REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
21-02-2007

2. REPORT TYPE
Final Performance

3. DATES COVERED *(From - To)*
01-06-2005 - 31-08-2006

4. TITLE AND SUBTITLE
Estimation of Information Hiding Algorithms and Parameters

5a. CONTRACT NUMBER
FA9550-05-1-0440

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dr. Scott A. Craver

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Research Foundation of SUNY at Binghamton
4400 Vestal Parkway East
Binghamton, NY 13902

8. PERFORMING ORGANIZATION REPORT NUMBER
1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
USAF,
Air Force Office of Scientific Research
875 North Randolph Street, RM3112
Arlington, VA 22203
Dr Robert Herklotz/NM

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR/PK3

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approve for Public Release: Distribution Unlimited

AFRL-SR-AR-TR-07-0102

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The goal of this project is the development of a system of useful tools for reverse-engineering covert channels and information hiding systems. This includes new algorithms for detection and estimation of certain hiding systems, and the statistical artifacts they leave behind. One of our main observations is that severe false alarms leak a great deal of information about a watermark detector algorithm. The tendency to admit certain extreme false alarms, a property we call superrobustness, is an exploitable weakness in a detector.
Using the techniques developed in this project, we participated in and won an international contest to defeat an unknown watermarking system. We did this by reverse-engineering the algorithm through the yes/no output of the watermark detector. Likewise, the participation in the contest spurred new research, in particular the "noise caliper" technique of plumbing a detection region by growing false positives.

15. SUBJECT TERMS
Information hiding, reverse-engineering, steganography, steganalysis, watermarking

16. SECURITY CLASSIFICATION OF:
a. REPORT
b. ABSTRACT
c. THIS PAGE

17. LIMITATION OF ABSTRACT

18. NUMBER OF PAGES

19a. NAME OF RESPONSIBLE PERSON

19b. TELEPHONE NUMBER *(Include area code)*

FINAL PROJECT REPORT

Estimation of Information Hiding Algorithms and Parameters

Award No. FA9550-05-1-0440

Awarded to THE RESEARCH FOUNDATION OF
STATE UNIVERSITY OF NEW YORK (CAGE code 3GRK1)
SUNY AT BINGHAMTON
4400 Vestal Pkwy E
Binghamton NY 13902-6000

Principal Investigator: Dr. Scott A. Craver
Assistant Professor, Department of Electrical and Computer Engineering
University of Binghamton
Binghamton, NY 13902-6000

Objectives

The goal of this project is the development of a system of useful tools for reverse-engineering covert channels and information hiding systems. This includes new algorithms for detection and estimation of certain hiding systems, and the statistical artifacts they leave behind. We also proposed an end-to-end system implementing our various research efforts in order to assist a specialist in breaking a covert communication system given very little information. Since it is likely for steganography to be used on very large multimedia files, e.g. audio and video, there are substantial issues to be addressed on the implementation end of such a system as well as the theoretical end.

Our project followed two tracks: as we conduct basic research in detection and estimation which comprises the primary objective of this project, we also pursued a test bed for implementing, comparing and demonstrating new algorithms. Initially we focused on audio steganalysis, but our theoretical results were generic, and for external reasons we aimed our efforts at image steganalysis.

Status of effort

Since the project's inception, we have laid the groundwork for an audio analysis testbed for testing our detection methods. This effort consumed the first few months of the project, but became less relevant as we applied our methods to image steganalysis.

Starting in February of 2006, we became aware of an academic challenge, the international Break Our Watermarking System (BOWS) contest. We decided that our methods could be tested in this environment, and we employed our existing theoretical methods to break it. As a result, our research team won the contest, achieving the highest quality images that passed the challenge's detector.

Both before and because of this contest participation, we have developed techniques for reverse-engineering detectors using a detector output, and by carefully constructing false-alarm images. Our existing research suggested that severe false alarms are very informative about a watermarking algorithm.

Finally, we have begun a project to taxonomically classify watermarking algorithms based on common approaches described in the peer-reviewed literature. We intend to provide a basis set of features to test, in order to distinguish between, say, frequency-domain and time-domain steganography, or between additive or multiplicative embedding. This is to aid in reverse-engineering; however, within this model we have developed some watermarking techniques that are not easily classified or attacked from this framework. In particular we developed steganographic and watermarking algorithms based on non-linear time warping of audio signals.

Accomplishments/New Findings

Our findings and accomplishments are:

- Development of techniques to reverse-engineer a watermarking algorithm based on experimental queries to a detector. Our generic technique has been presented as an invited lecture at the Wavila Challenge workshop in Geneva, Switzerland, and is to be presented at the SPIE Electronic Imaging conference in San Jose in January of 2007.
- Our team defined and explored the new concept of "superrobustness," the property of a detector to admit certain severe false alarms---which in turn leak information about the detector. Superrobustness is a valuable tool for an attacker, and a property that must be avoided in a secure detection algorithm.
- Our team used our techniques in the real world to win the international Break Our Watermarking System (BOWS) contest, achieving the highest-quality attacks over all participants. We feel that this justifies the theoretical direction of our research, and shows its practical value as well. A paper describing our methods is to be presented at a special session of SPIE in January of 2007.
- We developed detection methods for fingerprinting based on time warping. This contributes a new method of embedding to the field of watermarking, and if may provide a simple and reliable way to tag audio communications in the field. As part of our larger project, it is relevant to our greater plan of reverse-engineering systems to understand how such an embedding scheme could be detected.

The Noise Calipers Technique

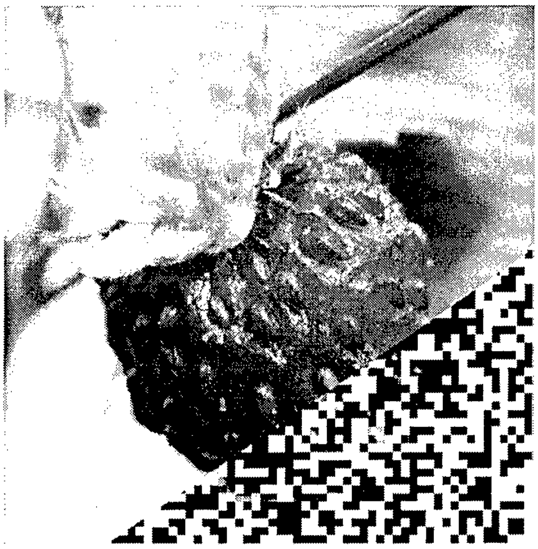

Figure 1: A challenge image from the BOWS contest, superimposed with one of our experimental attacks.

Suppose that we have a watermark detector, any generic detector that we want to reverse-engineer. We can attempt to submit experimental images, whose output will help us deduce the algorithm's inner workings. This *operational information leakage* is difficult to avoid, even if the algorithm itself can be kept secret.

In a more recent challenge, the PI and his students had three months to reverse-engineer and break an image watermarking system. On the right is one of the watermarked images, superimposed with an experimental image. This attack exploited what we now call *super-robustness* of watermarking systems. Watermark detectors sometimes admit extreme false positives, which leak information about the algorithm. Such a severe change as illustrated should break a watermark, but it won't break watermarks that are embedded in 8-by-8 pixel blocks. Hence the mark's survival tells us about the detector.

