ESTIMATING THE PROBABILITY OF FALSE ALARM FOR A ZERO-BIT WATERMARKING TECHNIQUE

T. Furon

Thomson Security Lab
Avenue Belle-Fontaine, Cesson-Sévigné

C. Jégourel, A. Guyader*, F. Cérou[†]

INRIA Rennes Bretagne Atlantique, ASPI
Campus de Beaulieu, Rennes

ABSTRACT

Assessing that a probability of false alarm is below a given significance level is a crucial issue in watermarking. We propose an iterative and self-adapting algorithm which estimates very low probabilities of error. Some experimental investigations validates its performance for a rare detection scenario where there exists a close form formula of the probability of false alarm. Our algorithm appears to be much quicker and more accurate than a classical Monte Carlo estimator. It even allows the experimental measurement of error exponents.

Index Terms— Watermarking, False alarm, Rare event analysis.

1. INTRODUCTION

Watermarking establishes a durable link between a piece of digital content and some meta-data by embedding the latter deeply in the former. In order to be useful, especially for digital long-term preservation application, a watermarking technique must be reliable. We introduce here the concept of *reliability* as the guarantee that not only watermark decoding errors very rarely happen, but also that their frequency or their probability is assessed to be below a given level.

In this paper, we focus on zero-bit watermarking which hides the presence of a secret mark in host contents such as still images. No message is embedded at the encoding side. The blind detector just looks for the presence or the absence of the mark in order to state whether the observed content is protected. This particular subclass of watermarking is used for instance in copy-protection application.

A big issue concerning watermark detection is to evaluate the probability of false alarm P_{fa}, *i.e.* how often the detector deems as watermarked a content which has not been. This figure of merit appears at the top on the list of requirements. The good thing with this feature is that one doesn't have to know how to watermark a content in order to estimate this probability. For instance, even if the embedder of a zero-bit

watermarking technique is not yet finalized, one can still estimate the probability of false alarm whenever the parameters at the detection side are fixed.

The bad thing about this feature is that its value is typically very low especially for digital preservation scenarios, usually never bigger than 10^{-6}. Experimental assessments then need millions of images and last very long, slowing the fine-tuning of the watermark detector. Technology providers claiming any result concerning this feature, actually have either very high probabilities (which are measurable with accuracy), either low probabilities with strongly inaccurate measures.

A last problem is that everything is a matter of trade-off in watermarking: If a technique is not ranked first in a robustness benchmark, its designers can complain and pretend this is because its probability of false alarm is lower than for the other competitors. Nobody can verify this statement if their order of magnitude is lower than 10^{-6}. This explains why benchmarking watermarking techniques is so difficult, and why efforts towards watermarking standardization have always failed until now.

This issue gave birth to a collaboration between a team of statisticians experts in rare event analysis and watermarkers. This paper presents a general framework for experimentally assessing the probability of false alarm of a wide class of watermarking techniques. For illustration purpose, we apply it to the well known normalized correlation watermark detector. Sec. 2 presents our main assumptions and typical estimations so far used by the watermarking community. Sec. 3 presents the main algorithm and its use on synthetic data. Sec. 4 validates its correctness and stresses its excellent performance. The last section tackles the experimental measurement of error exponents, which is, as far as we know, a first time in the watermarking literature.

2. PROBLEM STATEMENT

2.1. Assumptions

Our assumptions focus on the structure of the watermark detector. This algorithm takes as inputs an image $\mathcal{I}$ and a secret key K and yields a binary decision: $D(\mathcal{I}, K) = 1$ if the im-

Fig. 1. Structure of a watermark detector.

age is deemed as watermarked, 0 else. Define $\mathcal{H}_0$ the assumption that the image $\mathcal{I}$ is not watermarked. The probability of false alarm is defined by $P_{fa} = \text{Prob}[D(\mathcal{I}, K) = 1|\mathcal{H}_0]$.

We assume the detection consists in three steps (Fig. 1):

- L real features are extracted from the input image, and stored in a vector $\mathbf{x} \in \mathbb{R}^L$,

- From this vector, a score $s(\mathbf{x}) \in \mathbb{R}$ is calculated. It represents the likelihood that the input image is indeed watermarked: the greater the score, the more confident the detector is to yield a positive decision,

- The final decision is the comparison of this score with a threshold τ: $D(\mathcal{I}, K) = \mathbb{1}_{(s(\mathbf{x}) > \tau)}$.

The secret key serves during the feature extraction and/or the likelihood calculation. We assume that τ is the only parameter in the detector tackling the probability of false alarm. Hence, for a given extraction procedure and a given score function, the issue is to know the map: $P_{fa} = f(\tau)$. These assumptions cover many watermark detectors, because this structure is indeed advised by the detection theory, and especially the Neyman-Pearson theorem [1].

2.2. Prior Art

The problem is easily solved when the probability density function p_S of the score is known under hypothesis $\mathcal{H}_0$. The map is then just the integration of the tail of the pdf: $P_{fa} = \int_\tau^{+\infty} p_S(s)ds$. However, this first choice is almost never possible: a simple statistical model doesn't capture the reality, mathematical derivations are too cumbersome with a complex model. The score often writes as a sum of many and more or less independent random extracted features. This explains the abusive resort to the Central Limit Theorem to evaluate P_{fa} in literature. However, the convergence rate to the Gaussian law is very crucial and depends on the third moment of the extracted features (in the most simple case) as stated by the Berry-Esséen bound [2]. Roughly speaking, a small probability of error amounts to integrate the tail of the pdf, where the CLT approximation by a Gaussian law is indeed very bad.

