Adaptive Steganography by Oracle (ASO)

Sarra Kouider, Marc Chaumont, William Puech

ADAPTIVE STEGANOGRAPHY BY ORACLE (ASO)

Sarra Kouider [#1], *Marc Chaumont* [§#2], *William Puech* [#3]

[§] University of Nîmes, F-30021 Nîmes Cedex 1, France

[#] LIRMM Laboratory, UMR 5506 CNRS, University of Montpellier II
161 rue Ada, 34095, Montpellier Cedex 05, France

{kouider,chaumont,puech}@lirmm.fr

ABSTRACT

HUGO [1] and MOD [2] are the most secure adaptive embedding algorithms of 2011. These algorithms strive to hide a secret message, while minimizing an ad hoc embedding impact. They use a detectability map, which, if properly defined, is correlated to the security. In this paper, we present a new adaptive embedding scheme: Adaptive Steganography by Oracle (ASO). It is based on an oracle used to calculate the detectability map, and this oracle use the *Kodovský's ensemble classifiers* [3]. Our approach preserves both cover image and sender's database distributions during the embedding process, which improves the security. In addition, it offers to the sender the opportunity to choose the most reliable image(s), during his secret communication. Experimental results show that our embedding scheme presents good security performances, since the detection error of ASO is much higher than that of HUGO.

Index Terms— Steganography, Steganalysis, Oracle, Detectability map, Security

1. INTRODUCTION

Steganography is the art of secret communication. The goal is to hide a secret message in an unsuspicious object in such a way that no one can detect it. With the Internet spread and the emergence of digital supports (audio files, videos or images), several philosophies of designing steganographic methods were proposed. One of the most usual embedding methods used with real digital images is the steganography by modification of the original cover object, which is based on the principle of minimizing embedding impact.

Let $\mathbf{x} = (x_1, ..., x_n)$ be a cover support composed of n elements. The goal of steganography by minimizing the embedding impact[1] is to communicate a secret message $\mathbf{m} = (m_1, ..., m_m)$ by making small perturbations of cover object $\mathbf{x}$ to produce a stego object $\mathbf{y} = (y_1, ..., y_n)$. For this, we define a distortion function $D(\mathbf{x}, \mathbf{y})$ that we minimize under the constraint of a fixed payload. This distortion function is generally based on the use of a detectability map $\rho \in \mathbb{R}_+^n$ which assigns to each cover element x_i with $i \in \{1, ..., n\}$, a detectability cost $\rho_i \in \mathbb{R}_+$ that models the impact on the security caused by the modification of the i^{th} element.

Most proposed steganographic algorithms by minimizing the embedding impact use the information of current cover image to calculate their detectability map. The HUGO[2] algorithm [1] used during the BOSS[3] competition [5] uses a detectability map, which attributes to each pixel of the cover image a detectability cost $\rho_i \in [0, \infty]$, as suggested in [6]. The calculation of detectability cost is based on the use of high-dimensional features, which are calculated from the cover image. These features correspond to the conditional probabilities in each pixel of the filtered image. The MOD[4] algorithm proposed in 2011 [7], extends the HUGO proposal by defining a parametric detectability cost $\rho_i \in [0, \infty]$, which is parametrized by a high number of parameters. Through the *downhill simplex* optimization algorithm, the search of the parameters leading to the highest level of security is performed by repeating iteratively message embedding, and parameter modification. The authors use at each iteration the size of the margin of SVM[5] as a criterion for evaluating the security level.

In this paper, we propose an Adaptive Steganographic scheme by Oracle (ASO) in spatial domain based on the use of an oracle for the computation of the detectability map. During the computation of the detectability map, ASO scheme takes into account not only the model distribution of the current cover image, but also the sender's database distribution. It thus preserves both of the distributions during the embedding process. Unlike Fillers approach [7] that uses a para-

[1] The principle of minimizing the embedding impact was proposed in 2007 [4]. It is based on the adaptivity of the embedding operation by the use of a detectability map.

[2] HUGO: Highly Undetectable steGO [1].