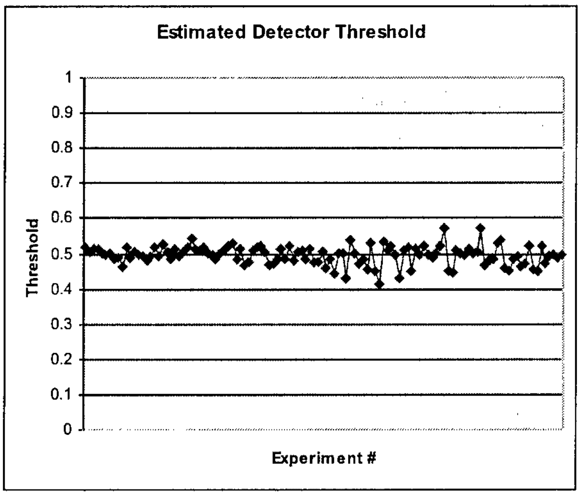

Figure 2: a detection threshold of 0.5, estimated by an average of 1016 detector queries per experiment. With 500 detector features, this detector has an asymptotic false alarm rate of 2.39×10^{-33}.

Extending these results, we have designed general techniques to force a watermark detector to leak specific information about its secret algorithm. If the watermark uses normalized correlation in its detection, we can deduce parameters such as the number of watermark features and the watermark detector threshold. In an interesting experiment, we were able to estimate the false alarm rate of a detector by querying it 1,000 times---even though the false alarm rate was on the order of 10^{-33}. This exploits the same super-robustness principle: we iteratively grow long noise vectors under which the watermark remains detectable, and when they grow to sufficient length they tell us properties of the detector, such as a normalized correlation threshold. A similar experiment tells us the number of features used.

The technical details of the BOWS contest and the Noise Calipers techniques are to be published in two separate papers, drafts of which are attached.

Personnel Supported

Scott A. Craver, principal investigator.
Assistant Professor, Department of Electrical and Computer Engineering
University of Binghamton

Idris Atakli, graduate student
Jun Yu, graduate student.

Idris Atakli is a master's candidate who is hired primarily to contribute to the software implementation of our experimental test bed. Jun Yu is a Ph.D. student whose primary focus is theoretical development ,of hypothesis tests and new information hiding systems.

Publications:

We presently have three publications and one invited talk from this research, two to appear in 2007. They are:

1. S. Craver, J. Yu and I. Atakli, "How we Broke the BOWS Watermark." In Proceedings of SPIE, Security and Watermarking of Multimedia Contents IX, to appear January 2007.
2. S. Craver and J. Yu and I. Atakli, "Reverse-engineering a Watermark with False Alarms." In Proceedings of SPIE, Security and Watermarking of Multimedia Contents IX, to appear January 2007.

3. S. Craver and J. Yu, "Fingerprinting with Wow," In Proceedings of SPIE, Security and Watermarking of Multimedia Contents VIII, January 2006.
4. (invited talk) S. A. Craver, "Noise Calipers: a Technique for Reverse-Engineering Watermarks." In WaCha, the 2nd Wavila Challenge Workshop, Geneva, Switzerland, Nov 2006.

We expect to submit at least one journal article in the near future, summarizing our findings more rigorously.

Interactions/Transactions:

We have provided consultative and advisory functions to AFRL in Rome, NY. As part of a summer faculty program the PI has performed work in audio steganalysis relevant to this award.

New inventions or patent disclosures:

There have been no invention or patent disclosures.

Honors/Awards:

The research team (consisting of the PI and his two students) have won the first International BOWS contest, using research techniques developed as part of this project.

How we broke the BOWS watermark

Idris Atakli, Scott Craver and Jun Yu[a]

[a]Binghamton University, Binghamton, NY, USA

ABSTRACT

From December 2005 to March of 2006, the Break Our Watermarking System (BOWS) contest challenged researchers to break an image watermark of unknown design. The attacked images had to possess a minimum quality level of 30 dB PSNR, and the winners would be those of highest average quality over three images. Our research team won this challenge, employing the strategy of reverse-engineering the watermark before any attempts to attack it in earnest. We determined the frequency transform, sub-band, and an exploitable quirk in the detector that made it sensitive to noise spikes. Of interest is our overall methodology of reverse-engineering through severe false alarms, and we introduce a new concept, "superrobustness," which despite its positive name is a security flaw.

Keywords: watermarking,reverse-engineering,steganalysis,oracle attacks

1. INTRODUCTION

From December 15, 2005 to March 15, 2006, the Break Our Watermarking System (BOWS) contest challenged researchers to break an unnamed image watermarking system, by rendering a watermark undetectible in three separate images.[1] Doctored images had to meet sufficient quality standards to be considered successfully attacked, and all three had to be successfully attacked to qualify for the prize. The prize went to the team which achieved the highest quality level by March 15, 2006. Our team from Binghamton University was fortunate enough to have the highest quality level on the 15th, winning the prize of 300 dollars and a digital camera.

1.1. Goals of the Contest

The specific goal was to alter three separate 512-by-512 grayscale images, so that all three fail to trigger a watermark detector. Relative to the original, the quality of a successful attack had to exceed 30db PSNR, and the winner was the team who achieved the highest PSNR – although the average PSNR of all three was computed from the average MSE:

$$PSNR = 10\log_{10}\frac{255^2}{\frac{1}{3}\sum_{k=1}^{3}\frac{1}{262144}\sum_{i,j}(I_k[i,j]-J_k[i,j])^2}$$

The three images are illustrated below.

1.2. A Comparison to the SDMI Challenge

It is inevitable to compare this watermarking challenge to a previous one, the SDMI challenge of 2000.[2] In this contest, the Secure Digital Music Initiative (SDMI) challenged "hackers" to break four audio watermarking technologies and two supplemental digital signature technologies. Several teams passed the preliminary phase of the challenge for all four watermark technologies, although representatives of Verance, Inc, say that their own watermarking technology was not broken by any participant.[3–5]

There are some specific facets of the two contests which are worth comparing and contrasting:

- The SDMI challenge was three weeks, and the BOWS contest was three months. The organizers of the BOWS contest also arranged a post-challenge period of three months after the watermarking algorithm was made public.

E-mail: scraver@binghamton.edu

Figure 1. The three images from the BOWS contest.

- The BOWS contest oracle took seconds, whereas the SDMI challenge oracle often took hours. The SDMI challenge oracle presumably involved watermark tests and listening tests by a human being, and the delay between submission and results could take varying amounts of time.[3]
- The BOWS contest oracle's behavior was well-defined. For example, the minimum quality metric of the BOWS contest is explicitly defined at 30.00 dB PSNR, and the detector was automated: it would accept any properly formatted image, even a blank image or wrong image. This allowed reverse-engineering attacks of the variety we employed. In the SDMI challenge, the human's behavior was unclear. Some researchers submitted properly signed TOC files to the signature verification oracle, and received a rejection from the oracle; they could not tell if the oracle was defective or if the human had a policy of rejecting trivial submissions.[3]
- The BOWS oracle told the user both the detector output and the image quality (which could also be computed at the user's end, since the quality metric was explicitly defined.) The SDMI oracle would not reveal the specific reason for a rejection: either the detector found the watermark or the audio quality was poor, or both.
- The SDMI challenge provided watermarked and unwatermarked samples; the BOWS contest only provided watermarked samples with no reference images. It is likely the SDMI challenge would not be won if there were no unwatermarked samples. With few possible opportunities to query an oracle, one could only analyze the difference between before and after clips, and possibly look for anomalies in the watermarked clips.

One major difference is that the SDMI challenge was not amenable to oracle attacks. The BOWS contest allowed potentially millions of queries in serial, versus hundreds of queries in serial over the lifetime of the SDMI challenge.

1.3. Results

Our team achieved a PSNR of 39.22 by the deadline, winning the contest.[1] We achieved this first by reverse-engineering the algorithm from experimental oracle submissions, and then by exploiting an observed sensitivity to noise spikes in the feature space. One John Earl of Cambridge University deserves special recognition for rapidly increasing his PSNR results just before the deadline. If we extrapolate from his submissions, he could have beaten us were the contest just a few hours longer.[1]

The algorithm was then revealed to be one published by M. L. Miller, G. J. Doerr and I. J. Cox in 2004.[6] We confirmed that our guess for the image transform and subband matched the paper, while their detector partially explained the weakness we used to defeat the detector.

