Watermarking Schemes Evaluation

Algorithms Need Common Benchmarks

Fabien A.P. Petitcolas

©Stock Illustration Source.

Digital watermarking has been presented as a solution to the copy protection of multimedia objects and dozens of schemes and algorithms have been proposed. But two main problems seriously darken the future of this technology. First, the large number of attacks and weaknesses, which appear as fast as new algorithms are proposed, emphasizes the limits of this technology and in particular the fact that it may not match users' expectations. Second, the requirements, tools, and methodologies to assess the current technologies are almost nonexistent. The lack of benchmarking of current algorithms is blatant. This confuses rights holders as well as software and hardware manufacturers and prevents them from using the solution appropriate to their needs. Indeed basing long-lived protection schemes on badly tested watermarking technology does not make sense.

Our focus in this article is how one could solve the second problem by having a public benchmarking service. We will examine the challenges behind such a service.

Background

Digital watermarking remains a largely untested field, and only very few large industrial consortiums have published requirements against which watermarking algorithms should be tested [1], [2]. For instance, the International Federation for the Phonographic Industry led one of the first large-scale comparative testing of watermarking algorithms for audio. In general, a number of broad claims have been made about the "robustness" of various digital watermarking or fingerprinting methods but very few researchers or companies have published extensive tests on their systems.

1053-5888/00/$10.00©2000IEEE

The growing number of attacks against watermarking systems (e.g., [3]–[5]) has shown that far more research is required to improve the quality of existing watermarking methods so that, for instance, the JPEG 2000 (and other new multimedia standards) can be more widely used within electronic commerce applications.

We already pointed out in [6] that most articles have used their own limited series of tests, their own pictures, and their own methodology and that, consequently, comparison was impossible without reimplementing the methods and trying to test them separately. But then the implementation might be very different from—and quite possibly weaker than—the original authors'. This suggested that methodologies for evaluating existing watermarking algorithms were urgently required, and we proposed a simple benchmark for still-image marking algorithms.

With a common benchmark, authors and watermarking software providers would just need to provide a more or less detailed table of results, which would give a good and reliable summary of the performances of the proposed scheme. Then this would allow end users to check whether their basic requirements are satisfied and researchers to compare different algorithms to see how a method can be improved or whether a newly added feature actually improves the reliability of the whole method; the industry could properly evaluate risks associated with the use of a particular solution by knowing which level of reliability can be achieved by each contender. Watermarking system designers would be able to use such an evaluation to identify possible weak points during the early development phases of the system.

Evaluation per se is not a new problem, and significant work has been done to evaluate, for example, image compression algorithms or security of information systems [7], and we believe that some of it may be reused for watermarking.

In the following section we will explain the scope of the evaluation we envisage. In the third section we will review the type of watermarking schemes that an automated evaluation service could manage. (Such service is the logical continuation of the existing StirMark benchmark.) In the fourth section we will review the basic functionalities that need to be evaluated. Next, we will examine how each functionality can be tested. In the final section we will argue the need for a third-party evaluation service and briefly sketch its architecture.

Scope of Evaluation

Watermarking algorithms are often used in larger systems designed to achieve certain goals (e.g., prevention of illegal copying). For instance, Herrigel et al. [8] presented a system for trading images; this system uses watermarking technologies but relies heavily on cryptographic protocols.

Basing long-lived protection schemes on badly tested watermarking technology does not make sense.

Such systems may be flawed for reasons other than watermarking itself; for example, the protocol—which uses the watermark—may be wrong or the random number generator used by the watermark embedder may not be good. [The "watermark," or "mark," is what is actually imperceptibly added to the cover signal to convey the hidden data. The "cover signal" is the audiovisual signal (still image, audio track, video) in which one wishes to hide information (the work, in many cases).] In this article we are only concerned with the evaluation of watermarking (thus the signal-processing aspects) within the larger system, not the effectiveness of the full system, to achieve its goals.

Target of Evaluation

The first step in the evaluation process is to clearly identify the target of evaluation, that is, the watermarking scheme (set of algorithms required for embedding and extraction) subject to evaluation and its purpose. The purpose of a scheme is defined by one or more objectives and an operational environment. For instance, we may wish to evaluate a watermarking scheme that allows automatic monitoring of audio tracks broadcast over radio.