A better way is to establish upper bounds (*e.g.* Chernoff's bound, union bound). The tightness of the bound, which is usually good only over a small range of parameter values, is then an issue. Numerical approximations of the probability formula also exist like the Beaulieu and the DFT methods used when the score is the sum of i.i.d. random variables [3].

When these approaches are not possible, then the last choice is the experimental estimation. However, many watermarking articles were only running the Monte Carlo (MC)

method, which is very inefficient for a low P_{fa}. This naive approach consists in running n experiments and to count the number of times k that the detector failed. Then, P_{fa} is estimated by the error frequency: $\hat{P}_{fa} = k/n$. This estimator is unbiased ($E[\hat{P}_{fa}] = P_{fa}$) and its variance, $\text{Var}[\hat{P}_{fa}] = P_{fa}(1 - P_{fa})/n$, asymptotically goes to zero. However, one needs around P_{fa}^{-1} experiments to make it work (*i.e.* $k \neq 0$), and even worse, its relative standard deviation is given by $\sqrt{\text{Var}[\hat{P}_{fa}]}/E[\hat{P}_{fa}] \approx 1/\sqrt{P_{fa}n}$. For a decent accuracy, n must be several times bigger than P_{fa}^{-1}: the smaller the probability, the harder its estimation.

3. OUR ALGORITHM

Our algorithm pertains to the field of rare event analysis under static distribution. We present it when a relevant statistical model of $\mathbf{x}$ is available.

3.1. Key idea

The key idea is to factorize a probability into a product of bigger probabilities. Let $A_N = A$ be the rare event, and A_{N-1} a related event such that when A_N occurs, A_{N-1} has also occured. However, when A_{N-1} occurs, it doesn't imply that A_N is true. Hence, A_{N-1} is less rare an event than A_N. This justifies the first equality in the following equation, the second one being just the Bayes rule:

$$\begin{aligned} \text{Prob}[A_N] &= \text{Prob}[A_N, A_{N-1}] \\ &= \text{Prob}[A_N|A_{N-1}].\text{Prob}[A_{N-1}]. \quad (1) \end{aligned}$$

Repeating the process, we finally obtain:

$$\begin{aligned} \text{Prob}[A_N] &= \text{Prob}[A_N|A_{N-1}]\text{Prob}[A_{N-1}|A_{N-2}] \\ &\quad \dots \text{Prob}[A_2|A_1]\text{Prob}[A_1] \quad (2) \end{aligned}$$

provided that $\{A_j\}_{j=1}^N$ is a sequence of nested events. Knowing that estimation of a probability is easier when its value is bigger, we have succeeded in decomposing a hard problem into N much easier problems. In our case, the rare event A_N occurs when $\mathbf{x} \in \mathcal{A}_N$. A sequence of nested events translates then in a sequence of subsets $\mathcal{A}_N \subset \mathcal{A}_{N-1} \dots \subset \mathcal{A}_1$. The indicator function of these sets is as follows: $\mathbf{x} \in \mathcal{A}_j$ if $s(\mathbf{x}) > \tau_j$. Nested events are created for a sequence of increasing thresholds: $\tau_1 < \tau_2 < \dots < \tau_N = \tau$.

The algorithm estimates $\text{Prob}[s(\mathbf{x}) > \tau_1]$ as $\hat{p}_1$, and the $N-1$ conditional probabilities $\text{Prob}[s(\mathbf{x}) > \tau_j|s(\mathbf{x}) > \tau_{j-1}]$ as $\hat{p}_j$ for $2 \leq j \leq N$. It returns $\hat{P}_{fa} = \hat{p}_1\Pi_{j=2}^N\hat{p}_j$. The difficulty is now to give the appropriate values to the thresholds $\{\tau_i\}_1^{N-1}$. The probabilities to be estimated must not be very weak in order to maintain a reasonable complexity. Moreover, it can be shown that the variance of $\hat{P}_{fa}$ is minimized when the probabilities $\{p_i\}_i^N$ are equal [4]. However, to set the correct value of the thresholds, we would need the map

$\tau = F^{-1}(p)$ which we have not. Otherwise, we would already know what the value of $P_{fa} = F(\tau)$ is. The idea is to set them adaptively.

3.2. Description of the adaptive levels estimator

3.2.1. Requirements

Our algorithm needs two random processes. The GENERATE process creates random vectors statistically independent and distributed as $p_\mathbf{X}$. The MODIFY process has two inputs: a random vector $\mathbf{x}$ and the strength $\mu \in \mathbb{R}^+$. It randomly modifies $\mathbf{x}$ to create a vector output $\mathbf{y}$, such that:

- $\partial \mathbb{E}[d(\mathbf{x}, \mathbf{y})]/\partial \mu > 0$, for a given distance $d(.,.)$ in $\mathbb{R}^L$.

- $p_\mathbf{Y} = p_\mathbf{X}$. The MODIFY process lets the pdf invariant.

