to (9).

$$f_P = \frac{\sum_{X_k \in Clean} w^k \mu_{R_j}(X_k)}{\sum_{X_k \in Clean} w^k} \tag{7}$$

$$f_N = \frac{\sum_{X_k \in Stego} w^k \mu_{R_j}(X_k)}{\sum_{X_k \in Stego} w^k} \tag{8}$$

$$fitness(R_j) = w_P f_P - w_N f_N \tag{9}$$

where, f_P is the rate of positive training samples covered by rule R_j (correctly covered), f_N is the rate of negative training samples covered by rule R_j (wrongly covered), w^k is a weight which reflects the frequency of the sample X_k in the training database, w_P is the weight of rule's positive power, and w_N is the weight of rule's negative power.

Each stego image database and clean image database are shown to the rule induction algorithm (we set the parameters of MB, PQ, and YASS to construct stego image databases with variety of payloads). Afterward, a clean image rule set is resulted considering the effects of a steganography method on images. We have three types of stego image databases (MB, PQ, and YASS), therefore, three sets of rules are inducted. Putting all the rules of clean images in a clean image rule base, we obtain the signature of clean images.

3.4 Determining steganography capacity of an image

To determine the steganography capacity of an image an incremental embedding routine is applied. It steadily increases embedding rate until the stego image does not move away from the signature of clean images. Figure 1(b) illustrates the block diagram of incremental embedding routine.

For a given rule base *S*, in order to determine steganography capacity of an image with feature vector $X_p = (x_{p1}, x_{p2}, \ldots, x_{pn})$ is reliable to host a secret data, two parameters τ_{Clean} and τ_{Stego} are computed using equations (10) and (11). After hiding the secret data in the cover image, τ_{Stego} is computed based on the features $X_{ps} = (x_{ps1}, x_{ps2}, \ldots, x_{psn})$ of the produced stego image.

$$\tau_{Clean} = \sum_{R_j \in S} \mu_j(X_p)\, CF_j \tag{10}$$

$$\tau_{Stego} = \sum_{R_j \in S} \mu_j(X_{ps})\, CF_j \tag{11}$$

Generally, equation (12) is valid for a pair of clean and stego images because the rule base S contains rules that are achieved from with regard to clean images. Consequently, a clean image is more compatible with these rules compared to its stego version. If τ, the difference of τ_{Clean} and τ_{Stego}, is lower than threshold T in equation (13), it means that the clean and stego images are not distinguishable and the security of the cover image is acceptable. Since we can say that a clean image is a stego image with hidden data size of zero, so for a cover image τ_{Clean} is equal to τ_{Stego}. While embedding in the image steadily and examining its features with the signature of clean images, τ_{Stego} decreases and so the value of τ increases little by little. Thus, for determining the steganography capacity of an images we embed in the image incrementally until $\tau < T$. Finally, the steganography capacity of the images is equal to the size of embedded data.

$$\tau_{Clean} \succ \tau_{Stego} \tag{12}$$

$$\tau = (\tau_{Clean} - \tau_{Stego}) \prec T \tag{13}$$

Considering capacity limit in steganography, the steganographer can embed in an image until the stego image is undetectable. To produce a secure stego image, the secret data is embedded in the image and the resulted stego image is evaluated according to the signature of clean images. If the stego image deviates from the signature significantly, the cover image is overloaded and the steganographer can reduce the payload.

4. Experiments

For signature extraction of clean images, we used Camera image database of Binghamton University (Cancelli et.al ,2008), which has 3164 grayscale images of size 512×512. To evaluate the performance of RABS approach, we obtained 1000 JPEG images from both Washington University image database (washington.edu) and Internet. All images are converted to grayscale and cropped to size of 512×512.

In this chapter, we measure the separation of feature vectors of cover and stego images by training a non-linear SVM classifier with RBF kernel and use the accuracy of steganalysis as a measure of detectability (Fridrich et.al ,2007). Detection accuracy being close to 50 implies nearly undetectable hiding, and as the detectability improves, detection accuracy increases towards 100. To make stego image databases, MB, PQ, and YASS steganography methods are employed. We set the parameters of these methods to different values to obtain three stego image databases with variety of payloads. Each stego database has 1000 stego images.

Our experiments were executed on a personal computer with a 2046 MB PIV processor using Matlab R2007.

In our implementation environment, the average time for incremental embedding in one image and determining the steganography capacity of the image is around 20 seconds. This time is around 2 minutes when the ABS is employed (Cordon et.al ,2004). As the result shows, RABS is less time-consuming than ABS. Therefore, using RABS a large secret data can be hidden in a cover image set more quickly compared to ABS. Since secure data embedding is the main goal of steganography and if any of the steganalyzers gets suspicious, the purpose of steganography is broken, it is worth to spend more time to make stego images more secure.

