Two Key Estimation Techniques for the Broken-Arrows Watermarking Scheme

Patrick Bas, Andreas Westfeld

Two Key Estimation Techniques for the Broken-Arrows Watermarking Scheme

Patrick Bas
CNRS - Lagis
Ecole Centrale de Lille , Avenue Paul Langevin
BP 48 , 59651 Villeneuve d'Ascq, France
Patrick.Bas@ec-lille.fr

Andreas Westfeld
HTW Dresden
University of Applied Sciences
01008 DRESDEN, PF 120701, Germany
andreas.westfeld@htw-dresden.de

ABSTRACT

This paper presents two different key estimation attacks targeted for the image watermarking system proposed for the BOWS-2 contest. Ten thousands images are used in order to estimate the secret key and remove the watermark while minimizing the distortion. Two different techniques are proposed. The first one combines a regression-based denoising process to filter out the component of the original images and a clustering algorithm to compute the different components of the key. The second attack is based on an inline subspace estimation algorithm, which estimates the subspace associated with the secret key without computing eigen decomposition. The key components are then estimated using Independent Component Analysis and a strategy designed to leave efficiently the detection region is presented. On six test images, the two attacks are able to remove the mark with very small distortions (between 41.8 dB and 49 dB).

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous—*Watermarking*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance Evaluation (efficiency and effectiveness)*; K.6.m [**Management of Computing and Information Systems**]: Miscellaneous—*Security*

General Terms

Security, Algorithms

Keywords

Zero-bit watermarking algorithm , Security, Attack, Subspace Estimation

1. INTRODUCTION

If watermarking robustness deals with the performance of a watermarking scheme against common processing operations (re-compression, transcoding, editing operations), the security of a watermarking scheme is addressed whenever an adversary is part of the game and tries to remove the watermark. There are two important families of security attacks:

- Sensitivity attacks [1] aim at removing the watermark by using the watermark detector as an oracle,
- Information leakage attacks [2] aim at estimating the secret key analysing contents watermarked with the same key.

In order to assess the security of a robust watermarking against information leakage attacks, the third episode of the BOWS-2 contest [3] was run during 3 months (the two first episodes were focused on robustness and sensitivity attacks). During the third episode, the adversary had access to the description of the embedding and detection watermarking schemes, this is compliant with the Kerckhoffs' principle [4] used in cryptanalysis. Moreover, 10,000 images watermarked with the same secret key were also available to the adversary and her ultimate goal was consequently to analyse these images in order to estimate the secret key and remove the watermark while minimizing the PSNR between the 3 original and watermarked images.

Classical information leakage attacks encompass key estimation using blind source separation schemes such as Principal Component Analysis [5], Independent Component Analysis [2] and clustering schemes such as set-membership approaches [6] or K-Means [7].

This papers presents and compares two attacks that have been used on the watermarking scheme called Broken-Arrows [8] used during BOWS-2. The first one, designed by Andreas Westfeld, was the most efficient one during the contest, and relies on a denoising step inspired from [9], a clustering step and an estimation step. The second attack has been designed by Patrick Bas later on and relies on the global estimation of the secret subspace using an inline PCA algorithm.

The paper is organised as follows: the next section presents a description of the main features of the Broken Arrows technique. The third section describes a first attack mixing denoising and clustering. The fourth section presents an alternative method to perform the attack by estimating the secret subspace using inline subspace tracking and estimate the secret key using independent component analysis. Finally, the results of the two attacks are presented and compared in Section 5.

2. BROKEN ARROWS IN A NUTSHELL

MM&Sec'09, September 07–08, 2009, Princeton, USA.

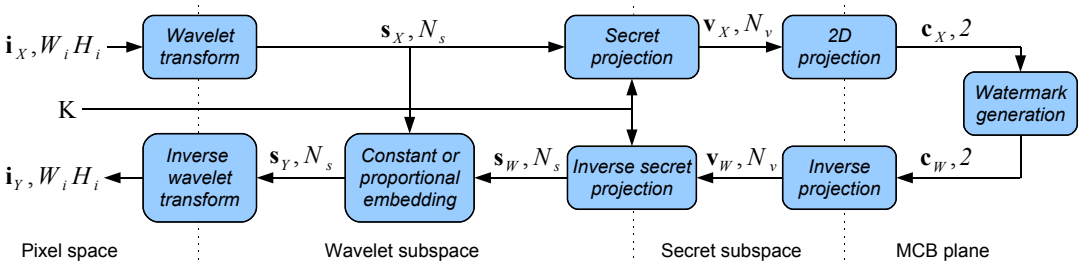

Figure 1: Diagram of the BA embedding scheme, each couple X,Y denotes respectively the vector and the size of the vector that is processed.

The whole diagram of the embedding scheme is depicted on Figure 1 and for an extended description of the watermarking system, the reader is invited to read [8]. The BA watermarking scheme first performs a wavelet decomposition of the image $\mathbf{I}_X$ and it watermarks all the components but the low frequency ones. For a 512×512 grey-level image, $N_s = 258048$ wavelet coefficients of 9 subbands are arranged in a vector $\mathbf{s}_X$ to be watermarked.

The security of the system relies on a secret projection on $N_v = 256$ pseudo-orthogonal vectors generated using a pseudo-random generator seeded using the key. The embedding is performed in this secret subspace by using both informed coding and informed embedding [10]. Informed coding is used by selecting the one vector that is closest to the host vector from a set of $N_c = 30$ pseudo-orthogonal vectors out of 256. This way the embedding distortion is minimised. Informed embedding is performed by pushing the host content as far as possible from the border of the detection region and looks at maximizing the robustness. Once the watermark vector $\mathbf{s}_W$ is generated, two embeddings are possible, a *constant* embedding which does not consider psychovisual requirements corresponding to

$$\mathbf{s}_Y(i) = \mathbf{s}_X(i) + \mathbf{s}_W(i), \tag{1}$$

and a *proportional* embedding that acts as a psychovisual mask:

$$\mathbf{s}_Y(i) = \mathbf{s}_X(i) + |\mathbf{s}_X(i)|\mathbf{s}_W(i), \tag{2}$$

where $\mathbf{s}_X(i)$ and $\mathbf{s}_Y(i)$ denote respectively the original and watermarked wavelet coefficients.

In the end, most of information about the secrecy of BA relies in the set of N_c vectors $\mathbf{c}_i$ of size N_s. The $N_v - N_c$ other vectors are used during the embedding, but their contributions are very small and as we will see in Section 5, powerful removal attacks can already be devised by estimating only the subspace of size N_c.

3. A CLUSTERING APPROACH BASED ON DENOISING

3.1 The denoising process

There are several kinds of noise that we should distinguish:

1. the watermark $\mathbf{s}_W$, which is a random vector that is independent of the image content,
2. the image noise that is *not* independent of the local surrounding in the image, and
3. the estimation error that is added by the denoising process described in this section.

These three kinds of noise may have similar spectral properties. Our denoising process is used to weaken the content-independent watermark $\mathbf{s}_W$ as much as possible while keeping a maximum of the content $\mathbf{s}_X$. In contrast to the usual meaning of the word "denoising"—the reduction of random visual image artefacts—this denoising process will not reduce any visible noise and might even increase such artefacts. So this procedure is rather a de-watermarking process than a denoising. It was developed during the first episode of BOWS-2 [9]. In the figurative sense it is comparable to the self similarities attack [11]. Parts from the image are restored from the surrounding. Because locally close values in images strongly depend on each other, but the elements of the watermark do not, the image can be preserved by estimation from the surrounding while the watermark is completely removed (cf. Figure 2). We use simple linear regression to

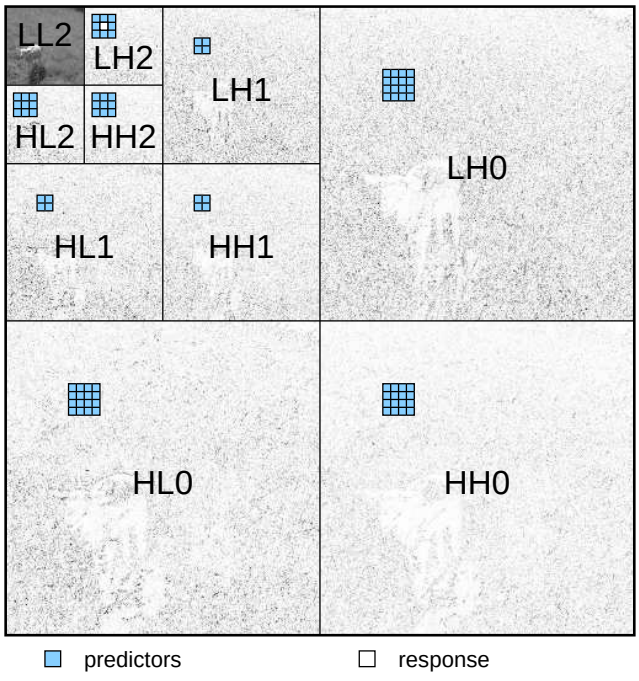

Figure 2: Model for estimating the absolute value of wavelet coefficients in LH2 from the surrounding

Table 1: Number of terms k for prediction

response in	number of terms from level 0	level 1	level 2	**total terms**
level 0	$3 \times 9 - 1$	0	0	**26**
level 1	3×4	$3 \times 9 - 1$	0	**38**
level 2	3×16	3×4	$3 \times 9 - 1$	**86**

predict the absolute value of N_s wavelet coefficients $\mathbf{s}_Y(j)$ from k "neighbouring" coefficients $\mathbf{s}_Y(i_1), \ldots, \mathbf{s}_Y(i_k)$:

$$|\mathbf{s}_Y(j)| \quad = \quad \beta_1 |\mathbf{s}_Y(i_1)| + \cdots + \beta_k |\mathbf{s}_Y(i_k)| + \epsilon_j. \qquad (3)$$

The number of terms k depends on the decomposition level that the coefficient belongs to (cf. Table 1). The regression model collects the local dependencies between the wavelet coefficients. We determine the predictor parameters $\hat{\beta}_1, \ldots, \hat{\beta}_n$ for which we find the minimum sum of squared residuals $\sum_{s=1}^{n} \epsilon_j^2$ (ordinary least squares). This condition is equivalent to the maximum PSNR, which is a logarithmic measure based on the mean squared error (MSE). We predict the unmarked coefficient by prediction of its absolute value and take the sign from the marked original[1]:

$$\hat{\mathbf{s}}_X(j) \quad = \quad \operatorname{sign}(\mathbf{s}_Y(j)) \cdot (\hat{\beta}_1 |\mathbf{s}_Y(i_1)| + \cdots + \hat{\beta}_k |\mathbf{s}_Y(i_k)|). \qquad (4)$$

Figure 2 marks a predicted coefficient in LH2 and the corresponding terms used for prediction. Every coefficient in LH2 is estimated from

- its direct neighbours in LH2,
- its counterpart in the subbands HL2 and HH2 together with their direct neighbours,
- its superior counterparts from the first and second level of decomposition (4 and 16 per subband, respectively).

