The Blind Newton Sensitivity Attack*

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

Signal Theory and Communications Department, University of Vigo, Vigo 36310, Spain.

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 tamper other watermarking algorithms, 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, thus being suitable for attacking most known watermarking methods, both for tampering watermarked signals and obtaining forgeries. The soundness of this new approach is tested by empirical results.

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 a digital media contains a given watermark or not, so they can be considered as a binary hypothesis testing problem, termed *watermark detection*.† In a general setup, the watermarking 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.,

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

where $f_{\mathbf{Z}|H_i}(\mathbf{z}|H_i)$ is the pdf of $\mathbf{Z}$ conditioned to 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

*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.

Further author information: (Corresponding author: P.C.)
P.C.: E-mail: pcomesan@gts.tsc.uvigo.es, Telephone: +34 986 812683
L.P-F.: E-mail: lpfreire@gts.tsc.uvigo.es, Telephone: +34 986 812683
F.P-G.: E-mail: fperez@gts.tsc.uvigo.es, Telephone: +34 986 812124

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

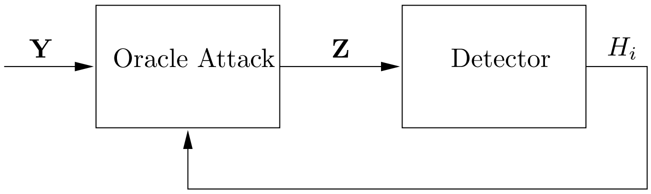


Figure 1. Block diagram of oracle attacks.

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

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

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 $l(\mathbf{z})$.

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 raise 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 is depicted in Fig. 1.

The most popular oracle attack is the so-called *sensitivity attack*.[1] At the time this attack was proposed, *additive spread spectrum* methods[2] 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:

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

Essentially, the sensitivity attack[1] (specialized to the case of digital images) for this kind of detectors consists in 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 to be not 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 a public detector available, 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 it in order to obtain 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 practical scenarios is given in Section 4.

2. PREVIOUS WORK AND IMPROVEMENTS

Starting from its original formulation,[1] the sensitivity attack was extensively studied for detectors based on linear correlation,[3,4] i.e., those given by (2). Simultaneously, countermeasures were proposed[3] 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. Anyway, estimation of the watermark is still possible and the information leakage about the watermark provided by the detector outputs can be quantified in an information-theoretic sense, in such a way that the shape of the optimum randomization function that minimizes the information leakage can be derived.[3] 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.[4]

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*. In view of the security flaws presented by traditional spread spectrum methods under sensitivity-like attacks, researchers put their efforts in the design of *asymmetric* schemes.[5] One of the advantages offered by asymmetric 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; four of those asymmetric methods were analyzed under a unified framework,[5] showing that their detection function can be written in terms of a quadratic form in $\mathbb{R}^n$, i.e.

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

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,[6] 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.[‡] Indeed, this makes more difficult the sensitivity attack, but obviously this is not the ultimate solution: for example, a N-th order detection boundary can still be described by estimating n^N points on such boundary. A possible solution to this problem was proposed by means of non-parametric decision boundaries,[7] i.e., by using boundaries that can not be described by a finite number of parameters. An example of such decision boundaries are those given by

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

fractal curves like the Peano curve, which is used to replace the original linear detection boundary in a spread spectrum scheme.[7] With a proper design, the proposed method was said to be able to invalidate sensitivity attacks with slight degradations in robustness.

Recently, a rigorous formulation of the sensitivity attack was presented[8]: first, the convergence of one of the previously proposed sensitivity attack algorithms[4] is proven, using the law of large numbers; thereafter, a new non-iterative sensitivity attack for detectors based on linear correlation is presented. This new algorithm is also suitable for estimating continuous-valued watermarks, whereas former approaches[3, 4] assumed that the watermark could only take discrete values. This algorithm completely discloses the decision boundary by computing n points on it, based on the fact that the decision boundary is a hyperplane. Another remarkable contribution of this work[8] is the extension of the sensitivity attack 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 any 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 $\boldsymbol{\theta}$.[§] 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 $g : \mathbb{R}^m \to \mathbb{R}$.

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

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

[§]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.

[¶]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.

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 belongs to the set of attacking vectors yielding H_1 (which we will denote by $\mathcal{B}$),[‖] i.e., $\mathbf{t}_0 \in \mathcal{B} \triangleq \{\mathbf{t} : g \circ f(\mathbf{t} + \mathbf{y}) > \eta\}$. Then, denoting by $\partial\mathcal{B}$ the boundary of $\mathcal{B}$ 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 tantamount 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 function depends on the secret key, which is unknown to the attacker. Actually, he/she will have only access to the binary output of the detector. In the Appendix 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 the boundary of $\mathcal{B}$, and $h_{\mathbf{y}}$ is a surjection which maps $\mathbb{R}^n$ onto the boundary of the decision region.

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 should 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 also needs the convexity of $\mathcal{B}^c$;[9] nevertheless, the experimental results in Section 4 will show that this suboptimal solution is usually satisfactory enough. Concerning the iterative method, in this paper we will use an adaptation of Newton's method,[10] where the updated vector in the $(k+1)$-th iteration is computed as

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

where $\xi_k \in \mathbb{R}^+$ is the step-length, whose computation requires (in general) a line search[9, 10]: a small value of ξ_k will imply a slow convergence, but with a large one convergence cannot be assured. When the boundary to be estimated is known to be an hyperplane, and the distortion measure is the squared Euclidean norm, we can adopt $\xi_k = 1$. This observation is based on the fact that the projecting function $h_{\mathbf{y}}$ is almost the identity and the squared Euclidean norm is a second order polynomial, so it can be completely characterized by its gradient and Hessian; therefore, Eq. (8) with $\xi_k = 1$ will give the value of $\mathbf{s}$ which achieves the global minimum in just one iteration.

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

$$\begin{aligned}
\frac{\partial(d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})}{\partial s_i}(\mathbf{s}) &= \frac{(d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s} + \delta\mathbf{e}_i) - (d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s})}{\delta} + O(\delta), \text{ and} \qquad (9)\\
\frac{\partial^2(d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})}{\partial s_i \partial s_j}(\mathbf{s}) &= \frac{(d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s} + \delta\mathbf{e}_i + \delta\mathbf{e}_j) - (d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s} + \delta\mathbf{e}_i)}{\delta^2}\\
&+ \frac{-(d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s} + \delta\mathbf{e}_j) + (d^{\star}_{\mathbf{y}} \circ h_{\mathbf{y}})(\mathbf{s})}{\delta^2} + O(\delta), \qquad (10)
\end{aligned}$$

‖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.

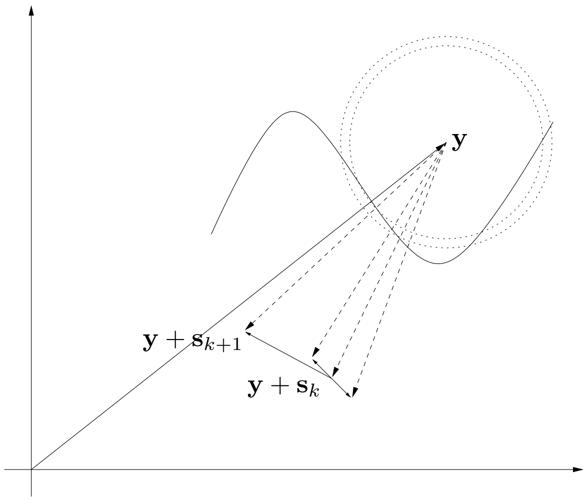


Figure 2. Example of an iteration of the algorithm. Given a watermarked signal $\mathbf{y}$ and the attacking vector in the k-th iteration $\mathbf{s}_k$, the last one is slightly modified to estimate the gradient and Hessian of $d^\star_{\mathbf{y}} \circ h_{\mathbf{y}}(\mathbf{s}_k)$. Once the descent 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. $||\alpha_{k+1}\mathbf{s}_{k+1}||^2 < ||\alpha_k\mathbf{s}_k||^2$.

