GMD Report 85

GMD –
Forschungszentrum
Informationstechnik
GmbH

Jana Dittmann, Klara Nahrstedt,
Petra Wohlmacher (Eds.)

Multimedia and Security

Workshop at ACM Multimedia'99

Orlando, Florida, USA

October 30 - November 5 1999

December 1999

© GMD 1999

GMD –
Forschungszentrum Informationstechnik GmbH
Schloß Birlinghoven
D-53754 Sankt Augustin
Germany
Telefon +49 -2241 -14 -0
Telefax +49 -2241 -14 -2618
http://www.gmd.de

In der Reihe GMD Report werden Forschungs- und Entwicklungsergebnisse aus der GMD zum wissenschaftlichen, nichtkommerziellen Gebrauch veröffentlicht. Jegliche Inhaltsänderung des Dokuments sowie die entgeltliche Weitergabe sind verboten.

The purpose of the GMD Report is the dissemination of research work for scientific non-commercial use. The commercial distribution of this document is prohibited, as is any modification of its content.

Anschrift der Verfasser/Address of the authors:
Jana Dittmann
Dr. Klara Nahrstedt
Petra Wohlmacher
Institut für Integrierte Publikations- und Informationssysteme
GMD – Forschungszentrum Informationstechnik GmbH
Dolivostraße 15
D-64293 Darmstadt
E-mail: Jana.Dittmann@gmd.de
Klara.Nahrstedt@gmd.de
Petra.Wohlmacher@gmd.de

ISSN 1435-2702

Abstract

Recently security has become one of the most significant problems for spreading new information technology. Digital data can easily be copied and multiplied without information loss. This requires security solutions for such fields as distributed production processes and electronic commerce, since the producers seek to provide access control mechanisms to prevent misuse and theft of material. The workshop analyses specific security problems of multimedia systems and multimedia material in the digital environment. Based on our discussion in the workshop at the ACM MM'98 in Bristol we want to continue with the state of the art evaluation and discuss future needs for the design of MM Security and legal aspects. We understand that the interest and importance of security was reflected in the great number of participants from all over the world in Bristol.

Objectives

Based on these excellent experiences the objective of the workshop is to see the advantages in the field of multimedia and security. Especially in the field of copyright protection we will evaluate the progress of digital watermarking, the robustness and the practical usage for authentication and also for integrity checks. Beside technical approaches legal requirements, the identification of design and acceptance problems for security solutions are further topics. In the workshop we want also address the topic, that existing media security mechanisms are address multimedia. Thus, the discussion is extend to the use watermarking for multimedia to perform a combined security. Furthermore based on the discussions on security in multimedia environments we want to analyse interactive multimedia tools which strengthen the producers acceptance to use available security features. The intention of the workshop is to bring together experienced researchers, developers, and practitioners from academia and industry for a state of the art evaluation and discussions of topics and problems for multimedia security environments for the next century. The workshop reflects the strength and weaknesses of what the multimedia community has to offer to meet the needs of secure multimedia environments.

Jana Dittmann
Klara Nahrstedt
Petra Wohlmacher

Keywords: multimedia, security, digital watermarking, copyright protection, manipulation recognition, privacy.

Abstrakt

Die Problematik Sicherheit spielt in vielen IT-Anwendungen eine wesentliche Rolle, da digitale Daten einfach ohne Qualitätsverlust kopiert und vervielfältigt sowie leicht manipuliert werden können. Sicherheitslösungen werden deshalb beispielsweise im Bereich verteilter Mediaproduktionsumgebungen und Ecommerce notwendig, um Zugriffskontrollen zu realisieren sowie Diebstahl und Mißbrauch zu vermeiden. Der Workshop analysiert spezifische Probleme in Multimediaapplikationen und mit Mediendaten. Aufbauend auf dem Workshop auf der ACM MM ’98 in Bristol wollen wir die Diskussionen zum Stand der Forschung und zukünftiger Problemfelder fortsetzen sowie die rechtliche Aspekte einbeziehen. Die Vielzahl der Teilnehmer in Bristol hat uns gezeigt, daß die Sicherheitsproblematik bei Multimedia auch international von großem Interesse ist.