[3] BOSS (Break Our Steganography System) is the first challenge on Steganalysis. The challenge started the September 9th 2010 and ended the 10th of January 2011. The goal of the player was to figure out, which images contain a hidden message and which images do not. The steganographic algorithm was HUGO [1]. http://www.agents.cz/boss/BOSSFinal/.

[4] MOD: Model Optimized Distortion [2].

[5] SVM: Support Vector Machine.

metric method to reduce the SVM margin separating the covers and the stegos, we propose a non-parametric method that uses the *Kodovský's ensemble classifiers* [3] as an oracle to calculate the detectability map. Thus, both cover image information and sender's database information are fully exploited. Moreover, the proposed ASO approach manages to resolve the complexity problem of [7] when using high-dimensional feature spaces. It has a good numerical stability and scales well as the feature space increases, which is not the case with MOD [7].

One should point out that even if the proposed scheme can be confused with the FCM[6] approach [8], these two embedding algorithms are very different. The FCM approach strives to preserve the model distribution of the original cover image by performing a feature restoration of the used model. The DCT coefficients of the cover JPEG image are split into two disjoint sets. The first set is used to embed the secret message, while the second set is used to restore the feature vector, by making additional modification. In contrast to FCM, ASO brings the stego image distribution closer to cover images distribution. There is no features restoration as FCM. The detectability cost calculated for each pixel fosters the modification of pixels implying a displacement of the stego feature vector towards the covers distribution. Through this paper we introduce a general methodology for oracle based approaches. Of course, such an approach may suffer from the incompleteness of the used feature space [8], but the algorithm may easily been improved by increasing the number of well chosen feature sets.

The rest of this paper is organized as follows. In Section 2, we recall some preliminary notions. In Section 3, we introduce and describe our adaptive steganographic algorithm by oracle. The experimental results are given in Section 4. The paper is concluded in Section 5 with a discussion of possible future directions.

2. PRELIMINARIES

In this section, we recall some fundamental concepts. For sake of simplicity, we denote by $\mathbf{x} = (x_1, ..., x_n) \in \mathcal{X} = \{0, ..., 255\}^n$ and $\mathbf{y} = (y_1, ..., y_n) \in \mathcal{Y} = \{0, ..., 255\}^n$ grayscale cover and stego images with n pixels.

2.1. Minimizing embedding impact

All practical steganographic algorithms by minimizing embedding impact strive to hide a given message $\mathbf{m} = \{0, 1\}^m$ in a cover support $\mathbf{x}$, while minimizing an ad hoc embedding impact [4, 6]. To achieve this goal, it is important to establish a distortion measure D that can model the statistical detectability caused by the embedding.

The authors in [6] propose to model the embedding impact by an additive function $D : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$ that is defined by:

$$D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \rho_i |x_i - y_i|, \tag{1}$$

where $0 \leq \rho_i \leq \infty$ is the cost of changing the i^{th} cover pixel x_i to y_i, and such that the additivity of the distortion function D implies that the embedding changes do not interact between each other. In other words, the modification of a cover element does not affect the detectability of neighboring elements.

For an additive distortion function (Eq. 1), and binary embedding changes, i.e. $|x_i - y_i| \leq 1$, the solution to the problem of minimizing embedding impact, under the payload-limited sender constraint, takes the following form [4]:

$$\min D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} p_i \rho_i, \tag{2}$$

where p_i is the probability of modification of the i^{th} pixel is defined by [4]:

$$p_i = \frac{e^{-\lambda \rho_i}}{1 + e^{-\lambda \rho_i}}. \tag{3}$$

The parameter λ is obtained by solving the following equation:

$$-\sum_{i=1}^{n} \Big(p_i \log_2 p_i + (1 - p_i) \log_2 (1 - p_i) \Big) = m. \tag{4}$$

This formalization of adaptive steganography by minimizing embedding impact allows us to split the embedding process into two successive steps: a) the computation of a detectability map, and b) the embedding step by a practical adaptive algorithm. The major advantage resulting from this separation is that the security evaluation of the detectability map does not require to use a classical algorithm for the embedding. In practice, if we want to insert a given message while minimizing the embedding impact (which means that the detectability map ρ is known) under the fixed-payload constraint, it is possible to simulate the optimal embedding by: a) looking for the parameter λ (Eq. 4), and then b) flipping each pixel x_i with probability p_i as defined in Eq. 3.