Typical objectives found across the watermarking and copy protection literature include the following:

▲ *Persistent identification of audiovisual signals:* The mark carries a unique identification number (similar to an ISBN), which can be used as a pointer in a database. This gives the ability to manage the association of digital content with its related descriptive data, current rights holders, license conditions, and enforcement mechanisms. This objective is quite general as it may wrap many other objectives described below. However, one may wish to have the data related to the work stored into the work itself rather than into a central database to avoid connection to a remote server.

▲ *Proof of creatorship, proof of ownership:* The embedded mark could be used to prove to a court who created or holds the rights to the work.

▲ *Auditing:* The mark carries information used to identify parties present in a transaction involving the work (the distributors and the end users). This audit trail shows the transfer of work between parties. Marks for identifying users are usually referred to as "fingerprints."

▲ *Copy-control marking:* The mark carries information regarding the number of copies allowed. Such marks are used in the digital versatile disk copy protection mecha-

nisms. In this system a work can be copied, copied once only, or never copied [9].

▲ *Monitoring of multimedia object usage:* Monitoring copyright liability can be achieved by embedding a license number into the work and having, for instance, an automated service constantly surfing the web or listening to the radio, checking the licensing, and reporting infringement.

▲ *Tamper evidence:* Special marks can be used in a way that allows detection of modifications introduced after the mark has been added.

▲ *Labeling for user awareness:* This type of mark is typically used by right holders to warn end users that the work they "have in hand" is copyrighted. For instance, whenever an end user tries to save a copyrighted image opened in a web browser or an image editor, he may get a warning encouraging him to purchase a license for the work.

▲ *Data augmentation:* This is not really in the scope of "digital watermarking," but a similar evaluation methodology can be applied to it.

▲ *Labeling to speed up search in databases.*

Basic Functionalities

The objectives of the scheme and its operational environment dictate several immediate constraints (a set of minimal requirements) on the algorithm. In the case of automated radio monitoring, for example, the watermark should clearly withstand distortions introduced by the radio channel. Similarly, in the case of MPEG video broadcast, the watermark detector must be fast (to allow real-time detection) and simple in terms of number gates required for hardware implementation. One or more of the following general functionalities can be used.

Perceptibility

One does not wish that the hidden mark deteriorates the perceived quality of the medium too much.

Level of Reliability

There are two main aspects to reliability:

▲ *Robustness* and *false negatives* occur when the content was previously marked but the mark could not be detected. The threats centered on signal modification are robustness issues. Robustness can range from no modification at all to destruction of the signal. (Complete destruction may be too stringent a requirement. Actually, it is not clear what it means. Instead one could agree on a particular quality measure and a maximum quality loss value.) This requirement separates watermarking from other forms of data hiding (typically steganography). Without robustness, the information could simply be stored as a separate attribute.

Robustness remains a very general functionality as it may have different meanings depending on the purpose of the scheme. If the purpose is image integrity (tamper evidence), the watermark extractor should have a different output after small changes have been made to the image while the same changes should not affect a copyright mark.

In fact, one may distinguish at least the following main categories of robustness:

1. The threats centered on modifying the signal to disable the watermark (typically a copyright mark), wilfully or not, remain the focus of many research articles that propose new attacks. By "disabling a watermark" we mean making it useless or removing it.
2. The threats centered on tampering of the signal by unauthorized parties to change the semantic of the signal are an integrity issue. Modification can range from the modification of court evidences to the modification of photos used in newspapers or clinical images.
3. The threats centered on anonymously distributing illegal copies of marked work are a traitor-tracing issue and are mainly addressed by cryptographic solutions [10].
4. Watermark cascading—that is, the ability to embed a watermark into an audiovisual signal that has been already marked—requires a special kind of robustness. The order in which the marks are embedded is important [11] because different types of marks may be embedded in the same signal. For example, one may embed a public and a private watermark (to simulate asymmetric watermarking) or a strong public watermark together with a tamper evidence watermark. As a consequence, the evaluation procedure must take into account the second watermarking scheme when testing the first one.

▲ At last, false positives occur whenever the detected watermark differs from the mark that was actually embedded. The detector could find a mark A in a signal where no mark was previously hidden, in a signal where a mark A was actually hidden with the same scheme, where a mark B was hidden with another scheme.

Capacity

Knowing how much information can reliably be hidden in the signal is very important to users, especially when the scheme gives them the ability to change this amount. Knowing the watermarking access unit (or granularity) is also very important; indeed, spreading the mark over a full sound track prevents audio streaming, for instance. (A "watermark access unit" is the smallest part of a cover signal in which a watermark can be reliably detected and the payload extracted.)

Speed

As we mentioned earlier, some applications require real-time embedding and/or detection.