Ziele

Aufbauend auf den Erfahrungen in Bristol 1998 wollen wir die Fortschritte auf dem Gebiet Multimedia und Sicherheit aufzeigen und diskutieren. Speziell auf dem Gebiet des Urheberschutzes sollen digitale Wasserzeichentechniken betrachtet und deren Robustheitsaspekt und praktische Nutzungsmöglichkeiten zur Urheberkennzeichnung sowie Manipulationserkennung evaluiert werden. Neben den technischen Möglichkeiten und Grenzen sollen rechtliche Aspekte, Design- und Akzeptanzproblem dargelegt werden. Weiterhin soll diskutiert werden, inwieweit die Lösungen auch für Multimedia angewendet werden können, da bisher lediglich meist auf einem einzigen Medium gearbeitet wird. Ein weiterer Aspekt sind interaktive Arbeitsumgebungen, die es dem Benutzer erlauben, die Sicherheitsfunktionen zu bedienen und anzuwenden.
Ziel des Workshops ist es, Wissenschaftler und Entwickler aus Forschung und Industrie zusammenzubringen, um den Stand der Technik zu präsentieren sowie Probleme zu diskutieren, die zukünftig eine wesentliche Rolle spielen werden. Der Workshop reflektiert die Stärken und Schwächen der Techniken, die die Multimedia Community gegenwärtig anbieten kann, um den Ansprüchen an sichere Multimedia-Umgebungen nachzukommen.

Jana Dittmann
Klara Nahrstedt
Petra Wohlmacher

Schlagworte: Multimedia, Sicherheit, digitale Wasserzeichen, Copyright Protection, Manipulationserkennung, Schutz der Privatsphäre.

Contents

A Survey of Multimedia Security

Jana Dittmann
GMD – IPSI
Darmstadt, Germany
jana.dittmann@gmd.de

Klara Nahrstedt
University of Illinois
at Urbana-Champaign
klara@cs.uiuc.edu

Petra Wohlmacher
University of Klagenfurt
Klagenfurt, Austria
petra.wohlmacher@uni-klu.ac.at

ABSTRACT

Regarding security particularly in the field of multimedia, the requirements on security increase. If and in which way security mechanisms can be applied to multimedia data and their applications needs to be analyzed for each purpose separately. This is mainly due to the structure and complexity of multimedia.

Based on the main issues of IT-security, this paper introduces the most important security requirements, which must be fulfilled by today's multimedia systems. Furthermore it describes the security measures used to satisfy these requirements. These measures are based on modern cryptographic mechanisms and digital watermarking techniques as well as on security infrastructures.

KEYWORDS

Security requirements, security measures, security mechanisms, cryptographic mechanisms, digital watermarkings, multimedia, confidentiality, data integrity, data origin authenticity, entity authenticity, originality, non-repudiation.

1 Introduction

Recently security has become one of the most significant and challenging problems for spreading new information technology. Since digital data can easily be copied and multiplied without information loss as well as manipulated without any detection, security solutions are required, which encounter these threats. Security solution are especially of interest for such fields as distributed production processes and electronic commerce, since their producers provide only access control mechanisms to prevent misuse and theft of material. By increasing both the requirements for efficiency and the possibilities of IT-systems the needs for security and trustworthiness also enlarges. These needs are particularly important for security-relevant applications as well as for applications processing sensitive personal data.

In order to assess trustworthiness of IT-systems in general, catalogues for security criteria have been published [5, 13, 26, 27, 29]. One of the most important one is the Europe-wide valid ITSEC catalogue of criteria [13], which contains criteria for evaluating the security of IT-systems. This catalogue defines security criteria within different classifications regarding the following three basic threats:

- threat of confidentiality (unauthorized revealing of information),
- threat of integrity (unauthorized modification of data),
- threat of availability (unauthorized withholding of information or resources).