3.2.2. Initialization

Our algorithm starts by estimating $p_1 = \text{Prob}[s(\mathbf{x}) > \tau_1]$ with a classical MC approach. GENERATE creates n vectors $\{\mathbf{x}_i\}_{i=1}^n$, and their scores $s(\mathbf{x}_i)$ are stored in a vector $\mathbf{sx}$. Instead of returning $\hat{p}_1$ for a given threshold τ_1, we indeed act the other way around. We set $p_1 = p = k/n$ for an integer $k < n$, parameter of the algorithm, and we return $\hat{\tau}_1$ as the k-th biggest score.

3.2.3. Iteration

The j-th iteration starts by selecting good vectors: once the intermediate threshold $\hat{\tau}_j$ is set to the value of the k-th biggest score in $\{s(\mathbf{x}_i)\}_{i=1}^n$, the k vectors with the biggest scores are stored in a pool $\mathcal{P}_k$. These good vectors are then duplicated: a vector $\mathbf{w}$ is picked up at random in $\mathcal{P}_k$ and MODIFY transforms it into a vector $\mathbf{z}$. If its score is still bigger than $\hat{\tau}_j$, then the modification is successful and $\mathbf{z}$ enters the pool $\mathcal{P}_n$. Else, the modification is rejected and $\mathbf{w}$ enters in $\mathcal{P}_n$. This is repeated n times and the $(j+1)$-th iteration starts with a population of n vectors in $\mathcal{P}_n$.

3.2.4. Ending

The algorithm ends when the intermediate threshold is above τ. Suppose this happens at the N-th iteration: $\hat{\tau}_N > \tau$. The closing step counts the number k' of vectors in $\mathcal{P}_n$ whose scores are bigger than τ, and computes the estimation $\hat{P}_{fa} = (k/n)^{N-1} k'/n$. If this stopping condition hasn't been met, the algorithm stops after $N_{\max}$ iterations, and $\hat{P}_{fa} = 0$. Algorithm 1 summarizes the estimator in pseudo-code.

3.3. Properties

In expectation, the expected number of iterations is

$$E[N] = \lfloor \log P_{fa}^{-1} / \log p^{-1} \rfloor + 1, \qquad (3)$$

and the number of calls to the score function is Nn, proportional to $\log P_{fa}^{-1}$. Hence, our estimator is far less complex than a classical MC.

From [5], the method inherits the asymptotic properties of consistency and normality as proven in [6]. With equations:

$$\hat{P}_{fa} \xrightarrow[n \to +\infty]{a.s.} P_{fa}, \qquad (4)$$

$$\sqrt{n}(\hat{P}_{fa} - P_{fa}) \xrightarrow[n \to +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma^2), \qquad (5)$$

with

$$\sigma^2 \gtrsim P_{fa}^2 \left((N-1)\frac{n-k}{k} + \frac{n-k'}{k'} \right). \qquad (6)$$

We can also show that, in the asymptotic regime, the bias decreases inversely proportional with n:

$$E\left[\frac{\hat{P}_{fa} - P_{fa}}{P_{fa}} \right] = \frac{1}{n} \frac{N(n-k)}{k} + o(n^{-1}), \qquad (7)$$

which means that $E[(\hat{P}_{fa} - P_{fa})/P_{fa}] \gtrsim \alpha n^{-1}$, where α is always a positive number. A remarkable fact is that the bias is positive, so that estimations tend to over-estimate the probability of rare event. In concrete situations, the rare event often corresponds to a catastrophic scenario to be prevented, and over-estimating is then a nice property.

Algorithm 1: Estimation of $\text{Prob}[s(\mathbf{x}) > \tau]$

Data: τ, k, n, $N_{\max}$, statistical model $\mathbf{x} \sim p_\mathbf{X}$
begin
  for $i = 1$ *to* n **do**
    $\mathbf{x}_i = \text{GENERATE}(p_\mathbf{X})$; $sx_i = \text{SCORE}(\mathbf{x}_i)$;
  $N = 1$;
  $\hat{\tau}_N = \text{HIGHER\_SCORE}(\mathbf{sx}, k)$;
  while $\hat{\tau}_N < \tau$ *and* $N < N_{max}$ **do**
    $t = 1$;
    for $i = 1$ **to** n **do**
      if $sx_i \geq \hat{\tau}_N$ **then**
        $\mathbf{y}_t = \mathbf{x}_i$; $sy_t = sx_i$; $t = t+1$;
    $\Pi = \text{RAND\_PERM}(k)$;
    for $i = 1$ **to** n **do**
      $j = \Pi(\mod(i,k)+1)$;
      $\mathbf{z} = \text{MODIFY}(\mathbf{y}_j, \mu)$;
      if $SCORE(\mathbf{z}) > \hat{\tau}_N$ **then**
        $\mathbf{x}_i = \mathbf{z}$; $sx_i = \text{SCORE}(\mathbf{z})$;
      else
        $\mathbf{x}_i = \mathbf{y}_j$; $sx_i = sy_j$;
    $N = N+1$; $\hat{\tau}_N = \text{HIGHER\_SCORE}(\mathbf{sx}, k)$;
  $k' = 0$;
  for $i = 1$ **to** n **do if** $sx_i > \tau$ **then** $k' = k'+1$;
  return $\hat{P}_{fa} = \frac{k' k^{N-1}}{n^N}$;
end