One of the key properties of this denoising process is its non-interactivity. The attacked images are produced without submitting trials to the detector. All computations can be done locally on the attacker's side.

3.2 The clustering process

In this section we will cluster the images into $N_c = 30$ bins, depending on the version $v = 1 \ldots N_c$ of the watermark $\mathbf{s}_{W(v)}$ that has been selected during the informed coding stage. In principle these bins are ordered, since the versions of the watermark are consecutive chunks of N_s bits from the pseudorandom number generator that was seeded with the secret key. However, since the clustering works without this key, the order of the bins is determined by this process and might be different. The version v that is used depends on the feature vector $\mathbf{s}_X$ to be marked:

$$v = \underset{i=1 \ldots N_c}{\operatorname{argmax}} \; |\operatorname{cor}(\mathbf{s}_{W(i)}, \mathbf{s}_X)| \qquad (5)$$

Since the image content in the marked feature vector $\mathbf{s}_Y$ is much stronger than the embedded watermark $\mathbf{s}_W$ (PSNR of the watermark is about 42.5 dB), it is impossible to correctly decide whether two images $\mathbf{I}_1$ and $\mathbf{I}_2$ that are marked using the same key belong to the same or different bins, based on the (Pearson) correlation of their feature vectors $\mathbf{s}_{Y1}$ and $\mathbf{s}_{Y2}$ alone. However, the chances are higher, if we can take an estimate of the embedded watermark(s) instead and decide based on their correlation. The difference between the marked original $\mathbf{s}_Y$ and the dewatermarked image from the denoising process $\hat{\mathbf{s}}_X$ forms such an approximation of $\mathbf{s}_W$:

$$\hat{\mathbf{s}}_W = \mathbf{s}_Y - \hat{\mathbf{s}}_X. \qquad (6)$$

We can pick one image of the BOWS-2 database $\mathcal{D}$ with the approximated watermark $\hat{\mathbf{s}}_{Wi}$ and determine the absolute correlation c with all $\hat{\mathbf{s}}_{Wj}$:

$$c \quad = \quad |\operatorname{cor}(\hat{\mathbf{s}}_{W(i)}, \hat{\mathbf{s}}_{W(j)})|. \qquad (7)$$

We picked the "Sheep" image, which is one of the three to be attacked during Episode 3. Let's call this image the leader of bin 1. The clustering started with $i = 3661$, which is the index of Sheep in the BOWS-2 database, and $j = \{1 \ldots |\mathcal{D}|\}$. We expect a "strong" absolute correlation ($c \approx 0.01$) if two images belong to the same bin and a weaker ($c \approx 0.002$) if they don't.

We tried to define the membership of a bin by c, which exceeds a certain threshold. This first approach did not work very well, because c of the bin members and its leader ranges from 0.03 almost to 0 (cf. Figure 3).

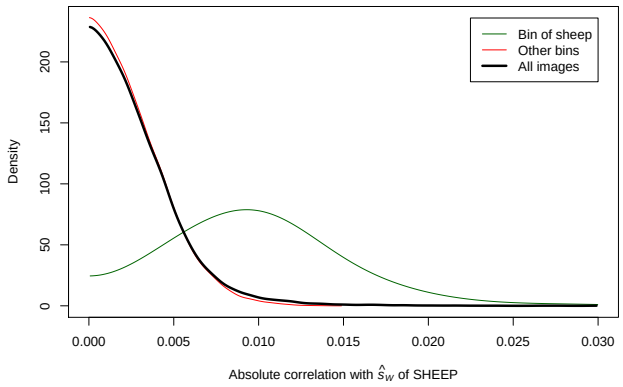

Figure 3: Density of the absolute correlation between the approximated watermarks of the BOWS-2 database and the image "Sheep"

A better approach, which we finally used for clustering, works "by exclusion." The idea is to select the image with the smallest c to lead the next bin (cf. Algorithm 1). The more leaders are selected (with growing k in the algorithm), the clearer the bins are clustered. Step 5 updates the current leader in each bin by new leader, that

Algorithm 1 Cluster by exclusion

1: $\ell_1 := 1$ (we used $\ell_1 := 3661$, image Sheep, without restricting generality)
2: **for** $k = 2 \ldots N_c - 1$ **do**
3: $\quad \ell_{k+1} := \operatorname{argmin}_{j=1}^{|\mathcal{D}|} \max_{i=1}^{k} (|\operatorname{cor}(\hat{\mathbf{s}}_{W(\ell_i)}, \hat{\mathbf{s}}_{W(j)})|)$
4: $\quad$ **for** $m = 1 \ldots k$ **do**
5: $\quad\quad \ell_m := \operatorname{argmax}_{j=1}^{|\mathcal{D}|} |\operatorname{cor}(\hat{\mathbf{s}}_{W(\ell_m)}, \hat{\mathbf{s}}_{W(j)})|$ for $j \neq \ell_m$
6: $\quad$ **end for**
7: **end for**

