# Explicit Optimization of min max Steganographic Game

Solène Bernard, Patrick Bas, John Klein, Tomáš Pevný

# Explicit Optimization of $\min\max$ Steganographic Game

Solène Bernard, Patrick Bas, *Senior Member, IEEE,* John Klein, *Senior Member, IEEE,* and Tomáš Pevný, *Member, IEEE,*

***Index Terms*—Steganography, Steganalysis, Game Theory, Distortion function**

***Abstract*—This paper proposes an algorithm which allows Alice to simulate the game played between her and Eve. Under the condition that the set of detectors that Alice assumes Eve to have is sufficiently rich (e.g. CNNs), and that she has an algorithm enabling to avoid detection by a single classifier (e.g adversarial embedding, gibbs sampler, dynamic STCs), the proposed algorithm converges to an efficient steganographic algorithm. This is possible by using a $\min\max$ strategy which consists at each iteration in selecting the least detectable stego image for the best classifier among the set of Eve's learned classifiers. The algorithm is extensively evaluated and compared to prior arts and results show the potential to increase the practical security of classical steganographic methods. For example the error probability $P_{err}$ of XU-Net on detecting stego images with payload of 0.4 bpnzAC embedded by J-Uniward and QF 75 starts at 7.1% and is increased by +13.6% to reach 20.7% after eight iterations. For the same embedding rate and for QF 95, undetectability by XU-Net with J-Uniward embedding is 23.4%, and it jumps by +25.8% to reach 49.2% at iteration 3.**

## I. Introduction

Since the formal definition of the prisoners' problem by Simmons [1], steganography and steganalysis have been considered as a *hide and seek* game, where the goal of the steganographer, a.k.a. Alice, is to embed messages into cover objects while being undetectable by the steganalyst, a.k.a. Eve. For image steganography, Alice embeds a message by modulating either pixels or quantized DCT coefficients for JPEG images. As opposed to Alice, Eve wishes to detect the presence of a possibly hidden message using an image model, which can be either built using machine learning techniques or by domain knowledge (see [2] for an introduction to history of steganography and steganalysis). Historically, this game played in academia led to a succession of better and better steganography schemes (for example for spatial domain steganography we can cite the evolution from LSB replacement, to LSB matching, Hugo [3], S-Uniward [4], HILL [5], MiPod [6]) but also to better steganalysis methods (e.g. Sample-pair steganalysis [7], SPAM features [8], SRM features [9] and deep-learning methods such as Yedrouj-Net [10], XU-Net [11] or SRNet [12]).

In this paper, we ask ourselves if playing this cat and mouse game virtually without human intervention can lead to a more secure steganographic algorithm. We answer by the affirmative in this paper and show that it is possible to achieve this goal by using convolution neural networks (CNNs) which encompass a general and very complete class of detectors. The experimental results support this finding, since the error of a detector implemented by XU-Net detecting messages with payload 0.4 hidden in JPEGs QF 95 has increased to $49.2\%$, which is by $25.8\%$ more than J-Uniward, presently still considered to be a state of the art.

On an historical note, the possibility of playing the steganographic game explicitly has been already investigated but it was rarely played (exceptions known to us are listed in subsection I-A on related works), as attempts to explicitly defeat a particular (set of) detector(s) were considered as an extremely dangerous practice due to the belief that fixing one security hole might generate other unknown ones. For example in the JPEG domain, Model-Based steganography [13], contrary to F5 [14], was designed in order to offer no statistical distortion between marginal distributions of DCT coefficients before and after embedding, but was more detectable than F5 using features capturing joint distributions such as SPAM features [8].

This paper therefore investigates a trend, which we believe will become prevalent in upcoming years, where a generic class of steganalyzers are used to implicitly design a distortion function used during the embedding of a message. The security of the resulting steganographic algorithm depends on the assumptions that (i) the class of detectors used by Alice to learn the image model is general enough to not offer other security holes[1], (ii) Alice knows how to embed a message while being undetected by a single detector.

### A. Relation to prior art

Most state of the art steganographic algorithms relies on adaptive embedding [16], where each image coefficient is modified according to its assumed impact on the detectability. This impact is captured in so-called embedding costs, which are designed to be approximately additive [17].

A wide class of algorithms compute these costs using heuristic principles, for example in HILL [5], UNIWARD [4] or UERD [18]. In these algorithms, the formula used to compute costs has free parameters which are empirically tweaked in order to maximize the detection error w.r.t a large set of detectors. Notice that although the game between Alice and Eve is not formally defined, it is actually played through

[1] We believe this holds reasonably well for state-of-the-art deep-learning schemes such as [15] or [12].

many iterative adjustments during which the researcher tweaks the formula used to compute costs. Note that these iterations are not automatic and they can exhibit a large computational burden, which is typically unknown by the user of the method, as one typically publishes the solution without reporting the failed attempts.

Yet, a more sophisticated class of steganographic algorithms derive costs by explicitly taking into account Eve's capabilities. Part of theses strategies derive costs from the detector performance, computed through a proxy such as the deflection of the likelihood ratio test for MiPod [6], the distance between cover and stego feature sets for Hugo [3] and Gibbs constructions [19], the explicit assumption that Eve knows Alice's embedding probabilities [20], or mimicking the distribution of the sensor noise for Natural Steganography [21].

Other strategies explicitly target Eve's detector to embed a message while defeating it. A terminology akin to present Machine Learning topics would call these "adversarial embeddings". To the best of our knowledge the first attempt in this direction was ASO [22], which derives embedding costs from an ensemble of Fisher Linear Discriminants [23] computed using SRM features [9]. For a pixel $i$, the cost of adding 1 to the pixel value is proportional to $\sum_k (f_k(x_i+1) - f_k(x_i))$, $f_k$ being the function returning the soft output of a weak classifier of the ensemble. By doing this, the embedding scheme succeeds to defeat Eve's classifier after the first iteration, but not in successive ones. Note that for this scheme empirical costs computed using Hugo [3] are used only to train the first classifier and the costs after the first iteration are directly computed using the classifier outputs.

Another more recent strategy, called ADV-EMB [24] (for adversarial embedding), attacks a detector implemented by some CNN and relies on the gradient of the loss function to modify the costs (since the attack from this work is used here as well, it is reviewed in more details in Section II). The very same work also proposes an algorithm creating a sequence of embeddings and classifiers in the hope to obtain a better classifier and better embeddings. Interestingly, both ADV-EMB and ASO adopt a similar strategy to generate stego images through iterations which consists in selecting the image defeating Eve's last trained classifier. One important difference between our work and ASO or ADV-EMB is the choice of classifiers-to-attack / images-to-use to train a new detector. As will be seen in the experimental section, this has an important effect on the security.

Furthermore, there is a prior art explicitly linking Steganography and Game theory. To the best of our knowledge, Ker in the context of batch steganography [25] proposes to solve the problem of how Alice should spread the message among a large number of covers while Eve anticipates this and tries to detect the existence of at least one secret message (pooled steganalysis). In this game, Alice chooses the number of images carrying the total payload and Eve sets her detector threshold. Ref. [26] studies the optimal strategy of both parties where Alice uses adaptive steganography while taking Eve's knowledge on Alice embedding strategy into account. Finally [27] studies the optimum strategy for choosing $\beta$ (the ratio of adversarial coefficients) in ADV-EMB [24] by setting a zero-sum game.

On a more general note, an inspiring work comes from the domain of adversarial machine learning to train robust classifiers [28], where the goal is to train a classifier by iteratively performing on one side efficient adversarial attacks (the $\max$ strategy), using the Projected Gradient Descent algorithm on the evolving loss, and on the other side to learn using new adversarial examples to train a more secure classifier (the $\min$ strategy). One difference between this reference and the proposed scheme is that we are not actually interested in training a better classifier but in designing a better steganographic embedding (even if a more robust classifier may be obtained as a bonus). Also, the iterative procedure [28] does not operate under the hidden message embedding constraints of the steganographic game.

Another popular class of models in the field of adversarial machine learning are Generative Adversarial Networks (GANs) [29], which have been recently used in steganography to learn the cost function [30]. The algorithm proposed in this paper shares to some extent some ideas with GANs. Indeed, GANs have a game theoretic setting in which two neural networks (called the generator and the discriminator) are opponents in a min-max optimization problem. In theory [31], the solution of GANs corresponds to a Nash equilibrium of a zero-sum game where the cost function is the loglikelihood of the discriminator. As will be later exposed, our algorithm converge to the Stackelberg equilibrium of a game, where Alice and Eve are rivals and Alice is a leader.

In addition to having different game theoretical grounds, a major difference between our approach and steganographic GANs such as [30] lies in the definition of the generator. In [30], the generator is implemented *explicitly* by an optimized network assigning costs to individual coefficients, which are then used in the embedding simulator. This is in sharp contrast to this paper, where the generator is *implicit* and corresponds to an attack of a particular classifier (or a set of them). The convergence (or training) of our algorithm should be therefore simpler, more stable, and Alice is saved from the hassle of designing an architecture of the generator. Indeed, Ref. [30] reports the error of detection by XU-Net of a GAN JPEG stego images with payload 0.4 bpnzAC equal to $11.8\%$, whereas that of the steganography proposed here is equal to $20.7\%$, which makes the proposed approach twice more secure.