Statistical Undetectability

For some private watermarking systems—that is, a scheme requiring the original signal—one may wish to have a perfectly hidden watermark. In this case it should not be possible for an attacker to find any significant statistical

differences between an unmarked signal and a marked signal. As a consequence an attacker could never know whether an attack succeeded or not; otherwise he could still try something similar to the "oracle" attack [12]. Note that this option is mandatory for steganographic systems.

Asymmetry

Private-key watermarking algorithms require the same secret key both for embedding and extraction. They may not be good enough if the secret key has to be embedded in every watermark detector (that may be found in any consumer electronic or multimedia player software); then malicious attackers may extract it and post it to the Internet allowing anyone to remove the mark. In these cases the party that embeds a mark may wish to allow another party to check its presence without revealing its embedding key. This can be achieved using asymmetric techniques. Unfortunately, robust asymmetric systems are currently unknown, and the current solution (which does not fully solve the problem) is to embed two marks: a private one and a public one.

Other functionality classes may be defined but the ones listed above seem to include most requirements used in the recent literature. The first three functionalities are strongly linked together, and the choice of any two of them imposes the third one. In fact, when considering the three-parameter watermarking model (perceptibility, capacity, and reliability), the most important parameter to keep is the imperceptibility. ("Capacity" is the bit size of a payload that a watermark access unit can carry.) Then two approaches can be considered: emphasize capacity over robustness or favor robustness at the expense of low capacity. This clearly depends on the purpose of the marking scheme, and this should be reflected in the way the system is evaluated.

Evaluation

A full scheme is defined as a collection of functionality services to which a level of assurance is globally applied and for each of which a specific level of strength is selected. So a proper evaluation has to ensure that all the selected requirements are met to a certain level of assurance.

The number of levels of assurance cannot be justified precisely. On the one hand, it should be clear that a large number of them makes the evaluation very complicated and unusable for particular purposes. On the other hand, too few levels prevent scheme providers from finding an evaluation close enough to their needs. We are also limited by the accuracy of the methods available for rating. Information technology security evaluation has been using six or seven levels for the reasons we just mentioned above but also for historical reasons. This seems to be a reasonable number for robustness evaluation.

For perceptibility we preferred to use fewer levels and, hence, follow more or less the market segmentation for electronic equipment. Moreover, given the roughness of existing quality metrics it is hard to see how one could reasonably increase the number of assurance levels.

Following we discuss possible methods to evaluate the functionalities listed above.

Perceptibility

Perceptibility can be assessed to different levels of assurance. The problem here is very similar to the evaluation of compression algorithms. The watermark could be just slightly perceptible but not perceptible under domestic/consumer viewing/listening conditions. Another level is nonperceptibility in comparison with the original under studio conditions. Finally, the best assurance is obtained when the watermarked media are assessed by a panel of individuals who are asked to look or listen carefully at the media under the above conditions (see Table 1).

As it is stated, however, this cannot be automated, and one may wish to use less stringent levels. In fact, various levels of assurance can also be achieved by using various quality measures based on human perceptual models. Since there are various models and metrics available, an average of them could be used. Current metrics do not really take into account geometric distortions, which remain a challenging attack against many watermarking schemes.

Reliability

Although robustness and capacity are linked in the sense that schemes with high capacity are usually easy to defeat, we believe that it is enough to evaluate them separately. Watermarking schemes are defined for a particular application, and each application only requires a certain fixed payload so we are solely concerned with the robustness of the scheme for this given payload.

Robustness

The robustness can be assessed by measuring the detection probability of the mark and the bit error rate for a set of criteria that are relevant for the application that is considered.

The levels of robustness range from no robustness to provable robustness (e.g., [7], [13]).

For *level zero,* no special robustness features have been added to the scheme apart from the one needed to fulfill the basic constraints imposed by the purpose and operational environment of the scheme. So if we go back to the radio-monitoring example, the minimal robustness feature should make sure that the mark survives the distortions of the radio link in normal conditions.

The *low level* corresponds to some extra robustness features added but which can be circumvented using simple and cheap tools publicly available. These features are provided to prevent "honest" people from disabling the mark during normal use of the work. In the case of watermarks used to identify owners of photographs, the end users should be able to save and compress the photo, resize it, and crop it without removing the mark.

Moderate robustness is achieved when more expensive tools are required, as well as some basic knowledge on watermarking. So if we use the previous example, the end user would need tools such as Adobe Photoshop and apply more processing to the image to disable the mark.