4.1 Performance of RABS

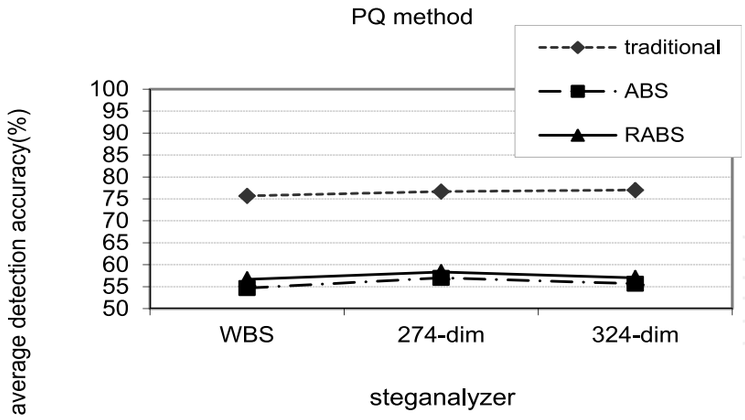
RABS can apply every existing steganography method. Table 2 shows the detection accuracy of four steganalyzers on the proposed RABS, ABS, and the traditional usage of PQ, MB, and YASS steganography methods. Employing WBS, 274-dim, 324-dim, and YASS steganalysis (denoted by YASS analyzer in the table) methods, to train the SVM classifier of each steganalyzer, 1200 (600 clean and 600 stego) images from the image database were used. The remaining 800 images are used for testing. As we see in Table 2, the results obviously show that the stego images, which are produced by RABS, are less detectable than the stego images constructed by traditional usage of steganography methods. In providing security for stego images, RABS approach operates close to ABS.

It should be noted that in ABS, the steganography capacity of an image is determined without considering the steganalysis method of YASS.

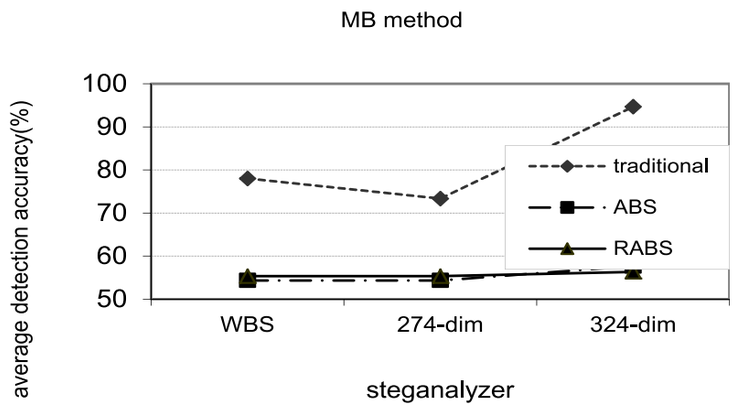
Figure 2 shows the average accuracy of steganalyzers in detection of stego images with different payloads produced by traditional usage of steganography methods, ABS, and RABS. In most of the cases, ABS' and RABS' undetectability are very close to each other. Both, ABS and RABS produce stego images with much higher security than traditional usage of steganography methods.

Steganography method	Average Payload (bits)	Steganalysis detection accuracy (%)											
		Traditional steganography method				ABS				RABS			
		WBS	274-dim	324-dim	YASS analyzer	WBS	274-dim	324-dim	YASS analyzer	WBS	274-dim	324-dim	YASS analyzer
PQ	2000	72	74	57	-	53	53	52	-	55	56	53	-
	6000	76	77	83	-	55	58	56	-	56	58	58	-
	10000	79	79	91	-	56	60	59	-	59	61	60	-
MB	2000	71	67	89	-	51	54	59	-	51	54	54	-
	6000	77	72	96	-	56	52	56	-	56	52	56	-
	10000	86	81	99	-	56	57	58	-	59	60	59	-
YASS	2000	55	57	59	72	52	56	57	72	52	55	55	56
	6000	62	63	57	86	56	58	57	86	59	59	57	61
	10000	61	69	65	97	59	60	59	97	59	60	59	65

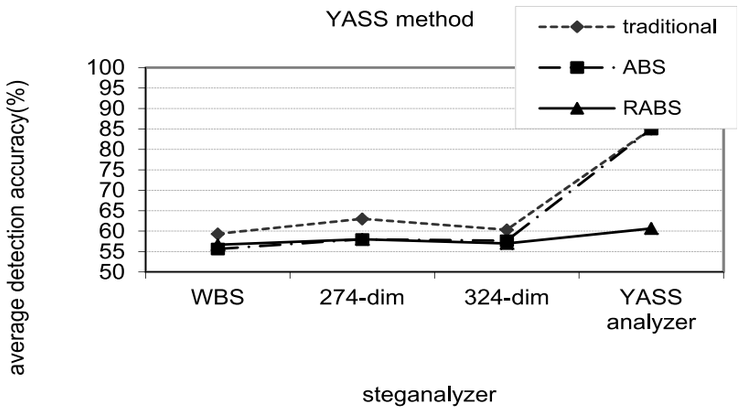
Table 2. Accuracy of steganalysis methods in detection of stego images produced by different steganography methods (in percent



(a)



(b)



(c)

Fig. 2. Average detection accuracy of steganalysis methods on (a) PQ, (b) MB, and (c) YASS steganography methods.