2.2. The Kodovský's ensemble classifiers

Modern steganographic schemes, such as HUGO [1], are designed to approximately preserve a high-dimensional representation of covers[7], which constitutes a real problem for steganalysis. To address the curse of dimensionality, Kodovský *et al.* [3] propose a new *machine learning tool* alternative to steganalysis tools such as SVM or neural networks. Their classifier is composed by a set of weak classifiers of low computational complexity. They use for learning and classification a set $\mathcal{F} = \{F_1, ..., F_L\}$ of binary FLD[8] classifiers.

[6] FCM: Feature Correction Method.

[7] The dimensionality of the feature set used by HUGO is about 10^7.

[8] FLD: Fisher Linear Discriminant.

Let $\mathcal{A} = \{\mathbf{f}_i, c_i\}_{i=1}^{i=N}$ be a training base of size N with cover and associated stego images, where $\mathbf{f}_i \in \mathbb{R}^d$ is a vector of dimension d characterizing the i^{th} image, and $c_i \in \{0, 1\}$ is the associated class number (0 for a cover image, and 1 for a stego image).

During the learning phase, each FLD classifier learns to associate to each feature vector $\mathbf{f}_i$, the correct class number c_i:

$$\begin{array}{rccc} F_l: & \mathbb{R}^d & \rightarrow & \{0,1\} \\ & \mathbf{f}_i & \rightarrow & F_l(\mathbf{f}_i). \end{array}$$

For this, each FLD classifier uses the training base $\mathcal{A}$ to calculate the vector $\mathbf{w}^{(l)}$ orthogonal to the hyperplane separating the two classes.

One should note that each FLD classifier performs its learning on a subspace of d_{red} dimension, with $d_{red} \ll d$. In practice, each classifier pseudo-randomly selects some features from the feature vector $\mathbf{f}_i \in \mathbb{R}^d$.

The classification of an input observation $\mathbf{f} \in \mathbb{R}^d$ is made by merging all the votes of the FLD classifiers. The final decision is obtained by a majority vote [3]:

$$\begin{array}{rccc} R: & \mathbb{R}^d & \rightarrow & \{0,1\} \\ & \mathbf{f} & \rightarrow & R(\mathbf{f}), \end{array}$$

$$\text{where:} \quad R(\mathbf{f}) = \begin{cases} 1 & \text{if} \quad \sum_{l=1}^{L} F_l(\mathbf{f}) > L/2, \\ 0 & \text{otherwise.} \end{cases}$$

The decision threshold of the FLD base learners is adjusted to minimize the total detection error, under equal priors on the training data [3]:

$$P_E = \min_{P_{FA}} \frac{1}{2} \left(P_{FA} + P_{MD}\left(P_{FA}\right) \right),$$

where P_{FA} and P_{MD} are the probabilities of false alarm and missed detection. In this paper, the probability of error (P_E) is used to report the efficiency of detection.

3. THE ASO SCHEME

3.1. The detectability map

Our steganographic strategy is based on the principle of minimizing embedding impact (section 2.1). The adaptive steganographic algorithm by oracle (ASO), that we propose is based on the adaptivity of the embedding by the use of a detectability map $\rho = \{\rho_i \in [0, \infty[\}_{i=1}^{n}$, which is calculated by an oracle. The functionalities of *Kodovský's ensemble classifiers* [3] and the acquired information of the learning phase are exploited during the computation of the detectability map. By using the information of the sender's database, we seek to increase the security of the embedding process.

Let us consider a grayscale cover image $\mathbf{x} = (x_1, ..., x_n)$ with n pixels, a feature vector $\mathbf{f_x}$ characterizing the image $\mathbf{x}$, an additive distortion function D as defined in Eq. 1, and an adaptive *LSB-matching*[9] embedding.

We wish to calculate the detectability map $\rho \in \mathbb{R}^n$ that assigns a detectability cost ρ_i to each pixel x_i. To do this, we define the detectability cost ρ_i as in HUGO [1] by:

$$\rho_i = min\left(\rho_i^{(+)}, \rho_i^{(-)}\right), \tag{5}$$

with $\rho_i^{(+)}$ (resp. $\rho_i^{(-)}$) the detectability cost of changing the i^{th} pixel by $+1$ (resp. -1).