However, an advantage of GAN-based steganography is that it can directly estimate embedding costs, a possibility that the proposed scheme does not offer yet.

### *B. Contributions of the paper*

This paper proposes an algorithm which can be used to automatically iterate through Alice's embedding strategies in order to find better ones in a simple steganographic game. The algorithm is general and can be applied on any adversarial-embedding-scheme/detector pair. This paper uses the adversarial attack presented in section II as a potential candidate, and two different CNNs, namely XU-Net [15] and SRNet [12], as detectors. It is an extension of the paper published in [32] in following directions:

- Theoretical and practical convergence issues with links to other iterative strategies;
- Evaluations of avoidance of classifiers with different architectures, which are here combined using an appropriate calibration function (novel), which has further improved the quality of the algorithm;
- Analysis of the transferability of ADV-EMB attacks between classifiers;
- Evaluation of the protocol on different quality factors, embedding rates, initial distortion functions, and also in spatial domain;
- Study of the increasing robustness of the learned classifier, it shows that this algorithm enables both to increase the security of the embedding scheme but also the robustness of the trained detector.

### *Notations*

In the following, letters in bold are used to represent vectors. The corresponding non bold letters are used for vector elements. The calligraphic letters are used for sets. Cover and stego objects are respectively denoted as $\mathbf{x} = (x_i)^{H\times W}$ and $\mathbf{y} = (y_i)^{H\times W}$ where $H$ and $W$ are the height and width of the image. We use $\mathbf{z} = (z_i)^{H\times W}$ to denote the stego objects that will be communicated by Alice. Note that $\mathbf{z}$ is a special type of $\mathbf{y}$. The corresponding sets are denoted as $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ respectively. $\omega \in \{0,1\}$ denotes the class of a object which is either cover ($\omega = 0$) or stego ($\omega = 1$). $N$ is the number of objects in the data base.

More specifically, the next section uses the following additional notations. A steganographic algorithm is any pair of functions $h_{\mathrm{emb}}(\mathbf{x}, m, k) : \mathcal{I}\times\mathcal{M}\times\mathcal{K} \to \mathcal{I}$ and $g_{\mathrm{ext}}(\mathbf{x}, k) : \mathcal{I}\times\mathcal{K} \to \mathcal{M}$ for which it holds that $g_{\mathrm{ext}}(h_{\mathrm{emb}}(\mathbf{x}, m, k), k) = m$ for all $m \in \mathcal{M}$, $k \in \mathcal{K}$, and $\mathbf{x} \in \mathcal{I}$. Spaces $\mathcal{I}$, $\mathcal{M}$, and $\mathcal{K}$ are respectively the space of all images, messages and keys with appropriate distributions, $\mathcal{P}_{\cdot}$, defined over them. Furthermore a steganographic detector is any function $f_{\mathrm{det}}(\mathbf{x}) : \mathcal{I} \to \{\mathrm{cover}, \mathrm{stego}\}$, although it is more convenient to assume $f_{\mathrm{det}}(\mathbf{x}) : \mathcal{I} \to \mathbb{R}$ and $\mathbf{x}$ is assigned to $\mathrm{stego}$ class if the output is greater than some threshold $\tau$. Lastly, we define a a distribution of stego images induced by the above mentioned distributions of cover images, messages, keys, and embedding algorithm $h_{\mathrm{emb}}$ as $\mathcal{P}_{\mathcal{Y}}^{(h_{\mathrm{emb}})}$.

## II. The ADV-EMB scheme

The pioneering work of Goodfellow et al. [33] demonstrated that classifiers based on neural networks can be forced to mis-classify an image by adding a specific signal of small amplitude. While in the field of computer vision the attacker has the freedom to change the image, attacking a steganographic detector is more difficult due to the constraint that the resulting stego image has to carry a particular message. Note that this property was known in Steganography before long before (see [19], [22] and recently also [17]).

A method ADV-EMB inpired by those in field of adversarial learning was proposed in [24] and due to low computational complexity it is used in this paper. ADV-EMB modifies costs of DCT coefficients, such that changes of coefficients during embedding are correlated with the gradient of the soft output of a CNN steganalyzer. Since the embedding function $h_{\mathrm{emb}}$ is not differentiable with respect to costs, ADV-EMB [24] comes with an heuristic which proposes modify costs $\rho_i^+$ and $\rho_i^-$ for increasing and decreasing the $i^{\mathrm{th}}$-DCT coefficient obtained by some existing cost function $\rho$ (e.g. J-Uniward, UERD) in the following way:

$$\rho_i^{+,new} = \begin{cases} \rho_i^+/\alpha & \text{if } \frac{\partial f}{\partial x_i}(\mathbf{x}) < 0, \\ \rho_i^+ & \text{if } \frac{\partial f}{\partial x_i}(\mathbf{x}) = 0, \\ \rho_i^+\alpha & \text{if } \frac{\partial f}{\partial x_i}(\mathbf{x}) > 0, \end{cases} \tag{1}$$

and

$$\rho_i^{-,new} = \begin{cases} \rho_i^-/\alpha & \text{if } \frac{\partial f}{\partial x_i}(\mathbf{x}) > 0, \\ \rho_i^- & \text{if } \frac{\partial f}{\partial x_i}(\mathbf{x}) = 0, \\ \rho_i^-\alpha & \text{if } \frac{\partial f}{\partial x_i}(\mathbf{x}) < 0, \end{cases} \tag{2}$$

where $\frac{\partial f}{\partial x_i}$ is the partial derivative of $f$ with respect to the value of the $i^{\mathrm{th}}$-DCT coefficient at its current value $x_i$ and $\alpha$ is a parameter set to recommended value 2.

The partial derivative of $f$ is computed for a cover object $\mathbf{x}$ that we wish to be assigned to a low probability of being stego (i.e. a small $f(\mathbf{x})$) after embedding. Therefore, for instance if $\frac{\partial f}{\partial x_i}(\mathbf{x}) > 0$, a positive increment on $x_i$ would increase this probability and consequently, this situation is penalized by increasing the corresponding modification cost $\rho_i^+$ by a factor $\alpha$.

Since steganalyzers are usually not good models of cover images,[2] modulating costs of all coefficients would probably lead to very detectable models. The solution adopted by ADV-EMB is to dispatch DCT coefficients into *common* and *adjustable* groups, $\mathcal{L}_c$ / $\mathcal{L}_a$, corresponding to $(1-\beta)$ / $\beta$ fractions of coefficients, and then modify only coefficients in the adjustable group. By minimizing $\beta$, ADV-EMB changes costs of a minimal number of coefficients. ADV-EMB finds the minimal $\beta$ by exhaustive search in $\beta \in \{0.1, 0.2, \ldots, 1.0\}$. The gradient used to modulate costs is calculated after coefficients from the common group are used for embedding a $1-\beta$ fraction of bits of the message $m$.

In this paper, the spatial version of ADV-EMB is tested as well; the idea is exactly the same in the explanation before, but where $i$ designates the index of pixel instead of DCT coefficient.

## III. Optimizing undetectability by using worst-case detectors

The ADV-EMB algorithm embeds a message while decreasing detectability w.r.t. a particular fixed detector $f$. The fundamental question answered in this section is:

*how to use this general attack to design a more secure steganographic scheme?*

[2] Steganalyzer models discriminate cover from stego images, but they do not model cover images themselves.

### A. Kerckhoffs' principle and Game Theory

Kerckhoffs' principle [34] states that in a security game, the adversary should know everything except the shared key. Applied in the context of steganography, it means that we can assume that Eve knows Alice's steganographic algorithm, distribution of messages, keys, and it is also customary to assume that Eve knows the length of message possibly hidden by Alice.

Assuming a worst case attack from Eve, Alice should consequently select an algorithm $h_{\mathrm{emb}}$ minimizing the utility of Eve's best detector $f_{\mathrm{det}}$. Without loss of generality it is assumed here the utility to be given by the accuracy under equal prior,[3] i.e. the average of the true positive rate and true negative rate, but other scores such as FP50 [35] (the false positive rate when the false negative rate equals 50%) or MD5 [36] (the miss detection rate when the false positive rate equals 5%) could also be used. Alice consequently wishes to solve the following optimization problem:

$$\arg\min_{h_{\mathrm{emb}}} \max_{f_{\mathrm{det}}} \mathbb{E}_{\mathbf{x}\sim P_{\mathcal{X}}}[f_{\mathrm{det}}(\mathbf{x}) < \tau] + \mathbb{E}_{\mathbf{y}\sim \mathcal{P}_{\mathcal{Y}}^{(h_{\mathrm{emb}})}}[f_{\mathrm{det}}(\mathbf{y}){\geq}\tau]. \tag{3}$$

The solution of Alice's problem (3) coincides with the Stackelberg equilibrium [37] of a sequential game with Alice being the leader defined as:

**Definition 1.** *the steganographic game, denoted $\mathcal{G}$ is a tuple $(\mathfrak{N}, \mathcal{A}_a, \mathcal{A}_e, u)$ where :*

- *$\mathfrak{N}$ is a set of 2 players, indexed by $p$ where $p \in \{a, e\}$ (for Alice and Eve)*
- *$\mathcal{A}_a, \mathcal{A}_e$ is a possibly infinite set of actions of Alice and Eve.*
- *$u = (u_a, u_e)$ where $u_p : \mathcal{A}_a \times \mathcal{A}_e \to \mathbb{R}$ is a real-valued utility function for player $p$.*

In the optimization problem (3), Eve's action consists in picking a detector in $\mathcal{A}_e$[4] and her payoff function is the detector accuracy

$$u_e(h_{\mathrm{emb}}, f) = \mathbb{E}_{\mathbf{x}\sim P_{\mathcal{X}}}[f(\mathbf{x}) < \tau] + \mathbb{E}_{\mathbf{y}\sim \mathcal{P}_{\mathcal{Y}}^{(h_{\mathrm{emb}})}}[f(\mathbf{y}){\geq}\tau].$$

Alice's action consists in picking an embedding function in $\mathcal{A}_e$ and her payoff function is $u_a(h_{\mathrm{emb}}, f) = -u_e(h_{\mathrm{emb}}, f)$. In general, strategies of Alice and Eve and can be stochastic, in which case, the Stackelberg equilibrium (3) coincides with the more popular Nash equilibrium, as the game is zero-sum.