Figure 4 shows the relationship between the number of members in a cover image set and the total steganography capacity of it. White columns are computed by square root law, as described by Ker et al. in (Ker, 2007). Light gray columns show the average capacity of the cover image sets when their members are selected randomly. Dark gray and black columns show the average capacity of the cover image sets when their members are selected using ABS and RABS respectively from images having top 40% higher capacity in the database. As the Figure 4 shows, random selection columns are close to the square root law columns. In square root law, we suppose that cover images have equal steganography capacities. However, selection of cover images based on steganography capacity (as suggested by ABS and RABS methods) breaks this law.

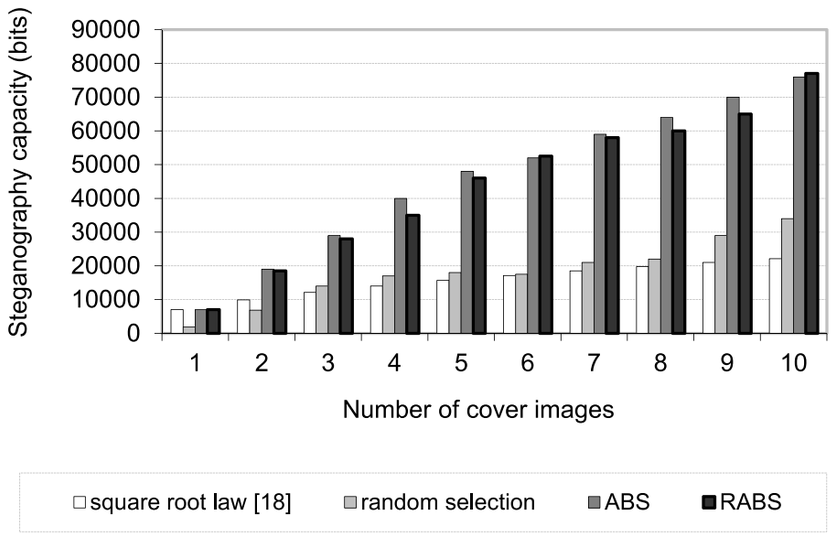


Fig. 3. Relationship between the number of images in cover image sets and total steganography capacity of them when the cover images are selected in random, using ABS, or RABS comparing to square toot law that shows the total steganography capacity in theory.

Figure 3 implies that the square root law of steganographic capacity that is true theoretically, is not always true in practice. We observed that the steganography capacity of a cover image set is approximately a constant, which is multiplied by the number of cover images in the set. In square root law, increasing the number of cover images in a set has a slight influence in total capacity of it. In both ABS and RABS approaches, the total steganography capacity of an image set is approximately the sum of the steganography capacity of all its members.

5. Conclusions

We investigated batch steganography, which is hiding secret data in more than one cover images. This chapter proposes a new adaptive batch steganography approach for hiding a large secret data in multiple cover images by defining and using the steganography capacity of images. Images have various properties due to their different contents. Therefore, for a certain size secret data they could result in stego images with unequal degree of undetectability. In this chapter, we proposed a novel approach to estimate the steganography capacity of images based on signature of clean images, which is achieved by analyzing the similarity between features of cover images. In this regard, an evolutionary fuzzy algorithm is employed to induct fuzzy if-then rules from features of clean images and form the signature of clean images. After discovering the signature of clean images, in the next step, the steganographer can selects the proper cover images from the database. A proper cover image is the one that after embedding, its effective features do not deviate from the signature of clean images. According to the obtained results, our approach reduces the detection rate of steganalyzers compared to the traditional use of steganography methods. The advantage of our proposed approach is that in appearance of new steganalyzer methods, the fuzzy rule base can be upgraded and thus the signature of clean images can become more trustable. By employment of our proposed RABS, one can hide a large secret data securely and quickly in an image set with the least number of images.

6. References

Cachin, C. (1998). An information-theoretic model for steganography, *Proceeding of 2nd Int. Workshop on Information Hiding,* LNCS 1525, pp. 306–318.

Cachin. C. (1998) An information-theoretic model for steganography, *Proceeding of 2nd International Workshop on Information Hiding*, vol. 1525, pp. 306-318.