Moderately high means tools are available but special skills and knowledge are required and attempts may be unsuccessful. Several attempts and operations may be required and one may have to work on the approach.

High robustness means all known attempts have been unsuccessful. Some research by a team of specialists is necessary. The cost of the attempt may be much higher than what it is worth and its success is uncertain.

Provable robustness means it should be computationally (or even more stringent: theoretically) infeasible for a willful opponent to disable the mark. This is similar to what we found for cryptography where some algorithms are based on some difficult mathematical problem.

The first levels of robustness can be assessed automatically by applying a simple benchmark algorithm similar to [6]:

▲ For each medium in a determined set:

1. Embed a random payload with the greatest strength that does not introduce annoying effects. In other words, embed the mark such that the quality of the output for a given quality metric is greater than a given minima.

2. Apply a set of given transformations to the marked medium.

Table 1. Summary of the Possible Perceptibility Assurance Levels.*

Level of Assurance	Criteria
Low	—PSNR (when applicable**) —Slightly perceptible but not annoying
Moderate	—Metric based on perceptual model —Not perceptible under domestic conditions, that is, using mass-market consumer equipment
Moderate high	Not perceptible in comparison with original under studio conditions
High	Evaluation by a large panel of persons under strict conditions

*These levels may seem vague but this is the best we can achieve as long as we do not have good and satisfactory quality metrics.

**PSNR is a very restrictive quality metrics; it does not take into account any properties of the human visual model. This includes the usual masking properties but also the large tolerance to geometric distortions. By using PSNR one excludes immediately watermarking schemes based on geometric distortions. Unfortunately, we are not aware of any metric taking those distortions into account.

▲ For each distorted medium try to extract the watermark and measure the certainty of extraction. Simple methods may just use a success/failure approach, that is, to consider the extraction successful if and only if the payload is fully recovered without error. The measure for the robustness is the certainty of detection or the bit error rate after extraction.

This procedure must be repeated several times since the hidden information is random and a test may be successful by chance.

Levels of robustness differ by the number and strength of attacks applied and the number of media on which they are measured. The set of test and media will also depend on the purpose of the watermarking scheme and are defined in evaluation profiles. An evaluation profile sample is given in Table 2. For example, schemes used in medical systems need only to be tested on medical images while watermarking algorithms for owner identification have to be tested on a large panel of images.

The first levels of robustness can be defined using a finite and precise set of robustness criteria (e.g., S.D.M.I., IFPI, or E.B.U. requirements) and one just needs to check them.

False Positives

False positives are difficult to measure, and current solutions use a model to estimate their rate. This has two major problems: first, "real world" watermarking schemes are difficult to model accurately, and second, modeling the scheme requires access to details of the algorithm. Despite the fact that not publishing algorithms breaches Kerckhoffs' principles [14], details of algorithms are still considered trade secrets, and getting access to them is not always possible. (In 1883, Auguste Kerckhoffs enunciated the first principles of cryptographic engineering, in which he advises that we assume the method used to encipher data is known to the opponent, so security must lie only in the choice of key. The history of cryptology since then has repeatedly shown the folly of "security-by-obscurity"—the assumption that the enemy will remain ignorant of the system in use.)

So one (naïve) way to estimate the false-alarm rate is to count the number of false alarms using large sample of data. This may turn out to be another very difficult problem, as some applications require one error in 108 or even 1012.

Capacity

In most applications the capacity will be a fixed constraint of the system so robustness testing will be done with a random payload of a given size. While developing a watermarking scheme, however, knowing the tradeoff between the basic requirements is very useful and graphing with two varying requirements—the others being fixed—is a simple way to achieve this. In the basic three-parameter watermarking model, for example, one

can study the relationship between robustness and strength of the attack when the quality of the watermarked medium is fixed between the strength of the attack and the visual quality or between the robustness and the visual quality [6]. The first one is probably the most important graph. For a given attack, and a given visual quality, it shows the bit error rate as a function of the strength of the attack. The second graph shows the maximum attack that the watermarking algorithm can tolerate. This is useful from a user's point of view: the performance is fixed (we want only 5% of the bits to be corrupted so we can use error correction codes to recover all the information we wanted to hide) and so it helps to define what kind of attacks the scheme will survive if the user accepts such or such quality degradation.

Speed

Speed is dependent on the type of implementation: software or hardware. In the automated evaluation service we propose in the next section, we are not concerned with hardware implementations. For these, the complexity is an important criteria and some applications impose a limitation on the maximum number of gates that can be used, the amount of required memory, etc. [15].