4. EXPERIMENTAL INVESTIGATIONS

This section applies the algorithm with a normalized correlation scoring: $s(\mathbf{x}) = \mathbf{x}^T \mathbf{u}/\|\mathbf{x}\|$, with $\mathbf{u}$ a secret unitary vector. This function is widely used in the watermarking literature (*e.g.* in the watermark detector of the last international challenge BOWS-2 [7]). Moreover, its probability of false alarm has been widely studied when $p_{\mathbf{X}}$ is isotropic (*e.g.* a white Gaussian law): M. Miller and J. Bloom propose an algorithm based on solid angle numerical evaluation [8], P. Comesaña *et al.* [9] use asymptotic development. Strangely enough, nobody found that P_{fa} has indeed a simple closed form expression. With a change of basis and the definition of F-distribution [10], we have:

$$P_{fa} = 1 - I_{\tau^2}(1/2, (L-1)/2), \qquad (8)$$

where I is the regularized incomplete beta function. The GENERATE process is the Mersenne Twister pseudo-random generator coded in Matlab `randn` command so that $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_L)$. The MODIFY process is $\mathbf{y} = (\mathbf{x} + \mu\mathbf{n})/\sqrt{1 + \mu^2}$ with $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_L)$.

4.1. The role of the modification strength μ

The main shortcoming of our algorithm is that the parameter μ needs a manual fine-tuning. The algorithm as described above works fine for the problem studied in this section when $\mu = 0.2$. The strength of the modification fixes the dynamic of the system. There is a trade-off to be found between two undesirable effects. The goal of this subsection is to experimentally show and explain these two effects and to find a trick to circumvent this manual fine-tuning shortcoming. The others parameters are set as follows: $L = 20$, $\tau = 0.95$, $n = 6400$. This gives $P_{fa} = 4.704 * 10^{-11}$. A greater or a lower value than 0.2 have negative impacts as we shall see.

As the estimator goes, the set $\mathcal{A}_j$ is smaller and smaller, and the modification process is more and more likely to move vectors out of this set when the strength is too big. Let us define the filtering rate of the modification process as the ratio of rejected modification. Figure 2 shows this filtering rate along the iteration number. Typically, a factor μ greater than 0.5 (red curves) yields a filtering rate of 100% for the last iterations. This implies that the vectors in the stacks are not renewed any longer. Thus, threshold $\hat{\tau}_j$ saturates and the algorithm does not converge. It stops thanks to the constraint on the maximum number of iterations.

We seize the opportunity of this case study where the true map $P_{fa} = F(\tau)$ is known to plot the relative error along the ROC curve $(p^j - F(\hat{\tau}_j))/F(\hat{\tau}_j)$ in Figure 3. We observe that, when the filtering rate is too high, the relative error has a peak followed by an exponential decay towards -1. The peak is explained by the fact that the vectors and their scores are no longer renewed, so that the thresholds quickly converge towards the supremum of these scores. Once the thresholds

saturate to this supremum, $F(\hat{\tau}_j)$ became fixed, and the relative error has an exponential decay due to the term p^j. When this latter becomes negligible compared to $F(\hat{\tau}_j)$, the relative error tends to -1.
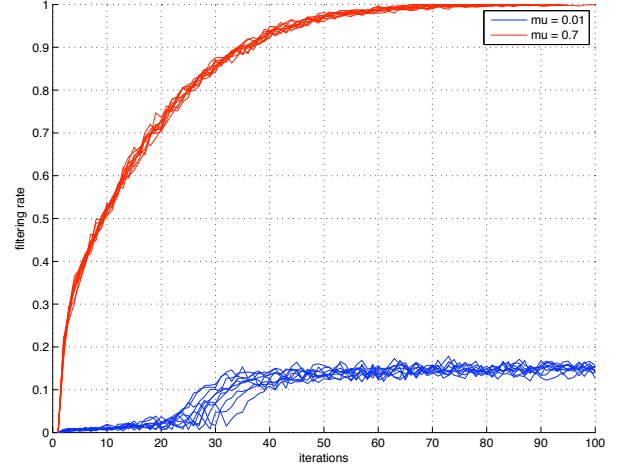

Fig. 2. Filtering rate for 10 estimator runs, $\mu \in \{0.7, 0.01\}$.

The impact of a small μ is not noticeable in the filtering rate which is far below the saturation phenomenon (see Fig. 2). Yet, Fig. 3 shows very strong relative errors (blue curves) in the first iterations. Factor μ is so weak that modified particles are almost located at the same place as the previous ones. This prevents us from exploring the space due to a low dynamic and from moving the vectors towards the acceptance region. Hence, the scores of the modified particles are almost the same scores than the previous ones. This is almost as if $\mu = 0$, *i.e.* classical Monte Carlo. The behavior of the relative error is then strongly dependent on the initialization process which yields the first stack of vectors. The selection process keeps a thiner and thiner portion p^j of this initial cloud of particles and the intermediate thresholds converge to the maximum of the initial scores. Once this is achieved, the intermediate thresholds saturate to this maximum value, and we again observe an exponential decay toward -1 (Fig. 3 - blue curves).