Cancelli, G.; Doerr, G.; Cox, I. J. & Barni, M. (2008). Detection of ±1 LSB steganography based on the amplitude of histogram local extrema, *Proceeding of International Conference on Image Processing*.

Chandramouli, R. & Memon, N.D. (2003). Steganography Capacity: A Steganalysis Perspective, *Proceeding of SPIE Security and Watermarking of Multimedia Contents*, vol. 5020, pp. 173-177.

Chen, C.; Shi, Y. Q. & Xuan, G. (2006). Statistical Moments Based Universal Steganalysis Using JPEG-2D Array and 2-D Characteristic Function, *Proceeding of International Conference on Image Processing*, pp. 105-108.

Cordon, O.; Gomide, F.; Herrera, F.; Hofmann, F. & Magdalena, L. (2004). Ten years of genetic fuzzy systems current framework and new trends, *Fuzzy Sets and System*, pp. 5–31.

Fridrich, J.; Pevny, T. & Kodovsky, J. (2007). Statistically Undetectable JPEG Steganography: Dead Ends, Challenges, and Opportunities, *Proceeding of ACM Multimedia and Security Workshop*.

Fridrich, J.;Goljan, M. & Soukal, D. (2004). "Perturbed quantization steganography with wet paper codes," Proceeding of ACM, Multimedia Workshop, German.

http://www.cs.washington.edu/research/imagedatabase.

Hu, Y.; Chen, R. & Tzeng, G. (2003). Finding fuzzy classification rules using data mining techniques, *Pattern Recognition Letters*, pp. 509–519.

Ker, A. D. (2006). Batch steganography and pooled steganalysis, *Proceeding of Information Hiding Workshop*, vol. 4437, pp. 265–281.

Ker, A. D. (2007). A Capacity Result for Batch Steganography, *IEEE Signal Processing Let.* vol. 14, no. 8, pp.525-528, 2007

Ker, A. D. (2008). Perturbation Hiding and the Batch Steganography Problem, *Proceeding of 10th Information Hiding Workshop.*

Li, B.;. Shi, Y. Q . & Huang, J. (2008). Steganalysis of YASS, *Proceeding of ACM Multimedia and Security Workshop*, 2008.

Lyu, S. & Farid, H. (2002). Detecting hidden messages using higher-order statistics and support vector machines, *Proceeding of 5th International Workshop on Information Hiding.*

Moulin P. & Mihcak, M. K. (2002). A framework for evaluating the data hiding capacity of image sources, *IEEE Trans. Image Processing*, vol. 11, pp. 1029–1042.

Munuera,C. (2007). Steganography and error-correcting codes, *Signal Processing*, vol. 87, pp. 1528–1533.

Pevný T. & Fridrich, J. (2007). Merging Markov and DCT Features for Multi-Class JPEG Steganalysis, *Proceeding of SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents.*

Pevny, T. & Fridrich, J. (2008). Novelty Detection in Blind Steganalysis, *Proceeding of ACM Multimedia and Security Workshop*, pp. 167-176.

Pevny, T.; Fridrich, J. & Ker, A.D. (2009). From Blind to Quantities Steganalysis, *Proceeding of SPIE, Electronic Imaging, Security and Forensics of Multimedia Contents*, pp. 0C 1-0C 14.

Sajedi, H. & Jamzad, M. (2008). Adaptive Steganography Method Based on Contourlet Transform, *Proceeding of 9th International Conference on Signal Processing*, pp. 745-748.

Sajedi, H. & Jamzad, M. (2009). Evolutionary Rule Generation for Signature-based Cover Selection Steganography, *Neural Network World*, Special Issue on Intelligent Computing for Multimedia Assurance.

Sajedi, H. & Jamzad, M. (2009). Secure steganography based on embedding capacity, *Journal of Information Security*, Springer, vol. 8, no. 6, pp. 433-445.

Sajedi, H.& Jamzad, M. (2009). ABS: Adaptive batch steganography, *Journal of Optical Engineering*, SPIE Publishing, vol.48, no.8, pp. 087002-1– 087002-10.

Sallee, P. (2003).Model-based steganography, *Proceeding of International Workshop on Digital Watermarking*.

Solanki, K.; Sarkar, A. & Manjunath, B. S. (2007). YASS: yet another steganographic scheme that resists blind steganalysis, *Proceeding of 9th International Workshop on Information Hiding*.

Westfeld, A. (2001). F5-a steganographic algorithm: high capacity despite better steganalysis, *Proceeding of 4th International Workshop on Information Hiding*.

Wu Y. T., & Shih, F. Y. (2006). Genetic Algorithm Based Methodology for Breaking the Steganalytic Systems, *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 36, no. 1.

IntechOpen

IntechOpen

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen