Blind Newton Sensitivity Attack

Pedro Comesaña, Luis Pérez-Freire and Fernando Pérez-González *

9th April 2006

Abstract

Until now, the sensitivity attack was considered as a serious threat to the robustness and security of spread-spectrum-based schemes, since it provides a practical method of removing watermarks with minimum attacking distortion. Nevertheless, it had not been used to remove the watermark from other watermarking algorithms, such as those which use side-information. Furthermore, the sensitivity attack has never been used to obtain falsely watermarked contents, also known as forgeries. In this paper a new version of the sensitivity attack based on a general formulation is proposed; this method does not require any knowledge about the detection function nor any other system parameter, but just the binary output of the detector, being suitable for attacking most known watermarking methods. The new approach is validated with experiments.

1 Introduction

In its early years, digital watermarking was conceived as a solution to the problems of illegal copy control and intellectual property rights (IPR) protection. The aim of these applications is to distinguish whether digital media contains a given watermark or not, so they can be considered as a binary hypothesis testing problem, termed *watermark detection*.[1] In a general setup, the wa-

*Signal Theory and Communications Department, University of Vigo, Vigo 36310, Spain. E-mail: {pcomesan, lpfreire, fperez}@gts.tsc.uvigo.es.

[1] Watermark detection and *watermark decoding* must be regarded to different problems, since in the latter the objective is to decode the embedded message.

termarking of a digital document $\mathbf{x}$, that is arranged as a column vector of dimension n, can be expressed as $\mathbf{y} = \mathbf{x} + \mathbf{w}$, with $\mathbf{w}$ the watermark, which may be made key-dependent in order to improve the security of the system. Hence, the hypothesis testing problem can be written as

$$\begin{aligned} H_0 : &\quad \mathbf{y} = \mathbf{x} \\ H_1 : &\quad \mathbf{y} = \mathbf{x} + \mathbf{w}. \end{aligned}$$

Since the watermarked signal could have been attacked, the detector should adapt this test to take the attack into account; this attack will be modeled as the addition of a vector $\mathbf{t}$, yielding a signal $\mathbf{z} = \mathbf{y} + \mathbf{t}$. The optimal solution to the hypothesis test is given by the likelihood ratio test, i.e.,

$$l(\mathbf{z}) = \frac{f_{\mathbf{Z}|H_1}(\mathbf{z}|H_1)}{f_{\mathbf{Z}|H_0}(\mathbf{z}|H_0)} \underset{H_0}{\overset{H_1}{\substack{> \\ \leq}}} \eta, \tag{1}$$

where $f_{\mathbf{Z}|H_i}(\mathbf{z}|H_i)$ is the pdf of $\mathbf{Z}$ conditioned on hypothesis H_i and η is a threshold which can be adjusted so as to optimize a certain criterion (Neyman-Pearson, Bayes, etc.). The output of the detector will be denoted by $D \in \mathcal{H} = \{H_0, H_1\}$. The detection function given by (1) divides the subspace $\mathbb{R}^n$ in two disjoint regions, $\mathcal{R}$ and $\mathcal{R}^c$, termed *acceptance* or *detection region* and *rejection region*, respectively, such that $\mathbb{R}^n = \mathcal{R} \cup \mathcal{R}^c$. These regions are defined as

$$\mathcal{R} = \{\mathbf{z} \in \mathbb{R}^n : l(\mathbf{z}) > \eta\}; \qquad \mathcal{R}^c = \{\mathbf{z} \in \mathbb{R}^n : l(\mathbf{z}) \leq \eta\}.$$

Since analytical derivation of the likelihood ratio test is not always feasible, a more general family of detection functions, given by

$$f(\mathbf{z}, \boldsymbol{\theta}) \underset{H_0}{\overset{H_1}{\substack{> \\ \leq}}} \eta,$$

is usually considered, where $\boldsymbol{\theta}$ is the secret key used in the detection process. Of course, the resulting detector will be optimal only when $f(\mathbf{z}, \boldsymbol{\theta})$ coincides with the likelihood ratio $l(\mathbf{z})$.

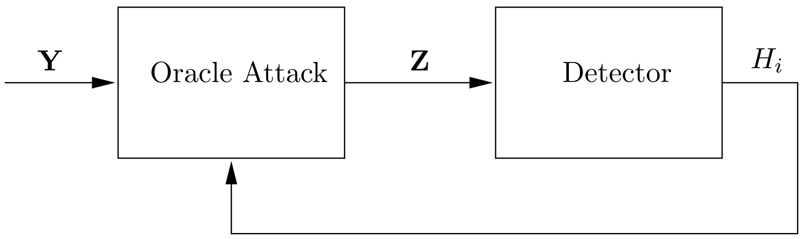


Figure 1: Block diagram of oracle attacks.

In the considered scenarios, i.e. copyright protection and copy control, the watermark detector is often made public, generally in the form of a tamper-proof black box which only provides binary outputs, in such a way that an observer can check whether $f(\mathbf{z}, \boldsymbol{\theta})$ is larger or smaller than η, but he/she can not know its actual value. This scenario gives rise to the so-called *oracle attacks*, where the attacker uses the detector outputs to some selected inputs in order to gain knowledge about secret information used in the detection process (for instance, the detection key). Intuitively speaking, the detector acts as an oracle, responding *yes* or *no* to the inputs provided by the attacker. A block diagram of the oracle attacks can be found in Fig. 1.

The most popular oracle attack is the so-called *sensitivity attack*, introduced for the first time in [4]. At the time this attack was proposed, *additive spread spectrum* methods [3] constituted the state of the art in digital watermarking, so this attack was suited to this particular scenario. For additive spread spectrum under the assumption of a Gaussian host, the likelihood ratio is given by the linear correlation $l(\mathbf{z}) = \mathbf{z}^T\mathbf{w}$, so the optimal detector in this case must apply the following test:

$$\mathbf{z}^T\mathbf{w} \underset{H_0}{\overset{H_1}{\substack{>\\ \leq}}} \eta. \tag{2}$$

Essentially, the sensitivity attack [4] (specialized to the case of digital images) for this kind of detectors consists of obtaining a new image $\mathbf{z}$ near the boundary of $\mathcal{R}$, in such a way that modifying one-by-one the pixels of the new image one can observe how the sign of this modifications affects the outputs of the detector; combining these observations, a rough estimate $\hat{\mathbf{w}}$ of the watermark vector can be computed. Therefore, the attacker will disclose the boundary of the detection region, which is supposed not to be known by unauthorized users. With this knowledge at hand, the attacker could generate an attacked image $\mathbf{z}$ with small distortion, capable of fooling the detector, just by subtracting a suitably scaled version of $\hat{\mathbf{w}}$. The importance of the sensitivity attack relies on disclosing the watermark with a complexity $O(n)$, instead of the $O(2^n)$ due to the *brute force* approach, which was previously believed to be the solution to this estimation problem.

This paper is concerned with the generalization of the sensitivity attack to any watermarking scheme with an available public detector, providing a formulation that encompasses most known watermark detection schemes with parameterizable and differentiable (but unknown to the attacker) detection boundaries. Furthermore, we will also use the attack for generating false positives, i.e. from a watermarked signal and an original unwatermarked different signal, a watermarked version of the latter will be constructed. The rest of the paper is organized as follows: Section 2 provides an overview of previous works dealing with this attack and the countermeasures proposed to increase the security of a watermarking system where public detectors are available. In Section 3, our new formulation of the problem is presented, and its application to some examples is given in Section 4.

2 Previous work and improvements

The sensitivity attack for detectors based on linear correlation, i.e., those given by (2), was extensively studied in [12] and [11]. Starting from the formulation of the attack given in [4], which was explained in Section 1, the work in [12] proposes a first countermeasure based on the randomization of the detection boundary: the basic idea is to define a region around the points that satisfy $\mathbf{z}^T\mathbf{w} = \eta$, where the decision of the detector is made random, in order to reduce the sensitivity of the detector to small changes in its inputs. The internal behavior of the detector is such that its outputs are deterministic, i.e., the response of the detector is always the same for a fixed input signal $\mathbf{z}$, in order to avoid the estimation of the randomization probability simply by feeding the same $\mathbf{z}$ to the detector repeatedly. In [12], the information leakage about the watermark provided by the detector outputs is quantified in an information-theoretic sense, and the shape of the optimum randomization function that minimizes the information leakage is given. Anyway, estimation of the watermark is still possible; it is easy to see that this countermeasure complicates the sensitivity attack, but its complexity still remains linear with the dimensionality of the images. In fact, a practical method for estimating the watermark in this framework was devised in [11].

