RANDOMIZED DETECTION FOR SPREAD-SPECTRUM WATERMARKING: DEFENDING AGAINST SENSITIVITY AND OTHER ATTACKS

Ramarathnam Venkatesan and Mariusz H. Jakubowski

Microsoft Research
One Microsoft Way, Redmond, WA 98052
{venkie, mariuszj}@microsoft.com

ABSTRACT

Spread Spectrum (SS) has been a well-studied technique in signal processing. As a tool for watermarking in an adversarial context, however, this methodology needs caution and new variations. We suggest SS variants where the detection rule is randomized in the sense of having the watermark detector use secret coin flips to choose subsets of the watermarked data and perform correlation tests. We then form a pool of such estimates and pick the median value. We study the effect of such detection methods on sensitivity and estimation attacks, which suggest that randomization is a necessary tool to prevent these types of potentially debilitating adversarial methodologies. We also present other schemes for improving the robustness of SS methods, along with experimental results. Though we recognize the limitations of SS in the face of adversarial attacks, our methods attempt to maximize the potential of SS watermarking in such scenarios.

1. INTRODUCTION

Spread Spectrum (SS) is a popular means of implementing image watermarking (WM) [1]. Via engineering tricks and clever implementation, SS has proven reasonably effective at withstanding image-manipulation and other non-adversarial attacks [2]. Unfortunately, SS is less effective against cryptanalytic attacks [3]. While the ultimate security of SS watermarking is questionable, various methods can be used to extract maximum performance from SS in the face of cryptanalytic adversaries. Our goal in this paper is to present such methodology and analyze its effectiveness in both theory and practice.

2. ALGORITHMS AND ENHANCEMENTS

To embed a WM, SS adds a pseudorandom sequence of small values, or chips, to coefficients in some image representation, typically wavelet- or DCT-based [4]. For detection, SS computes a normalized inner product of that same sequence and the marked coefficients. Techniques such as chip repetition, error correction, and embedded synchronization patterns are typically used to harden SS against common distortions and signal-processing attacks While such methods can resist StirMark [2] and similar attacks, the enhancements may also open the door to adversaries [3].

We summarize our general methodology [4], which aims towards robustness against adversarial attacks. Some of the techniques also help against StirMark-type signal-processing attacks, but this is not our focus.

Embedding in a specially chosen domain: We insert watermark data into the DCT [1] or wavelet transform of an entire image, and we choose a *random subset* of coefficients with the highest power among the middle frequencies. The subsets deflect averaging attacks that collect many distinct images watermarked with the same secret and use averaging to estimate (and possibly reduce) the watermark.

Detection randomized by subset computations: We compute correlations over *pseudorandom subsets* of the watermark data to generate many different watermark responses $c_1, ..., c_p$. We return the median of the p responses, which helps to defeat sensitivity-type attacks [5], as described later.

Pseudorandom chips: The chip values we add to coefficients are selected pseudorandomly from the range $[-D, D]$, where D is a small constant. This differs from classical SS WM, where each chip usually has the value $+D$ or $-D$.

Image-dependent WM keys: We use an image hash [6] as part of the WM key. This helps avoid averaging attacks, which can estimate WM chips by averaging coefficients of many images all watermarked with the same key.

Our scheme uses several other techniques. To amplify a watermark embedded in high-power, low- to middle-frequency DCT coefficients, we apply histogram equalization to an image before we attempt watermark detection. To counter moderate amounts of resizing and cropping, we rescale images before watermarking, either to a standard size or to some quantized dimensions (e.g., rounded to the nearest 20 pixels), and then restore original size. Finally, we embed separate watermarks into randomly overlapping regions of the image. During detection, we use the responses for all regions simultaneously.

The randomizing features of our algorithms seek to minimize the assumptions on how input images are generated [7]. We believe this is important for watermarking techniques to work well across a range of images with varying characteristics, including images traditionally difficult to watermark robustly. A combinatorial approach to formulating and analyzing the problem at hand is in progress and will appear elsewhere.