In the subsequent three-month challenge with a known algorithm, Andreas Westfeld achieved an astonishing 58.07 dB, a quality level exceeding even the minor error incurred by digitizing the image in the first place.[7]

2. REVERSE ENGINEERING

The reverse-engineering of the algorithm was performed in two steps: reverse-engineering the feature space by submitting experimental images, and reverse-engineering the sub-band of the feature space by carving out selected portions of the feature space once it was known.

2.1. Determining the Feature Space

In retrospect, the feature space was very clear given the obvious block artifacts in the images (see figure 2,) but we had to be certain. Our deducing of the feature space followed the following principle: *For each suspected image feature space, find a severe attack that leaves that feature space untouched.*

The philosophy behind this attack is that a severe modification of an image should break all other watermarks, providing a test with high selectivity. Curiously, our approach was the opposite of the contest goals: we were to damage the image quality as much as possible, while failing to remove the watermark.

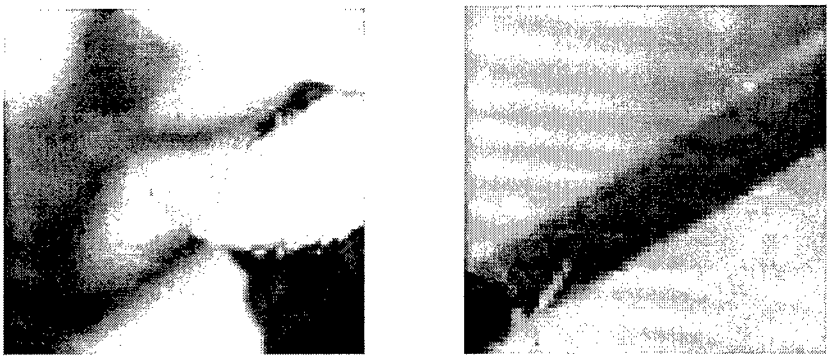

Figure 2. Some visible artifacts in the image, suggesting either severe compression or a block-based watermark.

For example, we would invert the image in luminance (this broke the watermark,) add DC offsets to the pixels (did not break the watermark,) mirror-flip the image, or blocks of the image, et cetera. After a small set of tests, we submitted a test image which convinced us that the watermark was embedded in 8-by-8 DCT coefficients (figure 3.) Here, each 8-by-8 block was given the worst possible DC offset without changing any AC coefficients in an FFT or DCT of the block. This preserved the watermark, but reduced image quality to 3.94 dB PSNR.

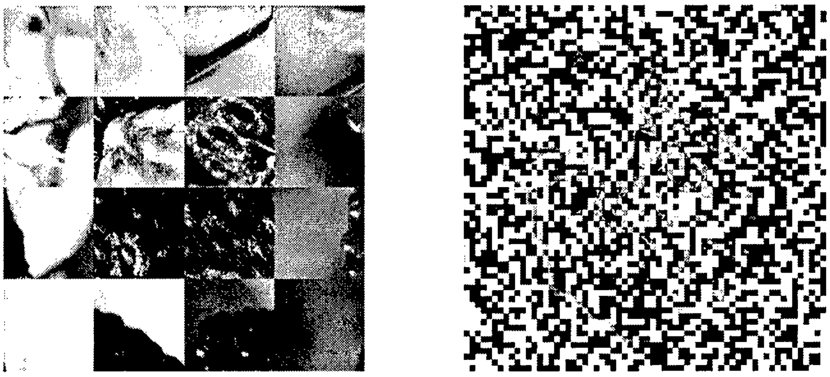

Figure 3. Some severe transformations of an image. Left, a shuffling of blocks aimed at preserving block-based FFT magnitudes. Right, a severe addition of 8-by-8 DC offsets. This preserved the watermark with only 3.94 dB PSNR.

Our attacks included flipping signs of common transform coefficients, reasoning that if a watermark is embedded multiplicatively rather than additively, then only changes in magnitude would damage the watermark. We found that changes in sign broke the watermark. Thus this approach could tell us not only what feature space was used, but also what kind of detector might be used. The precise dividing line between a choice of feature space and choice of detector is unclear: if only magnitude information is used in a detector, is this a restriction of the feature space?

2.2. Determining the Sub-band

Once we were convinced that the watermark was embedded in 8-by-8 AC DCT coefficients, we then submitted images with bands removed from each block. We took the largest bands we could remove without hurting the watermark, and computed the complement of their union. The result, illustrated below, matches the subband used in the watermarking algorithm.[6]

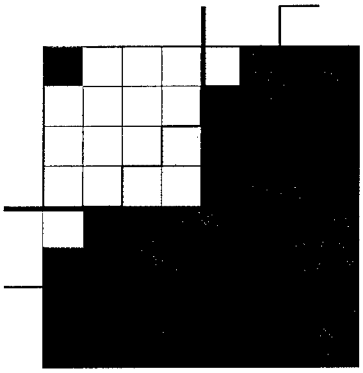

Figure 4. The sub-band (white,) determined by striking out diagonal and gnomonic subbands of the DCT coefficient matrix. The DC coefficient was ruled out by the DC-offset experiment.

In this process we were guided by knowledge of common watermarking algorithms. Watermarks have commonly resided in low-frequency and middle-frequency bands, a tactic described in the seminal paper by Cox *et al,* in order to make the watermark difficult to separate from the image content.[8] We expect common subbands to be upper-triangular regions, and square regions in the DCT coefficient matrix. Hence, our attacks struck out lower-triangular sections of the matrix, and gnomonic sections. Taking the union of the largest lower-triangular and the largest gnomonic section, we found the largest "pattern" we could remove without damaging the watermark, the largest region of geometric significance.

A second, parallel attempt at determining the sub-band focused on eliminating DCT coefficients arbitrarily until the watermark broke. This produced a contradictory result discussed in the next section: far fewer coefficients would need to be damaged to eliminate the watermark, than our sub-band estimates (and later, the published watermarking algorithm) would suggest.

3. BREAKING THE WATERMARK

Once we determined the secret algorithm, we could focus our work directly on the feature space in hopes of inflicting maximum damage. Our initial approach was to experimentally damage DCT coefficients until the watermark was removed, and then develop a working set of coefficients and multiplier that incurred minimum damage.

We did not, however, adopt any sophisticated sensitivity attack, and performed only thousands of oracle queries, by hand.

3.1. A Curious Anomaly

Early on, we discovered by experiment that an unusually small amount of damaged coefficients could render the watermark undetectable. We tried the following algorithm:

```
1. Sort the DCT coefficients by magnitude.
2. Let $k \leftarrow 200, d \leftarrow 50$
3. while $d >= 1$:
    (a) Eliminate the $k$ largest coefficients.
    (b) If the watermark remains, $k \leftarrow k + d$.
    (c) If the watermark is removed, $k \leftarrow k - d; d \leftarrow \lfloor d/2 \rfloor$.
```

Initially, the elimination rule set each coefficient to 0. Later we improved our result by amplifying rather than zeroing each coefficient, for example multiplying each of k coefficients by 3. Doing so, we found that a few hundred coefficients were enough to destroy the watermark in all three images.

However, our previous analysis gave us a sub-band of 49152 coefficients (A 512-by-512 grayscale image, 4096 8-by-8 blocks, 12 AC coefficients taken per block.) It seems unlikely that a few hundred zeroed coefficients would destroy a signal of 49152 samples. We suspected that some aspect of the detector made it sensitive to noise spikes in the right locations.