Another approach for performing a successful sensitivity attack was presented in [14]. The method is able to estimate the boundary of the acceptance region by modeling the attack as a classical adaptive filtering problem: it is easy to realize that the linear detection function given in (2) for additive spread spectrum can be thought of in terms of filtering $\mathbf{z}$ with a filter $\tilde{\mathbf{w}}$ such that $\tilde{w}_i = w_{n+1-i},\ \forall\ i = 1, \cdots, n$; furthermore, the attacker knows that $\mathbf{z} * \tilde{\mathbf{w}} = f(\mathbf{z}, \boldsymbol{\theta})$, where $*$ denotes the convolution operator, so if he/she can access the values of $f(\mathbf{z}, \boldsymbol{\theta})$, then using this signal as reference he/she can manage to construct an estimate of $\tilde{\mathbf{w}}$. The authors propose in [14] the use of the Least Mean Squares (LMS) algorithm in order to iteratively construct these estimates. In a more realistic situation, the attacker only has access to the detector output D, so the authors introduce some modifications in the algorithm to work with just the binary output of the system. The authors also propose some modifications in order to cope with the countermeasure introduced in [12], which was explained in the above paragraph.

In view of the security flaws presented by traditional spread spectrum methods under sensitivity-like attacks, researchers put their effort in the design of *asymmetric* schemes [9].[2] One of the advantages offered by asymmetric schemes

[2]Watermarking techniques can be roughly classified according to the role of the secret key in the embedding/detection processes: those methods which use different keys for embedding and detection are termed *asymmetric*, otherwise they belong to the category of *symmetric schemes*.

against sensitivity attacks is the fact that the embedding and detection keys are different, thus the impact of a successful attack revealing the detection boundary is minimized (recall that disclosure of the watermark in traditional spread spectrum methods allows to unwatermark legal contents, as well as generating forged illegal documents). The other advantage of asymmetric watermarking is the use of more involved detection regions, complicating the description of the detection boundary; for instance, in [9], four asymmetric methods are analyzed under a unified framework, showing that the detection function can be written in terms of a quadratic form in $\mathbb{R}^n$ for all cases, i.e.

$$\frac{\mathbf{z}^T \mathbf{A} \mathbf{z}}{n} \underset{H_0}{\overset{H_1}{\underset{\leq}{>}}} \eta,$$

where $\mathbf{A}$ is a matrix which depends on the secret key $\boldsymbol{\theta}$.

The idea of increasing the security of the system against sensitivity attacks by complicating the detection region is exploited by the family of detection functions called JANIS [8], which use N-th order polynomial detection functions, i.e.

$$f(\mathbf{z}, \boldsymbol{\theta}) = \frac{1}{n} \sum_{k=1}^{n/N} \prod_{j=1}^{N} z_{p[(k-1)\cdot N+j](\boldsymbol{\theta})} \cdot a_{p[(k-1)\cdot N+j](\boldsymbol{\theta})}(\boldsymbol{\theta}),$$

where $\mathbf{a}$ is a pseudorandom ± 1 vector and $\mathbf{p}$ is a pseudorandom permutation vector, both of them depending on the secret key $\boldsymbol{\theta}$. Based on this detection function the watermark is obtained as $\mathbf{w} = \gamma \nabla f(\mathbf{x}, \boldsymbol{\theta})$, where γ is a parameter to adjust the embedding distortion and $\nabla(\cdot)$ is the gradient operator.[3] Indeed it makes more difficult the sensitivity attack, but obviously this is not the ultimate solution: for example, an N-th order detection boundary can still be described by estimating n^N points on such boundary. This point was addressed in [14], showing that the LMS attack can be properly modified in order to cope with this kind of detection boundaries. A possible solution to this problem

[3]Be aware that $\mathbf{a}$ is needed by the encoding algorithm, so JANIS is not an asymmetric method.

was proposed also in [14] by means of non-parametric decision boundaries, i.e., by using decision boundaries that can not be described by a finite number of parameters. An example of such decision boundaries are those given by fractal curves like the Peano curve, which is used in [14] to replace the original linear detection boundary in a spread spectrum scheme. With a proper design, the authors defend that the proposed method can invalidate sensitivity attacks with slight degradations in robustness.

A different approach is followed in [18] which proposes the use of a decision function which is randomized whenever a signal with a similar hash was previously input to the detector; in this way the attacker could not rely on the outputs of the detector when he/she inputs similar signals, as it is the case of the sensitivity attack. Furthermore, [18] proposed the combination of the latter strategy with a response delay as a possible countermeasure against oracle attacks.

Recently, a rigorous formulation of the sensitivity attack was presented in [7]: first, the convergence of the algorithm proposed in [11] is proven, using the law of large numbers; thereafter, a new non-iterative sensitivity attack for detectors based on linear correlation is presented.[4] The main steps of this new algorithm are outlined in the following:

1. As in the former algorithms, the first step is the construction of a signal $\mathbf{z}'$ near the boundary of the detection region.

2. Now consider the set of vectors $\{\mathbf{t}^i\}, \quad i = 1, \cdots, n$, defined by the canonical basis of $\mathbb{R}^n$. For each $\mathbf{t}^i$, a signal $\mathbf{z}'' = \mathbf{z}' + \alpha_i \mathbf{t}^i$ on the detection boundary is constructed, by properly selecting the scaling factor α_i. The search for this value of α_i must be accomplished by means of some numerical algorithm, so it will be surely the most costly part.

3. For the detector under consideration, it holds that $(\mathbf{z}'')^T \mathbf{w} = \sum_{k=1}^{n} z'_k w_k + \alpha_i w_i = \eta, i = 1, \cdots, n$ where η is the detection threshold, and $w_i =$

[4] As a further contribution, this new algorithm is also suitable for estimating continuous-valued watermarks, whereas the algorithms previously proposed in [12] and [11] assumed that the watermark could only take discrete values.

$(\mathbf{t}^i)^T\mathbf{w}$. Thus, a linear system with n equations and n unknowns has been defined. By taking into account the special structure of this system, it is easy to show that it can be solved in $n+1$ elemental operations.

Another remarkable contribution of [7] is the extension of the sensitivity attack in order to work with a more generic family of detection functions of the form $g(\mathbf{y},\mathbf{w})$; furthermore, this method has the advantage of returning an estimate of the watermark. Nevertheless, this approach presents several drawbacks: the attacker needs to know the detection function and even the inverse of the gradient of the detection function. Thus, the need for a new formulation which overcomes these problems is justified; in the next section we will try to solve this problem, achieving a solution which will be shown to work with a wider range of detection functions. The method proposed has the following characteristics:

- It does not require knowledge about the detection function; it just needs to know the binary output of the detection function for a given input. Due to this, our method is indeed able to deal with watermarking methods which use a secret detection key, in such a way that the attacker has no access to the decoding function. This is the case, for example, of asymmetric watermarking and JANIS.
- The gradient of the detection function does not need to be inverted. As it was said in the previous point, sometimes the detection function will not be known by the attacker, so he/she will not be able to invert its gradient.

3 The Blind Newton Sensitivity Attack (BNSA)

Focusing on watermark detection, we will describe the detector output through the function $f_{\text{binary}}:\mathbb{R}^n \to \mathcal{H}$, with $\mathcal{H}=\{H_0,H_1\}$. Without loss of generality, we can define the following functions

$$\begin{aligned} f &: \mathbb{R}^n \to \mathbb{R}^m \text{ and} \\ g_{\text{binary}} &: \mathbb{R}^m \to \mathcal{H}, \end{aligned} \tag{3}$$

with $m \leq n$, in such a way that $f_{\text{binary}} = g_{\text{binary}} \circ f$, and f is parameterized by the secret key $\boldsymbol{\theta}$.[5] This decomposition will be shown to be useful in the next sections, since some of the most popular watermarking techniques perform embedding/detection in a projected domain so f can be seen as the projection function. Furthermore, in the schemes studied in this paper the output of g_{binary} will be based on the output of a real function g and a threshold η, in such a way that

$$g_{\text{binary}}(\mathbf{x}) = \begin{cases} H_0, & \text{if } g(\mathbf{x}) \leq \eta \\ H_1, & \text{if } g(\mathbf{x}) > \eta \end{cases}, \tag{4}$$

with $g : \mathbb{R}^m \rightarrow \mathbb{R}$.