The best trade-off can be stated in the following terms: find the maximum value of μ such that the filtering rate is below a given level. We modify Alg. 1 as follows. μ is set to one at the beginning. For each iteration, we measure the filtering rate. If this latter is bigger than the level, we reduce the value of μ and repeat the iteration until the filtering rate is below the level. The value of μ is thus now found adaptively. However, the number of detection trials is no longer fixed. Experimentally, we decrease μ by a factor 1.1 anytime the filtering rate is above 0.7.
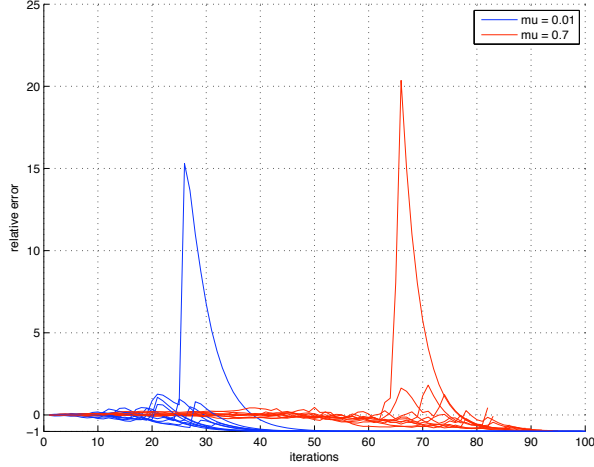
Fig. 3. Relative errors, same estimator runs as used in Fig. 2.

Fig. 5. Statistics over 1,000 estimation runs with $p = 3/4$.

4.2. The role of $p = k/n$

Parameter p strikes a trade-off between the speed and the accuracy of the estimator. (3) tells us that the lower p is, the faster is the estimation. However, (6) and (7) show that the relative variance and the bias are decreasing functions of p.

We keep the same experimental setup, and try two values for p (3/4 and 1/2) while increasing n. We run 1,000 estimations $\{\hat{P}_{fa}^{(i)}\}$ to measure the relative bias as $(\text{Mean}(\{\hat{P}_{fa}^{(i)}\}) - P_{fa})/P_{fa}$, the relative standard deviation $\text{Std}(\{\hat{P}_{fa}^{(i)}\})/P_{fa}$, and the relative maximum deviation $(\text{Max}(\{\hat{P}_{fa}^{(i)}\}) - P_{fa})/P_{fa}$. Figures 4 and 5 plot these values against the number of particles n.
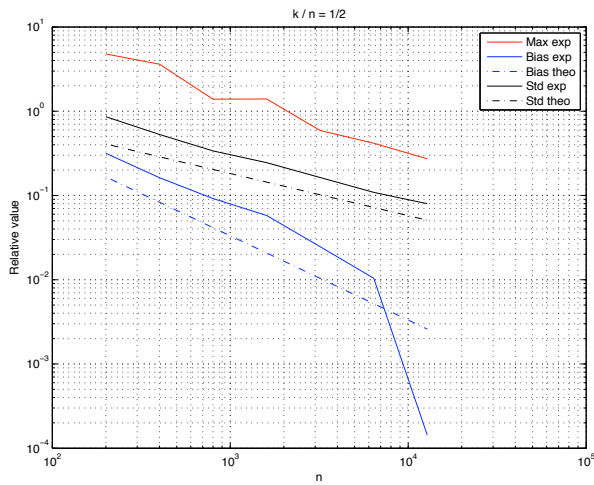

Fig. 4. Statistics over 1,000 estimation runs with $p = 1/2$.

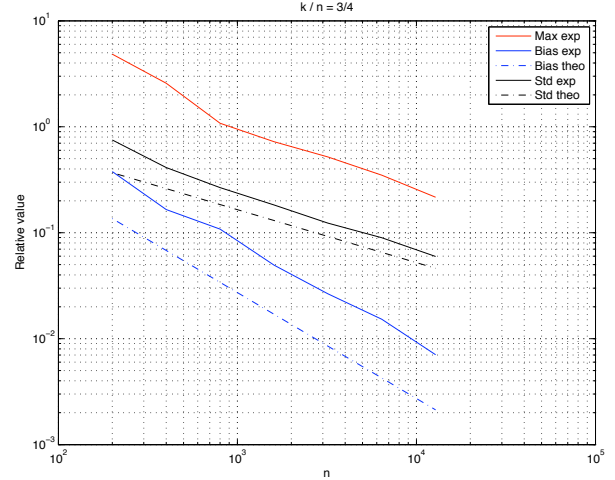Observe first the excellence of the estimator. $n = 12,800$

(last point on curves) represents around 1,000,000 detection trials for $p = 3/4$ or around 430,000 for $p = 1/2$. Any estimation yielded a result between $4.0*10^{-11}$ and $5.7*10^{-11}$ with $p = 3/4$, or between $3.6 * 10^{-11}$ and $6.0 * 10^{-11}$ with $p = 1/2$. The relative standard deviation represents less than 10%. A classical MC estimator would need more than 2.10^{12} detection trials to achieve such a precision!