The watermark detector, which we could not guess, was revealed to be a Viterbi decoder that extracted a long message from the 49152 coefficients, and compared that to a reference message.[6] We believe that this is the reason that the resulting detector could be derailed by a carefully chosen set of noise spikes. Whatever the cause, we then sought to reduce the number of samples we zeroed or amplified.

3.2. The Rest of the Algorithm

Once we could eliminate the watermark by amplifying the k highest-magnitude coefficients, we then reduced the set size incrementally:

```
1. Choose $k$ and $D$, so that multiplying the $k$ highest-magnitude DCT coefficients by $D$ destroys the watermark.
2. For $j = 1 \cdots k$:
    (a) Reinstate the original value of coefficient $j$
    (b) If the watermark becomes detectable, return coefficient $j$ to its damaged state.
```

We therefore wiped from our list of damaged coefficients all those which were not needed to remove the watermark. The resulting set was surprisingly small, reducing hundreds of coefficients to less than ten, per image, which we then reduced by hand, increasing the amplification factor.

Figures 3.2 and 3.2 illustrate the reduction process for images 1 and 2: raising the value D from 3 to 3.55, we found that some coefficients could be removed from the attack set for an overall increase in PSNR. We stopped when removing the next coefficient required an amplification that reduced the PSNR.

(Coefficient numbers)*amplification	PSNR
(12,69,107,127,132,140,141)*3.4	37.53dB
(12,69,127,132,140,141)*3.4	38.19dB
(12,69,127,140,141)*3.4	38.97dB
(12,69,127,141)*3.55	39.67dB

Table 1. Successive attacks for image 1. By amplifying a few AC coefficients by the right value, the detector will fail.

	AC coefficients								
Multiplier	206	216	223	236	279	286	310	314	316
3.1	R							R	
3.2	R		R					R	R
3.2	R		R	R		R		R	R
9.5	R		R	R	R	R		R	R

Table 2. Successive attacks for image 2. We start with 9 AC coefficients. 'R' denotes that the coefficient is no longer needed when the gain increases.

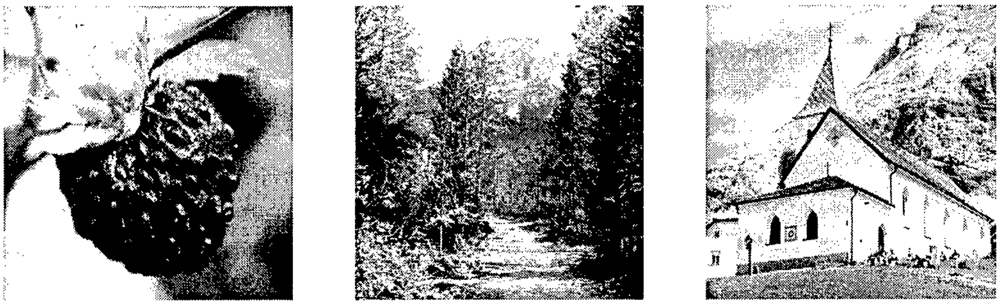

Figure 5. The three images after the attack. Note the obvious block artifacts.

4. SUPERROBUSTNESS AND ORACLE ATTACKS

For purposes of reverse-engineering, we adopted the strategy of making severe changes that will not effect one type of watermark. For example, if a watermark is embedded in block AC DCT coefficients, a random DC offset to each block has no effect on the watermark, even though it is an extreme amount of noise which renders the image useless.

Watemarks are typically designed so that the watermark remains detectable even if the image is reasonably degraded. A layperson might also expect the watermark to become undetectable if image distortion is ridiculously severe—however, some carefully chosen forms of degredation have no effect whatsoever on a detector, for example if they miss the feature space.

We call this property *superrobustness:* the property of a watermarking algorithm to survive select types of quality degredation far beyond what any reasonable person would expect. If we can render an image unrecognizeable and maintain the watermark, then we have found an attack to which it is superrobust.

This may seem like an admirable property for a watermark to possess, but it is in fact a security weakness. There is no need for a watermark to survive complete destruction of an image, and in doing so it only gives information to an attacker. The destructive operation is likely to destroy all other types of watermarks, giving an attacker a reverse-engineering tool with high selectivity.

4.1. Oracle Attacks Exploiting Superrobustness

The superrobustness attack exploits severe false alarms, by inflicting noise that would break most other watermarks. As human beings, we may use expertise and common sense in choosing such noise. A computer is not so gifted, and must determine severe false alarms blindly.

We developed an oracle attack based on the principle of superrobustness, in which an incremental amount of noise is added to an image as long as the watermark remains detectable.[9] We find that the noise component within the feature space quickly converges to the detection region boundary and stays there; several properties of this noise can be used to estimate the feature space size and detector threshold, if we guess in advance what type of detector is used. However, this takes far more oracle queries than we needed to break the more complicated BOWS watermark.

4.2. Mitigation Strategies

There is a simple strategy to prevent super-robustness attacks: prevent severe inputs to the detector. This may be easier to say than do, because it is not clear how to recognize a "severe" input. We offer several recommendations to the designer of a watermark detector:

- Make every effort to normalize an image before detection.
- Place hard limits on feature coefficient magnitude, and overall energy.
- Whenever possible, give the detector access to the watermarked image for comparison's sake. For example, a watermark can contain an identifier, which can be used to look up a copy of a watermarked image. If the test image does not resemble the marked image in some rudimentary way, then the oracle should declare the watermark undetectable.
- Reject inputs that fail to match a reasonable model of the image.
- Use "cocktail watermarking," as described in,[10] and write a detector that does not reveal which watermarks are missing after an attack.
- Choose the right application. In a fingerprinting or traitor-tracing application, an attacker doesn't have direct access to the detector at all. Indeed, attackers may not even know if content is watermarked, until someone gets in legal trouble. Meanwhile, for content-control watermarks, especially those detected in consumer electronics devices like television sets or portable music players, oracle attacks will be very difficult to prevent.

Note that these are properties of a detector, not the watermark itself. Any proper watermark feature sub-space will be super-robust to some attack—namely, destroying all image data outside the sub-space—so prevention must focus on masking this effect, using a sophisticated detector.

4.3. Other Advice to the Watermark Designer

Beyond the problem of superrobustness, several other exploitable weaknesses are worth highlighting. We provide some tips to the watermark designer who wishes to avoid our attacks.

1. *Do not use a steganographic algorithm to embed one bit.* If you simply wish to watermark an image with a fixed copyright label, which is either present or not (*e.g.,* a one bit message) do not adapt a stego algorithm intended to encode arbitrary long messages.

   It seems reasonable to adapt a stego algorithm into a watermark algorithm by embedding a long reference message, and later comparing that reference message to the output of the stego-decoder. However, the improved bitrate of a steganographic algorithm is often at the expense of robustness. Using that payload to send a single bit is suboptimal.

2. *Watch for security flaws introduced by detector components.* In this case, the watermark message was extracted using a Viterbi decoder, which we feel introduced an exploitable weakness. Since each chip of the signal was detected by correlation, why not simply detected the whole watermark with a single correlation? The answer is that the algorithm was intended to decode an unknown message, rather than search for one specific message known in advance.

3. *Consider how stereotypical the feature space is.* Many, many image watermarking algorithms use block DCT coefficients, or full-frame DCT coefficients. Many use straight correlators or normalized correlators, although the BOWS watermark did not. An attacker is going to guess these right from the start.

   We are not advocating security through obscurity, by asking a designer to pick an obscure transform; however, something like Fridrich's key-dependent random image transforms might increase the guessing entropy of the concealed algorithm, at least because the key-dependent image transform parameterizes more of the algorithm as key.[11]

5. CONCLUSIONS

The BOWS contest gave us an excellent opportunity to test new attack methodologies, and to further explore the process by which watermarks are broken in the real world.