3. SENSITIVITY ATTACKS

Against standard correlation-based watermark schemes, the *sensitivity attack* [5] can determine crucial data about watermark chips (i.e., values added to coefficients to encode watermarks). This is true even if the attacker has only black-box access to a detector; that is, the attacker can ask the detector only whether or not a given image is watermarked, possibly also obtaining the strength of the

watermark response. A variant of this idea is the following procedure to estimate and subtract out portions of an image watermark:

1. Transform the image or a portion thereof into the domain where the watermark is embedded. In our case, this transform plane P is the DCT or wavelet transform of some color or intensity plane of the image.
2. Choose a random subset $C = \{c_1, c_2, ..., c_k\}$ of k coefficients within the domain. Typically, $1 \leq k \leq 15$ for performance reasons. These are coefficients the attacker will try to guess.
3. Choose a value D that is at least an order of magnitude larger than typical coefficient values in C.
4. Consider the 2^k tuples of the form $\{d_1, d_2, ..., d_k\}$, where each $d_i = +D$ or $-D$, and d_i corresponds to c_i for $i = 1...k$. For each such tuple S_j, do the following:
   (a) Create a transform plane A with dimensions the same as those of P. Each coefficient in A is either 0 or d_i, depending on whether the coefficient's coordinates correspond to those of some c_i. We refer to A as an *attack plane*.
   (b) Create a new image I by transforming A to the image domain. We refer to I as an *attack image*.
   (c) Use the black-box detector to attempt watermark detection in I. Keep track of the corresponding sequences S_j for which watermark detection was successful or strongest.
5. The sequences S_j for which watermark detection succeeded provide an estimate of the signs of watermark chips added to the image. The attacker can repeat this procedure to guess the signs of as many coefficients as desired.
6. Once the attacker has estimated enough chip signs, he can use trial and error to estimate the magnitudes of the chips. Thereafter, subtracting the estimated chips from the embedding domain should degrade the watermark response to the point of detector failure.

We have implemented the above attack for our DCT-based scheme. As expected, the procedure allows us to make accurate guesses of watermark chips if the detector returns an overall correlation as the watermark response. Assuming the black-box detector returns a value indicating watermark strength, and depending on image size, we obtain accurate chip signs by starting our guesses with $k = 2$ or 3 coefficients at a time. We can guess more each time, but the time complexity of this procedure is $O(2^k)$.

As we demonstrate in a later section, we have observed that the above attack does not work well if the watermark response is the median (or weighted median) of a number of subset correlations. In effect, our detection procedure treats the attack image I and the attack coefficients d_i as "outliers" that should neither destroy nor enhance the overall watermark response. Our experiments, described in a subsequent section, present empirical data on attacks that involve guessing $k = 10$ and $k = 32$ watermark chips.

We review some statistical facts needed for an analysis of watermark detection based on the median of subset correlations. First, we recall a standard trick of using the median as a good estimate for the average. Assume we are given an estimator algorithm Y for the average value of a random variable X such that

$$\Pr[|Y - E(X)| \geq \epsilon] \leq \delta.$$

For example, Y may have been obtained via sample averaging. The median method allows one to decrease δ exponentially. The constant $\frac{1}{4}$ in the lemma below can be replaced by any other constant that is bounded away from $\frac{1}{2}$.

Lemma: Let $Y_1, ...Y_n$ be the values produced by independent runs of the algorithm Y for which $|\delta - \frac{1}{2}| = \lambda$, where λ is a positive constant. Let Y_{med} be the median value of the Y_i's. For some constant c, we have