[1]The predicted absolute values were broadly positive. (This is not obvious, because the predictor is not aware of the constraint that we expect a non-negative response.)

might have a stronger discriminating power. We consolidated the BOWS-2 database by removing all clones. (We replaced the 6533.pgm, 7263.pgm, 7265.pgm, 7602.pgm, and 7856.pgm by 9998.pgm, 9999.pgm, 10000.pgm, 0.pgm, and .pgm; sheep.pgm was inserted as the missing 3661.pgm, so $\mathcal{D}$ contains 1.pgm ... 9997.pgm.) This algorithm makes some assumptions. Step 3 assumes that ℓ_{k+1} belongs to a new bin. This is sometimes not the case. At the end of this algorithm one bin was split, i.e., we had two bins with about 170 members and about 340 in all others. (We did not suppose a biased database and expected $|\mathcal{D}|/N_c \approx 333$ images in each bin.) So we continued the algorithm for $k = 30$ and $k = 31$, removed one leader of the split bin that is revealed by its unexpectedly low number of members, and finally rerun the loop in Steps 4...6 for all 30 bins, yielding all 30 bin leaders $\ell_1 \ldots \ell_{30}$. Based on these bin leaders we define an operator $\text{bin}(i)$ that maps an image with index i in the database to the index of its bin:

$$\text{bin}(i) := \underset{j=1\ldots N_c}{\text{argmax}} \; |\,\text{cor}(\hat{\mathbf{s}}_{W(\ell_j)}, \hat{\mathbf{s}}_{W(i)})|. \tag{8}$$

A posteriori we tested that the clustering defined by $\text{bin}(i)$ is correct: no image was assigned to the wrong bin.

3.3 The key estimation and removal process

The key estimation process for a particular image $\mathbf{I}_k \in \mathcal{D}$ combines all estimated watermarks belonging to $\text{bin}(k)$ in order to find an improved estimate $\mathbf{s}^*_{W(k)}$. The pairwise correlation of two members in the same bin can have a positive or negative sign. The element-wise sum of all estimated watermarks in the bin will be neutral if we do not watch the sign of their watermark.

$$\mathcal{I}_k = \{i | \mathbf{I}_i \in \mathcal{D}, \text{bin}(i) = \text{bin}(k)\} \tag{9}$$

$$\mathbf{s}^*_{W(k)} = \sum_{i \in \mathcal{I}_k} \text{sign}(\text{cor}(\hat{\mathbf{s}}_{W(i)}, \hat{\mathbf{s}}_{W(k)})) \cdot \hat{\mathbf{s}}_{W(i)} \tag{10}$$

Finally we remove the watermark from the feature space by subtracting a PN sequence that is scaled to the detection border:

$$\mathbf{s}^*_{X(k)} = \mathbf{s}_{Y(k)} - \gamma \cdot \text{sign}(\hat{\mathbf{s}}^*_{W(k)}). \tag{11}$$

"sign" returns the element-wise sign of the vector. The scalar value γ is optimised to produce the unmarked image that is closest to the detection boundary.

4. A SUBSPACE ESTIMATION APPROACH

In this section, we propose an approach based on a partial estimation of the secret projection used by the embedding algorithm (see Figure 1). Our rational relies on the fact that the embedding increases considerably the variance of the contents within the secret subspace, in particular along the axes of the N_c vectors $\mathbf{c}_i$ that are used during the embedding. To illustrate this phenomenon, Figure 4 depicts a comparison between the histograms of the absolute correlations for original and watermarked contents on 10,000 images during the BOWS-2 challenge (embedding distortion of 43 dB). This shows clearly an important increases of the variance within the secret subspace, consequently the strategy that is developed in this section is to estimate the subspace spanned by the vectors $\{\mathbf{c}_i\}$ by estimating the components of important variances from the observations.

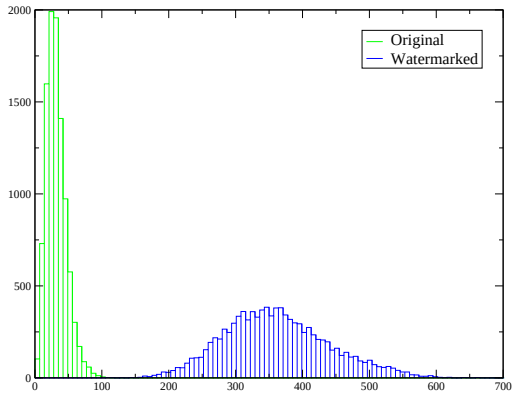

Figure 4: Histogram of the maximum of the 30 correlations (in absolute value) for 10 000 images (PSNR=43 dB), proportional embedding.

If such similar strategies have already been used for security analysis of watermarking systems [5, 2, 12], the estimation of the secret subspace in our case is challenging for different reasons:

- Contrary to the systems addressed in [5, 2, 12], the proposed method used a secret subspace of large dimension (30) in order to avoid basis estimation techniques such as averaging,
- The dimension of the host signal itself is very important (258048),
- The system is used is real-life conditions on 10 000 images on a watermarking scheme that fulfils the different constraints regarding robustness but also visual distortion.

In order to perform subspace estimation, one usually uses Principal Component Analysis (PCA) which can be performed by an Eigen Decomposition (ED) of the covariance matrix obtained using the different observations. In our practical context however, the ED is difficult to perform because of the following computational considerations:

- The covariance matrix if of size $N_s \times N_s$, which means that 248 gigabytes are required if each element of the matrix is stored as a float,
- The computation of the covariance matrix requires around $O(N_o N_s^2) \approx 10^{12}$ flops,
- The computational cost of the ED is $O(N_s^3) \approx 10^{15}$ flops.

Consequently, we have looked for another way to compute the principal components of the space of watermarked contents. One interesting option is to use a inline algorithm which compute the principal vectors without computing any $N_s \times N_s$ matrices.

4.1 The OPAST algorithm

The OPAST algorithm [13] (Orthogonal Projection Approximation Subspace Tracking) is a fast and efficient iterative algorithm that uses observations as inputs to extract

the N_p principal components (e.g. the component associated with the N_p more important eigenvalues). The goal of the algorithm is to find the projection matrix $\mathbf{W}$ in order to minimize the projection error $J(\mathbf{W}) = E(||\mathbf{r} - \mathbf{W}\mathbf{W}^t\mathbf{r}||^2$ on the estimated subspace for the set of observations $\{\mathbf{r}_i\}$. This algorithm can be decomposed into eight steps sum-up in Algorithm 2.

The notations are the following: the projection matrix $\mathbf{W}_0$ is $N_s \times N_p$ and is initiated randomly, the parameter $\alpha \in [0; 1]$ is a forgetting factor, $\mathbf{y}$, $\mathbf{q}$ are N_p long vectors, $\mathbf{p}$ and $\mathbf{p}'$ are N_s long vectors, $\mathbf{W}$ is a $N_s \times N_p$ matrix, $\mathbf{Z}$ is a $N_p \times N_p$ matrix.

Algorithm 2 OPAST algorithm

1: **for all** observations $\mathbf{r}_i$ **do**
2: $\mathbf{y}_i = \mathbf{W}_{i-1}^t \mathbf{r}_i$
3: $\mathbf{q}_i = \frac{1}{\alpha}\mathbf{Z}_{i-1}\mathbf{y}_i$
4: $\gamma_i = \frac{1}{1+\mathbf{y}_i^t\mathbf{q}_i}$
5: $\mathbf{p}_i = \gamma_i(\mathbf{r}_i - \mathbf{W}_{i-1}\mathbf{y}_i)$
6: $\mathbf{Z}_i = \frac{1}{\alpha}\mathbf{Z}_{i-1} - \gamma_i\mathbf{q}_i\mathbf{q}_i^t$
7: $\tau_i = \frac{1}{||\mathbf{q}_i||^2}\left(\frac{1}{\sqrt{1+||(p)_i||^2||\mathbf{q}_i||^2}} - 1\right)$
8: $\mathbf{p}'_i = \tau_i\mathbf{W}_{i-1}\mathbf{q}_i + \left(1 + \tau_i||\mathbf{q}_i||^2\right)\mathbf{p}_i$
9: $\mathbf{W}_i = \mathbf{W}_{i-1} + \mathbf{p}'_i\mathbf{q}_i^t$
10: **end for**

Step 6 is a recursive approximation of a the covariance matrix for the N_p principal dimensions. Steps 7 and 8 are the translations of the orthogonalisation process.

Since the complexity of OPAST is only $4N_sN_p + O(N_p^2) \approx 10^7$ flops per iteration, the use of the OPAST algorithm is possible in our context. Furthermore, it is easy to use and only relies on the parameter α for the approximation of the pseudo covariance matrix and does not suffer from instability.

4.2 Estimation assessment

In order to run experiments and to assess the behaviour of the subspace estimation algorithm we used the Square Chordal Distance (SCD) to compute a distance between two subspaces (the one coming from the secret key and the estimated subspace). The use of chordal distance for watermarking security analysis was first proposed by Pérez-Freire *et al.* [14] and is convenient because the SCD = 0 if the estimated subspaces are equal and SCD = N_p if they are orthogonal.

Given $\mathbf{C}$, a matrix with each column equal to one $\mathbf{c}_i$, the computation of the SCD is defined by the principal angles $[\theta_1 ... \theta_{N_c}]$ (the minimal angles between two orthogonal bases [15]) that are singular values of $\mathbf{C}^t\mathbf{W}$ (note that this matrix is only $N_c \times N_p$):

$$\text{SCD} = \sum_1^{N_c} \sin^2(\theta_i) \tag{12}$$

A geometric illustration of the principal angles is depicted on Figure 5

4.3 OPAST applied on Broken Arrows

We present here the different issues that we have encountered and are specific to the embedding algorithm: the impact of the weighting method, the influence of the host signal and the possibility to use several times the same observations to refine the estimation of the secret subspace.