Because we reverse-engineered the watermark through the oracle, we question the value of keeping a watermark algorithm secret in practice (although for the purposes of the contest, keeping the algorithm secret was an excellent idea.) If a watermark detector can be exploited as an oracle, then the algorithm is essentially leaked; the extent of this leakage is the subject of further study.

ACKNOWLEDGMENTS

We would like to thank the organizers of the BOWS contest, and we believe that contests of this type are of great value in spurring new academic research in steganalysis and digital forensics.

REFERENCES

1. "The Break Our Watermarking System (BOWS) contest." http://lci.det.unifi.it/BOWS/.
2. Secure Digital Music Initiative, "SDMI public challenge." http://www.hacksdmi.org, Sept. 2000.
3. S. Craver, M. Wu, Liu, A. Stubblefield, B. Swartzlander, D. S. Dean, and E. Felten, "Reading between the lines: Lessons learned from the SDMI challenge," *Proc. Usenix Security Symposium*, pp. 353–363, August 2001.
4. J. Boeuf and J. P. Stern, "An analysis of one of the SDMI candidates," *Lecture Notes in Computer Science* **2137**, pp. 395–??, 2001.
5. R. Petrovic, B. Tehranchi, and J. M. Winograd, "Digital watermark security considerations," in *MM&Sec '06: Proc. 8th Workshop on Multimedia and Security*, pp. 152–157, 2006.
6. M. L. Miller, G. J. Doerr, and I. J. Cox, "Applying informed coding and embedding to design a robust, high capacity watermark," *IEEE Trans. on Image Processing* **13**, pp. 792–807, June 2004.
7. A. Westfeld, "Lessons from the bows contest," in *MM&Sec '06: Proc. 8th Workshop on Multimedia and Security*, pp. 208–213, 2006.
8. I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Image Processing* **6**, pp. 1673–1687, 1997.
9. S. Craver and J. Yu, "Reverse-enineering a detector with false alarms," in *Proc. SPIE, Security, Steganography, and Watermarking of Multimedia Contents IX, to appear*, 2007.
10. C.-S. Lu, S.-K. Huang, C.-J. Sze, and H.-Y. M. Liao, "Cocktail watermarking for digital image protection," *IEEE Trans. on Multimedia* **2**, pp. 209–224, 2000.
11. J. Fridrich, "Key-dependent random image transforms and their applications in image watermarking," *Proc. International Conference on Imaging Science, Systems, and Technology, CISST '99*, pp. 237–243, June 1999.

Reverse-engineering a detector with false alarms

Scott Craver and Jun Yu[a]

[a]Binghamton University, Binghamton, NY, USA

ABSTRACT

Inspired by results from the Break Our Watermarking System (BOWS) contest, we explored techniques to reverse-engineer watermarking algorithms via oracle attacks. We exploit a principle called "superrobustness," which allows a watermarking algorithm to be characterized by its resistance to specific distortions. The generic application of this principle to an oracle attack seeks to find a severe false alarm, or a point on the watermark detection region as far as possible from the watermarked image.

For specific types of detection regions, these severe false positives can leak information about the feature space as well as detector parameters. We explore the specific case of detectors using normalized correlation, or correlation coefficient.

Keywords: watermarking,reverse-engineering,steganalysis,oracle attacks,sensitivity attacks

1. INTRODUCTION

Sensitivity attacks, or more generally oracle attacks, have been employed for years to break watermarking systems and have become increasingly sophisticated.[1,2] These attacks seek to remove a watermark by finding a point outside the watermark detection region that is as close as possible to the watermarked image.

Sometimes, however, we are not initially interested in breaking an unknown watermark, but learning its structure, or its detection algorithm. If we have an unknown algorithm, can an oracle be used experimentally to deduce it? The answer is yes, as has been demonstrated by recent watermarking challenges.[3,4] This use of an oracle for reverse-engineering is ad-hoc, however, and based on expert knowledge. An automated watermark reverse-engineer, which uses a detector as an oracle and analyzed its output, would be very useful, and we have begun to explore how components of this sytem could be built.

The problem is that embedding algorithms can be arbitrarily complex. However, if we have some general information about a watermark's generic structure, we can then seek to reverse-engineer further details by oracle attack. For example we might assume that a watermark is feature-based, extracting a vector f of unknown features and comparing them to a reference watermark by some popular detector structure.

1.1. Motivation

This research was primarily motivated by our success in reverse-engineering the unknown BOWS watermark.[4] In this project, we took watermarked images, distorted them in severe ways, and fed them to the provided watermark detector as experimental images.

Our goal was to test if a given feature space or detector was in use, by inflicting severe changes that would not hurt certain types of watermarks. This gave us a test with high selectivity for certain feature spaces, sub-bands and detector structures. Extending this concept, we sought to explore how the false positives of a detector leak information about the underlying watermarking algorithm.

E-mail: scraver@binghamton.edu

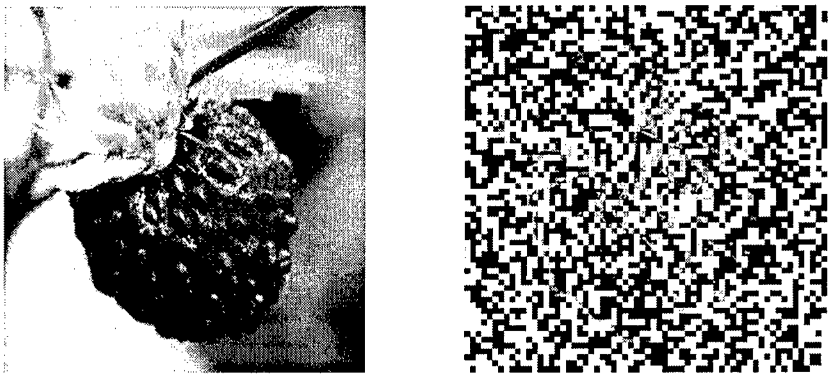

Figure 1. An image from the BOWS contest, and a severe distortion that preserves the watermark. From this experimental image we concluded that the watermark was embedded in non-DC 8-by-8 DCT coefficients.

2. SUPERROBUSTNESS AND FALSE POSITIVES

Superrobustness is the property of a watermark to survive specific extreme and absurd image distortions. Robust watermarks are of course intended to survive distortion if the image quality remains high; however, one rarely mandates the converse, that the watermark should break when the image is damaged far beyond its useful value.

Superrobustness may sound positive, but it is a security flaw. A watermark will be superrobust to operations specific to its feature space and detector, and this makes the watermark amenable to reverse-engineering, using the detector as an oracle. In other words, superrobustness results in a detector that *leaks information about the watermarking algorithm.*

2.1. Examples of Superrobustness

In the BOWS contest, we tested for several types of watermark algorithms by testing for superrobustness. Here are some properties we used to fashion experimental test images:

- If a watermark is embedded in AC frequency transform coefficients, it should be immune to DC offsets. For block-based transforms, an independent DC offset can be added to each block without hurting the watermark, although many other types of watermarks would be damaged. See figure 1 for an example.
- If a watermark is embedded multiplicatively rather than additively in a signal or a feature vector, then the signs of those features can be randomized without changing the embedded signal.
- Any watermark that uses a proper feature subset can be superrobust to the wiping out of all other information. For example, if a full-frame DCT watermark uses a sub-band of 50000 AC coefficients, erasing or randomizing all but those 50000 coefficients could render the image unreadable without hurting the mark.
- If a watermark detector uses linear correlation, an amplified signal should always trigger the detector while an attenuated signal eventually falls on the wrong side of the detection region. In contrast, a normalized correlation detector is robust to both severe amplification and attenuation.