We propose to calculate the detectability cost $\rho_i^{(+)}$ (resp. $\rho_i^{(-)}$) thanks to an oracle that is made of L *Kodovský's* FLD classifiers [3]. For this, we define the detectability cost $\rho_i^{(+)}$ (resp. $\rho_i^{(-)}$) as an unweighted sum of the detectability cost $\rho_i^{(l)}$ of each classifier F_l, $l \in \{1..,L\}$:

$$\rho_i^{(+)} = \sum_{l=1}^{L} \rho_i^{(l)(+)}, \qquad \text{and} \qquad \rho_i^{(-)} = \sum_{l=1}^{L} \rho_i^{(l)(-)}, \tag{6}$$

with $\rho_i^{(l)(+)}$ (resp. $\rho_i^{(l)(-)}$) the detectability cost provided by the l^{th} classifier.

For a classifier F_l, $l \in \{1, ..L\}$, we define the detectability cost $\rho_i^{(l)(+)}$, $l \in \{1.., L\}$, as:

$$\begin{aligned} \rho_i^{(l)(+)} &= \frac{\mathbf{w}^{(l)}.{\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(+)} - \mathbf{w}^{(l)}.{\mathbf{f}_{\mathbf{x}}}^{(l)}}{s^{(l)}} \\ &= \frac{\mathbf{w}^{(l)}.\left({\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(+)} - {\mathbf{f}_{\mathbf{x}}}^{(l)}\right)}{s^{(l)}}, \end{aligned} \tag{7}$$

In the same way, the detectability cost $\rho_i^{(l)(-)}$ is defined by:

$$\rho_i^{(l)(-)} = \frac{\mathbf{w}^{(l)}.\left({\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(-)} - {\mathbf{f}_{\mathbf{x}}}^{(l)}\right)}{s^{(l)}}, \tag{8}$$

with $s^{(l)} \in \mathbb{R}_+$ the normalization factor of the l^{th} classifier (F_l), $\mathbf{w}^{(l)}$ the vector orthogonal to the hyperplane separating the two classes calculated by the classifier F_l, ${\mathbf{f}_{\mathbf{x}}}^{(l)}$ the feature vector that we wish to classify by the classifier F_l, and ${\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(+)}$ (resp. ${\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(-)}$) the feature vector obtained after the modification of the i^{th} pixel by $+1$ (resp. -1).

Our goal is to obtain a low value of $\rho_i^{(l)(+)}$ (resp. $\rho_i^{(l)(-)}$), when the modification of pixel by +1 (resp. -1) causes a displacement $({\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(+)} - {\mathbf{f}_{\mathbf{x}}}^{(l)})$ (resp. $({\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(-)} - {\mathbf{f}_{\mathbf{x}}}^{(l)})$) towards the cover class. The strong assumption of our ASO scheme is that changing pixel brings closer the stego image to the cover images. By construction, the vector $\mathbf{w}^{(l)}$ is always directed in the direction cover to stego. By using Eq. 7 and Eq. 8 for computing $\rho_i^{(l)(+)}$ and $\rho_i^{(l)(-)}$, we get exactly the required behavior. Indeed, the detectability costs $\rho_i^{(l)(+)}$ and

[9] *LSB-matching*: Modification of each pixel by ± 1. It has been proved that this trivial modification of the LSB replacement is much harder to detect [9].

$\rho_i^{(l)(-)}$ are minimal when the vectors $\mathbf{w}^{(l)}$ and $({\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(+)} - \mathbf{f}_{\mathbf{x}}{}^{(l)})$ (resp. $({\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(-)} - \mathbf{f}_{\mathbf{x}}{}^{(l)})$) are co-linear and opposite in directions, i.e, when:

$$\mathbf{w}^{(l)}.\left({\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(+)} - \mathbf{f}_{\mathbf{x}}{}^{(l)}\right) < 0 \text{ or when } \mathbf{w}^{(l)}.\left({\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(-)} - \mathbf{f}_{\mathbf{x}}{}^{(l)}\right) < 0. \tag{9}$$

Furthermore, one can easily understand the necessity of a scaling factor for each classifier. Since the detectability costs are obtained by an unweighted sum of $\rho_i^{(l)(+)}$ or $\rho_i^{(l)(-)}$, each FLD classifier must provide a detectability cost $\rho_i^{(l)(+)}$ or $\rho_i^{(l)(-)}$ of the same order of magnitude.