For a software implementation, success also depends very much on the hardware used to run it but comparing performance results obtained on the same platform (usually the typical platform of end users) provides a reliable measure.

Statistical Undetectability

All methods of steganography and watermarking substitute part of the cover signal, which has some particular statistical properties, with another signal with different statistical properties; in fact, embedding processes usually do not pay attention to the difference in statistical properties between the original cover signal and the stegosignal. This leads to possible detection attacks [16].

As for false positives, evaluating such functionality is not trivial but fortunately very few watermarking schemes require it, so we will not consider it in the next section.

Methodology—Need for a Third Party

To gain trust in the reliability of a watermarking scheme, its qualities must be rated. This can be done by:

- trusting the provider of the scheme and his quality assurance (or claims);
- testing the scheme sufficiently oneself; or
- having the scheme evaluated by a trusted third party.

Only the third option provides an objective solution to the problem but the general acceptance of the evaluation methodology implies that the evaluation itself is as transparent as possible. This was the aim of StirMark, and this remains the aim of the project to build a next generation of StirMark benchmarks. This is why the source code and methodology must be public, so one can reproduce the results easily.

A question one may ask is: Does the watermarking system manufacturer need to submit any program at all, or can everything be done remotely using some interactive proof? Indeed, watermarking system developers are not always willing to give out software or code for evaluation or company policy for intellectual property prevents them from doing this quickly. Unfortunately there is no protocol by which an outsider can evaluate such systems using a modified version of the above robustness testing procedure. One could imagine that the verifier sends an image I to be watermarked to the prover. After receiving the marked images $\tilde{I}$, the verifier would apply a transformation f to the image and send either $J := f(I)$ or $J := f(\tilde{I})$ to the prover, who would just say "I can detect the mark" or "I cannot detect the mark." The verifier would always accept a "no" answer, but a "yes" answer only with a certain probability. After several iterations of the protocol the verifier would be convinced.

Unfortunately in this case, most fs are invertible or almost invertible—even if f is a random geometric distortion, such as the one implemented into StirMark, it can be inverted using the original image. So the prover can always approximate f^{-1} by comparing J to I and $\tilde{I}$ and try

Table 2. Evaluation Profile Sample.

	Level Zero	Low Level	Moderate
Standard JPEG compression quality	100 – 90	100 – 75	100 – 50
Color reduction (GIF)	256	256	16
Cropping	100 – 90%	100 – 75%	100 – 50%
Gamma correction		0.7 – 1.2	0.5 – 1.5
Scaling		1/2 – 3/2	1/3 – 2
Rotation		±0 – 2°	±0 – 5, 90°
Horizontal flip		✓	✓
Uniform noise		1 – 5%	1 – 15%
Contrast		±0 – 10%	±0 – 25%
Brightness		±0 – 10%	±0 – 25%
Median filter			3 × 3

to detect the mark in $f^{-1}(J)$ and so, always cheat. The conclusion of this is that the verifier must have at least a copy of the detection or extraction software.

So we propose, as a first step toward a widely accepted way to evaluate watermarking schemes, the implementation of an automated benchmark server for digital watermarking schemes. The idea is to allow users to send a binary library of their scheme to the server, which in turn runs a series of tests on this library and keeps the results in a database accessible to the scheme owner and/or to all "watermarkers." One may consider this service as the next generation of the StirMark benchmark: fully automated evaluation with real-time access to data.

To be widely accepted, this service must have a simple interface with existing watermarking libraries; in the implementation we propose we have exported only three functions (scheme information, embedding, and detection). The service must also, as we described earlier, take into account the application of the watermarking scheme by proposing different evaluation profiles (tests and set of media samples) and strengths; this will be achieved by the use of different evaluation profiles' configuration files. The service must be easy to use:

- the client sends a library (which follows our general interface) to be evaluated and specifies the evaluation profile and level of assurance to be used;
- the StirMark benchmark service automatically starts hundreds of tests on the library using its library of media;
- as soon as the test are finished the results are sent to the client and may later be published on the project website;
- at last all evaluation procedures, profiles, and code must be publicly available.

Although our current implementation only supports image-watermarking schemes, the general architecture we have chosen will allow us to support other media in the near future.

Conclusions and Future Work

In this article we have used a duality approach to the watermarking evaluation problem by splitting the evaluation criteria into two (independent) groups: functionality and assurance. The first group represents a set of requirements that can be verified using agreed series of tests; the second is a set of levels to which each functionality is evaluated. These levels go from zero or low to very high.