Note that *superrobustness is a property of the detector, not of the watermark embedder.* Ultimately the detector is responsible for admitting a false positive. While a certain feature space or embedding method may suggest a superrobustness attack, a detector does not have to blindly test an attacker's experimental images. This suggests some remedies for examples of super-robustness described above: build a smarter detector, one that can recognize or rule out certain severe false positives.

2.2. Generic Superrobustness

Suppose we have a generic watermarking algorithm, whose detector follows this basic framework:

1. Extract a vector of n features $f = \{f_1, f_2, \cdots f_n\}$ from the image.
2. Compare this vector to a watermark vector, for example by normalized correlation:

$$\frac{f \cdot w}{\|f\|\|w\|} \lesseqgtr \tau \in (0,1)$$

3. Report a successful detection if the correlation exceeds a chosen threshold.

In the case of normalized correlation, the detection region is a cone, with its tip at the origin, its axis equal to the watermark vector w, and its interior angle determined by the threshold τ.[5] Specifically, the cone angle is $\theta = \cos^{-1} \tau$.

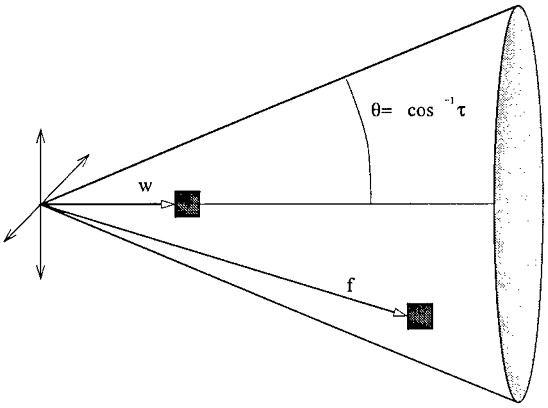

Figure 2. A normalized correlation detection region with threshold τ.

A key fact about this and may other detection regions is that it is unbounded. If one can efficiently travel within it, then one can find images that set off the watermark detector that are very distant from the original. This gives us a means to construct severe false positives, from a starting point within the detection region: gradually add noise to an image, as long as the detector continues to recognize the watermark.

3. MEASURING A DETECTION REGION

3.1. Noise Snakes

Our generic approach to generating false positives is to grow a "noise snake" using incremental uniform noise vectors. We do not know the watermark vector, hence the direction that we should be traveling; however, with an unbounded detection region we expect that an expanding noise vector will move outward into distant climes of the detection region, providing some useful information about its shape and orientation.

Our noise snakes are constructed via the following algorithm:

1. Start with test image I, treated here as a vector.
2. Initialize our snake vector to $J \leftarrow I$.
3. Do for $k = 1, 2, \cdots K$:

(a) Choose a vector uniformly over the n-dimensional unit hypersphere $\mathbb{S}_n$.
This can be accomplished by constructing an n-dimensional vector X of i.i.d.$\mathcal{N}(0,1)$ Gaussians, and scaling the vector to unit length.

(b) Choose an appropriate scaling factor α, which for normalized correlation is ideally proportional to the length of J.

(c) If $J + \alpha X$ still triggers the watermark detector, $J \leftarrow J + \alpha X$.

(d) Else, discard X and leave J unchanged.

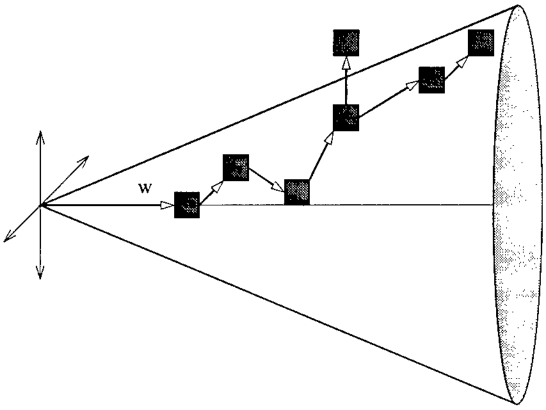

Figure 3. The construction of a noise snake.

We have observed that in high dimensions, a noise snake quickly converges to the detection region boundary, and grows gradually outward. Thus a pair of these snakes, constructed separately, can provide useful information about the detection surface.

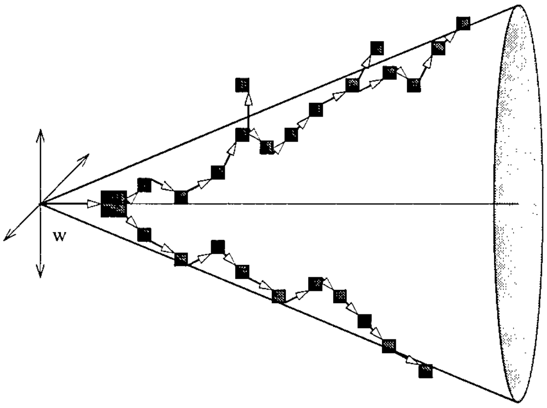

Figure 4. A pair of constructed noise snakes, used to plumb the detection cone.

3.2. The Growth Rate of a Noise Snake

When extending a noise snake by adding a uniform noise vector, we must wrestle with two conflicting factors: large noise vectors are more likely to drop us out of the detection region, but small noise vectors contribute little length per iteration. The optimal growth factor α is the one that maximizes the expected increment $\|\alpha X\| \cdot \Pr[\delta(J + \alpha X) = 1]$.

In the specific case of snakes on a cone, the growth factor α is proportional to the length of the snake as it grows. This can be seen by a simple and obvious geometric argument: the cone is congruent to scaled versions of itself. Thus if there is an optimal length α to extend a snake of length 1, then $M\alpha$ is optimal to extend a snake of length M. We need only determine the appropriate α for a snake of unit length.

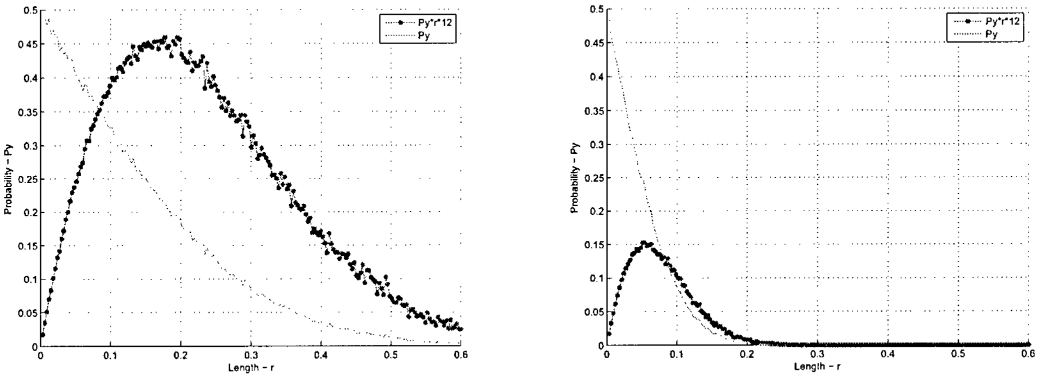

Figure 5. Optimal growth factors for a snake of unit length. On the left, a cone angle of $\pi/3$ and $n = 1000$. On the right, $\pi/3$ and $n = 9000$.

The good news is that theoretically, the growth rate is exponential in the number of queries. The bad news is two-fold: first, the best growth factor depends on the dimension and cone angle, which we do not know. Secondly, the best growth rate is nevertheless slow: For feature sets in the thousands, the best α ranges from 0.16 to 0.06. For larger feature sets, growth is small. We find in our experiments that for realistic feature sizes, a snake of useful length requires a number of queries roughly proportional to the dimension n.

3.3. Information Leakage from Snakes

We now explain how noise snakes can be used to estimate parameters of a watermark detector. First, we establish a fact about uniform noise vectors.