Thus in order to bring the values within the interval [0,1], we define the scale factor $s^{(l)}$ as:

$$s^{(l)} = (\mu_1^{(l)} - \mu_0^{(l)})\mathbf{w}^{(l)} + 2(\sqrt{\mathbf{w}^{(l)T}\Sigma_0^{(l)}\mathbf{w}^{(l)}} + \sqrt{\mathbf{w}^{(l)T}\Sigma_1^{(l)}\mathbf{w}^{(l)}}) \tag{10}$$

with $\mu_0^{(l)}$ (resp. $\mu_1^{(l)}$) the mean vector of the cover (resp. stego) class, and $\Sigma_0^{(l)}$ (resp. $\Sigma_1^{(l)}$) the covariance matrix of the cover (resp. stego) class.

Note that for an FLD classifier, the vector $\mathbf{w}^{(l)}$ and the normalization factor $s^{(l)}$ are calculated during the learning phase. We thus do not need to calculate them during the computation of $\rho^{(l)(+)}$ and $\rho^{(l)(-)}$ (Eq. 7 and Eq. 8). The computational complexity of the construction of the detectability map, only comes from the computation of ${\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(+)}$ and ${\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(-)}$. To address this problem, we do not calculate separately the feature vectors ${\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(+)}$ and ${\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(-)}$, we only calculate, on a reduced image area, the variation $({\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(-)} - \mathbf{f}_{\mathbf{x}}{}^{(l)})$ and $({\mathbf{f}_{\mathbf{x}\sim x_i}}^{(l)(+)} - \mathbf{f}_{\mathbf{x}}{}^{(l)})$ introduced by the modification +1 or -1 of the i^{th} pixel.

At the end of the process, we obtain a detectability map $\rho \in \mathbb{R}$ consisting of positive and negative values. To get a positive detectability map $\rho = \{\rho_i \in [0, \infty[\}_{i=1}^n$, we translate the set of the detectability costs by $\rho_{min} = min(\rho)$, where ρ_{min} is the smallest cost of the detectability map ρ. The final detectability map can then be used for the embedding process, which will be either simulated as explained in section 2.1 (Eq. 3 and Eq. 4) or, done by using the STC[10] approach [6].

3.2. The proposed adaptive embedding scheme

Through our adaptive steganographic scheme by oracle (ASO), we propose a new concept of steganography, which is *the steganography by database*. During the embedding process, the proposed algorithm takes into account not only the model distribution of the cover image, but also the distribution of the sender's database. Moreover, it allows us to obtain a set of stego images, at the output of system, instead of just one image. During the transmission, the sender can then choose the most secure image(s) to communicate his secret message.

[10]STC: Sydrome Trellis Codes.

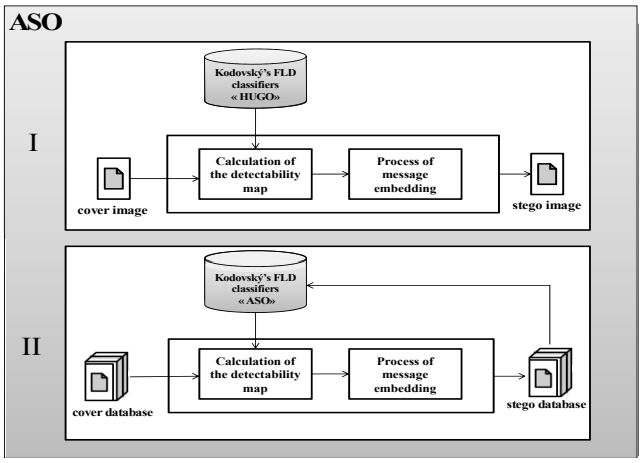

Fig. 1. General scheme of the Adaptive Steganography by Oracle (*ASO*).