$$\Pr[|Y_{\text{med}} - E(X)| \geq \epsilon] \leq e^{-cn}.$$

This lemma is simple and standard enough, but its security implications seem little known. Now let us imagine an attacker who changes one of the coefficients in the DCT plane to an arbitrary value of his choosing, which he can do easily, since there is no requirement that the resulting image not have significant perceptual distortion. In fact, there exist many DCT coefficients that can be changed significantly with acceptable perceptual distortions. We say that this DCT, as a perceptual characteristic, is *locally unstable*. Let k be the size of the random subset S from the set of all possible n coefficients. The probability p that the coefficient the attacker picked will be included in S is $\frac{k}{n}$. The following lemma states that the detector values before and after the attack remain unchanged, unless the attacker changes too many coefficients. If the attacker does not change enough coefficients, he gains very little information; on the other hand, if the attacker has to change 32 coefficients before the detector value changes, then he has 2^{32} possible values for the signs of the spread-spectrum chips. We call this an *exhaustive-search* strategy, which works only for a limited number of coefficients. Note that we can insert delays into a black-box detector, so that the attacker will be forced to expend a given amount of time for each guess of k coefficients (e.g., 0.1 seconds), no matter how fast a machine he is using.

Lemma (Threshold Phenomenon): Consider a watermarked image, and set $p = \frac{k}{n}$. Assume the attacker changes ζ co-efficients in the DCT plane, and $|p\zeta - \frac{1}{2}| \geq \dot{\lambda}$. Let $S_i, i \leq n$, be the random subsets choosen by the detector. Let D and $\widetilde{D}$ denote the detector values that are output to the attacker. For every $\rho > 0$, we have

$$\Pr_{\Omega}[|D - \widetilde{D}| \geq \rho] \leq e^{-cn}$$

for some constant c, where Ω is the space of coin-flips used by the detector.

Remark: If $p < \frac{1}{2}$, the case when $p\zeta - \frac{1}{2} \geq \lambda$ forces the attacker to change more coefficients than in the case when $p\zeta < \frac{1}{2} - \lambda$, and consequently the attacker gains even less information about signs of the SS chips for a given query to the detector as a black-box oracle.

Remark: The space Ω of the detector's coin-flips need not be known even to the embedder. Thus, there is no need to fix these coin-flips, and the detector may choose them independently on each trial (and even use a hardware noise generator that, unlike a keyed pseudo-random generator, has no reproducible results).

Remark: By the last remark, the attacker gains little advantage (except by exhaustive strategy) from accumulating information by correlated queries to the black-box oracle for detection. Thus, the expected number of trials for a successful sensitivity attack is at least $\min(2^{\zeta}, 2^{cn})$.

4. OTHER ATTACKS

Since SS is locally unstable as a measure of perceptual characteristics, some designers have used repetition as a way of increasing robustness. For example, the scheme in [8] has excellent performance against signal-processing attacks, but fails against estimation attacks [3]. In general, even without repetition one may be able to estimate watermark chips by using correlations in the host signal. For example, if an image is expected to yield relatively constant or predictable DCT coefficients at location (i, j), then one may estimate the watermark coefficient at this location using the average in a neighborhood as an estimate for the original; one may then subtract the estimate from the watermarked image. However, our scheme prevents black-box oracle methods from allowing the attacker to guess which coefficients are used in the process.

A *swap attack* [9, 10] locates perceptually similar regions of a signal and copies one such region to another. There are many variations on this theme, including shifting around pieces of signals to foil watermark detection, estimating and copying watermark data between signals to create false positives, and others. This procedure can be applied across different signals; for example, the attacker may keep a database of non-watermarked images, and copy similar-looking areas, such as small rectangles, from these images into a watermarked image under attack. For watermarking schemes that use local signal features, such as the 8x8 DCT blocks in JPEG compression, such attacks can be effective. However, our experiments have not yielded satisfactory results against our schemes, which embed watermark data more globally (i.e., into the DCT or wavelet coefficients of the entire image or large portions thereof). We have run searches to find and swap small, rectangular regions in both the intensity and DCT domains. This has led to detection failure, but only in cases where the image itself was corrupted. This attack may need further study.