IT-systems are commonly used for different kinds of applications, increasingly applications dedicated to multimedia. In this context, IT-systems are named multimedia systems. Obviously, secure and trustworthy actions and interactions are also important requirements for multimedia systems. Whether or not a multimedia system fulfils these requirements will have a substantial influence on the acceptance of the relatively new medium multimedia.

Starting from these three threats the basic requirements for the security of a given multimedia system may derive. Security requirements are met by security measures, which generally consist of several security mechanisms. Security services can be made available by security mechanisms.

The remainder of this paper deals with the most important security requirements of today's multimedia systems. Additionally, security measures and security mechanisms, which are fulfilling these requirements, are presented, and problems with media data are discussed.

2 Requirements and Measures

The following security requirements are essential for multimedia systems. These requirements can be met by the succeeding security measures:

- Confidentiality: Cipher systems are used to keep information secret from unauthorized entities.
- Data integrity: The alteration of data can be detected by means of one-way hash functions, message authentication codes, digital signatures (especially content-based digital signatures), fragile digital watermarkings, and robust digital watermarkings.
- Data origin authenticity: Message authentication codes, digital signatures, fragile digital watermarkings, and robust digital watermarkings enable the proof of origin.

- Entity authenticity: Entities taking part in a communication can be proven by authentication protocols. These protocols ensure that an entity is the one it claims to be.
- Non-repudiation: Non-repudiation mechanisms prove to involved parties and third parties whether or not a particular event occurred or a particular action happened. The event or action can be the generation of a message, the sending of a message, the receipt of a message and the submission or transport of a message. Non-repudiation certificates, non-repudiation tokens, and protocols establish the accountability of information. These mechanisms are based on message authentication codes or digital signatures combined with notary services, timestamping services and evidence recording.

The security measures mentioned above use cryptographic mechanisms and digital watermarking techniques. A short introduction to both approaches is given in the next section. The focus is also concentrated on the problems with multimedia data deriving from applying cryptographic mechanisms.

3 Security Mechanisms

Security mechanisms applied for multimedia systems are based on cryptographic mechanisms as well as on digital watermarking techniques.

3.1 Cryptographic Mechanisms

Modern cryptographic mechanisms are mainly based on different not proven assumptions of the complexity theory concerning "easy" and "hard" computability of functions by algorithms. In this context, the term "easy" computability stands in contrast to the term "hard" computability. Intuitively, "easy" and "hard" computable functions can be described as follows:

A function $f: X \rightarrow Y$ is called to be "easy" to compute if any feasible algorithm exists, which calculates for all $x \in X$ the image $f(x)$. A function $f: X \rightarrow Y$ is named to be "hard" to compute if no feasible algorithm is known, which computes even for a small amount of elements of X their image $f(x)$. Additionally, there are reasons to assume, that such an algorithm cannot be found. Obviously, these are no exact definitions in a mathematical sense, since "known" and "not found" can not be defined in more detail.

There are attempts to specify these intuitive understandings in a mathematical way, but this seems to be very difficult and leads to unsolved problems concerning logic and theory of algorithms. First approaches for defining "easy" and "hard" computable functions are deriving from the complexity theory by examining the need for time and storage depending on the amount of input.

In cryptography two types of functions are used in particular: one-way functions and trapdoor one-way functions.

- A one-way function $f: X \rightarrow Y$ has the property that $f(x)$ is easy to compute for all $x \in X$ but for most $y \in \text{Image}(f)$ it is hard to find any $x \in X$ such that $f(x)=y$. For example, the following function belongs to this type of functions: Define $f: P \times P \rightarrow \mathbb{N}$ where $(p,q) \rightarrow p \cdot q$, it is easy to calculate the product n by multiplying two large primes p and q each possessing more than 100 decimals. But it is computationally infeasible (hard) to find two primes such that $n = p \cdot q$. This problem is named factorization problem. Another one-way function is the hash function which is described in section 5.
- A trapdoor one-way function is a one-way function $f: X \rightarrow Y$ with the additional characteristic that knowing a specific information (namely trapdoor information) it is easy to compute for any $y \in \text{Image}(f)$ an element $x \in X$ such that $f(x)=y$. With respect to the factorization problem this trapdoor information is represented by the prime factors of n.