### B. Assumed Eve's detectors

The term *Eve's detectors* is ambiguous, as it can refer to the set of detectors $\mathcal{A}_e$ used by Alice during the optimization of Equation (3), and the set of detectors used by real Eve to detect Alice's steganography. The mismatch between these two can be devastating and the whole history of steganography is precisely about this mismatch. To clarify, the former will be called *assumed Eve's detectors* and denoted $\mathcal{A}_e$ and the latter *real Eve's detectors* denoted $\tilde{\mathcal{A}}_e$.

In the optimization of Equation (3), the set of detectors $\mathcal{A}_e$ has to be reasonably complete, otherwise no security guarantees can be given about the resulting embedding algorithm. In this paper, $\mathcal{A}_e$ contains all detectors having the architecture of XU-Net [15] and/or SRNet [12]. The current limitation is the feasibility of the ADV-EMB attack, which can attack only differentiable detectors. However very general approaches [17], [19] can be used to alleviate this limitation at the expense of computational complexity. Another approach would be to train differentiable surrogates, but any investigation in this direction is currently outside of the scope of this paper.

### C. Solving the simple steganographic game

Solving optimization problem in (3) is difficult, because the expectation in the utility function does not have an analytical formula (because distributions are unknown) and the inner maximum and outer minimization are both over infinite sets.

To make the problem tractable, we propose an iterative algorithm utilizing the fact that when Alice is searching a suitable algorithm, the classifier in (3) does not have to be workable in practice. Specifically in the case of this work, the classifiers are (unreasonably) assumed to be selected for each image separately (technically the choice of a detector depends on the knowledge of cover image). This assumption is not realistic and overly pessimistic[5] for Alice, since it lower bounds the actual detectability achievable by Eve, but it decreases the computational complexity.

At $k^{\mathrm{th}}$ iteration, the proposed algorithm consists on the two following macro-steps :

1) it creates a stego set $\mathcal{Y}^k$ by using an embedding function $h^k_{\mathrm{emb}}$ maximally secure with respect to the set of detectors $\mathcal{F}^{k-1} = \{f^0_{\mathrm{det}}, f^1_{\mathrm{det}}, \dots, f^{k-1}_{\mathrm{det}}\}$, i.e. for a given image $\mathbf{x}$ it uses function

$$h^k_{\mathrm{emb}} = \arg\min_{h_{\mathrm{emb}} \in \mathcal{A}_a} \max_{f \in \mathcal{F}^{k-1}} u_e(h_{\mathrm{emb}}, f) \tag{4}$$

2) it creates a new detector $f^k_{\mathrm{det}}$, which should be optimal for stego images produced in previous step from $h^k_{\mathrm{emb}}$ :

$$f^k_{\mathrm{det}} = \max_{f \in \mathcal{A}_e} u_e(h^k_{\mathrm{emb}}, f) \tag{5}$$

and appends it to the pool, i.e. $\mathcal{F}^k = \mathcal{F}^{k-1} \cup \{f^k_{\mathrm{det}}\}$.

Notice that the distortion function / embedding algorithm is not fixed, but it is implicitly defined by the set of detectors $\mathcal{F}^k$ and a set of $\mathcal{A}_a$ of Alice's strategies. This means that for each image Alice picks the most secure algorithm for a given image with respect to detectors $\mathcal{F}^k$ she believes Eve might own.

The above algorithm is general, as Alice can use any set of embedding functions, $\mathcal{A}_a$, even those evading a specific detector as discussed in the previous section, and she can

[3] Authors agree that this measure is not very realistic and in practice one would probably bound a false positive rate, it is nevertheless the widely accepted standard.

[4] Selecting detector mounts to choosing one function from a set of functions from an image space to $[0, 1]$. Such function can be for example a CNN with a particular architecture (e.g. SRNet) and with fixed weights. In theory, the set of possible function is not restricted. This is shown later in experimental section, where Eve can use any of XU-Net, or SR-Net, or linear classifers using DCTR or GFR features.

[5] It is unlikely that Eve should be able to train a perfect classifier selector that maps each image to the best classifier she possesses for this image.

assume any set of steganographic detectors $\mathcal{A}_e$. In practice, one can guess that the bigger are both sets, the more secure the resulting algorithm will be. The only caveat is that, due to minimizing output of detectors in Equation (4), detectors should have comparable outputs. This problem of calibration is described in detail in Section IV-B.

The next two subsections discuss particular choices of Alice's and Eve's strategies used in this paper and turning the remaining generalities in the above procedure into practical algorithmic steps.

### *D. Alice's strategy*

In this paper, the set of steganographic algorithms used by Alice is a union of ADV-EMB attacks against all Eve's detectors $f_{\mathrm{det}} \in \mathcal{A}_e$ with $\beta \in \{0.1, 0.2, 0.3, \ldots, 0.9, 1.0\}$ and J-Uniward. This set has theoretically an infinite size. But if the optimality of ADV-EMB attack against a given detector $f_{\mathrm{det}}$ is assumed,[6] it is sufficient during the $k^{\mathrm{th}}$ iteration to consider attacks against a limited set of detectors $\mathcal{F}^{k-1} = \{f^0, \ldots, f^{k-1}\}$, as we cannot do better against this set. Thus, the $\min\max$ problem in the step 1 of each iteration is over a finite set, and hence computationally feasible.

An important implementation detail here is that stego images created by attacking $\mathcal{F}^{k-1} = \{f^0, \ldots, f^{k-1}\}$ and outputs of all detectors on them can be cached and used in subsequent iterations. This means that at every iteration, Alice needs to (i) create (adversarial) stego images against the detector $f^{k-1}$ appended to $\mathcal{F}^{k-1}$ in the previous iteration, and (ii) calculate outputs of $f^{k-1}$ for stego images created in iterations $1, 2, 3, \ldots, k$. This significantly decreases the computational complexity.

Here, we emphasize on the differences w.r.t. strategies proposed in [24] which we call *last iteration* and *random* strategies:

- In *last iteration* strategy, Alice's embedding algorithm is ADV-EMB attacking only the last trained detector $f^{k-1}$.
- In *random* strategy, Alice's embedding algorithm is ADV-EMB attacking a detector $f \in \mathcal{F}^k$ where each stego image of the training set is sampled uniformly over the previous iterations.

Again, note that Alice in this paper behaves more strategically (and conservatively), as she uses the algorithm producing the least detectable stego image by an unrealistic detector (in the sense that we assume that Alice knows an information not accessible in practice). The rationale here is the fact that for the next iteration Eve might learn a better detector than the last trained, by for example by training a new classifier from the ensemble of already trained classifiers.

### *E. Operational embedding algorithm*

An operational version of the algorithm that relies on ADV-EMB is given by Algorithm 1 (in this case a single convnet architecture is used to train classifiers and calibration can be omitted). The first few steps of the algorithm are illustrated in Figure 1. One can observe how classifiers in $\mathcal{F}^k$ surrounds the distribution of cover images and thereby restricting the choice of embedding algorithms.

```
Data: $\mathcal{Z}^0$ initial stego base, $\mathcal{X} = \left\{\mathbf{x}_{(1)}, .., \mathbf{x}_{(N)}\right\}$
      cover base, set of detector $\mathcal{F}^0 = \left\{f^0\right\}$, $k_{\max}$
$k \leftarrow 1$;
while $k \leq k_{\max}$ do
    Obtain adversarial base $\mathcal{Z}^k = \left\{\mathbf{z}^k_{(1)}, .., \mathbf{z}^k_{(N)}\right\}$
      where $\mathbf{z}^k_{(n)} = \text{ADV-EMB}\left(\mathbf{x}_{(n)}, f_{k-1}\right)$;
    Create stego base $\mathcal{Y}^k = \left\{\mathbf{y}^k_{(1)}, .., \mathbf{y}^k_{(N)}\right\}$ to be least
      detectable with respect to detectors in $\mathcal{F}^{k-1}$,
      $\mathbf{y}^k_{(n)} = \underset{\mathbf{z} \in \{\mathbf{z}^0_{(n)}, .., \mathbf{z}^k_{(n)}\}}{\arg\min} \max_{f \in \mathcal{F}^{k-1}} f(\mathbf{z})$ ;
    Train a new classifier $f^k$ to discriminate $\mathcal{X}$ from
      $\mathcal{Y}^k$ ;
    $\mathcal{F}^k = \mathcal{F}^{k-1} \cup \{f^k_{\mathrm{det}}\}$.;
    $k \leftarrow k + 1$;
end
Return stego base $\mathcal{Y}^{k_{\max}}$
```

**Algorithm 1:** Operational algorithm executed by Alice to generate a stego base.

### *F. Convergence of the algorithm*

The following theorem proves the convergence of Algorithm 1 under mild conditions on $\mathcal{F}$.