As illustrated in Figure 1, the adaptive steganographic scheme by oracle is made of two steps:

The first step (labeled I in Figure 1), uses during the calculation of the detectability map (section 3.1) a set of FLD classifiers that have previously learn to distinguish between the cover images and the HUGO's stego images. Once calculated, the detectability map is then used to insert the secret message. This operation is performed by simulating the optimal embedding algorithm (section 2.1 Eq. 3 and Eq. 4). At the output of this first step we obtain a stego image ASO.

The second step (labeled II in Figure 1), is an iterative step that aims to increase the undetectability of the message. At each iteration, the computation of the detectability map is performed using a set of FLD classifiers that are trained to distinguish between the cover images and the ASO's stego images obtained during the previous iteration. The iterative process is repeated until the desired probability of error of classification is obtained. At the output of this second step, we obtain a database of ASO's stego images.

During the transmission, the sender can choose the most reliable image(s) for the transmission of his secret data. The transmission of his message is performed in a concrete manner by using the STC approach [6].

This paper is focused on the study of the first step of ASO (step I in Figure 1). The study of the iterative second step (step II in Figure 1) will be treated on a future work. The design of the first step of ASO was achieved using our own implementation of the *Kodovský's ensemble classifiers* [3] with $d_{red} = 250$, and $L = 30$ classifiers. The oracle learning phase was conducted on the BossBase v1.00 database[11] consisting of 10 000 512×512 grayscale cover images in the pgm format. 5000 covers and 5000 HUGO's stego images are

[11]BOSSBase v1.00: images database available on http://agents.cz/boss/BOSSFinal/.

Table 1. Parameters of the used MINMAX [10] features. s: the span of the difference used to compute the residual, q: the quantization step, m: the order of the co-occurrence matrices, T: the truncation threshold, and d: the resulting dimension.

s	q	m	T	d
3	2	3	3	686
3	2	4	2	1250
3	2	3	4	1458
4	2	3	3	686
4	2	4	2	1250

used during the learning, with the payload fixed by the user. To keep a balance between optimality and performance, we choose to represent each image by a vector of $d = 5330$ MINMAX features [10]. We highlight that although the small size of this feature vector, it has proven its effectiveness against the HUGO algorithm [10]. The parameters of the used MINMAX feature sets and their corresponding dimension are given in Table 1.

Finally we note that our implementation of the embedding process was parallelized by using the OpenMP[12] library. This was achieved using an architecture made of 8 processors *Quad-Core AMD Opeteron(tm) Processor 8384*, at 2.69 GHz; and all 32 cores are used.

4. EXPERIMENTAL RESULTS

First of all, recall that: a) the ASO's oracle construction, is made by using the 5330 MINMAX features, b) during the embedding process we use our own C++ ensemble classifiers with the fixed parameters $L = 30$ FLD classifiers, and $d_{red} = 250$ for the random subspace dimension (seeds are different for each weak FLD classifier), c) the oracle learning phase is performed on 5000 (BossBase-v1) covers and 5000 associated HUGOs stego images, and d) our embedding algorithm ASO takes 10000 covers (BossBase-v1) and generates 10000 stego images.

To test the performance of ASO, we compared the security of ASO[13] with that of HUGO, for five different payloads from 0.1 bpp to 0.5 bpp. For this, we steganalyzed ASO and HUGO using the original *Kodovský's ensemble classifiers* framework[14] with fully automatized search for d_{red} and L [3] ($L \neq 30$, $d_{red} \neq 250$, and seeds are different for the random subspaces). The experiments were done using the BossBase v1.00 database consisting of 10000 covers and the associated stego images. Each image was represented using the Spatial domain Rich Model SRMQ1 [11] made of 12753 features, the features are a merge of SPAM, MINMAX, SQUARE, and EDGE submodels with the fixed quantization $q = 1$ [11]. For each payload, the database is divided randomly into two halves training/testing images. The division is repeated five times with different seeds. The final performance (P_E) is averaged over the five splits.