On the other hand, a distortion measure has to be defined in order to quantify the impact of the attacking signal, termed $\mathbf{t}$, on the watermarked signal $\mathbf{y}$: [6]

$$\begin{aligned} d_{\mathbf{y}} : \mathbb{R}^n &\rightarrow \mathbb{R}^+ \\ \mathbf{t} &\rightarrow d_{\mathbf{y}}(\mathbf{t}). \end{aligned}$$

This distortion measure should be based on perceptual criteria (depending on the nature of the host signal), although very often, and for the sake of simplicity, the squared Euclidean norm of $\mathbf{t}$ is chosen (i.e., $d_{\mathbf{y}}(\mathbf{t}) = ||\mathbf{t}||_2^2$).

Recalling that the attacker tries to find the vector $\mathbf{t}$ which yields a "no watermark" decision (i.e., $f_{\text{binary}}(\mathbf{y}+\mathbf{t}) = H_0$) while minimizing the distortion measure $d_{\mathbf{y}}(\mathbf{t})$, his/her target can be formalized as

$$\arg \min_{\mathbf{t}: g \circ f(\mathbf{y}+\mathbf{t}) \leq \eta} d_{\mathbf{y}}(\mathbf{t}). \tag{5}$$

Let us assume that $d_{\mathbf{y}}(\mathbf{t})$ is a continuous and convex function of $\mathbf{t}$ (for a given watermarked signal $\mathbf{y}$), which achieves its global minimum value at $\mathbf{t}_0$ (the squared Euclidean norm obviously fulfills these conditions), a vector that

[5] Be aware that f is generalized with respect to its definition in the previous sections, in order to provide m-dimensional outputs, and the dependency with $\boldsymbol{\theta}$ is not explicitly shown for the sake of notational simplicity.

[6] Ideally, this measure should quantify the differences between the original host signal and its attacked version; nevertheless, the attacker will have to design his/her strategy taking into account the watermarked signal, since he/she has not access to the original one.

belongs to the set of attacking vectors yielding H_1 (which we will denote by $\mathcal{B}$),[7] i.e., $\mathbf{t}_0 \in \mathcal{B} \triangleq \{\mathbf{t} : g \circ f(\mathbf{t}+\mathbf{y}) > \eta\}$. Be aware that $\mathcal{B} \triangleq \mathcal{R} - \mathbf{y}$. Then, replacing $\mathcal{B}$ in (5), and denoting by $\partial\mathcal{B}$ its boundary and by $\mathcal{B}^c$ its complement, it is straightforward to show that $\arg\min_{\mathbf{t}\in\mathcal{B}^c} d_{\mathbf{y}}(\mathbf{t}) \in \partial\mathcal{B}$, so (5) is equivalent to

$$\arg \min_{\mathbf{t}: g\circ f(\mathbf{y}+\mathbf{t})=\eta} d_{\mathbf{y}}(\mathbf{t}). \tag{6}$$

This is a typical Lagrange multipliers problem, so the attacker could find a theoretical solution if both $d_{\mathbf{y}}$ and $g \circ f$ were known by him/her; nevertheless, this is not the case, since the latter depends on the secret key, which is unknown for the attacker. Actually, he/she will have only access to the binary output of the detector. In Appendix A we show that (6) is equivalent to

$$\arg \min_{\mathbf{s}\in\mathbb{R}^n} d^{\star}_{\mathbf{y}}(h_{\mathbf{y}}(\mathbf{s})), \tag{7}$$

where $d^{\star}_{\mathbf{y}}$ is the restriction of $d_{\mathbf{y}}$ to $\partial\mathcal{B}$, and $h_{\mathbf{y}}$ is a function verifying that: a) $h_{\mathbf{y}}$ is a surjection which maps $\mathbb{R}^n$ onto $\partial\mathcal{B}$, i.e. $h_{\mathbf{y}}(\mathbb{R}^n) = \partial\mathcal{B}$; b) $h_{\mathbf{y}}(\mathbf{s}) = \mathbf{s}$, for all $\mathbf{s} \in \partial\mathcal{B}$; c) $h_{\mathbf{y}}(\mathbf{s}) \in C^2$, i.e., its second derivative is continuous in a neighborhood of $\mathbf{s}$; and d) $h_{\mathbf{y}}(\mathbf{s})$ is estimated based on the binary output of the detector, without any other knowledge about the detection function.

Since theoretical solutions to (7) are not possible in general due to the lack of knowledge of the boundary of the decision region, numerical iterative methods should be applied by the attacker in order to find a solution. Due to this, in most cases the attacker will have to be satisfied with computing a local minimum of the function in (7), since the achievement of global minima is only ensured for convex problems, which need also the convexity of $\mathcal{B}^c$ [2]; nevertheless, in Section 4 the experimental results will show that this suboptimal solution is usually good enough. Concerning the iterative method, in this paper we will use an adaptation of Newton's method [16], where the updated vector in the $(k+1)$-th iteration is computed as

$$\mathbf{s}_{k+1} = \mathbf{s}_k - \xi_k \cdot \left[\nabla^2 (d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k)\right]^{-1} \cdot \nabla (d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k), \tag{8}$$

[7] Be aware that in most cases it is reasonable to consider that $\mathbf{t}_0 = \mathbf{0}$, since in that case the attacked signal will be the watermarked one, so the distortion is minimized; furthermore $\mathbf{t}_0$ is in $\mathcal{B}$, since $g \circ f(\mathbf{y})$ will yield H_1.

where $\xi_k \in \mathbb{R}^+$ is the step-length, whose computation requires (in general) a line search [16, 2]: a small value of ξ_k will imply a slow convergence, but with a large one convergence cannot be assured.

It is straightforward to see that $\nabla(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k)$ and $\nabla^2(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k)$ cannot be obtained in an analytic way, therefore they must be numerically approximated by taking into account that

$$\frac{\partial(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})}{\partial s_i}(\mathbf{s}) = \frac{(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}+\delta\mathbf{e}_i) - (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s})}{\delta} + O(\delta), \text{ and} \tag{9}$$

$$\begin{aligned}\frac{\partial^2(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})}{\partial s_i \partial s_j}(\mathbf{s}) &= \frac{(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}+\delta\mathbf{e}_i+\delta\mathbf{e}_j) - (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}+\delta\mathbf{e}_i)}{\delta^2} \\ &+ \frac{-(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}+\delta\mathbf{e}_j) + (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s})}{\delta^2} + O(\delta),\end{aligned} \tag{10}$$

with $\mathbf{e}_i$ the i-th vector of the canonical basis, and $\delta > 0$ the step size used by the approximation.

An alternative strategy to (8), which is especially suitable for large-scale problems, is based on replacing the Hessian by a diagonal matrix just keeping the diagonal elements; in this way, an iteration of the algorithm just requires $(2 \cdot n+1)$ evaluations of $(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s})$ and (8) is computed with n scalar divisions (if the complete matrix were used, a linear system with n equations and n variables should be solved). Another updating algorithm is given by

$$\mathbf{s}_{k+1} = \mathbf{s}_k - \xi_k \cdot \hat{\nabla}(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k), \tag{11}$$

where $\hat{\nabla}(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k)$ is the estimation of the gradient of $(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})$ based on (9), which only requires $n+1$ evaluations; for a small enough ξ_k, (11) will guarantee a decrease in $(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})$ as far as $(\hat{\nabla}(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k))^T \cdot \nabla(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k) > 0$, where the last condition is based on the Taylor series expansion of the objective function.