**Theorem 1.** *Let $\mathcal{F} = \{f : \mathcal{I} \to \mathbb{R}\}$ be a set of functions and let $\mathcal{F}^1, \mathcal{F}^2, \ldots, \mathcal{F}^k, \ldots$ be a sequence of subsets such that $\mathcal{F}^1 \subset \mathcal{F}^2 \subset \ldots \subset \mathcal{F}^k \subset \ldots \subset \mathcal{F}$. Suppose all functions $f \in \mathcal{F}$ are bounded by some constant $c$, i.e. $(\exists c \in \mathbb{R})(\forall f \in \mathcal{F})(\forall \mathbf{x} \in \mathcal{I})(f(\mathbf{x}) \leq c)$.*

*Then the limit $\hat{f}(\mathbf{x}) = \lim_{k\to\infty} \max_{f \in \mathcal{F}^k} f(\mathbf{x})$ exists.*

*Proof.* Define function $f^k_{\max}(\mathbf{x}) = \max_{f \in \mathcal{F}^k} f(\mathbf{x})$. Then for every $\mathbf{x} \in \mathcal{I}$ the sequence $f^1_{\max}(\mathbf{x}), f^2_{\max}(\mathbf{x}), \ldots, f^k_{\max}(\mathbf{x}), \ldots$ is non-decreasing and because of the boundedness assumption $\forall f \in \mathcal{F}$ , $f(\mathbf{x}) \leq c$, the sequence is bounded by $c$ as well. The monotone convergence theorem then states that the sequence $f^k_{\max}(\mathbf{x})$ converges to some value, which is denoted by $\hat{f}(\mathbf{x})$, which proves pointwise convergence of $f^k_{\max}$ to $\hat{f}$. □

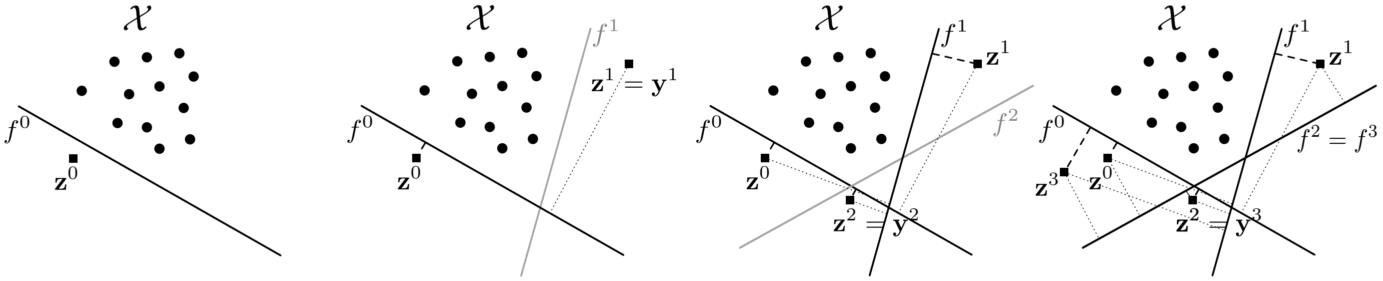
Note that the proof of the theorem holds for the important point that the subsets $\mathcal{F}^i$ are included in each other, so in other words the protocol converges because we are taking into account all previous classifiers among iterations when creating a new attack.

The above theorem implies that, when $k$ is large, the maximization w.r.t. $f \in \mathcal{F}^{k-1}$ is replaced by $\hat{f}$ (or a function $\epsilon$-close to $\hat{f}$). The algorithm defines detectability $\hat{f}(x)$ as a limit

$$\hat{f}(\mathbf{x}) = \lim_{k\to\infty} \max_{f \in \mathcal{F}^k} f(\mathbf{x}).$$

Note that the security of the resulting steganographic algorithm depends on two factors: (i) the set of all possible detectors $\mathcal{F}$; (ii) the attack quality on the classifier $f \in \mathcal{F}$.

[6] The optimality of the attack against a detector $f_{\mathrm{det}}$ here means that she cannot devise better attack against $f_{\mathrm{det}}$ by attacking different detector $f'_{\mathrm{det}}$


$\mathcal{X}$
$f^0$
$\mathbf{z}^0$
$f^1$
$\mathbf{z}^1 = \mathbf{y}^1$
$f^2$
$\mathbf{z}^2 = \mathbf{y}^2$
$\mathbf{z}^3$
$f^2 = f^3$
$\mathbf{z}^2 = \mathbf{y}^3$

(a) Iteration $k = 0$ (b) Iteration $k = 1$ (c) Iteration $k = 2$ (d) Iteration $k = 3$

Fig. 1: Initialization and the three first iterations of the algorithm with only one stego image. For simplification of representation, we assume that Euclidean distances (the dashed lines) between one adversarial image $\mathbf{z}$ and the boundary of the classifier $f^j$ represents the soft output of the classifier $f^j(\mathbf{z})$. The gray lines with bigger dash represent positive distance (so when the image is in the stego region), and smaller dash represent negative distances (when the image is in the cover region). For iteration $k$, all values $f^j(\mathbf{z}^i)$ (for $0 \leq i \leq k$ and $0 \leq j \leq k-1$) are computed, in order to select the stego $\mathbf{y}^k$ from $\{\mathbf{z}^0, .., \mathbf{z}^k\}$ according to the $\min\max$ strategy. Then $f^k$ (in grey shade) is trained to discriminate $\mathbf{y}^k$ from cover images.

Thus improving any of them should improve the quality of the scheme.

Theorem 1 assumes functions $f \in \mathcal{F}$ to be bounded. This condition can be trivially ensured for any function based on machine learning classifiers, as they are already bounded (e.g. Neural Networks), or they can be trivially bounded by applying some scaling or passing their output through a bounded and monotonous functions like sigmoid or $\tanh$.

Furthermore, the usual functions involved in a neural network are not only bounded but also Lipschitz continuous. Indeed, dot product,[7] convolution, max pooling, ReLU, sigmoid or tanh are all Lipschitz continuous and the sum and composition of such functions also are, i.e. neural networks are Lipschitz continuous. This observation leads to a stronger form of convergence.

If each $f \in \mathcal{F}$ is Lipschitz continuous with common constant, then it is known that $\hat{f}$ is also Lipschitz continuous with the same constant provided that $\hat{f}$ achieves a finite value for some $\mathbf{x}$. Since $\hat{f}$ is bounded, it is finite everywhere and thus Lipschitz continuous.

In addition, since all functions $f^k_{\max}$ and function $\hat{f}$ are defined on a compact subset of $\mathbb{R}^{H\times W}$ and the sequence $f^k_{\max}$ is monotonically increasing then Dini's theorem [38] applies which gives uniform convergence.

Roughly speaking, uniform convergence indicates that the series of functions on which the min-step of the algorithm operates converges everywhere in the input space at least with a rate that does not depend on $\mathbf{x}$. Although this rate lower bound is unknown, it can be argued that the algorithm could be stopped when $\|f^k_{\max} - f^{k-1}_{\max}\|$ is no greater than a given threshold. However, given the stochasticity of neural networks training and the time spend on it, it appears safer to stop after a predefined number of iterations as proposed in Algorithm 1.

## IV. Experimental settings

This section details the choices of $\mathcal{A}_a$ and $\mathcal{A}_e$ used in the experiments below, together with other important details such as calibration of classifier scores, database of images, etc.

### A. Steganograhic detectors $\mathcal{A}_e$

Since state-of-the-art steganalytic detectors are based on Convolutional Neural Networks (CNNs) [39], [40], [41] it should not be surprising that they are used here as well. We assume readers to be familiar with them, otherwise they are referred to [42] for a general introduction and to [39], [40], [41] for their uses in steganography.

For the purpose of this work, it is sufficient to view neural networks as an efficient procedure selecting $f$ from a large class of functions $\mathcal{F}$ minimizing the empirical error:

$$\hat{P}_{\text{err}}(f; \mathcal{X}, \mathcal{Y}) = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x}\in\mathcal{X}} \mathbb{1}\left\{f(\mathbf{x}) \geq \tau\right\} + \frac{1}{|\mathcal{Y}|} \sum_{\mathbf{x}\in\mathcal{Y}} \mathbb{1}\left\{f(\mathbf{x}) < \tau\right\}. \quad (6)$$

An important property of CNNs for ADV-EMB attack is their differentiability, which means that a gradient $\frac{\partial f}{\partial x}$ with respect to their inputs exists for almost every $\mathbf{x}$ and for every $f \in \mathcal{F}$.