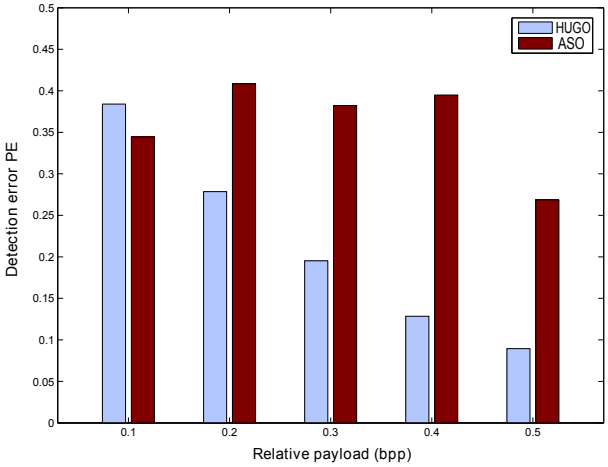

Fig. 2. Detection error P_E of ASO and HUGO for five relative payloads, using the SRMQ1 [11] model of 12753 dimension.

To summarize, the oracle and the steganalysis detector are different, the features and subspaces are also different.

Figure 2 reports the obtained results. We can notice that, for the payloads from 0.2 bpp to 0.5 bpp, the security of ASO is better than the HUGO's security. For instance, at 0.5 bpp, the detection error, P_E, of ASO is about 26.87% whereas it is only about 8.94% for HUGO. There is thus a difference of 17.93%. Similarly, the detection error, P_E, of ASO at 0.4 bpp is greater than that of HUGO. It is about 39.49%, compared with 12.85% for HUGO. We can say that ASO allows the sender to hide long messages with a better security than HUGO.

For small payloads, the ASO algorithm is less efficient than HUGO. At 0.1 bpp, the detection error, P_E, is about 34.45%, whereas it is about 38.40% for HUGO. For such payload, the oracle used for computing the detectability map (section 3.1), probably does not allow to distinguish between the cover and stego images. This thus influences the ASO's clairvoyance. In other words, ASO can not distinguish between secure and insecure areas.

Note that, between 0.1 bpp and 0.5 bpp, the ASO's behaviour is non-monotonous. We also observed this behavior by steganalyzing ASO with the 5330 MINMAX [10] features. At 0.4 bpp, the detection error P_E of ASO is greater than that at 0.3 bpp. It is about 39.49% for 0.4 bpp, compared with 38.22% for 0.3 bpp. This behavior is counter-intuitive and we would expect a monotonous decreasing of P_E. The first possible explication of those fluctuating performances is that we fixed $d_{red} = 250$ and $L = 30$ in our oracle C++ imple-

[12]OpenMP: a parallel programming library, available on http://openmp.org/wp/.

[13]In this test only one iteration is performed with an oracle that is trained to distinguish between covers and HUGO's stego images (Step I in Figure 1).

[14]The Kodovský's ensemble classifiers source is available on http://dde.binghamton.edu/download/ensemble/.

mentation. These parameters should probably not be the same for each payload. This implies that our approach may be improved in the future by the automatic search of the best parameters d_{red} and L. The additional explication about those fluctuating performances is that we have two contradictory expected effects: the less the payload is, the less the embedding process is detectable (because we modifies fewer pixels (or coefficients)), and at the same time, the less the payload is, the less is the oracle reliability, and thus the more ASO is detectable (because the oracle incorrectly fills the detectability map).

To summarize, the results show that ASO outperforms HUGO against the steganalysis with the Rich Model SRMQ1 [11] of 12753 features. This confirms that the ASO's stego images are build in such a way that their distribution is as close as possible to the cover class. In other words, the boundary separating the two classes cover/stego is kept to be very thin, which makes the detection a difficult task. Moreover, unlike HUGO, the ASOs embedding strategy is different from one image to another, which represents a real problem for machine learning algorithms.

Finally, one should remark that the completeness of ASO can not be guaranteed against other attacks (other feature sets). However, we stress that through this paper, we present a general concept that can be adapted for any other embedding algorithm, and that can be improved by using a more well chosen complete cover model [8, 11]. Moreover, we believe that additional iterations during the embedding process (step II in Figure 1) could significantly improve the security.