Although $h_{\mathbf{y}}(\mathbf{s})$ could be any function verifying the properties introduced above, in the experimental part of this paper we will use $h_{\mathbf{y}}(\mathbf{s}) = \alpha^* \cdot \mathbf{s}$, where $\alpha^* = \arg\min_{\alpha \in \mathbb{R}: \mathbf{y}+\alpha\cdot\mathbf{s} \in \partial\mathcal{B}} |\alpha|$. This computation of $h_{\mathbf{y}}(\cdot)$ is based on the fact that $\mathbf{0} \in \mathcal{B}$, since $\mathbf{y}$ is a watermarked signal, and also that for most known watermarking methods $\beta \cdot \mathbf{s} \in \mathcal{B}^c$ for large values of β, so α^* (equivalently $h_{\mathbf{y}}$) can be estimated by a bisection algorithm, where just the binary output of the detector is needed to determine what will be the extremes of the interval at the

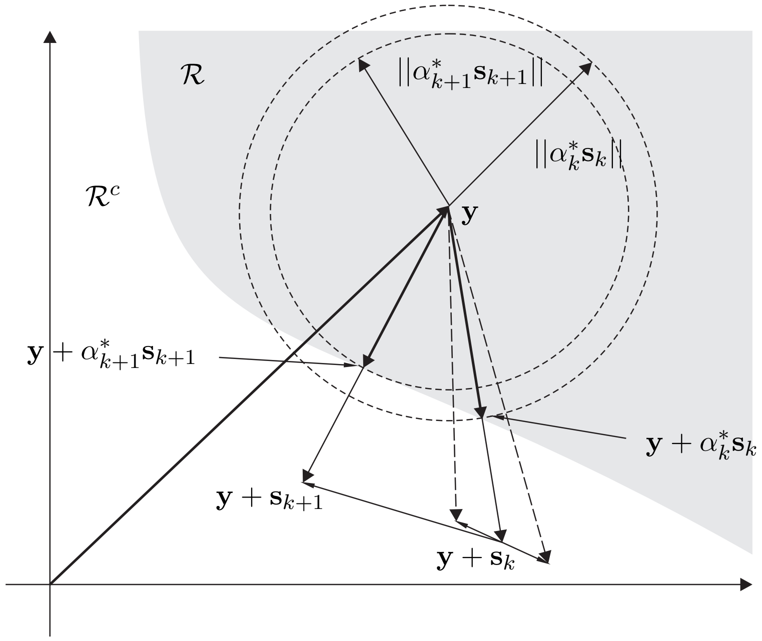


Figure 2: Example of an iteration of the algorithm. Given a watermarked signal $\mathbf{y}$ and the attacking vector in the k-th iteration $\mathbf{s}_k$, the last one is slightly modified to estimate the gradient and Hessian of $d^\star_{\mathbf{y}} \circ h_{\mathbf{y}}(\mathbf{s}_k)$. Once the descent direction and the step-length have been computed, $\mathbf{s}_{k+1}$ is obtained. It can be seen that $\mathbf{y}+\alpha^*_{k+1}\mathbf{s}_{k+1}$ is closer to $\mathbf{y}$ than $\mathbf{y}+\alpha^*_k\mathbf{s}_k$, i.e. $||\alpha^*_{k+1}\mathbf{s}_{k+1}||^2 < ||\alpha^*_k\mathbf{s}_k||^2$.

next iteration. Given that this method is based on the binary output of the detector, without any other knowledge about the detection function, the algorithm is said to be *blind*. An example of an iteration of the algorithm is plotted in Fig. 2, where the reduction from $||\alpha^*_k\mathbf{s}_k||^2$ to $||\alpha^*_{k+1}\mathbf{s}_{k+1}||^2$ is represented by the decrease of the radius of two spheres centered at $\mathbf{y}$. It is important to remark that the blindness of the proposed attack comes at the price of its iterative nature and thus typically requires many more calls to the oracle than other algorithms, e.g. [4], [11], [7], for which the attacker is assumed to perfectly know the shape of the detection function.

3.1 Implementation

In this section, the different steps of BNSA are enumerated, and some considerations on its implementation are made.

1. Initialization: in order to start to iterate, a first point on the boundary of the detection region is needed. We propose to use $\mathbf{z} = \gamma^* \cdot \mathbf{y}$, where $\gamma^* = \arg\max_{\gamma \in \mathbb{R}^+ : g \circ f(\gamma \mathbf{y}) = \eta}(\gamma)$. This choice is based on the fact that for most known watermarking methods $\mathbf{0}$ is in $\mathcal{R}^c$. Were this not the case, we assume that the attacker can compute a signal yielding in $\mathcal{R}^c$, and hence a document on the decision boundary can still be computed by simple convex combination of this signal and the watermarked one. The factor γ^* can be computed using a bisection algorithm.

   Therefore, the initialization value of $\mathbf{s}_k$ is given by $\mathbf{s}_0 = (\gamma^* - 1)\mathbf{y}$.

2. Updating: the update of the algorithm has the generic form

$$\mathbf{s}_{k+1} = \mathbf{s}_k - \xi_k \cdot \mathbf{B}^{-1} \cdot \hat{\nabla}(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k),$$

   where $\hat{\nabla}(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k)$ is the estimate of the gradient of $(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k)$, and its i-th component is computed as

$$[\hat{\nabla}(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k)]_i = \frac{(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s} + \delta \mathbf{e}_i) - (d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s})}{\delta}.$$

   This computation requires also to estimate $h_{\mathbf{y}}(\cdot)$, which is not known by the attacker. As it was mentioned above, the proposed strategy is to use $h_{\mathbf{y}}(\mathbf{s}) = \alpha^* \cdot \mathbf{s}$, where $\alpha^* = \arg\min_{\alpha \in \mathbb{R} : \mathbf{y} + \alpha \cdot \mathbf{s} \in \partial \mathcal{B}} |\alpha|$, although any function verifying the conditions described above could be used. Again, in the proposed implementation α^* is computed using a bisection algorithm.

   Concerning the matrix $\mathbf{B}$, different possibilities can be considered; they are chosen by the attacker based on his/her computational resources. First of all, we can consider an approximation to the Hessian, i.e. $\mathbf{B} = \hat{\nabla}^2(d^\star_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}_k)$, which yields an approximation to the classical Newton's method. The (i,j)-th component of that matrix is computed as

$$\begin{aligned}[\hat{\nabla}^2(d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s})]_{i,j} &= \frac{(d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}+\delta\mathbf{e}_i+\delta\mathbf{e}_j) - (d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}+\delta\mathbf{e}_i)}{\delta^2} \\ &+ \frac{-(d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s}+\delta\mathbf{e}_j) + (d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s})}{\delta^2}.\end{aligned}$$

This alternative is really computationally expensive, since it requires $O(n^2)$ calls to the detector per update of the algorithm.

Finally, the computation of ξ_k, which is done just once per iteration of the algorithm, can be calculated using one of-the-shelf line search algorithm (see [16, 2] for further references). In our implementation we have followed Armijo's rule, due to its simplicity.

3. Stopping: the stopping condition could be based on different criteria. For example, the attacker could stop when he/she computes a non-watermarked signal with a sufficiently good quality, or when the distortion measure function is not significantly reduced from one iteration to the next. Another alternative is to constrain the number of calls to the detector.

3.2 Computing forgeries

Let us suppose now that the attacker has access to a watermarked signal $\mathbf{y}$ and an original unwatermarked content $\mathbf{x}$, i.e. $f_{\text{binary}}(\mathbf{x}) = H_0$. A possible application of the proposed method is the computation of a signal $\mathbf{z} = \mathbf{x} + \hat{\mathbf{w}}$ which yields $f_{\text{binary}}(\mathbf{z}) = H_1$ while minimizing the distortion when it is compared with $\mathbf{x}$, i.e., we can create a watermarked version of a signal $\mathbf{x}$ whenever a watermarked content $\mathbf{y}$ and a detector are available.