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). Just 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, Eq. (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.

On the other hand, $h_{\mathbf{y}}(\mathbf{s})$ is usually based on scaling $\mathbf{s}$ by a factor $\alpha \in \mathbb{R}$, such that $\mathbf{t} = \alpha \cdot \mathbf{s} \in \partial\mathcal{B}$. The existence of such an α is based on the fact that for most of the known detection functions $\mathbf{0} \in \mathcal{B}$ and $\beta \cdot \mathbf{s} \in \mathcal{B}^c$, for large values of β, so α can be found by a dicothomy algorithm. Be aware that this method is based on the binary output of the detector, without any other knowledge about the detection function; this is why the algorithm is said to be *blind*. An example of an iteration of the algorithm was plotted in Fig. 2, where the reduction from $||\alpha_k\mathbf{s}_k||^2$ to $||\alpha_{k+1}\mathbf{s}_{k+1}||^2$ was symbolized by the decrease in the radius of the two spheres centered at $\mathbf{y}$.

3.1. Computing forgeries

Let us suppose now that the attacker has access to a watermarked signal $\mathbf{y}$ and some 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

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

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 way similar 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 tampering 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},** which typically is really small due to design criteria, the projecting function $h_{\mathbf{Y}}(\cdot)$ will be still a function of $\mathbf{y}$, since a point in the detection region is usually needed in order to compute a point on the detection boundary (one can think of the $h_{\mathbf{Y}}(\cdot)$ proposed in Section 3).

Finally, one must be aware that the decision regions could depend on the statistics of the input signal, as it is the case of the ML detector for a Generalized Gaussian distributed host; this dependency has a limited impact when the initial signal is not substantially modified, e.g. when the attacker tries to unwatermark a signal, but it could dramatically change the decision regions when the attacker tries to *move* from one image to another, possibly frustrating an attack.

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 methods.

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 is given by

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

in such a way that the boundary of the decision region ($\partial\mathcal{B}$) will be a hyperplane and just one iteration of BNSA will be needed to estimate its orthogonal vector, i.e. the projecting vector, which is a scaled version of the watermark itself.

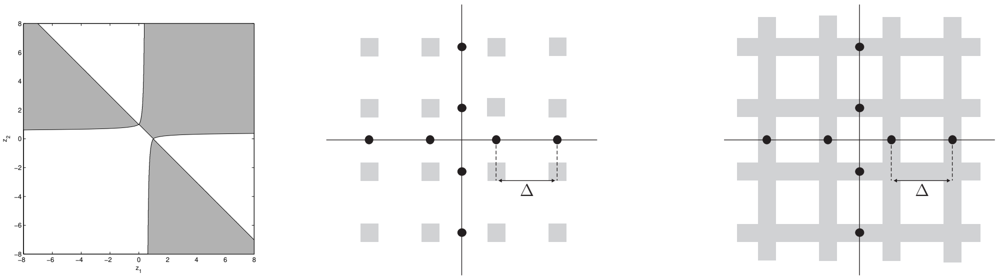
Another alternative for the detection function is that proposed by Cox *et al.*[11]; in that case, f quantifies the angle between the received signal $\mathbf{z}$ and the watermark vector $\mathbf{w}$, i.e. $f(\mathbf{z}) = \frac{\mathbf{z}^T \cdot \mathbf{w}}{||\mathbf{z}|| \cdot ||\mathbf{w}||}$, and g is again the identity function, yielding a decision region $\mathcal{B}$ which is a n-dimensional cone.

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[12] fulfills this requirement whenever the shape parameter c_k is such that $c_k < 1$. That 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).

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



(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.

4.2. Side-informed methods

In Section 2 JANIS methods were introduced. In order to make a comparison with other existing methods, we have fixed the order of the detection function to $N = 4$, 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 due to Eggers *et al.*,[13] who adapted the Scalar Costa Scheme to authentication purposes by embedding a fixed message, yielding the detection function $g(\mathbf{z}, \boldsymbol{\theta}) = \frac{f_{\mathbf{Y}}(\mathbf{z})}{f_{\mathbf{X}}(\mathbf{z})}$. Note that in this case the sensitivity attack is straightforward, since it can be done component-wise.