5. CONCLUSION

In this paper, we present an Adaptive Steganographic scheme by Oracle (ASO) that uses an oracle to calculate a detectability map. The *Kodovský's ensemble classifiers* [3] allows to split the features space in two regions (cover and stego regions). We use this separation as an oracle in order to define the detectability costs. Our detectability map is then defined such that changing pixels must bring closer the stego image to the cover images. Thus our proposed embedding scheme does not only strive to preserve the model distribution of the current cover image, but also preserves the model of the sender's database. It thus improves the security of the embedding process. An additional security feature of ASO, that can be point out, is that during the transmission phase, ASO allows the sender to choose the most undetectable stego image(s). Experimental results show that our embedding scheme presents good security performances. With only one iteration, ASO allows the embedder to hide long messages with a better security than HUGO. Future work will be focused on: (1) the problem of finding the exact number of iterations needed to improve the security of ASO, (2) extending the ASO scheme by working with more diverse feature spaces, and (3) the problem of the feature set completeness [2].

6. ACKNOWLEDGMENT

This work was supported by the "Ministry of Higher Education and Scientific Research of People's Democratic Republic of Algeria".

7. REFERENCES

[1] T. Pevný, T. Filler, and P. Bas, "Using High-Dimensional Image Models to Perform. Highly Undetectable Steganography," in *Information Hiding - 12th International Conference*, Berlin, Heidelberg, October 01 2010, vol. 6387 of *Lecture Notes in Computer Science, IH'10*, pp. 161–177, Springer-Verlag.

[2] J. Kodovský, J.J. Fridrich, and V. Holub, "On Dangers of Overtraining Steganography to Incomplete Cover Model," in *Multimedia and Security Workshop, MM&Sec '11 Proceedings of the thirteenth ACM multimedia*, Buffalo, NY, USA, September 29 - 30 2011, pp. 69–76, ACM.

[3] J. Kodovský, J.J. Fridrich, and V. Holub, "Ensemble Classifiers for Steganalysis of Digital Media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

[4] J.J. Fridrich and T. Filler, "Practical Methods for Minimizing Embedding Impact in Steganography," in *Security, Steganography, and Watermarking of Multimedia Contents IX, part of IS&T SPIE Electronic Imaging Symposium*, San Jose, CA, January 29-February 1 2007, vol. 6505, pp. 02–03.

[5] P. Bas, T. Filler, and T. Pevný, "Break Our Steganographic System — the ins and outs of organizing BOSS," in *Information Hiding - 13th International Workshop*, Prague, Czech Republic, May 18-20 2011, vol. 6958 of *Lecture Notes in Computer Science, IH'11*, pp. 59–70, Springer-Verlag.

[6] T. Filler, J. Judas, and J.J. Fridrich, "Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3-2, pp. 920–935, 2011.

[7] T. Filler and J.J. Fridrich, "Design of Adaptive Steganographic Schemes for Digital Images," in *Media Watermarking, Security, and Forensics XIII, part of IS&T SPIE Electronic Imaging Symposium*, San Francisco, CA, January 23-26 2011, vol. 7880, paper. 13, pp. F 1–14.

[8] J. Kodovský and J.J. Fridrich, "On Completeness of Feature Spaces in Blind Steganalysis," in *Multimedia and Security Workshop, MM&Sec '08 Proceedings of the tenth ACM multimedia*, Oxford, UK, September 22 - 23 2008, pp. 123–132, ACM.

[9] G. Cancelli, G. Doërr, M. Barni, and I.J. Cox, "A Comparative Study of ± 1 Steganalyzers," in *IEEE International Workshop on Multimedia Signal Processing*, 2008, pp. 791–796.

[10] J.J. Fridrich, J. Kodovský, V. Holub, and M. Goljan, "Breaking HUGO the Process Discovery," in *Information Hiding - 13th International Conference*, Tomás Filler, Tomás Pevný, Scott Craver, and Andrew D. Ker, Eds., Prague, Czech Republic, May 18-20 2011, vol. 6958 of *Lecture Notes in Computer Science, IH'11*, Springer.

[11] J.J. Fridrich and J. Kodovský, "Rich Models for Steganalysis of Digital Images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.