Following the approach introduced in the previous section, this problem can be formalized as

$$\arg \min_{\mathbf{t}: g\circ f(\mathbf{y}+\mathbf{t})>\eta} d_{\mathbf{x}}(\mathbf{y}+\mathbf{t}-\mathbf{x}), \tag{12}$$

i.e., we are trying to find a vector $\mathbf{t}$ such that when it is added to $\mathbf{y}$ yields a watermarked signal which is as similar as possible to $\mathbf{x}$, so $\mathbf{z} = \mathbf{x} + \hat{\mathbf{w}} = \mathbf{y} + \mathbf{t}$. Assuming again that $d_{\mathbf{x}}(\mathbf{t})$ is a continuous and convex function of $\mathbf{t}$, the solution

$\mathbf{t}^*$ to (12) will verify $g \circ f(\mathbf{y} + \mathbf{t}^*) = \eta + \epsilon$, where $\epsilon > 0$ is arbitrarily small. Therefore, from a practical point of view, we can think of $\mathbf{t}^*$ as being on $\partial\mathcal{B}$. Proceeding in a similar way to (7), we can rewrite (12) as

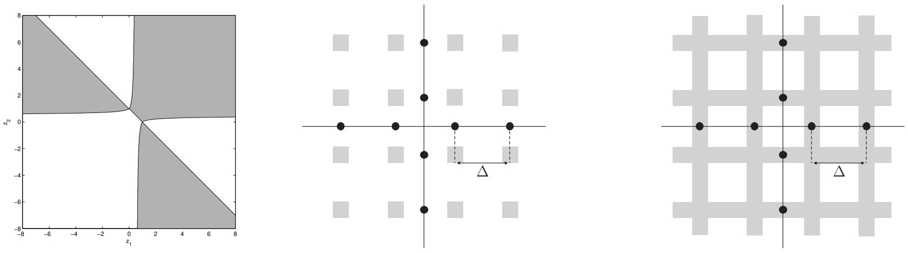
$$\arg\min_{\mathbf{s}\in\mathbb{R}^n} d_{\mathbf{x}}(\mathbf{y} + h_{\mathbf{y}}(\mathbf{s}) - \mathbf{x}). \tag{13}$$

The solution to the last problem will be based on numerical iterative methods, so in the general case of non-convex decision regions just a local minimum will be achieved, as it happened with the watermark removal problem.

Taking into account that the probability of randomly finding a point in the detection region is given by the probability of false alarm P_{fa},[8] which typically is really small due to design criteria, the attacker still needs to be given a watermarked signal $\mathbf{y}$ (which can be completely different of the document to be forged $\mathbf{x}$) in order to define the projecting function $h_{\mathbf{y}}(\cdot)$, which will allow him/her to compute documents on the detection boundary. Assuming that he/she has access to a watermarked signal $\mathbf{y}$, then $h_{\mathbf{y}}(\cdot)$ can be given by the expression proposed in Section 3.1, i.e. $h_{\mathbf{y}}(\mathbf{s}) = \alpha^* \cdot \mathbf{s}$, where $\alpha^* = \arg\min_{\alpha\in\mathbb{R}:\alpha\cdot\mathbf{s}\in\partial\mathcal{B}} |\alpha|$.

Finally, one must be aware that the decision regions could depend on the statistics of the input signal, as it is the case for the ML detector for a Generalized Gaussian distributed host, where the decision region depends on the shape parameter of the received signal. This dependency has a limited impact when the initial signal is not substantially modified, since in that case the statistics of that initial signal will be almost unaltered; this will be usually the case when the attacker tries to unwatermark a signal, but *the statistics* could dramatically change the decision regions when the attacker tries to *move* from one image to another, possibly frustrating an attack.

[8]The probability of false alarm P_{fa} is defined as $\Pr\{g_{\text{binary}} \circ f(\mathbf{x} + \mathbf{t}) = H_1|H_0\}$. On the other hand, the probability of missed detection P_m is defined as $\Pr\{g_{\text{binary}} \circ f(\mathbf{x} + \mathbf{w} + \mathbf{t}) = H_0|H_1\}$.



(a) (b) (c)

Figure 3: Examples of decision regions: (a) Decision regions obtained taking into account a l_c-norm when $c_1 = c_2 = 0.5$. (b) AND Region for QPD. (c) OR Region for QPD.

4 Application to real methods

In this section we will particularize the proposed algorithm to some of the most popular watermarking methods, showing the practical usefulness of this new attack and comparing the performance of the different schemes.

4.1 Spread Spectrum

Detection of standard Spread Spectrum methods is based on the correlation between the received signal $\mathbf{z}$ and the watermark $\mathbf{w}$. Therefore, the function f, defined in (3), projects $\mathbf{z}$ onto a one-dimensional domain ($m = 1$), i.e. $f(\mathbf{z}) = \mathbf{z}^T \cdot \mathbf{w}$, and g in (4) will be the identity function ($g(x) = x$, for all $x \in \mathbb{R}$), so the detection function is given by

$$\mathbf{z}^T \cdot \mathbf{w} \quad \underset{H_0}{\overset{H_1}{\substack{> \\ \leq}}} \quad \eta,$$

in such a way that the boundary of the decision region will be a hyperplane. We will denote this case by *SS-corr*.

Another alternative for the detection function is that proposed by Cox *et al.* in [5]; in that case, f quantifies the angle between the received signal $\mathbf{z}$ and the watermark vector $\mathbf{w}$, i.e. $f(\mathbf{z}) = \frac{\mathbf{z}^T \cdot \mathbf{w}}{||\mathbf{z}|| \cdot ||\mathbf{w}||}$, and g is again the identity function, yielding a decision region $\mathcal{B}$ which is an n-dimensional cone. This method will be named *SS-angle*, as it really measures the angle between the received signal and the watermark.

As a countermeasure against BNSA, one could design detection functions for which component-wise modifications produce bounded increments, since for this kind of functions the task of finding vectors on the boundary of the detection function is considerably complicated. Interestingly, the ML detection function for Generalized Gaussian distributed hosts (see [10]), denoted by *SS-GG*, fulfills this requirement whenever the shape parameter c_k is such that $c_k < 1$. Such ML detection function can be written as

$$f(\mathbf{z}) = \sum_{k=1}^{n} \beta_k^{c_k} \left(|Y_k|^{c_k} - |Y_k - \alpha_k s_k|^{c_k} \right), \tag{14}$$

where c_k is the shape parameter of the k-coefficient, $\beta_k = \frac{1}{\sigma_{X_k}} (\Gamma(3/c_k)/\Gamma(1/c_k))^{1/2}$, and α_k is the watermark strength for the aforementioned coefficient. An example of the resulting decision region is plotted in Fig. 3(a).

4.2 Side-informed methods

In Section 2 the JANIS methods were introduced. In order to make a comparison with the other existing methods, we have fixed the order of the detection function to 4, so

$$f(\mathbf{z}) = \frac{1}{n} \sum_{k=1}^{n/4} \prod_{j=1}^{4} z_{p[(k-1)\cdot N+j]} \cdot a_{p[(k-1)\cdot N+j]}.$$

Quantization-based methods have been shown to be useful for data hiding applications; nevertheless, and despite of their success in that application, very little has been said about their use in detection scenarios. To the best of our knowledge, the first work addressing the problem from this point of view is [6], where the Scalar Costa Scheme is adapted to authentication purposes by embedding a fixed message, yielding the detection function $g_{\mathrm{SCS}}(\mathbf{z}, \boldsymbol{\theta}) = \frac{f_{\mathbf{Y}}(\mathbf{z})}{f_{\mathbf{X}}(\mathbf{z})}$. Note that in this case the sensitivity attack is straightforward, since it can be done componentwise.

On the other hand, in [13] the received signal $\mathbf{z}$ is quantized with a lattice Λ and the decision is made upon the squared norm of the quantization error. Formalizing it, we can write $f(\mathbf{z}) = ||\mathbf{z} \mod \Lambda||^2$, and g is the identity function again. In this way, the acceptance region is the union of n-dimensional hyperspheres centered at the centroids of Λ. From the point of view of attacking such a system, this decision region assures that the attacker can produce a signal yielding H_0 by adding *any* noise vector with a given variance, as far as that noise vector is independent of the self noise. Therefore, a sensitivity attack is not really necessary in this case.