On the other hand, other works are available[14] where the received signal $\mathbf{z}$ is quantized with a lattice Λ and the decision is made upon the squared norm of the quantization error. In a formal way, 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),[15] where uniform scalar quantizers are used to quantify an 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

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

where Δ is the quantization step, a_{ij} are the secret projection matrix coefficients and m is the dimensionality of the projected subspace. The obtained decision regions were respectively plotted in Fig. 3(b) and Fig. 3(c) for $m = 2$. Obviously, the optimal attacking strategy will depend on the chosen decision region. The convergence of the algorithm introduced in Section 3 for finding the optimal attacking vector will be very much slower for

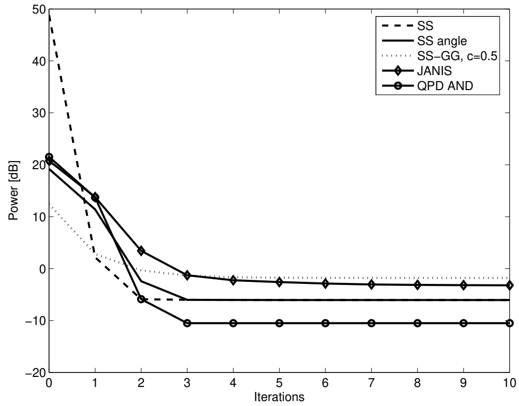


Figure 4. Power needed to yield an unwatermarked signal (compared with the embedding power and expressed in dB) averaged over 100 watermarked Gaussian vectors as a function of the number of iterations (0 corresponds to random attacking vectors), for different decision regions: SS based on a hyperplane, SS based on the angle, SS ML detector for Generalized Gaussian distributed hosts ($c = 0.5$), JANIS and QPD for AND regions.

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

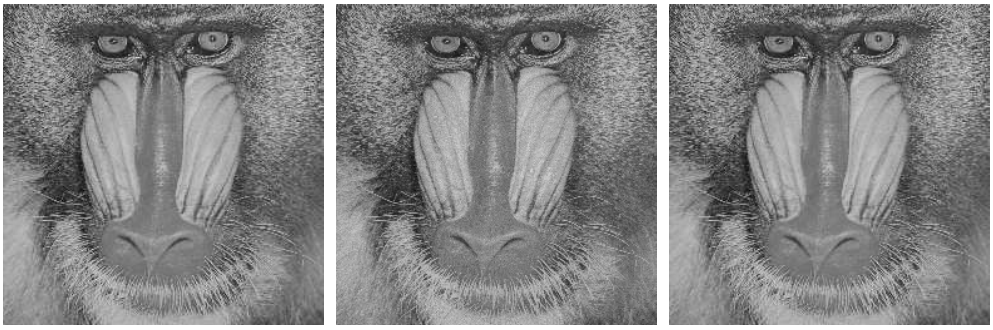
As a first step, we will compare the performance of the different methods under the BNSA when this method uses the diagonal estimation of the Hessian for removing the watermark of randomly generated host signals. In order to make a fair comparison, the value of the probability of false alarm P_{fa} will be 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 the studied methods.** The host signals were randomly generated following a Gaussian distribution, except when the ML detector for the Generalized Gaussian host is used, in which case we have chosen the shape parameter to be 0.5. In Fig. 4 the ratio between the power needed to achieve an unwatermarked signal and the embedding power is plotted versus the number of iterations of BNSA; we can see that the power needed at iteration 0 (just randomly generated attacking vectors) is much larger for hyperplane-based SS, but converges to that of angle-based SS when the number of iterations is increased. In the same way, the most robust method against the BNSA among those plotted in the figure is the ML detector for SS and a Generalized Gaussian distributed host ($c_k = 0.5$), 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 the this result is the -3.21 dB needed by JANIS, for which the power required for producing 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, the power needed to yield an unwatermarked signal is significantly smaller than in the other studied cases. Finally, for QPD-OR also with $m = 10$, the power required after 10 iterations is only -38 dB.