The set of classifiers $\mathcal{F}$ is also the set of Eve's actions $\mathcal{A}_e$ and is equal to all convolutional neural networks with a given set of architectures (here XU-Net [15] or SRNet [12]).

### B. Calibrating classifier's output

As has been mentioned above, the space of classifiers $\mathcal{A}_e$ can contain CNNs of different architectures and even classifiers based on a very different paradigm. In these cases, it is important to make their output comparable, as pointed in [43], such that $\min\max$ selection in the step 1 of each iteration (Equation (4)) compares meaningful quantities. A situation is illustrated in top row in Figure 2 showing histograms of outputs of two classifiers on cover and stego images. Clearly, the left tail of the empirical cumulative distribution on stego images of Classifier $B$ (denoted by $f_B(\mathcal{Y})$ for simplicity) is more spread than that of Classifier $A$, which means that the

[7] To make sure that each instance of an architecture has the same (maximal) Lipschitz constant, it is sufficient to add a regularization term to the objective function.

inner maximization in (4) would prefer Classifier $A$ over the Classifier $B$, although the latter is more precise.

We therefore propose to calibrate the output of a detector $f$ by its empirical distribution function $\hat{F} : [0,1] \rightarrow [0,1]$ estimated on cover images as

$$\hat{F}(t) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{f(\mathbf{x}_{(i)})\leq t},$$

where $\{\mathbf{x}_{(i)}\}_{i=1}^{N}$ are cover images. The calibrated detector, denoted $\hat{f}$, and is then defined as a composition

$$\hat{f}(\mathbf{x}) = \hat{F}(f(\mathbf{x})).$$

The effect of the calibration is shown in the bottom row in Figure 2. We can see that after the calibration, Classifier $B$ would be selected by $\min\max$ strategy as desired.

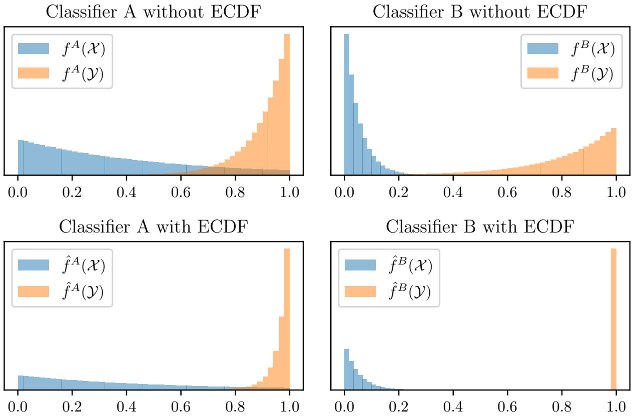


Fig. 2: Effect of calibration allowing to make the output of two classifiers comparable on the same data base for the $\min\max$ strategy.

## C. Other implementation and experimental settings

*a) Embedding:* The embedding algorithm used to initialize the algorithm and to calculate default costs for changing elements is J-Uniward [4], UERD [18] (for JPEG steganography) or HILL [5] (for spatial steganography). The experiments use the JPEG version of the popular BossBase database [44] of size $512 \times 512$ in grayscale format and compressed with Quality Factor (QF) 75 or 95, or same sized images in PGM format for spatial steganography. Unless stated otherwise, all images are embedded using an embedding rate of 0.4 bits per non-zero AC DCT coefficient (bpnzAC) at each iteration of the algorithm.

*b) Classification/Steganalysis:* our implementations of XU-Net and SRNet use TensorFlow [45] library[8]. In each iteration of the algorithm, a new steganalyzer $f^k$ is trained by classifying cover objects $\mathcal{X}$ and stego objects $\mathcal{Y}^k$ given at the second step of the loop of Algorithm 1. Those classifiers are trained using full-size images of $512 \times 512$ coefficients, $2 \times 4000$ Cover and Stego objects for training, $2 \times 1000$ for validation set and using remaining $2 \times 5000$ to estimate error rates. The training database is shuffled after each epoch. In each batch, we apply data augmentation based on random mirroring and rotation of the batch images by 90 degrees. 280 epochs are used for training using ADAM optimization algorithm. The configuration achieving the best validation accuracy is used as the result of training. For XU-Net, the classifier is trained starting with randomly initialized weights (zero mean Gaussian with standard deviation 0.01), initial learning rate is set to 0.001 and decreased after each 5000 steps to 0.9 times the current value. Remaining parameters of ADAM are kept to default setting. The size of mini-batch is 32 (16 cover-stego pairs). The configuration of SRNet is the one proposed in the paper [12], except the training which lasts for 280 epochs. The size of mini-batch is 16 (8 cover-stego pairs). The learning rate is set to 0.001 proposed in the paper. The experiments were run on an Nvidia GPU Quadro P6000 (24 GB of memory). Training XU-Net takes approximately 20 hours at each iteration $k$, SRNet 30 hours, and the generation of an adversarial data-base 5 hours multi-threaded on 36 cores.

*c) Attack:* The ADV-EMB attack adjusting costs of DCT coefficients is implemented as described in Section II. Because XU-Net / SRNet uses a spatial image without rounding as input, to compute partial derivatives $\frac{\partial f}{\partial x_i}$ with respect to the $i^{th}$-DCT coefficient, IDCT is treated as an additional layer placed before the first layer of XU-Net / SRNet. The partial derivative is consequently handled by automatic differentiation using the function `tf.gradient()` from the TensorFlow library, and differentiating with respect to the image coded in the JPEG domain.

Since there is a possibility that embedding using ADV-EMB fails for some images, which means that even when we modify all costs $\rho_i^+, \rho_i^-$ of all DCT coefficients, the corresponding stego image is classified as stego. As suggested in [24], in this case the costs are all set to their current values without any modification, which corresponds to setting $\beta = 0$ in ADV-EMB.

# V. Experimental comparison to prior art

This section summarizes an extensive experimental study of properties of the algorithm and comparison to the prior art. First the convergence of the algorithm is studied when the set of assumed and real Eve's detectors $\tilde{\mathcal{A}}_e$ and $\mathcal{A}_e$ are the same and when they are different. Then the proposed algorithm is compared to the prior art: the $\min\max$ strategy is compared to "last iteration" and "random iteration" of [24].

The steganalytic detectors used by *real Eve* inludes XU-Net, SRNet, and linear classifiers [46] with DCTR [47] and GFR [48] feature sets. This means that *real Eve* uses algorithms which Alice has not assumed during derivation of her embedding function, which proves that resulting embedding is not overoptimized.The reported error is probability of error under equal priors, $P_{\text{err}} = \min_{\text{Pr}_{\text{FA}}} \frac{1}{2}(\text{Pr}_{\text{FA}} + \text{Pr}_{\text{MD}})$, with $\text{Pr}_{FA}$ and $\text{Pr}_{MD}$ standing for the false-alarm and missed detection empirical probabilities.

[8] The codes for the experiments will be made available after publication.

Unless said otherwise, reported error rates always follows Kerkhoff's principle, which means that the detector is always trained after Alice publishes her embedding algorithm.

### *A. Results*

Error rate $P_{\text{err}}$ of XU-Net detector for eight iterations of the algorithm when Alice assumes that Eve will use XU-Net as detector is shown in the top row of Figure 3. Errors are shown for different payloads and different quality factors. The algorithm succeeds at significantly increasing the security in all cases. For example it makes the stego-images with payload $0.4$ bpnzAC in JPEGs with QF 95 undetectable by XU-Net. For other cases, the undetectability was not reached within eight iterations, but the improvement in security is still huge. Notice that the error is not strictly monotonically improving, which we attribute to (i) the training of detectors does not reach global minimum and (ii) the ADV-EMB attack might not succeed in avoiding all detectors — a phenomenon described in more details below in sections VI-B and VII.

The most interesting case occurs when the detectors assumed by Alice and those actually used by Eve differ as if models used by Alice are not sufficiently rich, she might anticipate a detectable embedding. Middle row in Figure 3 again shows error of XU-Net, SRNet, DCTR, GFR classifiers after first eight iterations of the algorithm when Alice assumes XU-Net (left) or SRNet (right) in her optimization. Notably, the algorithm still improves the security even in case of mismatch. We assume that this is due to the fact that both XU-net and SRNet are sufficiently rich models.

The two bottom rows in Figure 3 compare the proposed algorithm to "last" and "random" strategies proposed in [24]. We see that the proposed algorithm is markedly better than both prior art solutions. This should not be surprising as unlike them, it directly optimizes undetectability measured by Kerckhoffs' principle.

Figure 4 shows histograms of the iteration at which attacked detector of each stego image of Alice's stego-sets was created. This distribution is very far from the "last" strategy, which would contain a single peak at $k-1$ for iteration $k$. The distribution is more like a "random" strategy, which should be uniform on $\{0, 1, 2, \dots, k-1\}$ at iteration $k$. The reason, why the proposed algorithm is more secure than the "random" strategy is that stego images are not selected randomly, but deterministically conditioned by a given cover according to $\min\max$ criterion.

The security ($P_{\text{err}}$ of classifiers) of the algorithm when the ADV-EMB attack is initialized with UERD costs and $\mathcal{A}_e$ is learned using the XU-Net architecture is shown in Figure 5. The security is similar to that achieved when the ADV-EMB is initialized with J-Uniward costs. It improves in the case of mismatch between assumed and real detectors (left figure) and it also improves over the prior art (right figure).