LEMMA 3.1. *If W is chosen uniformly over the unit n-sphere $\mathbb{S}_n$, and v is an arbitrary vector, the probability* $\Pr[W \cdot v > \cos\theta]$ *is*

$$\frac{\mathcal{S}_{n-1}}{\mathcal{S}_n} \int_0^\theta \sin^{n-2} \rho d\rho$$

...where $\mathcal{S}_n$ is the surface volume of $\mathbb{S}_n$.

Proof. Since W is uniform, the probability of any subset of $\mathbb{S}_n$ is proportional to its measure. Our subset $\mathbb{S}_n$ is rotationally symmetric about v, so let us integrate over the v axis: consider point $t \in [-1, 1]$ representing the v component of the hypersphere. For each t we have a shell of radius $r = \sqrt{1 - t^2}$, contributing a total hypersurface measure of $\mathcal{S}_{n-1} r^{n-2} \sqrt{dt^2 + dr^2}$. For example, a sphere in three dimensions is composed of circular shells, and each a circular shell has contribution $2\pi r\sqrt{dt^2 + dr^2} = \mathcal{S}_2 r^1 \sqrt{dt^2 + dr^2}$. The sphere portion with angle beneath θ is

$$\begin{aligned}
\text{Area} &= \int_{r=0}^{r=\sin\theta} \mathcal{S}_{n-1} r^{n-2} \sqrt{dt^2 + dr^2} \\
&= \int_{r=0}^{r=\sin\theta} \mathcal{S}_{n-1} r^{n-2} dr \sqrt{1 + \frac{dt^2}{dr^2}} \\
&= \int_{r=0}^{r=\sin\theta} \mathcal{S}_{n-1} r^{n-2} dr \sqrt{1 + \frac{r^2}{t^2}} \\
&= \int_{r=0}^{r=\sin\theta} \mathcal{S}_{n-1} r^{n-2} \frac{dr}{t}
\end{aligned}$$

...since $2tdt + 2rdr = 0$. Substituting $r = \sin\rho$, we get $\frac{dr}{t} = d\rho$ and

$$\text{Area} = \mathcal{S}_{n-1} \int_0^{\theta} \sin^{n-2} \rho d\rho$$

And we divide by $\mathcal{S}_n$ to get the probability of hitting that region. □

The area of a unit hypersphere $\mathbb{S}_n$ is

$$\mathcal{S}_n = \begin{cases} \frac{2^{(n+1)/2}\pi^{(n-1)/2}}{(n-2)!!} & \text{for n odd} \\ \frac{2\pi^{n/2}}{(\frac{1}{2}n-1)!} & \text{for n even} \end{cases}$$

The surface fraction $C_n = \mathcal{S}_{n-1}/\mathcal{S}_{n-2}$ therefore has a closed form[6]

$$C_n = \begin{cases} \frac{1}{2}\frac{(n-2)!!}{(n-3)!!} & \text{for n odd} \\ \frac{1}{\pi}\frac{(n-2)!!}{(n-3)!!} & \text{for n even} \end{cases}$$

It might surprise the reader to know that the surface volume of a unit hypersphere actually decreases with n, for $n > 7$.

Altogether, this means that if $\theta < \frac{\pi}{2}$, then the probability drops exponentially with n. This is the "equatorial bulge" phenomenon in high dimensions: as the number of dimensions n gets large, the angle between two uniformly chosen direction vectors will be within ϵ of $\pi/2$ with high probability.

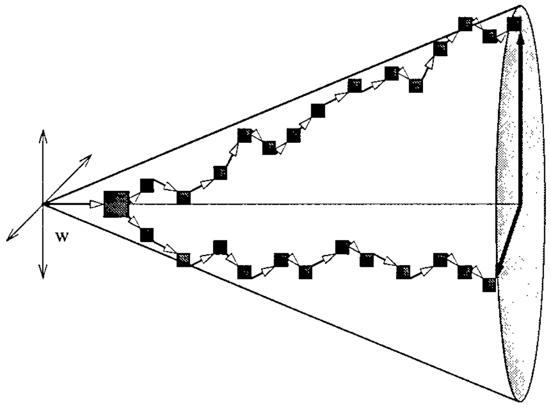

Figure 6. Lemma: two independently generated snakes have approximately perpendicular off-axis components.

COROLLARY 1. *Two independently generated noise-snakes have off-axis components S and T whose angle is very close to $\frac{\pi}{2}$.*

Proof. The density function of the set of noise-snakes is rotationally symmetric about the watermark axis. This is because each component, a uniform vector, has a rotationally symmetric density, and because if s is a valid noise snake, so is Ts, where T is any rotation holding the cone axis constant.

Because of this, the probability $\Pr[S \in F] = \Pr[TS \in TF]$. If we subtract w and then normalize each snake, the symmetric distribution implies that $W = (S-w)/\|S-w\|$ is uniformly distributed over the unit $n-1$ sphere. □

This observation gives us a simple means to estimate the cone angle from two constructed noise snakes X and Y of equal length r. Using trigonometry, $(\sqrt{2}r\sin\theta)^2 = r^2 + r^2 - 2rr\cos\phi$ where ϕ is the angle between the snakes (see figure 7.) Rearranging, we get:

$$\sin^2\theta = 1 - X \cdot Y$$

We can then estimate the cone angle and detector threshold by constructing two snakes of sufficient length, and computing their dot product.

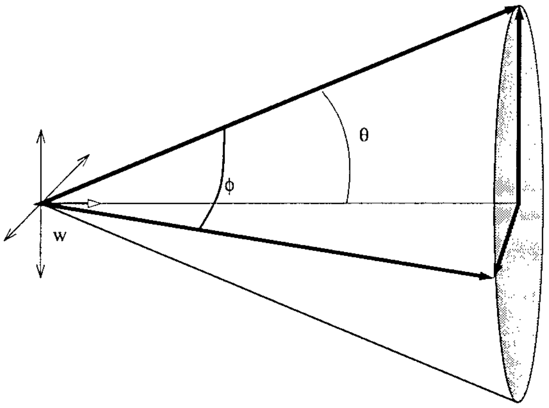

Figure 7. The relation between the cone angle and the dot product of two independently generated noise snakes.

3.4. Extracting the Size of the Feature Space

Once we have a decent estimate for the cone angle, we can use another technique to estimate the feature space size, a more useful piece of information. To accomplish this, we use the detection oracle again to deduce the error rate under two different noise power levels.

If we have a watermark vector w which falls squarely within the detection cone, and add a uniform noise vector r, the probability of detection is

$$\Pr[\delta = 1] = \frac{\mathcal{S}_{n-1}}{\mathcal{S}_n}\int_0^{\psi} \sin^{n-2} x dx \qquad (1)$$

$$\psi = \theta + sin^{-1}\left(\frac{\|w\|}{\|r\|}\sin\theta\right) \qquad (2)$$

Where θ is the cone angle, which we estimate using the technique described earlier. The second equation has one unknown, the watermark length $\|w\|$. The top equation has one unknown, n; the hit rate P_Y can be estimated by experiment.

If we then consider different detection rates P_Y for uniform noise vectors of length A, and then for noise of length B, we can combine these equations into the following identity:

$$\tan\theta = \frac{A\sin\psi_A - B\sin\psi_B}{A\cos\psi_A - B\cos\psi_B}$$

...where ψ_A and ψ_B are the integration limits in equation (1). This removes the unknown w, while n is implicit in the constants ψ_A and ψ_B. Here is our algorithm:

1. Pick two power levels A and B. They can be arbitrary, but make sure that the error rate under those noise levels is reasonably estimable in a few hundred trials (*e.g.,* 0.5 rather than 0.0000005.)
2. Use the watermark detector to estimate P_A and P_B, the detection rate under unform noise of length A and B, respectively.
3. For all suspected values of n:
   (a) Use the hypothesized n and estimated detection rates in equation (1) to estimate the parameters ψ_A and ψ_B. The integral does not have a simple closed form, but the integration limit is easily determined by Newton's method.
   (b) Compute $\epsilon_n \leftarrow \left|\tan\theta - \frac{A\sin\psi_A - B\sin\psi_B}{A\cos\psi_A - B\cos\psi_B}\right|$
4. Choose the value of n that minimizes the error ϵ.