Another approach to this problem is Quantized Projection based Detection (QPD) [17], where uniform scalar quantizers are used to quantify a m-dimensional projected version of the received signal $\mathbf{z}$ and the detection function depends on the quantization error, introducing two different strategies: the AND and OR detection regions, which can be formalized as

$$\begin{aligned} f_i(\mathbf{z}) &= \sum_{j=1}^{n} a_{ij} z_j, \ 1 \leq i \leq m, \\ g_{\mathrm{AND}}(f(\mathbf{z})) &= \max_{1 \leq i \leq m} |(f_i(\mathbf{z}) \mod \Delta) - \Delta/2|, \text{and} \\ g_{\mathrm{OR}}(f(\mathbf{z})) &= \min_{1 \leq i \leq m} |(f_i(\mathbf{z}) \mod \Delta) - \Delta/2|, \end{aligned}$$

where Δ is the quantization step, a_{ij} are the secret projection matrix coefficients and m the dimensionality of the projected subspace. The resulting methods are denoted as *QPD-AND* and *QPD-OR*, and the obtained decision regions are plotted in Fig. 3(b) and Fig. 3(c), respectively, for $m = 2$. The convergence of the algorithm introduced in Section 3 for finding the optimal attacking vector will be very much slower for the OR region, since the cost function has its minimum value at a non-differentiable point of g_{OR}. In fact, in such case we will follow a different strategy in which we try to estimate the m projecting vectors; this implies the complete disclosure of the secret key, and the optimal attacking vector can be computed as the sum of those vectors.

4.3 Comparison

This section shows the robustness of the watermarking methods introduced above to the BNSA attack. Experiments were carried out for both synthetic and real images, considering two different measures of robustness:

1. The ratio ρ (in dB) between the power needed to achieve an unwatermarked signal and the embedding power. If the watermarked image is $\mathbf{y} = \mathbf{x} + \mathbf{w}$ (with $\mathbf{x}$ the host and $\mathbf{w}$ the watermark) and the attacked image is $\mathbf{z} = \mathbf{y} + \mathbf{t}$ (with $\mathbf{t}$ the output of BNSA), then
$$\rho = 10 \log_{10} \left(\frac{||\mathbf{t}||^2}{||\mathbf{w}||^2} \right).$$
The average of ρ over a large number of realizations is a good indication of the robustness of each method against BNSA when used to remove the watermark. This will be the robustness measure used for synthetic images.

2. The Peak Signal to Noise Ratio (PSNR) of the attacked image, defined as
$$\text{PSNR}_{att} \triangleq 10 \log_{10} \left(\frac{(n \cdot 255)^2}{||\mathbf{v}||^2} \right),$$
where $\mathbf{v}$ is the vector added to the considered image. Notice that in the watermarking removal problem (see Eq. (7)), we have $\mathbf{v} = h_{\mathbf{y}}(\mathbf{s})$, whereas in the problem of computing forgeries, i.e Eq. (13), $\mathbf{v} = \mathbf{y} + h_{\mathbf{y}}(\mathbf{s}) - \mathbf{x}$. This will be the robustness measure used for real images.

4.3.1 Synthetic images

In this case, the host images are random vectors generated according to a Gaussian distribution, except when the ML detector for the Generalized Gaussian host is used, where we have chosen the shape parameter to be 0.5. In order to make a fair comparison, the value of the probability of false alarm P_{fa} has been fixed to 10^{-4}, $n = 2048$ and the document to watermark ratio to 16 dB (with $\sigma_W^2 = 1$) in order to ensure a reasonable probability of missed detection for all

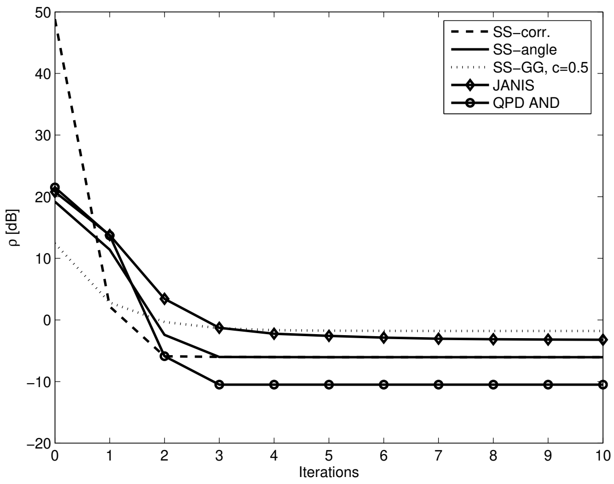


Figure 4: ρ averaged over 100 watermarked Gaussian vectors as a function of the number of iterations (0 when there is not attack), for different decision regions: SS-corr, SS-angle, SS-GG, JANIS and QPD-AND. Iteration 0 corresponds to random attacking vectors (without applying the proposed algorithm).

the studied methods. We have used the version of BNSA with the diagonal estimate of the Hessian.

Fig. 4 shows the robustness measure ρ versus the number of iterations of BNSA, for each of the considered watermarking methods. Iteration 0 means that no BNSA attack was applied, and as such it represents the value of ρ when the attacking vectors are randomly generated.

It can be seen that at iteration 0 SS-corr is much more robust than SS-angle, but the robustness of both methods converge to the same value when the number of iterations is increased. The most robust method against BNSA turns out to be SS-GG with shape parameter $c_k = 0.5$, even when the power required for producing an unwatermarked signal is reduced in 11.8 dB after just 3 iterations, achieving its minimum at -1.79 dB. Close to this result are the -3.21 dB needed by JANIS, for which the power required to produce an unwatermarked signal is reduced in 24 dB after 10 iterations.

For QPD-AND with $m = 10$, as soon as one of the projecting vectors has been estimated, its robustness against BNSA is significantly smaller than that

Figure 5: *Lena* watermarked by JANIS.

of the methods commented above. Finally, QPD-OR with $m = 10$ (not plotted in Fig. 4) shows the smallest robustness among the considered methods, since the value of ρ after 10 iterations is only -38 dB.

4.3.2 Real images

In order to reduce the computational cost, the updating algorithm described in (11) was implemented in this case. Once again, the probability of false alarm P_{fa} was fixed to 10^{-4} for all watermarking methods, for the sake of fairness.

Watermark removal. Image Lena 256×256, i.e. $n = 65536$, was watermarked using the above described methods with an embedding PSNR of 38.58 dB (with $\sigma_W^2 = 9$). The results are summarized in Table 1, which shows the value of PSNR_{att} for one iteration of BNSA. It can be seen the similar behavior of SS-corr and SS-angle, and on the other hand JANIS appears to be most robust method, supporting the conclusions drawn in Section 4.3.1. For illustration purposes, the image watermarked with JANIS is plotted in Fig. 5, whereas the unwatermarked result of applying BNSA can be seen in Fig. 6. Indeed, one

Figure 6: *Lena* watermarked by JANIS and attacked by BNSA.

can observe that the quality of the unwatermarked image obtained by BNSA is better than the quality of the watermarked one, suggesting that BNSA is successful in estimating the actual embedded watermark, i.e, the attacked image is closer to the original host image than its watermarked version.

Fig. 7 is included in order to provide an intuitive idea of the robustness of JANIS against blind attacks, showing the result of applying to the watermarked signal Additive White Gaussian Noise (AWGN) with the power necessary to push the watermarked signal out of the detection region. By comparing this figure to Fig. 6, one can realize that strong robustness against blind attacks does not imply at all robustness in the case of smarter attacks, as those represented by BNSA. The method labeled as *Trellis-based [15]* in Table 1 is considered here for similar reasons: it stands for the Trellis-based side-informed method proposed in [15], well known for its robustness against blind AWGN attacks; however, it is somewhat surprising to see that an unwatermarked version of Lena was obtained with a $\mathrm{PSNR}_{att} = 52.17$, just after one BNSA iteration. In any case, one must be aware that this method was firstly proposed for decoding scenarios, not for detection; we have adapted it to detection by comparing the

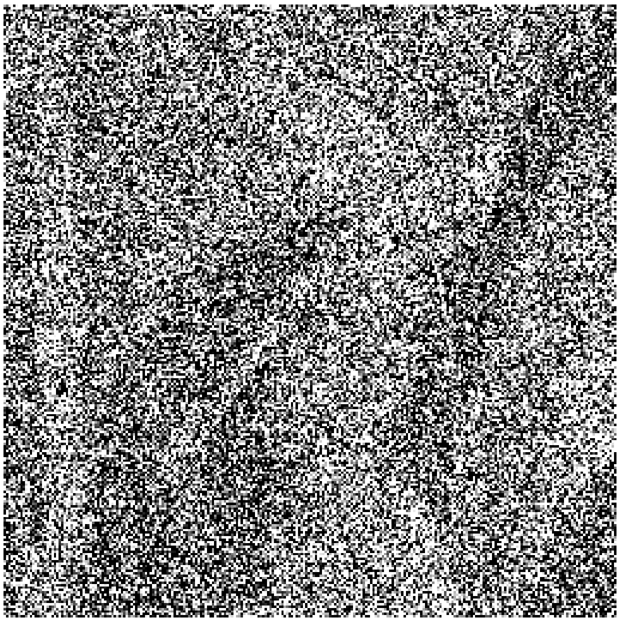


Figure 7: *Lena* watermarked by JANIS and attacked by AWGN.