Finally, the security of spatial steganography with this algorithm is evaluated on the experiment on figure 6. Here the algorithm is initialized with HILL [5] costs and $\mathcal{A}_e$ is learned using the SRNet architecture. For an embedding rate of 0.5 bpp, the error rate jumps from 12.5% to 20.8% at iteration 8, which gives an increase of +8.3%.

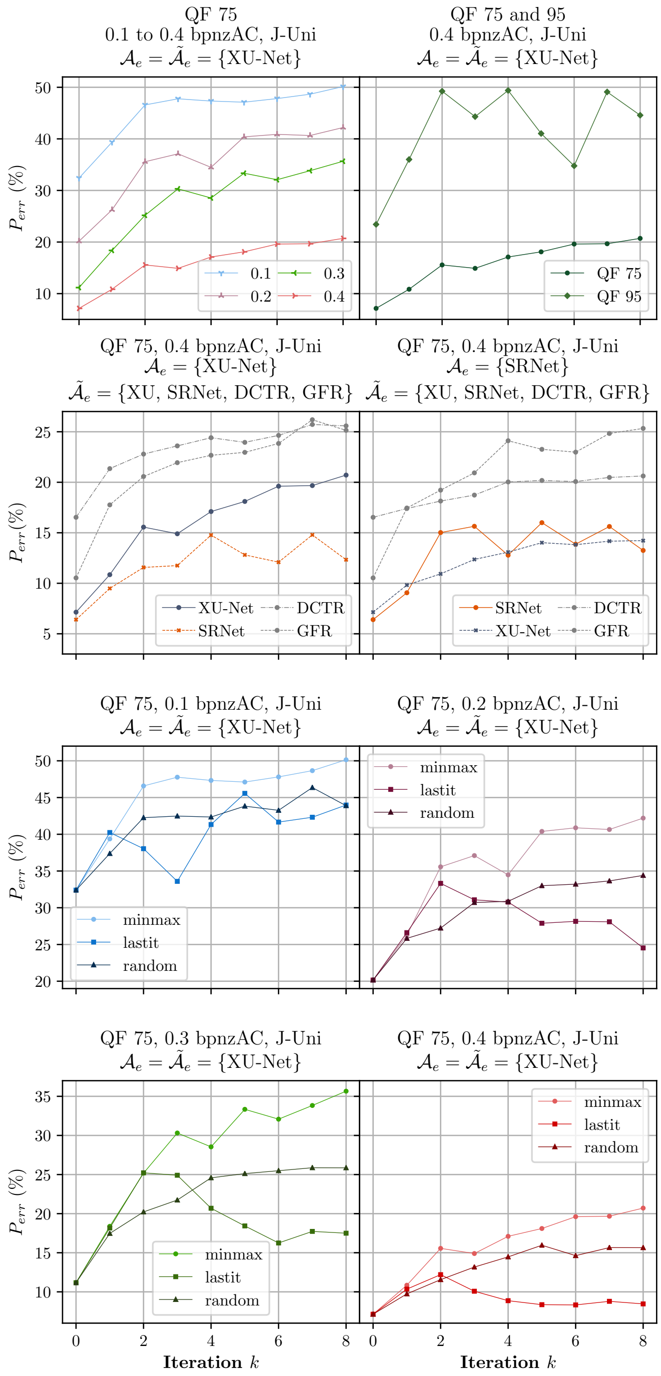


Fig. 3: (Top row) Evolution of the algorithm for an initialization of costs with J-Uniward, $\mathcal{A}_e = \tilde{\mathcal{A}}_e = \{$XU-Net$\}$ and (top left) QF75, and for four different embedding rates (0.1 to 0.4 bpnzAC); or (top right) 0.4 bpnzAC and two different quality factor (QF 75 and 95). (Second row) Two experiments with QF 75, 0.4 bpnzAC, J-Uniward, $\tilde{\mathcal{A}}_e = \{$XU-Net, SRNet, DCTR, GFR$\}$ but where (left) $\mathcal{A}_e = \{$XU-Net$\}$ and (right) $\mathcal{A}_e = \{$SRNet$\}$ (Two bottom rows) Four experiments for QF 75, J-Uniward, $\mathcal{A}_e = \tilde{\mathcal{A}}_e = \{$XU-Net$\}$ and three strategies $\min\max$, last iteration and random, for each 0.1, 0.2, 0.3 and 0.4 bpnzAC embedding rates.

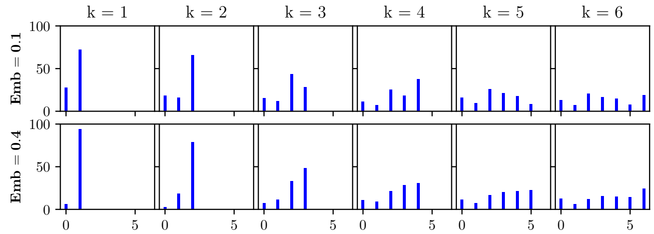


Fig. 4: Evolution of composition of the stego data set $\mathcal{Y}^k$ over $\min\max$ strategy iterations $k$ for two experiments : classifier XU-Net, QF 75, initialization of costs with J-Uniward, and for an embedding rate of 0.1 or 0.4 bpnzAC. For each plot, bars give the proportion (in %) of images taken from $\mathcal{Z}^i$ ($0 \leq i \leq k$) to generate $\mathcal{Y}^k$ (where where $i$ is on the x-axis).

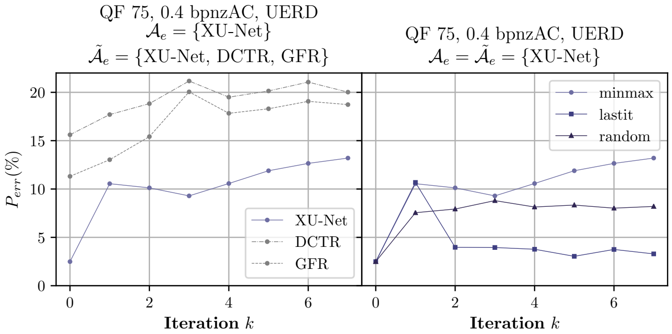


Fig. 5: Evolution of $P_{\text{err}}$ for XU-Net, an embedding rate of 0.4 bpnzac, QF 75, for an initialization of costs with UERD. (Left) Evolution of $P_{\text{err}}$ of the $\min\max$ strategy and of two blind steganalyzers based on GFR and DCTR features. (Right) Evolution for the 3 strategies $\min\max$, last iteration and random.

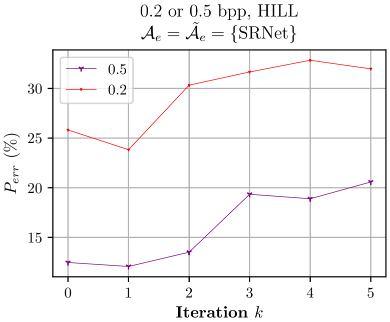


Fig. 6: Evolution of $P_{\text{err}}$ for spatial steganography, with an initialization of costs with HILL, $\mathcal{A}_e = \tilde{\mathcal{A}}_e = \{\text{SRNet}\}$ and for two different embedding rates of 0.2 and 0.5 bpp.

Note that the theorem doesn't prove that the $P_e$ is strictly increasing from one iteration to the next for the $\min\max$ strategy; but it holds that the protocol cannot enter an algorithmic pathologic behavior by looping on the same sequence of classifiers. In this case, the $P_e$ curves would exhibit strong oscillations and limited trends whereas, in our experimental results, oscillations are limited and a positive trend is always observed for the $\min\max$ protocol. In contrast, other protocols (last iteration and random) do exhibit such behaviors (see Fig 3).

## VI. Optimizing against more architectures

### A. Performance analysis

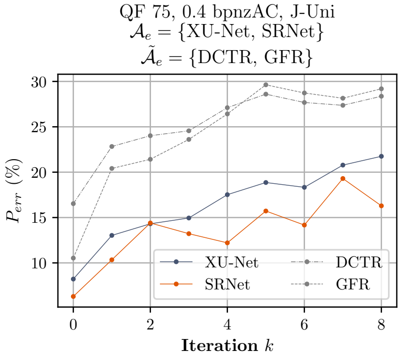


Fig. 7: Evolution of $P_{\text{err}}$ for the experiment with double adversary : where QF 75, 0.4 bpnzAC, J-Uniward, and where there are two assumed steganalyzers : XU-Net and SRNet, and with two more real steganalyzers based on DCTR and GFR features.

| $\tilde{\mathcal{A}}_e$ | J-Uniward | $\mathcal{A}_e$ (for QF 75, 0.4 bpnzac, J-Uniward) | | |
|---|---|---|---|---|
| | | **{XU-Net}** | **{SRNet}** | **{XU-Net, SRNet }** |
| {XU-Net} | 7.1% | **+ 13.6%** | + 7.1% | **+ 13.5%** |
| {SRNet} | 6.3% | + 5.9% | **+ 6.9%** | **+ 10.0%** |
| {DCTR} | 16.5% | + 9.1% | + 4.1% | **+ 11.8%** |
| {GFR} | 10.5% | + 14.6% | + 14.8% | **+ 18.7%** |

TABLE I: The first column shows the baseline detectability of J-Uniward : it gives the error rate of four types of detectors (in rows). Then, the next three columns show the gain in error rate $P_{\text{err}}(k = 8) - P_{\text{err}}(k = 0)$ for three experiments, QF 75 and embedding rate of 0.4 bpnzAC, J-Uniward, and with $\min\max$ strategy. First with $\mathcal{A}_e = \{\text{XU-Net}\}$, second with $\mathcal{A}_e = \{\text{SRNet}\}$ and third with $\mathcal{A}_e = \{\text{XU-Net, SRNet}\}$ (so with double adversaries). Bold: evolution of error rate when there is a match between $\mathcal{A}_e$ and $\tilde{\mathcal{A}}_e$ in the experiment. For each column, non-bold results are for mismatches.