4. RESULTS

We tested our techniques on a generic watermark detector with a varying number of DCT feature coefficients. We used normalized correlation with a detection threshold of 0.5. This requires some justification: a proper detector would probably have a much higher threshold, since for $\tau = 0.5$ the false alarm rate is unnecessarily low. However, in our experience watermark detectors are often designed in an ad-hoc manner, and designers are often prone to choosing round numbers, or in the case of thresholds, a value squarely between 0 and 1. In fact, a higher threshold would increase the growth rate of our noise snake.

We first generated noise snakes to deduce a detector threshold. Figure 8 shows our estimates for a detector with $\tau = 0.5$. This required an average of 1016 detector queries per experiment, to generate two snakes. Figure 9 shows the corresponding dimension estimates, once the threshold is deduced to be 0.5.

It is worth pointing out that in this detector, the asymptotic false alarm probability is approximately 2.39×10^{-33}. This means that we can rougly estimate a very low false alarm probability in only thousands of trials.

5. DISCUSSION AND CONCLUSIONS

5.1. Is it Useful?

Our experience reverse-engineering watermarks "by hand" shows us that only dozens of experimental queries are needed by an expert to identify a watermark using a common feature space, sub-band, and detector. In contrast, this technique is *generic*, and utilizes no expert knowledge or *a priori* information about likely watermarks. The number of queries is far larger.

We feel that techniques of this type can be combined with codified expert knowledge into an overall system that dismantles an unknown algorithm and estimates its parameters far more efficiently.

Another question is whether such detector parameters are very useful. So what if we know the rough number of features used by a watermark detector? The answer is that this conatins more information than one might initially think. For example, the BOWS watermark used around 50000 AC DCT coefficients.[7] Consider the significance of that number. In fact the exact value was 49152, the only genuinely "round" hexadecimal number near 50000. If one is told that an image watermark uses "around 50000" coefficients, 49152 is a very good guess for the actual value. It also turns out that 49152 = 12 × 4096, and the BOWS test images contained 4096 8-by-8 blocks. Hence from even a rough estimate of the feature space size, one can make a good guess that the algorithm is block-based, and amends 12 coefficients per block. This is, in fact, the algorithm used.

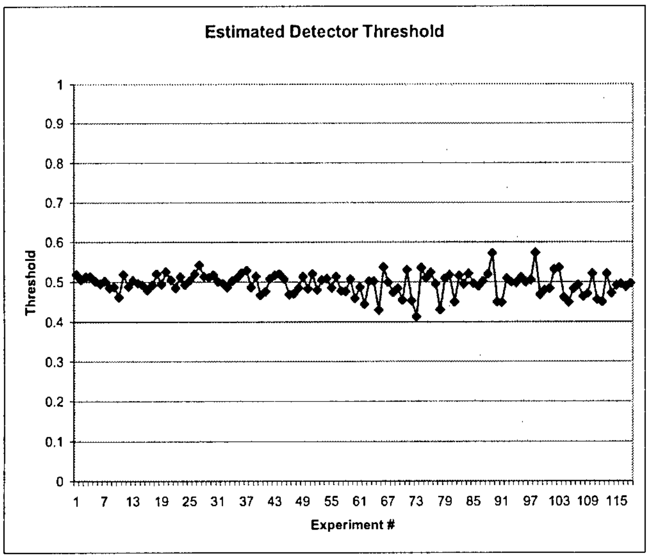

Figure 8. Estimated threshold using two noise snakes. Ground truth is $\tau = 0.5, n = 500$.

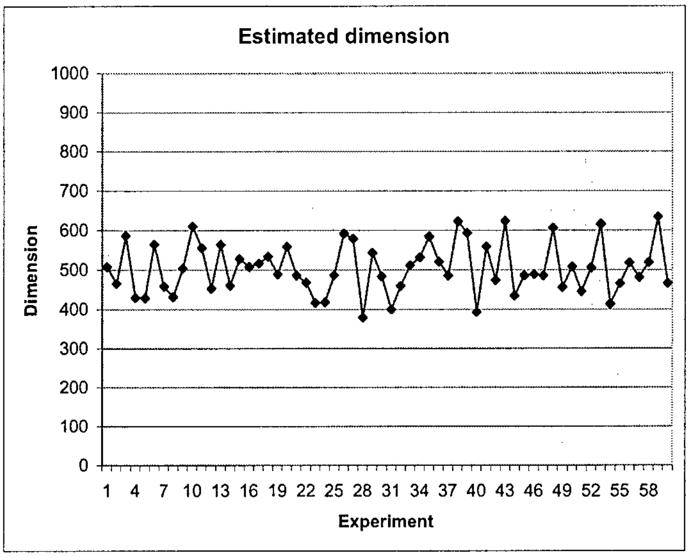

Figure 9. Estimated dimension using two noise snakes. Ground truth is $n = 500$.

5.2. Attack Prevention

As watermark designers, we might decide that enough is enough, and we have had it with these snakes on this cone. What are we going to do about it? The most obvious strategy is to avoid letting a detector be used as an oracle. Beyond that, preventing superrobustness in a detector could prevent these attacks.

In our geometric analogy, this means that *a detection region should be bounded.* Instead of a cone, for example, we can have a truncated cone by rejecting any image whose feature vector is overlong. The problem with this approach is that it only prevents superrobustness within the feature space: as long as some other information is ignored—that is, as long as the feature space is a proper subspace of the image—we can launch superrobustness attacks to deduce the feature space. When considering non-feature and feature spaces combined, the detection region is no longer a cone, but a hyper-cylinder with the cone as a base. *This* region remains unbounded.

Time will tell if these attacks can be prevented, or if superrobustness can be effectively utilized to reverse-engineer watermarking algorithms. We feel that great gains in efficiency can be made, and an overall system for automation of watermark reverse-engineering can use superrobustness as an exploitable flaw, and use oracle attacks such as those described here to deduce crucial unknown parameters of a data hiding system.

ACKNOWLEDGMENTS

This research was made possible by the generous support of the Air Force Office of Scientific Research, under grant FA9550-95-1-0440.

REFERENCES

1. J.-P. M. G. Linnartz and M. van Dijk, "Analysis of the sensitivity attack against electronic watermarks in images," *Lecture Notes in Computer Science* **1525**, pp. 258–272, 1998.
2. P. Comesana, L. Perez-Freire, and F. Perez-Gonzalez, "Blind newton sensitivity attack," *IEE Proceedings on Information Security* **153**, pp. 115–125, Sept. 2006.
3. S. Craver, M. Wu, Liu, A. Stubblefield, B. Swartzlander, D. S. Dean, and E. Felten, "Reading between the lines: Lessons learned from the SDMI challenge," *Proc. Usenix Security Symposium*, pp. 353–363, August 2001.
4. I. Atakli, S. Craver, and J. Yu, "How we broke the bows watermark," in *Proc. SPIE, Security, Steganography, and Watermarking of Multimedia Contents IX, to appear*, 2007.
5. I. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA, 2002.
6. "Hypersphere – from Wolfram MathWorld." http://www.mathworld.wolfram.com/Hypersphere.html.
7. M. L. Miller, G. J. Doerr, and I. J. Cox, "Applying informed coding and embedding to design a robust, high capacity watermark," *IEEE Trans. on Image Processing* **13**, pp. 792–807, June 2004.