Surprisingly enough, the measured variance and bias follow the laws (6) and (7) known for the asymptotic regime even for a small n [1]. Yet, the asymptotic regime is only achieved if the estimations are Gaussian distributed. An Arderson Darling test [11] reveals that this is the case only for the biggest values of n. This happens quicker for p closer to one: $\{\hat{P}_{fa}^{(i)}\}$ are deemed Gaussian distributed when n equals $6,400$ for $p = 3/4$ whereas this hypothesis is clearly rejected for $p = 1/2$. Fig.(6) shows that the empirical distribution of the estimations for a very large value of n exactly matches the distribution $\mathcal{N}(P_{fa}, \sigma^2/n)$ (with σ^2 given by (6)) except a positive offset due to the bias.
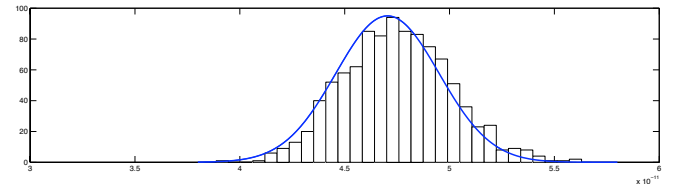

Fig. 6. Empirical distribution of 200 estimations for $n = 50,000$, $p = 3/4$ vs. the asymptotic distribution.

---

[1]The bias is not measured with enough precision with only 1,000 trials for $n = 12,800$ because its order of magnitude is 0.001 times the value of P_{fa}.

Our conclusions of this experiment are the following ones. There are two typical use cases of our algorithm. If the user looks for the order of magnitude of the probability to be estimated, then the choice $p = 1/2$ with around $n = 2,000$ particles gives a fast estimation (around 68,000 detection trials). This is especially true since the variance (6) and the bias (7) are not drastically bigger than the ones for $p = 3/4$. If the issue is to assess an estimation with a given accuracy and confidence range, then the estimator must be in the asymptotic regime where the pdf of the estimation error is known. This experiment shows that a ratio $3/4$ (*i.e.* closer to one) is advised. Each estimation lasts longer but, in the end, this is the quickest way to achieve the asymptotic regime.

A faster way to yield a confidence interval is to observe the number of iterations of several independent estimations. For $p = 1/2$ and $n \geq 800$, more than two thirds of the estimations end at $N = 34$ iterations (see Fig. 7), which gives a confidence interval of $[p^N, p^{N+1}] = [2.91, 5.82] * 10^{-11}$. For $p = 3/4$ and $n \geq 1,600$, more than two third of the estimations end at $N = 82$ iterations (see Fig. 8), which gives a confidence interval of $[p^N, p^{N+1}] = [4.26, 5.69] * 10^{-11}$. Once again, a bigger p provides more accurate results but at the cost of slower estimations.
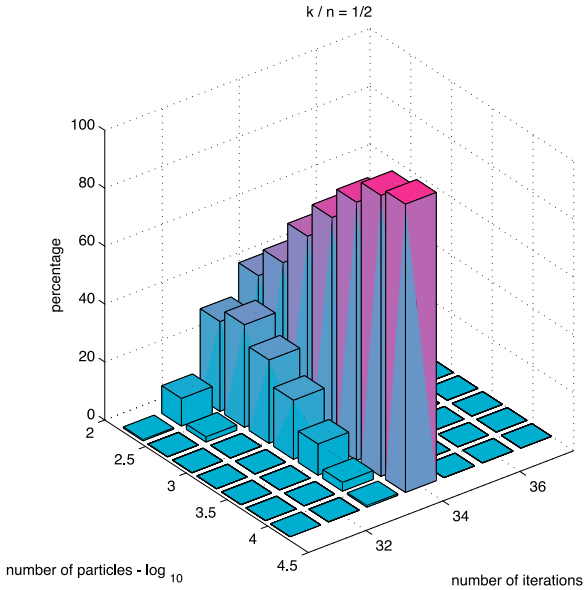

Fig. 8. Confidence intervals are smaller as n increases. Percentage of estimations over 1,000 runs for $p = 3/4$.

Fig. 7. Confidence intervals are smaller as n increases. Percentage of estimations over 1,000 runs for $p = 1/2$.

5. ERROR EXPONENTS MEASUREMENTS

A watermarking scheme is deemed as sound if its probability of false alarm and its probability of false negative decrease exponentially with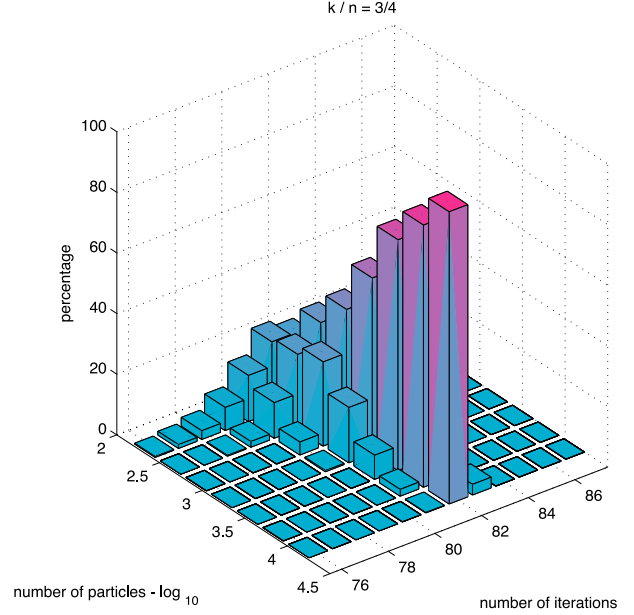 the dimension L of the signals under an embedding power constraint. Within this class, the comparison of two watermarking schemes can be based on their exponential decreasing rates, *i.e.* their error exponents defined as follows:

$$E_{fa}(\tau) = -\lim_{L \to +\infty} \frac{1}{L} \log P_{fa}, \qquad (9)$$

$$E_{fn}(\tau) = -\lim_{L \to +\infty} \frac{1}{L} \log P_{fn}. \qquad (10)$$

There are very few watermarking schemes where error exponents have closed form expressions [12]: For instance, the additive spread spectrum with a single nappe hypercone detection region, the improved sign embedder with a dual nappe hypercone detection region. Furthermore, these theoretical expressions do not foresee a noisy channel (*i.e.* attack) to calculate $E_{fn}(\tau)$. In practice, it is extremely hard to estimate these error exponents. As far as we know, it has never been proposed in the watermarking community. The reason is that huge values of L should imply very very low probabilities of errors impossible to be estimated. This is no longer a problem with our algorithm, and we simply estimate the error exponents by $\hat{E}_{fa}(\tau) = -\log \hat{P}_{fa}(\tau)/L$ and $\hat{E}_{fn}(\tau) = -\log \hat{P}_{fn}(\tau)/L$ with a big enough L.

For the false negative, the rare event is that a watermarked (and possibly attacked) vector has a score below a small threshold. At each step, the estimator sets $\hat{\tau}_j$ as the k-th highest scores. Hence, the intermediate thresholds are indeed decreasing. We can also study the impact of an attack on E_{fn} as soon as the attack vector $\mathbf{n}$ has a statistical model with the two following properties:

- We are able to generate vectors distributed as $p_{\mathbf{n}}$,

- There exists a modification process with a controllable strength that lets this distribution invariant.

We now work with couples of vectors $\{\mathbf{x}, \mathbf{n}\}$, and their score is the detection function applied to the attacked and watermarked vector: $s(\mathbf{w}(\mathbf{x}) + \mathbf{n})$, where $\mathbf{w}(.) : \mathbb{R}^L \mapsto \mathbb{R}^L$ is the watermark embedding function. The replication process changes both vectors in a couple, each one with its distribution invariant modification process. Another technical detail is that our algorithm is run only once, storing the intermediate thresholds in order to estimate the mapping $\{E_{fn}(\hat{\tau}_j), \hat{\tau}_j\}$. The same holds for the false alarm error exponents $\{E_{fa}(\hat{\tau}'_j), \hat{\tau}_j'\}$. An interpolation finally gives $\{E_{fa}(\hat{\tau}_j), E_{fn}(\hat{\tau}_j)\}$.

The experimental setup is the following: $L = 4000$, host vectors are Gaussian distributed with variance $\sigma_X = 1$. The embedding power equals $P_e = 0.1$. We test three watermarking schemes: Additive spread spectrum scheme with $s(\mathbf{x}) = \mathbf{x}^T \mathbf{u}/\|\mathbf{x}\|$, 'improved' sign embedder with $s(\mathbf{x}) = |\mathbf{x}^T \mathbf{u}|/\|\mathbf{x}\|$ as detailed in [12], and the JANIS scheme with order 2 [13]. For the first two schemes, the relationship between $E_{fa}(\tau)$ and the threshold τ is perfectly known [12]. However, there is no expression for $E_{fn}(\tau)$ under an attack (here a Gaussian white noise with variance $\sigma_N^2 = 0.1$). For the JANIS scheme, we have to estimate both $E_{fa}(\tau)$ and $E_{fn}(\tau)$. Fig. 9 shows the results.
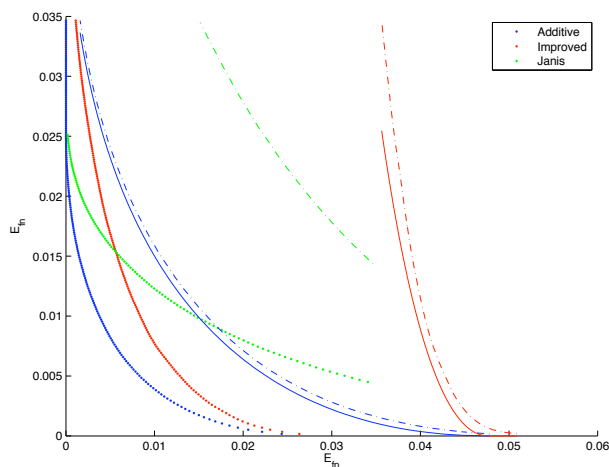

Fig. 9. Error exponents experimental measurements. E_{fn} against E_{fa}. Solid line: Theoretical curves (without noise). Dash-dot line: Experimental curve (without noise). Dotted line: Experimental curve (with AGWN $\sigma_N^2 = 10.P_e$).