We are investigating how evaluation profiles can be defined for different applications and how importance sampling techniques could be used to evaluate the false alarm rate in an automated way.

Hopefully this new generation of watermarking testing tools (in the continuation of the StirMark benchmark [17]) will be very useful to the watermarking community.

Fabien A.P. Petitcolas graduated from the École Centrale, Lyon, France, and received a Diploma in computer science and his Ph.D. from the University of Cambridge, England. He is now with Microsoft Research. His research interests include information hiding, multimedia security, and computer security engineering.

References

[1] *Request for Proposals–Embedded Signalling Systems Issue 1.0* , International Federation of the Phonographic Industry, London, U.K., June 1997.

[2] *Watermarking–Call for Systems* , European Broadcasting Union and Union Européenne de Radio Télévision, May 2000.

[3] J.K. Su and B. Girod, "Fundamental performance limits of power-spectrum condition-compliant watermarks," in *Proc. Electronic Imaging '99, Security and Watermarking of Multimedia Contents II*, vol. 3971. San Jose, CA, Jan. 24–26, 2000, pp. 314–325.

[4] M. Kutter, "Watermark copy attack," in *Proc. Electronic Imaging '99*, *Security and Watermarking of Multimedia Contents II*, vol. 3971. San Jose, CA, Jan. 24–26, 2000, pp. 371–380.

[5] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn, "Attacks on copyright marking systems," in *Second Workshop on Information Hiding* (Lecture Notes in Computer Science, vol. 1525), D. Aucsmith, Ed. Portland, OR, Apr. 14–17, 1998, pp. 218–238.

[6] M. Kutter and F.A.P. Petitcolas, "A fair benchmark for image watermarking systems," in *Proc. Electronic Imaging '99*, *Security and Watermarking of Multimedia Contents*, vol. 3657. San Jose, CA, Jan. 25–27, 1999, pp. 226–239.

[7] IT-Security Criteria, "Criteria for the evaluation of trustworthiness of information technology systems," German Information Security Agency (Zentralstelle für Sicherheit in der Informationstechnik) on behalf of the German Government, Köln: Bundesanzeiger, 1989.

[8] A. Herrigel, et al., "Secure copyright protection techniques for digital images," in *Second Workshop on Information Hiding* (Lecture Notes in Computer Science, vol. 1525), D. Aucsmith, Ed. Portland, OR, Apr. 16–18, 1998, pp. 218–238.

[9] J.A. Bloom, I.J. Cox, T. Kalker, J.-P.M.G. Linnartz, M.L. Miller, and C.B.S. Traw, "Copy protection for D.V.D. video," *Proc. IEEE*, vol. 87, pp. 1267–1276, July 1999.

[10] J.-H. Lee, "Fingerprinting," in *Information Hiding Techniques for Steganography and Digital Watermarking*, S.C. Katzenbeisser et al., Eds. Norwood, MA: Artech House, Dec. 1999, pp. 175–190.

[11] F. Mintzer and G.W. Braudaway, "If one watermark is good, are more better?" in *Proc. Int. Conf. Acoustics, Speech and Signal Processing* (ICASSP'99), Phoenix, AZ, 15–19 Mar. 1999, vol. 4, pp. 2067–2070.

[12] J.-P.M.G. Linnartz and M. van Dijk, "Analysis of the sensitivity attack against electronic watermarks in images," in *Second Workshop on Information Hiding* (Lecture Notes in Computer Science, vol. 1525), D. Aucsmith, Ed. Portland, OR, 25–27 Apr. 1998, pp. 218–238.

[13] D.G. Abraham et al., "Transaction security system," *IBM Syst. J.,* vol. 30, no. 2, pp. 206-229, 1991.

[14] A. Kerckhoffs, "La cryptographie militaire," *J. Sciences Militaires*, vol. 9, pp. 5–38, Jan. 1883.

[15] M.L. Miller, I.J. Cox, and J.A. Bloom, "Watermarking in the real world: An application to DVD," in *Proc. Multimedia and Security–Workshop at ACM Multimedia '98*, Bristol, U.K., Sept. 12–13, 1998, pp. 71–76.

[16] S.C. Katzenbeisser, "Principles of steganography," in *Information Hiding Techniques for Steganography and Digital Watermarking*, S.C. Katzenbeisser et al., Eds. Norwood, MA: Artech House, Dec. 1999, pp. 30–31.

[17] http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/