Experiments with real images were also performed: image *Lena* 256×256 (i.e., $n = 65536$) was watermarked using the above described methods with an embedding Peak Signal to Noise Ratio †† of 38.58 dB (with $\sigma_W^2 = 9$) and the probability of false alarm P_{fa} was fixed to 10^{-4}; for the creation of forgeries, the original unwatermarked signal was chosen to be *Baboon*. Trying to reduce the computational cost, the updating algorithm described in (11) was implemented. BNSA unwatermarked *Lena* achieving a PSNR with respect to the watermarked image of 40.25 dB in just one iteration both for hyperplane-based SS and angle-based SS, being the PSNRs with respect to the original image of 53.75 and 53.77 dB, respectively; these results show again the similar

††The PSNR between two 8-bit signals $\mathbf{x}_1$ and $\mathbf{x}_2$ is defined as $\mathrm{PSNR} \triangleq 10 \log_{10}(n \cdot (255)^2/||\mathbf{x}_1 - \mathbf{x}_2||^2)$.

(a) (b) (c)

Figure 5. (a)*Lena* watermarked by JANIS. (b) *Lena* watermarked by JANIS and attacked by BNSA. (c) *Lena* watermarked by JANIS and attacked by AWGN.



(a) (b) (c)

Figure 6. (a) Original *Baboon* 256×256. (b) Forgery of *Baboon* for the ML detector of Generalized Gaussian distributed hosts. (c)Forgery of *Baboon* for JANIS.

behavior of both methods. On the other hand, PSNRs of 38.31 dB with respect to the watermarked image and 58.48 dB with respect to the original one were needed to unwatermark *Lena* after one BNSA iteration when JANIS is used. These results are particularly promising, since the PSNRs with respect to the original image are larger for the image obtained by BNSA than for the watermarked one, meaning that BNSA is not just yielding an unwatermarked image, but it is also properly estimating and subtracting the watermark, as the embedding distortion is reduced. This is reflected in Figs. 5(a) and (b), where the signal watermarked with JANIS, and the result of applying BNSA can be compared. In order to provide an intuitive idea of the robustness of the system against other kind of attacks, in Fig. 5(c) one can see the result of applying to the watermarked signal Additive White Gaussian Noise (AWGN) with the power necessary to yield a *no watermark* decision. Strikingly, an unwatermarked version of *Lena* was obtained for a well-known dirty paper trellis-based watermarking scheme[16] with a PSNR of 52.17, also after one BNSA iteration, when a spreading factor of 4, 16 states and 8 arcs per state are used; needless to say the reported results could vary depending on the chosen parameters.

On the other hand, BNSA creates forgeries with high PSNRs, where the noise is the distortion introduced in the unwatermarked original *Baboon* (see Fig. 6(a)). The following results were obtained after 10 BNSA iterations. For example, for hyperplane-based SS a PSNR of 55.73 dB is necessary, similarly to the 55.23 dB needed for forging an angle-based SS watermarked version of *Baboon*. As it was previously discussed, the ML decision regions for a Generalized Gaussian distributed host will be modified when the statistics of the input image change; despite of this incovenient, BNSA gets a watermarked version of *Baboon* with 32.03 dB of PSNR (see Fig. 6(b)). Finally, the forged version of *Baboon* for JANIS plotted in Fig. 6(c) was produced by BNSA with a PSNR of 50.69 dB.

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:

- The power needed to push a watermarked signal out of the detection region after the BNSA can be seen as a measure of the robustness of a watermarking method against this attack: the larger the power needed, the more robust the method is. In this sense, at sight of Fig. 4 the ML detector for Generalized Gaussian distributed host could be said to be the most robust among the studied methods, whereas the QPD methods show quite poor performance. Note, however, 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.[15]
- 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,[17] where regardless of the domain in which 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[7] are also susceptible for being attacked by BNSA. Although the decision boundaries are non-parametric, the attacker could try to estimate its envelope, since this estimation will be usually enough to remove the watermark or creating 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. APPENDIX

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

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

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.,

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

and $h_{\mathbf{y}}(\mathbf{s})$ is a surjection from $\mathbb{R}^n$ to $\partial\mathcal{B}$,[‡‡] 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 $\mathbf{t}_2^* = h_{\mathbf{y}}(s_2^*)$, 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 looking for a function $h_{\mathbf{y}}$ and an algorithm which finds a solution to (15).