	SS-corr	SS-angle	JANIS	Trellis-based [15]
PSNR_{att} (dB)	40.25	40.25	38.31	52.17

Table 1: Values of PSNR_{att} for Lena image, after one iteration of BSNA for different watermarking methods.

decoded message with a reference one. Furthermore, the reported results could vary depending on the chosen parameters: in this case, the parameters were fixed to a spreading factor of 4, 16 states, and 8 arcs per state.

Generation of forgeries. For the creation of forgeries, the original unwatermarked signal $\mathbf{x}$ was chosen to be Baboon (Fig. 8), whereas the reference watermarked signal is Lena. The results shown in this section were obtained after 10 BNSA iterations, and are summarized in Table 2. We want to highlight here the result obtained for SS-GG, for which the detection regions are modified when the host statistics change, complicating the optimization problem: the resulting forgery can be found in Fig. 9. Likewise, the forged Baboon for JANIS detector is represented in Fig. 10.

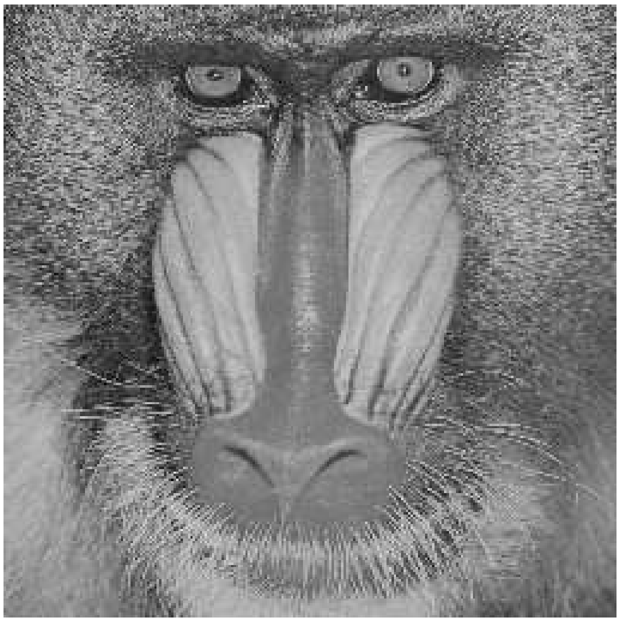


Figure 8: Original *Baboon* 256×256.

	SS-corr	SS-angle	SS-GG	JANIS
PSNR_{att} (dB)	55.73	55.23	32.03	50.69

Table 2: Values of PSNR_{att} for the forgeries of Baboon, after 10 iterations of BSNA for different watermarking methods.

4.4 Computational complexity

One question that needs to be answered is the amount of calls to the detector that are required to achieve the results presented so far. With this aim, the number of calls per dimension needed to perform one iteration of BNSA was averaged for all the images considered in Section 4.3.2. Interestingly, this number was about 58 for all the methods analyzed in this paper, independently of whether the attack consisted in removing the watermark or forging watermarked images. In this computation we used the initialization step and the definition of $h_{\mathbf{y}}(\cdot)$ proposed in Section 3.1, whereas matrix $\mathbf{B}$ was set to the identity and Armijo's rule was chosen for the line search. The tolerance for the bisection algorithm was set to 10^{-12}. This implies that each iteration of BNSA

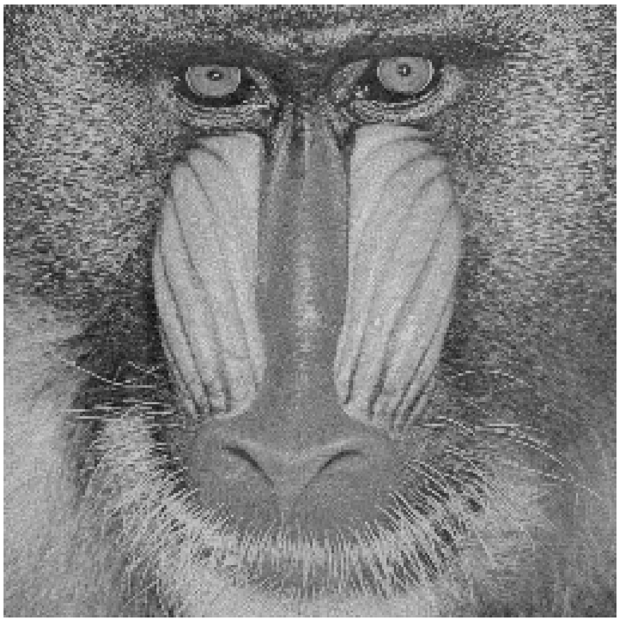


Figure 9: Forgery of *Baboon* for the ML detector of Generalized Gaussian distributed hosts.

for our test images needs about 3.8×10^6 calls to the detector. Although this large number could be frightening at first sight, the attack is still practical, as it is confirmed by the experimental part of this work. In any case, note that the number of calls to the detector required by the non-iterative method proposed in [7] will be similar to that needed by one iteration of BNSA, since in both cases n boundary points must be located using a binary search.

As a final remark, note that our conclusions do not necessarily generalize to other schemes; in fact, one possible countermeasure against BNSA could be based on the design of detection regions which need a large number of iterations of BNSA to converge to an acceptable solution.

5 Final remarks

Following are some guidelines on how to measure the robustness of watermarking methods against BNSA, and the application of BNSA to new scenarios:

- Although ρ can be seen as a measure of the robustness of a watermark-

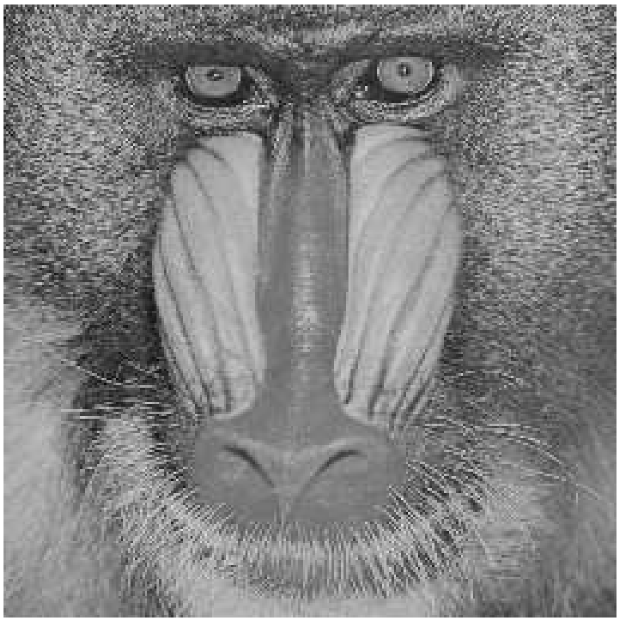


Figure 10: Forgery of *Baboon* for JANIS.

ing method against BNSA, note that this measure does not provide full information on the behavior of a particular method; for instance, QPD methods, which have been shown here to be quite weak against BNSA, have a very good Receiver Operating Characteristic in AWGN channels (see [17] for a comparison with SS-corr).

- Taking into account that it just needs the binary output of the detector, the BNSA is also suitable for detector based on zero-knowledge protocols [1], where, at the end, regardless of the domain where the detection function is computed, the detector will output a binary decision which can be used by the proposed algorithm to estimate the underlying detection region.

- The detection regions based on fractal curves proposed in [14] are also suitable for being attacked by BNSA. Although the decision boundaries are non-parametric, the attacker could try to estimate their envelope, since this estimation will be usually enough to remove the watermark or create forgeries. From a practical point of view, the envelope can be estimated by computing (9) and (10) with a large value of δ.