5. RESULTS

The top three graphs of fig. 1 show correlations over 500 watermark subsets in each of 10 images. Each plotted line shows the 500 sorted correlation values computed from random subsets of a watermark embedded in one image. Each subset contains 1.25 percent of the watermarked coefficients. From left to right, the three graphs show results for non-watermarked, watermarked, and attack images, respectively. The attack images contain 10 random DCT coefficients that have been set to large values with signs matching the corresponding watermark chips; thus, these images are the "correct" guesses that an attacker can make for 10 chip signs while trying all 2^{10} possibilities. Note that the subset medians in both the non-watermarked and attack images are similar and close to 0; however, the overall watermark correlation (or average of many watermark subsets) in each attack image is closer to 1, or within the threshold required for successful watermark detection. This latter fact is not shown in these graphs, but in the ones we describe next.

The middle three graphs in fig. 1 show averages and medians of subset correlations over 10 non-watermarked, watermarked, and attack images, respectively. Note that both of these statistics hover around 0 for non-watermarked images and around 0.75 for watermarked images (which have undergone middle-quality JPEG compression). The averages also indicate high watermark response in the attack images, thus allowing the attacker to conclude that he correctly guessed the 10 watermark chips in each image. However, the corresponding medians are still close to 0, indicating no watermark in any of the attack images. Thus, usage of the median for reporting watermark response has foiled the attack.

The two bottom-left graphs show results when the attacker is correctly guessing 32 watermark chips. This means the attacker must have performed an exhaustive search over 2^{32} calls to the black-box detector, making this attack impractical. The medians of the subset correlations are on the threshold of incorrectly showing watermarks. For our images, complete success of the attack required guessing 64 to 128 coefficients. However, the shapes of the curves in the graphs can be used to detect this kind of attack; note the irregularities on the right sides of the top-right and bottom-left graphs, as compared to the results for non-attack images. These irregularities reflect the small number of correctly guessed and artificially emphasized watermark chips used to enhance correlation.

6. CONCLUSION

We presented techniques for hardening image watermarks against cryptanalytic adversaries. We did not address the more commonly studied signal-processing distortions or "presentation" attacks [2]. Though the true security of SS watermarking is not certain, our methods attempt to maximize the potential of such methods.

7. REFERENCES

[1] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "A secure, robust watermark for multimedia," in *1st Info. Hiding Workshop*, Univ. of Cambridge, England, May 1996.

[2] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *2nd Info. Hiding Workshop*, Portland, OR (USA), Apr. 1998.

[3] M. K. Mıhçak, R. Venkatesan, and M. Kesal, "Cryptanalysis of discrete-sequence spread spectrum watermarks," in *5th Info. Hiding Workshop*, Noordwijkerhout, The Netherlands, Oct. 2002.

[4] R. Venkatesan and M. H. Jakubowski, "Image watermarking with better resilience," in *ICIP 2000*, Vancouver, BC (CA), Sept. 2000.

[5] J.-P. M. G. Linnartz and M. van Dijk, "Analysis of the sensitivity attack against electronic watermarks in images," *LNCS*, vol. 1525, pp. 258–272, 1998.

[6] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *ICIP 2000*, Vancouver, BC (CA), Sept. 2000.

[7] M. K. Mıhçak, R. Venkatesan, and M. Kesal, "Watermarking via optimization algorithms for quantizing randomized statistics of image regions," in *40th Allerton Conf.*, Monticello, IL, Oct. 2002.

[8] D. Kirovski and H. S. Malvar, "Spread-spectrum watermarking of audio signals," *IEEE Trans. on Signal Processing*, vol. 51, pp. 1020–1033, Apr. 2003.

[9] M. Holliman and N. Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes," *IEEE Trans. on Image Processing*, vol. 9, no. 3, pp. 432–441, Mar. 2000.

[10] D. Kirovski and F. A. P. Petitcolas, "Blind pattern matching attack on watermarking systems," *IEEE Trans. on Signal Processing*, vol. 51, no. 4, pp. 1045–1053, Apr. 2003.