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

ACKNOWLEDGMENTS

Thanks are due to Prof. I. J. Cox and Dr. G. Doërr who kindly provided us with the source code of their algorithm.[16]

REFERENCES

1. I. J. Cox and J. P. M. G. Linnartz, "Public watermarks and resistance to tampering," in *IEEE International Conference on Image Processing ICIP'97*, **3**, pp. 3–6, (Santa Barbara, California, USA), October 1997.
2. I. J. Cox, J. Killian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing* **6**, pp. 1673–1687, December 1997.
3. J. P. M. G. Linnartz and M. van Dijk, "Analysis of the sensitivity attack against electronic watermarks in images," in *2nd International Workshop on Information Hiding, IH'98*, D. Aucsmith, ed., *Lecture Notes in Computer Science* **1525**, pp. 258–272, Springer Verlag, (Portland, OR, USA), April 1998.
4. T. Kalker, J. P. Linnartz, and M. van Dijk, "Watermark estimation through detector analysis," in *IEEE International Conference on Image Processing, ICIP'98*, pp. 425–429, (Chicago, IL, USA), October 1998.
5. T. Furon, I. Venturini, and P. Duhamel, "An unified approach of asymmetric watermarking schemes," in *Security and Watermarking of Multimedia Contents III*, Edward J. Delp III and P. W. Wong, eds., **4314**, pp. 269–279, SPIE, (San Jose, California, USA), January 2001.
6. 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*, **3**, pp. 153–156, (Rochester, NY, USA), September 2002.
7. M. F. Mansour and A. H. Tewfik, "LMS-based attack on watermark public detectors," in *IEEE International Conference on Image Processing, ICIP'02*, **3**, pp. 649–652, September 2002.
8. M. El Choubassi and P. Moulin, "New sensitivity analysis attack," in *Security, Steganography and Watermarking of Multimedia contents VII*, Edward J. Delp III and P. W. Wong, eds., pp. 734–745, SPIE, January 2005.
9. D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.
10. J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 1999.
11. I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital watermarking*, Multimedia Information and Systems, Morgan Kauffman, 2002.
12. J. R. Hernández, M. Amado, and F. Pérez-González, "DCT-domain watermarking techniques for still images: Detector performance analysis and a new structure," *IEEE Trans. on ImageProcessing* **9**, pp. 55–68, January 2000. Special Issue on Image and Video Processing for Digital Libraries.
13. J. J. Eggers and B. Girod, "Blind watermarking applied to image authentication," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, **3**, pp. 1977–1980, May 2001.
14. T. Liu and P. Moulin, "Error exponents for one-bit watermarking," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, **3**, pp. 65–68, 6-8 April 2003.
15. L. Pérez-Freire, P. Comesaña, and F. Pérez-González, "Detection in quantization-based watermarking: performance and security issues," in *Security, Steganography, and Watermarking of Multimedia Contents VII*, Edward J. Delp III and P. W. Wong, eds., **5681**, pp. 721–733, SPIE, (San Jose, California, USA), January 2005.
16. M. L. Miller, G. J. Doërr, and I. J. Cox, "Applying informed coding and embedding to design a robust high-capacity watermark," *IEEE Transactions on Image Processing* **13**, pp. 792–807, June 2004.
17. A. Adelsbach and A.-R. Sadeghi, "Zero-knowledge watermark detection and proof of ownership," in *4th International Workshop on Information Hiding, IH'01*, I. S. Moskowitz, ed., *Lecture Notes in Computer Science* **2137**, pp. 273–288, Springer Verlag, (Pittsburgh, PA, USA), April 2001.