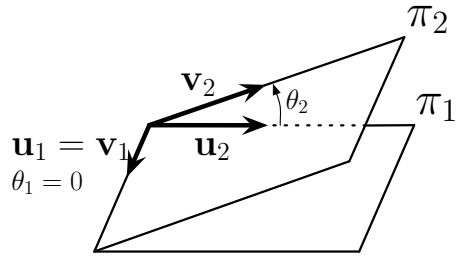

Figure 5: Principal angles between 2 plans π_1 and π_2.

4.3.1 Constant vs Proportional embedding

We have first compared the impact of the embeddings given by the constant embedding (Eq.1) and proportional embedding (Eq.2). The behaviour of the OPAST algorithm is radically different for these two strategies since the estimated subspace is very close to the secret subspace for constant embedding and nearly orthogonal to it for proportional embedding. The evolution of the SCD in both cases is depicted on Figure 6.

Such a problematic behaviour can be explained by the fact that the variance of the contents in the secret subspace is more important using constant embedding than using proportional embedding (compare Figure 6 of [8] with Figure 4). The second explanation is the fact that the proportional embedding acts as a weighting mask which is different for each observations. This makes the principal directions less obvious to find since the added watermark is no more collinear to one secret projection.

4.3.2 Calibration

One solution to address this issue is to try to decrease the effect of the proportional weighting and to reduce the variance of the host signal. This can be done by feeding the OPAST algorithm with a calibrated observation $\hat{\mathbf{s}}_Y$ where each sample is normalised by a prediction of the weighting factor $|s_X(i)|$ according to the neighbourhood $\mathcal{N}$:

$$\hat{s}_Y(i) = s_Y(i)/|\hat{s}_X(i)|, \tag{13}$$

where

$$|\hat{s}_X(i)| = \frac{1}{N}\sum_{\mathcal{N}} |s_Y(i)|. \tag{14}$$

The result of the calibration process on the estimation of the secret subspace is depicted on Figure 6 using a 5×5 neighbourhood for each subband. With calibration, the SCD decreases with the number of observations.

4.3.3 Principal components induced by the subbands

Whenever watermarking is performed on non-iid signals like natural images, the key estimation process can face issues regarding interferences from the host-signals [16]. Figure 7 depicts the cosine of the principal angles for $N_p = 30$ and $N_p = 36$ and one can see that all principal angles are small only for $N_p = 36$. For $N_p = 30$, only 25 out of 30 basis vectors of the subspace were accurately estimated.

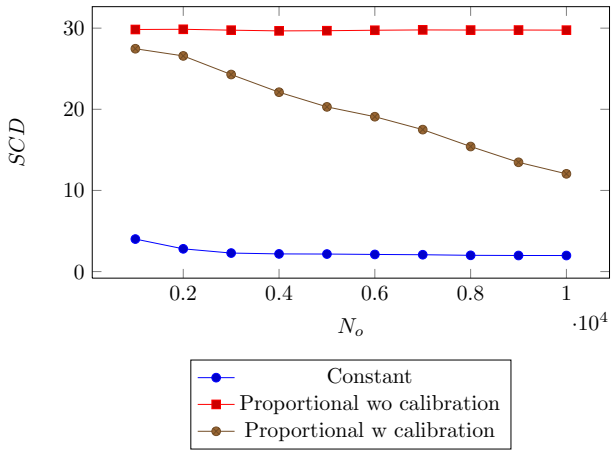

Figure 6: SCD for different embeddings and calibrations (PSNR=43 dB, $N_p = 36$)

Consequently, depending on the embedding distortion, one might choose $N_p > N_c$.

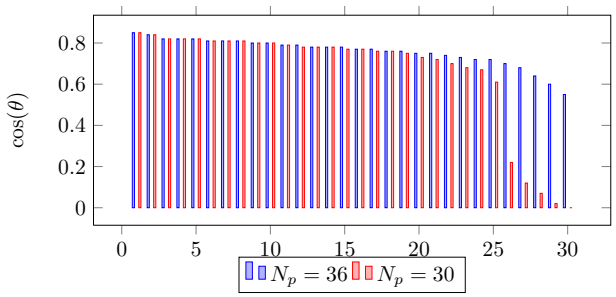

Figure 7: $\cos\theta_i$ for $N_P = 36$ and $N_p = 30$, Embedding PSNR = $43dB$.

4.3.4 *Multiple runs*

In order to improve the estimation of the subspace, another option is to use the contents several times and consequently improve the estimation of the pseudo-covariance matrix in the OPAST algorithm. Figure 8 shows the evolution of the SCD after three multiple runs. We can notice that if the SCD decreases significantly between 10^4 and 2.10^4 observations, the gain for using a third run is poor though.

4.4 Cone estimation using ICA

The last step of the key estimation process is to estimate each $\mathbf{c}_i$ by $\hat{\mathbf{c}}_i$. Since all the variances along the different cone axes are equal, one solution to estimate the direction of each axis is to look for independent directions using Independent Component Analysis (ICA). This strategy has already been used in watermarking security by former key estimation techniques and more information on the usage of ICA in this context can be found in [12].