In the previous section, the set of classifiers $\mathcal{A}_e$ used by Alice contained neural networks with the same architecture differing only in weights. But as was many-time emphasized, the set $\mathcal{A}_e$ should be as complete as possible, which means convnets with different architectures. Below, $\mathcal{A}_e$ contains all neural networks with XU-Net and SRNet architectures. This was achieved by extending the set $\mathcal{F}^k$ in each iteration by two networks, one with XU-Net and one with SRNet architecture. Outputs of these detectors are always calibrated as was described above in IV-B. Otherwise, the experimental settings and the algorithm are unchanged.

The error of four steganalyzers (DCTR and GFR are not in $\mathcal{A}_e$) for the first eight iterations when the algorithm optimized embedding message with payload 0.4 bpnz in JPEG images with QF 75 is shown in Figure 7. The behavior is similar as observed above as the algorithm improves significantly the undetectability. Table I summarizes the increase of the undetectability (error rate) against J-Uniward as measured by different classifiers (in rows) when $\mathcal{A}_e$ contains either XU-Net, SRNet, or both (in three last columns). In line with theoretical expectations, Algorithm 1 with $\mathcal{A}_e$ containing both architectures achieves highest undetectability (minus noise) with respect to all four tested detectors. Specifically, the detectability by SRNet presently considered the most powerful detector jumps from 6.3% to 16.3%, which is almost a three fold improvement in security. This also means that, at the time of writing, the proposed algorithm delivers the most secure steganographic algorithm.

### B. *Compositions of training sets*

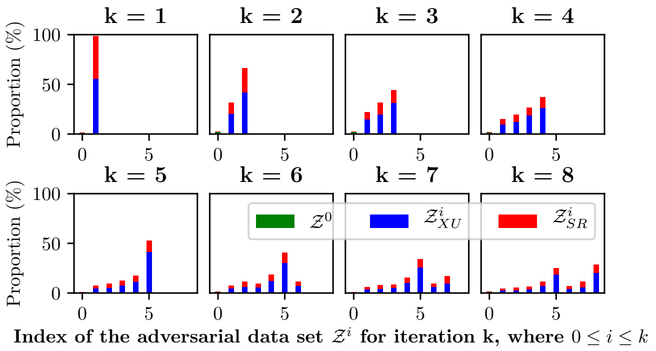


Fig. 8: Composition of $\mathcal{Y}^k$ set with two steganalyzers, an embedding rate of 0.4 bpnzac, QF 75. Green bar represent the proportion of stego chosen among $\mathcal{Z}^0$ (so with the costs of J-Uniward). For $i > 0$, blue (resp. red) bars represent the proportion of stego chosen among $\mathcal{Z}^i_{XU}$ (resp. $\mathcal{Z}^i_{SR}$), i.e. the adversarial stego contents attacking $f^{i-1}_{XU}$ (resp.$f^{i-1}_{SR}$).

Figure 7 showing $P_{\text{err}}$ of all steganalyzers with respect to iteration on the algorithm suggests that Alice should be sending stego-images created by ADV-EMB attacking SRNet, as these detectors produces lowest error. Figure 8 shows distribution of algorithms used to create stego-images for each iteration. Surprisingly, even though SRNet has lower error rate, stego-images are consistently created by attacking XU-Net, which is everything but intuitive.

We believe that this problem stems from the weakness of ADV-EMB attack, which calculates gradients of detectors outputs only once during embedding (see details in Section II. This can lead to cases, when by trying to evade one classifier (e.g. SRNet) it can make the image detectable by a different classifier (e.g. XU-Net). This is measured in terms of *transferability* of attacks, defined as

- $T^k_{XU} = P\big(f^{k-1}_{SR}(\mathbf{z}^k_{XU}) < 0.5 \big| f^{k-1}_{XU}(\mathbf{z}^k_{XU}) < 0.5\big)$,
- $T^k_{SR} = P\big(f^{k-1}_{XU}(\mathbf{z}^k_{SR}) < 0.5 \big| f^{k-1}_{SR}(\mathbf{z}^k_{SR}) < 0.5\big)$,

which expresses the probability that stego-images created by ADV-EMB against XU-Net will be undetectable by SRNet and vice versa. Transferability shown in Figure 9 for each iteration is generally very low, but that of ADV-EMB attacking XU-Net is generally higher than that of attacking SRNet. This implies that ADV-EMB has to be somehow adapted to attack a set of classifiers instead of just one, but this work is clearly outside of the scope of this paper.Figure 9 shows the probability of failure of ADV-EMB attack, which generally increases as the algorithm progresses, as the distribution of cover images (and its support) is better captured as illustrated in Figure 1.

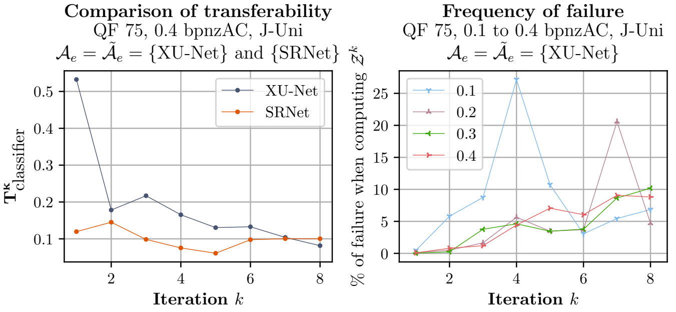


Fig. 9: (Left) Evolution of the transferability $T^k_{XU}$, $T^k_{SR}$ over iteration $k$ of adversarial data bases $\mathcal{Z}^k_{XU}$, $\mathcal{Z}^k_{SR}$. (Right) Evolution of the frequency of failure for multiple embedding rates

## VII. Note on the initialization of CNNs

Ref. [43] introduced to steganography a concept of Curriculum Learning (CL), which is a technique used to improve the learning of a classifier for low payloads by training them on easier problems. We have experimented this approach by initializing learning of a classifier at iteration $k$ with parameters of the classifier trained at previous iteration $k-1$. Alternatively and as used in all experiments above, the classifiers were initialized completely at random.

Figure 10 shows error rate of both algorithms when $\mathcal{A}_e$ contains only XU-Net detectors. The experiment used JPEG images with QF 75, 0.4 bpnzAC, and costs in ADV-EMB were initialized by J-Uniward. The experimental results show that the algorithm where curriculum learning is not used achieved lower error rate than the one using it. We believe that the detectors with curriculum learning might be stuck in suboptimal local minimum.

## VIII. Conclusion and perspectives

This paper builds upon equivalence of Kerckhoffs' principle and Stackelberg equilibrium of a game, where Alice is the leader, and which corresponds to the optimization problem where Alice minimizes the accuracy of the best detector of Eve — a $\min\max$ optimization. Since direct optimization of this criterion is computationally infeasible, we simplify the optimization problem by giving Eve an unrealistic advantage — she can choose her detector after she observes Alice's image. We advocate this simplification to be fair, as it is used exclusively by Alice during optimization of her steganographic

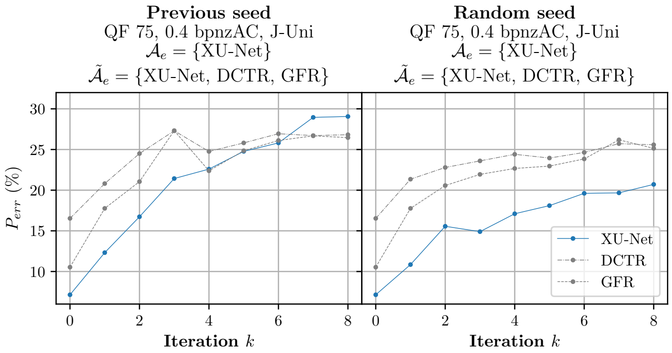


Fig. 10: Effect of the initialization of CNN on the $P_{\text{err}}$ of the algorithm (with XU-Net, initialization of costs with J-Uniward, QF75, $\min\max$ strategy and an embedding rate of 0.4 bpnzAC). (Left) $f^k$ was seeded by $f^{k-1}$, and (right) $f^k$ is randomly seeded.

scheme and the evaluation of the security of resulting algorithm is fair and done as is standard within the field.