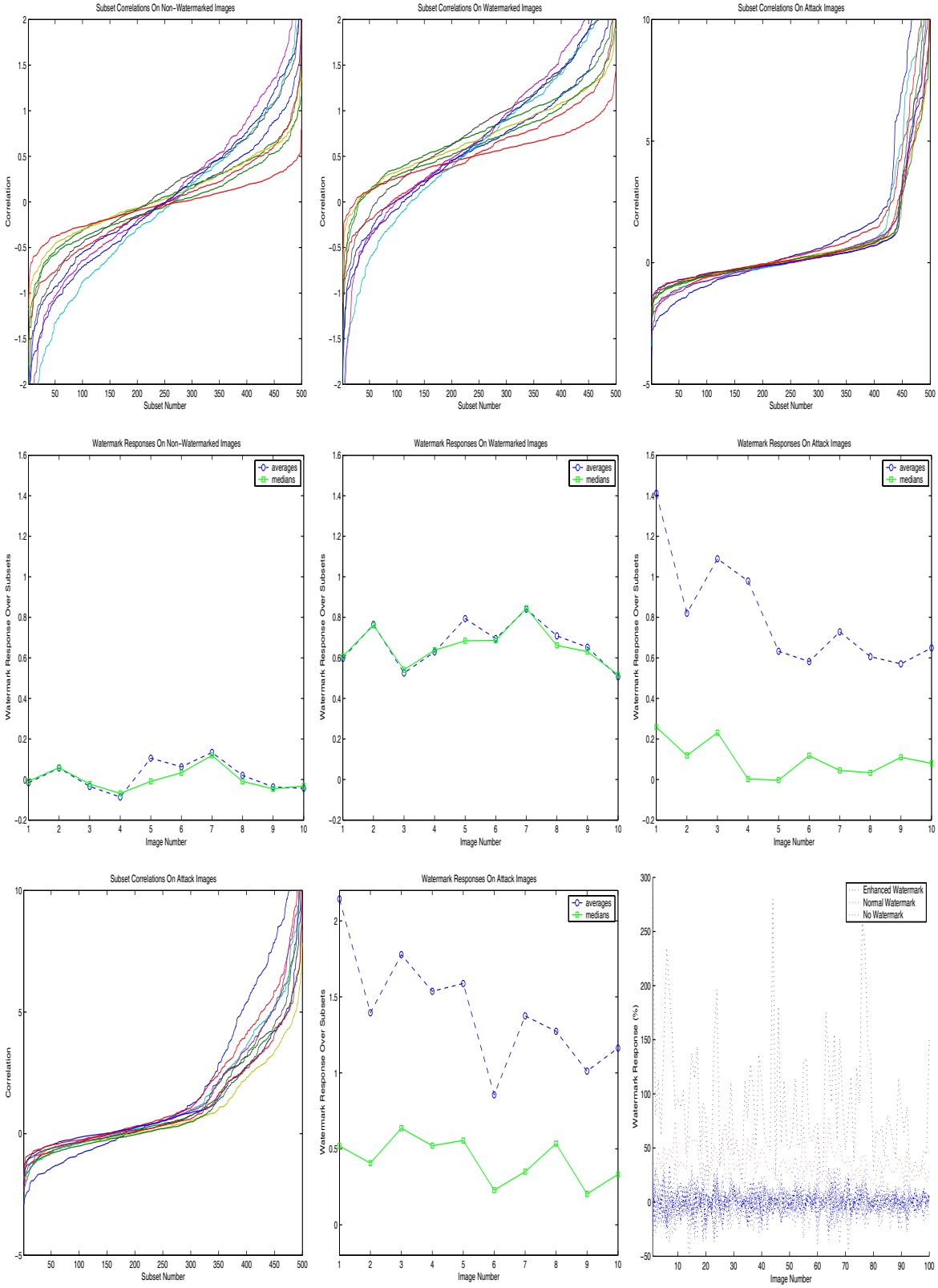

Fig. 1. **Top:** Sorted correlations over 500 watermark subsets in each of 10 images. From left to right, the graphs show results for non-watermarked, watermarked, and attack images. **Middle:** Averages and medians of subset correlations on 10 non-watermarked, watermarked, and attack images. **Bottom:** The two rightmost graphs show averages and medians of subset correlations in 10 images on the threshold of failed detection. The leftmost graph shows normal and enhanced WM responses for 100 images, each watermarked and then distorted by medium JPEG compression and the StirMark default attack.