- As a final remark, the approach presented in this paper can be also used in the case of data-hiding systems, since the decoding process is nothing but a multiple hypothesis test. In this case, any change of the decoder output should be interpreted as if it were done by a change in the detector output; this is equivalent to have the following binary hypothesis: a) the decoded message is changed; b) the decoded message is unaltered.

6 Acknowledgments

Thanks are due to Prof. I. J. Cox and Dr. G. Doërr who kindly provided us with the source code of their algorithm [15].

This work was partially funded by *Xunta de Galicia* under projects PGIDT04 TIC322013PR and PGIDT04 PXIC32202PM; MEC project DIPSTICK, reference TEC2004-02551/TCM; FIS project IM3, reference G03/185 and European Comission through the IST Programme under Contract IST-2002-507932 ECRYPT. ECRYPT disclaimer: The information in this paper is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

A Appendix

In this Appendix we will show that (6) is equivalent to

$$\arg\min_{\mathbf{s}\in\mathbb{R}^n} d^{\star}_{\mathbf{y}}(h_{\mathbf{y}}(\mathbf{s})), \tag{15}$$

with $d^{\star}_{\mathbf{y}}(\mathbf{t})$ the restriction of $d_{\mathbf{y}}(\mathbf{t})$ to those $\mathbf{t}\in\partial\mathcal{B}$, i.e.,

$$\begin{array}{rcl} d^{\star}_{\mathbf{y}}(\mathbf{t}) : \partial\mathcal{B} & \rightarrow & \mathbb{R}^+ \\ \mathbf{t} & \rightarrow & d_{\mathbf{y}}(\mathbf{t}), \end{array}$$

and $h_{\mathbf{y}}(\mathbf{s})$ is a surjection from $\mathbb{R}^n$ to $\partial\mathcal{B}$,[9] i.e., $h_{\mathbf{y}}(\mathbf{s}) : \mathbb{R}^n \rightarrow \partial\mathcal{B}$, such that $h_{\mathbf{y}}(\mathbb{R}^n) = \partial\mathcal{B}$, verifying that $h_{\mathbf{y}}(\mathbf{s}) = \mathbf{s}$ for all $\mathbf{s} \in \partial\mathcal{B}$; we will also assume that $h_{\mathbf{y}}(\mathbf{s}) \in C^2$, i.e., its second derivative exists and is continuous, in a neighborhood of $\mathbf{s}$ (this last point is related to the differentiability of $g \circ f$). Note that $h_{\mathbf{y}}(\mathbf{s})$ just maps the vector $\mathbf{s}$ to a point on $\partial\mathcal{B}$; following this approach the constraint in (6) is straightforwardly verified and we no longer have to care about it. In this way, if $\mathbf{t}_1^*$ is a solution to (6), it will verify $g \circ f(\mathbf{y} + \mathbf{t}_1^*) = \eta$, so $\mathbf{t}_1^* \in \partial\mathcal{B}$ and we can define the set of vectors $\mathcal{S}_1 \triangleq \{\mathbf{s}_1^* \in \mathbb{R}^n : h_{\mathbf{y}}(\mathbf{s}_1^*) = \mathbf{t}_1^*\}$. Taking into account that $h_{\mathbf{y}}$ is a surjection there will be at least one such vector $\mathbf{s}_1^* \in \mathcal{S}_1$, so $d^{\star}_{\mathbf{y}}(h_{\mathbf{y}}(\mathbf{s}_1^*)) = d_{\mathbf{y}}(\mathbf{t}_1^*)$, and $\mathbf{s}_1^*$ is a solution to (15). On the other hand, if $\mathbf{s}_2^*$ is a solution to (15), we can define $\mathbf{t}_2^* = h_{\mathbf{y}}(\mathbf{s}_2^*)$, which minimizes $d^{\star}_{\mathbf{y}}(\mathbf{t})$ over $\partial\mathcal{B}$, so $\mathbf{t}_2^*$ also minimizes $d_{\mathbf{y}}(\mathbf{t})$ for all $\mathbf{t} \in \partial\mathcal{B}$, and is a solution to (6).

Therefore, a vector $\mathbf{s}$ is a solution to (15) if and only if $h_{\mathbf{y}}(\mathbf{s})$ is a solution to (6), in such a way that we can restrict our problem to look for a function $h_{\mathbf{y}}$ and an algorithm which finds a solution to (15).

References

[1] André Adelsbach and Ahmad-Reza Sadeghi. Zero-knowledge watermark detection and proof of ownership. In Ira S. Moskowitz, editor, *4th International Workshop on Information Hiding, IH'01*, volume 2137 of *Lecture Notes in Computer Science*, pages 273–288, Pittsburgh, PA, USA, April 2001. Springer Verlag.

[2] Dimitri P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1995.

[3] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6:1673–1687, December 1997.

[4] I. J. Cox and J. P. M. G. Linnartz. Public watermarks and resistance to tampering. In *IEEE International Conference on Image Processing*

[9] This means that for all $\mathbf{b} \in \partial\mathcal{B}$, there is an $\mathbf{a} \in \mathbb{R}^n$ such that $h_{\mathbf{y}}(\mathbf{a}) = \mathbf{b}$.

ICIP'97, volume 3, pages 3–6, Santa Barbara, California, USA, October 1997.

[5] I. J. Cox, M. L. Miller, and J. A. Bloom. *Digital watermarking*. Multimedia Information and Systems. Morgan Kauffman, 2002.

[6] J. J. Eggers and B. Girod. Blind watermarking applied to image authentication. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, pages 1977–1980, May 2001.

[7] M. El Choubassi and P. Moulin. New sensitivity analysis attack. In Edward J. Delp III and Ping W. Wong, editors, *Security, Steganography and Watermarking of Multimedia contents VII*, pages 734–745. SPIE, January 2005.

[8] T. Furon, B. Macq, N. Hurley, and G. Silvestre. JANIS: Just Another N-order side-Informed watermarking Scheme. In *IEEE International Conference on Image Processing, ICIP'02*, volume 3, pages 153–156, Rochester, NY, USA, September 2002.

[9] T. Furon, I. Venturini, and P. Duhamel. An unified approach of asymmetric watermarking schemes. In Edward J. Delp III and Ping W. Wong, editors, *Security and Watermarking of Multimedia Contents III*, volume 4314, pages 269–279, San Jose, California, USA, January 2001. SPIE.

[10] Juan R. Hernández, Martín Amado, and Fernando Pérez-González. DCT-domain watermarking techniques for still images: Detector performance analysis and a new structure. *IEEE Trans. on Image Processing*, 9(1):55–68, January 2000. Special Issue on Image and Video Processing for Digital Libraries.

[11] T. Kalker, J. P. Linnartz, and M. van Dijk. Watermark estimation through detector analysis. In *IEEE International Conference on Image Processing, ICIP'98*, pages 425–429, Chicago, IL, USA, October 1998.

[12] J. P. M. G. Linnartz and M. van Dijk. Analysis of the sensitivity attack against electronic watermarks in images. In D. Aucsmith, editor, *2nd International Workshop on Information Hiding, IH'98*, volume 1525 of *Lecture Notes in Computer Science*, pages 258–272, Portland, OR, USA, April 1998. Springer Verlag.

[13] T. Liu and P. Moulin. Error exponents for one-bit watermarking. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, pages 65–68, 6-8 April 2003.

[14] M. F. Mansour and A. H. Tewfik. LMS-based attack on watermark public detectors. In *IEEE International Conference on Image Processing, ICIP'02*, volume 3, pages 649–652, September 2002.

[15] Matt L. Miller, Gwenaël J. Doërr, and Ingemar J. Cox. Applying informed coding and embedding to design a robust high-capacity watermark. *IEEE Transactions on Image Processing*, 13(6):792–807, June 2004.

[16] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 1999.

[17] Luis Pérez-Freire, Pedro Comesaña, and Fernando Pérez-González. Detection in quantization-based watermarking: performance and security issues. In Edward J. Delp III and Ping W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 721–733, San Jose, California, USA, January 2005. SPIE.

[18] Ilaria Venturini. Oracle attacks and covert channels. In Mauro Barni, Ingemar Cox, Ton Kalker, and Hyoung Joong Kim, editors, *Fourth International Workshop on Digital Watermarking*, volume 3710 of *Lecture Notes in Computer Science*, Siena, Italy, September 2005. Springer.