4.5 Leaving the detection region

The last step is to modify the watermarked content in order to push it outside the detection region of the hypercone of normalised axis $\hat{\mathbf{c}}_k$ which is selected such that:

$$|\mathbf{s}_Y^t \hat{\mathbf{c}}_k| \geq_{j \neg k} |\mathbf{s}_Y^t \hat{\mathbf{c}}_i|.$$

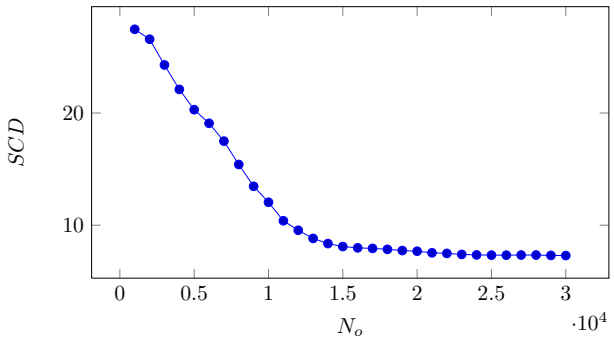

Figure 8: Evolution of SCD after 3 runs (30,000 observations, PSNR = 43 dB, $N_p = 36$).

Theoretically this is possible by cancelling the projection between $\hat{\mathbf{c}}_i$ and $\mathbf{s}_Y$ to create the attacked vector $\mathbf{s}_Z$:

$$\mathbf{s}_Z = \mathbf{s}_Y - \gamma \mathbf{s}_Y^t \hat{\mathbf{c}}_k \hat{\mathbf{c}}_k. \quad (15)$$

However, practically $\hat{\mathbf{c}}_k$ may not be accurate enough to be sure that $\mathbf{s}_Z^t \mathbf{c}_k = 0$, especially if the coordinates of the watermarked content are close to the cone axis. On Figure 9, we can see the effect of this strategy (called "Strat. 1") on two images of the BOWS-2 contest Sheep and Casimir inside the MBC plan (the plan that includes $\mathbf{c}_k$ and the watermarked content $\mathbf{s}_Y$). Another more efficient strategy is to push the content also to the directions that are orthogonal to $\hat{\mathbf{c}}_k$, this can be done by increasing the projection of all the components except for the cone axis:

$$\mathbf{s}_Z = \mathbf{s}_Y - \gamma \mathbf{s}_Y^t \hat{\mathbf{c}}_k \hat{\mathbf{c}}_k + \sum_{j \neq k} (\beta \mathbf{s}_Y^t \hat{\mathbf{c}}_j - 1) \hat{\mathbf{c}}_j. \quad (16)$$

γ and β are constant factors specifying the amount of energy put in the directions which are respectively collinear and orthogonal to the cone axis. This second strategy (called Strat. 2) is depicted on Figure 9 and the PSNR between the watermarked and attacked images for Sheep and Casimir are respectively equal to 41.83 dB and 48.96 dB.

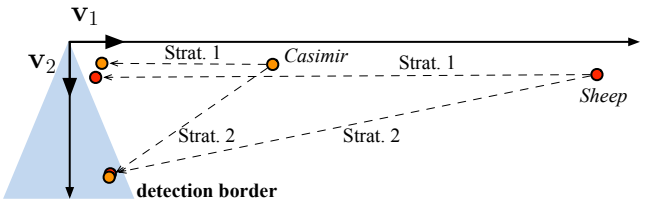

Figure 9: Effects of the different strategies on the MCB plan for Casimir and Sheep.

5. RESULTS

5.1 Attacks after the first approach

Building upon the watermark estimates from a regression-based approach, the clustering perfectly separates all images of the BOWS-2 database in 30 bins defined by the version of the watermark that has been selected by the informed coding step. Within these bins the watermark is simply determined by element-wise averaging of the watermark estimates, but

Table 2: Final PSNR for the three images under attack in Episode 3 (γ represents the scale of the PN sequence, cf. Eq. 11)

Image	PSNR	γ	MCB coord. after attack
Sheep	45.58 dB	1.360	(70.0, 196.5)
Bear	46.64 dB	1.202	(17.8, 50.9)
Horses	46.48 dB	1.226	(35.2, 99.7)

two cases will be considered: positive and negative correlation with the watermark to be removed. The element-wise sign of the averaged watermark forms a PN sequence that is used to eliminate the watermark in the image under attack. Here the detector is needed only a small number of times to find the optimal scale γ of the PN sequence to just remove the watermark with the highest PSNR.

It takes about a minute to find a watermark estimate using the regression-based approach. So for 10,000 images it may easily take a week on a single computer. We assigned this task to a PC farm that returned the result in minutes. The clustering took about 24 hours on a single computer[2], the key estimation took about one minute per key (only three for the three given images are needed, but all 30 could be estimated).

5.2 Attacks after the second approach

Using the attack based on subspace estimation, the subspace is estimated on the 10 000 images provided by BOWS-2 contest. Each image is watermarked with a PSNR between 42.5 dB and 43 dB. As for Episode 3, proportional embedding is used.

OPAST is run using calibration on a 5×5 neighbourhood for each subband (see 4.3.2), $N_p = 36$ (see 4.3.3, and 2.10^4 observations (e.g. two runs, see 4.3.4), and the forgetting factor α is set to 1.

The ICA step was performed using fastICA [17, 18], with a symmetric strategy and the tanh function to estimate negentropy. All the other parameters are set to defaults values.