Although the proposed algorithm is general, the realization used in this paper relied on two recent innovations: convolutional neural networks implementing a general class of steganographic detectors and adversarial embedding capable of embedding a message while being undetectable by a given detector. The extensive experimental results demonstrate the superiority of the proposed algorithm with respect to prior art in JPEG domain. Specifically, the most secure version increases the undetectability of messages with payload 0.4 hidden in JPEGs of SRNet by $10\%$ comparing to J-Uniward: it jumps from 6.3% to 16.3%. This increase in the security should not be surprising, since the presented algorithm just automatically plays the game, which the community plays implicitly since the birth of the field.

A weakness of this protocol is its dependency to the source of images : for example, if we execute the protocol for a constant size of images, we can't generate stego images for images with different size; we would have to re-run the entire protocol.

This paper just scratched the possibilities and we expect that stronger attacks than ADV-EMB, whose some limitations were identified, will lead to more secure steganography. In the same time, the proposed algorithm can be used to verify the completeness of steganalyzers, as when coupled with powerful attacks [19], [17], it can automatically identify their weaknesses.

## Acknowledgments

This work has been funded in part by the French National Research Agency (ANR-18-ASTR-0009), ALASKA project: https:// alaska.utt.fr, and by the French ANR DEFALS program (ANR-16- DEFA-0003). The authors acknowledge the support of the OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" and a support by Czech Ministry of Education 19-29680L. This work was granted access to the HPC resources of IDRIS under the allocation 2019-AD011011259 made by GENCI (Grand Equipement National de Calcul Intensif).

## References

[1] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Advances in Cryptology*. Springer, 1984, pp. 51–67.

[2] J. Fridrich, *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press, 2009.

[3] T. Pevny, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Information Hiding 2010*, 2010.

[4] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 1, 2014.

[5] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 4206–4210.

[6] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, 2016.

[7] S.Dumitrescu, X.Wu, and Z.Wang, "Detection of LSB steganography via sample pair analysis," in *IEEE transactions on Signal Processing*, 2003, pp. 1995–2007.

[8] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on information Forensics and Security*, vol. 5, no. 2, pp. 215–224, 2010.

[9] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[10] M. Yedrouj, F. Comby, and M. Chaumont, "An efficient CNN for spatial steganalysis," in *ICASSP*, 2018.

[11] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.

[12] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.

[13] P. Sallee, "Model-based methods for steganography and steganalysis," *International Journal of Image and graphics*, vol. 5, no. 01, pp. 167–189, 2005.

[14] A. Westfeld, "High capacity depsite better steganalysis: F5- a steganographic algorithm," in *Fourth Information Hiding Workshop*, 2001, pp. 301–315.

[15] G. Xu, "Deep convolutional neural network to detect j-uniward," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2017, pp. 67–73.

[16] T. Filler, J. Judas, and J. Fridrich, "Minimizing embedding impact in steganography using trellis-coded quantization," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2010, pp. 754 105–754 105.

[17] T. Pevný and A. D. Ker, "Exploring non-additive distortion in steganography," in *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2018, pp. 109–114.

[18] L. Guo, J. Ni, W. Su, C. Tang, and Y.-Q. Shi, "Using statistical image model for jpeg steganography: Uniform embedding revisited," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2669–2680, 2015.

[19] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 705–720, 2010.

[20] A. D. Ker, T. Pevný, and P. Bas, "Rethinking optimal embedding," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2016, pp. 93–102.

[21] T. Taburet, P. Bas, W. Sawaya, and J. Fridrich, "A natural steganography embedding scheme dedicated to color sensors in the jpeg domain," in *Electronic Imaging 2019*, Burlingame, United States, Jan. 2019.

[22] S. Kouider, M. Chaumont, and W. Puech, "Adaptive steganography by oracle (aso)," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2013, pp. 1–6.

[23] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 2, pp. 432–444, 2012.

[24] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "Cnn-based adversarial embedding for image steganography," *IEEE Transactions on Information Forensics and Security*, 2019.

[25] A. D. Ker, “Batch steganography and the threshold game,” in *Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505. International Society for Optics and Photonics, 2007, p. 650504.

[26] P. Schottle and R. Bohme, “Game theory and adaptive steganography,” *Trans. Info. For. Sec.*, no. 4, pp. 760–773, Apr.

[27] X. Shi, B. Tondi, B. Li, and M. Barni, “Cnn-based steganalysis and parametric adversarial embedding: a game-theoretic framework,” *arXiv preprint arXiv:1906.00697*, 2019.

[28] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” *arXiv preprint arXiv:1706.06083*, 2017.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[30] J. Yang, D. Ruan, X. Kang, and Y.-Q. Shi, “Towards automatic embedding cost learning for jpeg steganography,” in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, 2019, pp. 37–46.

[31] F. A. Oliehoek, R. Savani, J. Gallego-Posada, E. Van der Pol, E. D. De Jong, and R. Groß, “Gangs: Generative adversarial network games,” *arXiv preprint arXiv:1712.00679*, 2017.

[32] S. Bernard, T. Pevný, P. Bas, and J. Klein, “Exploiting Adversarial Embeddings for Better Steganography.”

[33] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[34] A. Kerckhoffs, “La cryptographie militaire,” *Journal des Sciences Militaires*, pp. 5–38, 1883.

[35] A. D. Ker and T. Pevnỳ, “The steganographer is the outlier: Realistic large-scale steganalysis,” *IEEE Transactions on information forensics and security*, vol. 9, no. 9, pp. 1424–1435, 2014.

[36] R. Cogranne, Q. Giboulot, and P. Bas, “The ALASKA Steganalysis Challenge: A First Step Towards Steganalysis ”Into The Wild”.”

[37] J. Cermak, B. Bosansky, K. Durkota, V. Lisy, and C. Kiekintveld, “Using correlated strategies for computing stackelberg equilibria in extensive-form games,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[38] U. Dini, *Fondamenti per la teoria delle funzioni di variabli reali [Foundations of the theory of functions of real variable]*, Pisa, 1878.

[39] L. Pibre, J. Pasquet, D. Ienco, and M. Chaumont, “Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch,” *Electronic Imaging*, vol. 2016, no. 8, pp. 1–11, 2016.

[40] J. Ye, J. Ni, and Y. Yi, “Deep learning hierarchical representations for image steganalysis,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2545–2557, Nov 2017.

[41] G. Xu, H. Wu, and Y. Shi, “Structural design of convolutional neural networks for steganalysis,” *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, May 2016.

[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[43] Y. Yousfi, J. Butora, J. Fridrich, and Q. Giboulot, “Breaking alaska: Color separation for steganalysis in jpeg domain,” in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, 2019, pp. 138–149.

[44] P. Bas, T. Filler, and T. Pevný, “”break our steganographic system”: The ins and outs of organizing boss,” in *International Workshop on Information Hiding*, vol. 6958, LNCS. Springer Berlin Heidelberg, 2011, pp. 59–70.

[45] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

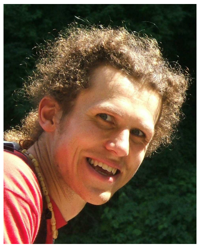
[46] R. Cogranne, V. Sedighi, J. Fridrich, and T. Pevný, “Is ensemble classifier needed for steganalysis in high-dimensional feature spaces?” in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015, pp. 1–6.

[47] V. Holub and J. Fridrich, “Low-complexity features for jpeg steganalysis using undecimated dct,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219–228, 2015.

[48] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, “Steganalysis of adaptive jpeg steganography using 2d gabor filters,” in *Proceedings of the 3rd ACM workshop on information hiding and multimedia security*. ACM, 2015, pp. 15–23.

**Solène Bernard** received the Engineering diploma from the Ecole Centrale de Lille - France, in 2018. She is currently in his third year of PhD under the supervision of Patrick Bas, John Klein and Tomáš Pevný. The subject of her thesis focuses on steganography.

**Patrick Bas** received the Electrical Engineering degree from the Institut National Polytechnique de Grenoble, France, in 1997, and then the Ph.D. degree in signal and image processing from Institut National Polytechnique de Grenoble, France, in 2000. He has co-organized the 2nd Edition of the BOWS-2 contest on watermarking in 2007, and the BOSS and Alaska contests on steganalysis respectively in 2010 and 2019.



**Tomáš Pevný** received the master’s degree in computer science from the School of Nuclear Sciences and Physical Engineering, Czech Technical University, Prague, in 2003, and the Ph.D. degree in computer science from the State University of New York, Binghamton, in 2008. From 2008 to 2009, he held a Post-Doctoral position at Gipsa-lab, Grenoble, France. He is with Artificial Intelligence Center at Czech Technical University in Prague. His research interests are applications of non-parametric statistics (machine learning, density modeling) with a focus on computer security, steganography, and steganalysis.

**John Klein** obtained the habilitation à diriger les recherches in computer sciences from the University of Lille in 2017 and a Ph.D. in information sciences from the University of Rouen in 2008. Prior to that, he was an intern in Beijing University and obtained a Master degree in signal processing from the University of Bordeaux and an engineering degree from ENSEIRB in telecommunications. His research interests include several aspects of artificial intelligence on both symbolic (approximate reasoning and uncertainty models) and data driven (ensembling and deep learning) sides. His works are also frequently applied to image processing tasks such as image segmentation, object tracking, biomedical image analysis and multimedia security.