From an experimental point of view, the measurements are good with only a small inaccuracy. We blame two shortcomings. L is not big enough and the ratio $L^{-1} \log P_{fa}$ (*idem* with the false negative exponent) does not reflect the rate of the exponential decay. A better way would be to estimate $\log(P_{fa})$ for several values of L and to estimate the exponent with a linear regression. Second, these plots were obtained very rapidly with our algorithm working with $n = 3,200$ and $k = n/2$. Therefore, the accuracy of the estimation of P_{fa} itself is not at best. But, we are indeed interested in showing that error exponents can be measured very rapidly: The experimental curves for the additive and 'improved' sign embedder have the right shape (in particular for the 'improved' sign scheme, E_{fn} goes to infinity when E_{fa} goes to zero). In the same way, the range of the measurements is limited by the $N_{\max}$, which is here set to 200.

From a watermarking point of view, it is quite difficult to announce which scheme performs better. All of them share the same detection complexity. The improved scheme has the advantage of an infinite E_{fn} when there is no attack. JANIS performances curve seems to be better only at high E_{fa}. Yet, performances of course collapse with the presence of an attack, but JANIS seems to be the most robust of the three compared schemes.

6. APPLICATION

The probability of false alarm of the technique [7] used for the international challenge BOWS-2 (Break Our Watermarking Technique, 2nd edition) has been experimentally assessed with this algorithm. The detector extracts some wavelet transform coefficients and makes N_c normalized correlations $\{x_i\}_{i=1}^{N_c}$ with secret vectors. The detection is positive if $s(\mathbf{x}) = \max_i |x_i|$ is above a threshold. Hence, the acceptance region is composed of several dual hypercones, possibly intersecting. Denotes $P_{fa}(1)$ the probability of being in one dual hypercones, a union bound limits the total probability of false alarm: $P_{fa} \leq N_c P_{fa}(1)$, with equality if no intersection. Considering that the secret vectors are independent over the all secret key ensemble, the expected total probability of false alarm is given by $\mathbb{E}_K[P_{fa}] = 1 - (1 - P_{fa}(1))^{N_c}$. However, for a given secret key K, we didn't find any other way than our experimental assessment to evaluate the probability of false alarm [7].

7. CONCLUSION

We presented an efficient estimator of probability of rare event defined as $\text{Prob}[s(\mathbf{x}) > \tau | \mathbf{X} \sim p_{\mathbf{X}}]$ knowing the distribution $p_{\mathbf{X}}$. The performances of the algorithm have been validated with a scenario where this probability has a closed form expression. This framework is very useful for experimentally assessing the probability of false alarm of zero-bit watermarking technique. However, the estimation is accurate provided that the statistical model of the extracted features match the reality. Our future works investigate whether it is possible to directly apply our estimator on images in order to get rid off this assumption.

8. REFERENCES

[1] H. Vincent Poor, *An introduction to signal detection and estimation*, vol. 2nd edition, Springer, 1994.

[2] Janos Galambos, *Advanced probability theory*, vol. 10 of *Probability: Pure and Applied*, Marcel Dekker Inc., New York, second edition, 1995.

[3] P. Comesaña, *Side-Informed Data Hiding: Robustness and Security Analysis*, Ph.D. thesis, Universidade de Vigo, 2006.

[4] A. Lagnoux, "Rare event simulation," *PEIS*, vol. 20, no. 1, pp. 45–66, jan 2006.

[5] F. Cérou and A. Guyader, "Adaptive multilevel splitting for rare event analysis," *Stochastic Analysis and Applications*, vol. 25, no. 2, pp. 417–443, 2007.

[6] A. Guyader, F. Cérou, T. Furon, and P. Del Moral, "Rare event simulation for a static distribution," Tech. Rep. RR-6792, INRIA, 2009.

[7] T. Furon and P. Bas, "Broken arrows," *EURASIP Jounal on Information Security*, vol. 2008, no. ID 597040, pp. doi:10.1155/2008/597040, 2008.

[8] M. Miller and J. Bloom, "Computing the probability of false watermark detection," in *Proc. of the third Int. Workshop on Information Hiding*, A. Pfitzmann, Ed., Dresden, Germany, September 1999, pp. 146–158, Springer Verlag.

[9] P. Comesaña, M. Barni, and N. Merhav, "Asymptotically optimum embedding strategy for one-bit watermarking under gaussian attacks," in *Security, Steganography and Watermarking of Multimedia contents VIII*, San Jose, CA, USA, jan 2008, vol. 6819 of *Proc. of SPIE-IS&T Electronic Imaging, SPIE*.

[10] Milton Abramowitz and Irene A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, vol. 55 of *National Bureau of Standards Applied Mathematics Series*, Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 1964.

[11] Henry C. Thode, Jr., *Testing for normality*, vol. 164 of *Statistics: Textbooks and Monographs*, Marcel Dekker Inc., New York, 2002.

[12] N. Merhav and E. Sabbag, "Optimal watermarking embedding and detection strategies under limited detection resources," *IEEE Trans. on Inf. Theory*, vol. 54, no. 1, pp. 255–274, Jan 2008.

[13] T. Furon, G. Silvestre, and N. Hurley, "JANIS: Just Another N-order side-Informed Scheme," in *Proc. of Int. Conf. on Image Processing ICIP'02*, Rochester, NY, USA, September 2002, vol. 2, pp. 153–156.