Watermark removal (see 4.5) uses normalised estimated vectors $\hat{\mathbf{c}}_i$ orientated such that $\mathbf{s}_Y^t \hat{\mathbf{c}}_i > 0$. The second strategy is used and the parameters are set to $\gamma = 1.1 + 0.1i$ (where i is a number of iterations) and $\beta = 50$.

The attack was performed on the five images used during the contest and available on the BOWS-2 website.

Figure 9 shows the effects of the attacks in the MCB plan for "Casimir" and "Sheep".

Table 3 presents the PSNRs after the attack and the number of necessary iterations. The coordinate of the original images in the MCB plane are also presented. As can be seen, the distortion is between $41.8dB$ and $49dB$, which yields very small or imperceptible artefacts. Since the norm of the attacking depends of $\mathbf{s}_Y^t \hat{\mathbf{c}}_i$, the farther the images are from the detection boundary, the more important the attacking distortion is.

6. CONCLUSIONS AND PERSPECTIVES

We point out the weaknesses of a very robust watermarking scheme in terms of security. Theses weaknesses comes from the facts that:

1. It is possible to filter most of the image components using regression-based denoising and consequently to increase the watermark to content ratio,
2. The embedding increases significantly the variance of the data in the secret subspace and subspace estimations techniques can consequently be used,
3. The number of hypercones N_c used to create the detection region is rather small, which makes the estimation easier.

The future directions will consequently try to address these different issues in order to increase the security of the analysed algorithm. However, one has also to consider the inevitable trade-off between robustness and security.

[2]AMD Athlon 64 Processor 3200+ at 2.2 GHz

Table 3: PSNR after successful attack using subspace estimation (i represents the number of iteration necessary to obtain a successful attack).

Image	PSNR	i	MCB coord.	MCB coord. after attack
Sheep	41.83 dB	1	(925,48)	(62,223)
Bear	44.21 dB	0	(532,47)	(88,253)
Horses	41.80 dB	0	(915,20)	(77,233)
Louvre	48.95 dB	0	(321,194)	(96,317)
Fall	46.76 dB	0	(553,250)	(116,370)
Casimir	48.96 dB	0	(352,31)	(59,234)

7. ACKNOWLEDGEMENTS

Patrick Bas is supported by the National French projects Nebbiano ANR-06-SETIN-009, ANR-RIAM Estivale, and ANR-ARA TSAR.

8. REFERENCES

[1] P. Comesaña, L. Pérez Freire, and F. Pérez-González. Blind newton sensitivity attack. *IEE Proceedings on Information Security*, 153(3):115–125, September 2006.

[2] F. Cayre, C. Fontaine, and T. Furon. Watermarking security: theory and practice. *IEEE Trans. Signal Processing*, 53(10), oct 2005.

[3] P. Bas and T. Furon. Bows-2. http://bows2.gipsa-lab.inpg.fr, July 2007.

[4] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX, 1883.

[5] G. Doërr and J-L. Dugelay. Security pitfalls of frame-by-frame approaches to video watermarking. *IEEE Transactions on Signal Processing*, 52, 2004.

[6] L. Pérez-Freire, F. Pérez-González, Teddy Furon, and P. Comesaña. Security of lattice-based data hiding against the Known Message Attack. *IEEE Transactions on Information Forensics and Security*, 1(4):421–439, December 2006.

[7] P. Bas and G. Doërr. Evaluation of an optimal watermark tampering attack against dirty paper trellis schemes. In *ACM Multimedia and Security Workshop*, Oxford, UK, Sept 2008.

[8] T. Furon and P. Bas. Broken arrows. *EURASIP Journal on Information Security*, 8, 2008.

[9] Andreas Westfeld. A regression-based restoration technique for automated watermark removal. In *MM&Sec*, pages 215–220, 2008.

[10] M. L. Miller, G. J. Doërr, and I. J. Cox. Applying informed coding and embedding to design a robust, high capacity watermark. *IEEE Trans. on Image Processing*, 6(13):791–807, 2004.

[11] Christian Rey, Gwenaël Doërr, Gabriella Csurka, and Jean-Luc Dugelay. Toward generic image dewatermarking? In *IEEE International Conference on Image Processing ICIP 2002*, volume 2, pages 633–636, New York, NY, USA, September 2002.

[12] F. Cayre and P. Bas. Kerckhoffs-based embedding security classes for woa data-hiding. *IEEE Transactions on Information Forensics and Security*, 3(1), March 2008.

[13] K. Abed-Meraim, A. Chkeif, and Y. Hua. Fast orthogonal past algorithm. *IEEE Signal Processing Letters*, 7(3), March 2000.

[14] L. Pérez-Freire and F. Pérez-González. Spread spectrum watermarking security. *IEEE Transactions on Information Forensics and Security*, To appear 2008.

[15] A. V. Knyazev and M. E. Argentati. Principal angles between subspaces in an a-based scalar product. In *SIAM, J. Sci. Comput.*, volume 23, pages 2009–2041. Society for Industrial and Applied Mathematics, apr 2002.

[16] P. Bas and J. Hurri. Vulnerability of dm watermarking of non-iid host signals to attacks utilising the statistics of independent components. *IEE proceeding, transaction on information security*, 153:127–139, 2006.

[17] A. Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[18] A. Hyvärinen. The fastica package for matlab. http://www.cis.hut.fi/projects/ica/fastica, July 2005.