The most important cryptographic mechanisms are implemented by use of cryptosystems. These systems consist of two sets of functions, a set of keys, parameterizing these functions, and sets, on which these functions operate. Cryptosystems are subdivided into private-key cryptosystems and public-key cryptosystems:

- In private-key cryptosystems the communicating entities share a key K, which must strictly be kept secret. Due to this requirement the key is called secret key. The size of the key space, where K is chosen from, must be large enough to make it hard to find the right key K, for instance by exhaustive search.
- Public-key cryptosystems are based on trapdoor one-way functions. Each entity holds a key-pair (PK,SK). This pair consists of a private key SK and a public key PK corresponding to SK. The key SK must strictly be kept secret, the key PK may be made public, e.g. in a public-key directory. Given a public key PK it is computationally infeasible to find the private key SK if the trapdoor information is unknown. In other words, even with the most powerful computers it is computationally infeasible to deduce PK from SK during a given period of time.

Usually cryptographic mechanisms take each bit of data for input to calculate the output, which is needed to provide the security mechanism. Additionally, the mechanisms have the property that if one bit of the input or output is changed, the encryption or even the validation i.e. that data is authentic or has integrity will fail in most cases.

Because multimedia applications need a high performance and their data can be altered due to transmission errors, higher compression rates or scaling operations during transmission or life time or even due to allowed operations such as scaling and conversion of picture formats, it seems to be difficult to define a suitable input for the cryptographic mechanisms. If cryptographic mechanisms are applied directly to the whole amount of media data some problems may occur. These problems and their solutions will be described in more detail

within each section accordingly to the presented cryptographic mechanism.

3.2 Digital Watermarking Techniques

Digital watermarking techniques based on steganographic systems offer the possibility to embed information directly into the media data. Beside cryptographic mechanisms watermarking represents an efficient technology to ensure both data integrity and data origin authenticity. Watermarking techniques usually used for digital imagery and now also used for audio and 3D-models are relatively young and their amount is growing at an exponential rate. It is a highly multidisciplinary field that combines image and signal processing with cryptography, communication theory, coding theory, signal compression, and the theory of visual perception. Copyright, customer or integrity information is embedded by use of a secret key into the media data as transparent patterns. Because the security information is integrated into the media data one cannot ensure confidentiality of the media data itself but for the security information by use of the secret key.

Based on application areas for digital watermarking the following watermarking classes are defined:

- Authentication watermark: Ensures copyright protection by watermarking the data with an owner or producer identification.
- Fingerprint watermark: Ensures copyright protection by watermarking the data with customer identifications to track and trace legal or illegal copies.
- Copy control or broadcast watermark: Ensures copyrights with customer rights protocols, for example for copy or receipt control.
- Annotation watermark: Ensures copyright protection by annotations or capturing of the media data. This kind of watermark is also used to embed descriptions of the value or content of the data.
- Integrity Watermark: Beside the authentication of the author or producer, it ensures integrity of the data and recognizes manipulations.

The most important properties of digital watermarking techniques are robustness, security, imperceptibility / transparency, complexity, capacity and the possible verification procedure.

- Robustness describes if the watermark can be reliably detected after media operations. We emphasize that robustness does not include attacks on the embedding scheme that are based on the knowledge of the embedding algorithm or on the availability of the detector function. Robustness means resistance to "blind", non-targeted modifications, or common media operations.
- Security describes if the embedded watermarking information cannot be removed beyond reliable detection by targeted attacks based on a full knowledge of the embedding algorithm and the detector, except the secret key, and the knowledge of at least one watermarked data. The concept of security includes procedural attacks, such as the IBM attack [3], or attacks based on a partial knowledge of the carrier modifications due to message embedding [10] or embedding of templates [37]. The security aspect also addresses the false positive detection rates.
- Transparency is based on the properties of the human visual system or the human auditory system. A transparent watermark causes no artefacts or quality loss.
- Complexity describes the effort and time needed for watermark embedding and retrieval like for encoding and decoding of JPEG images or MPEG streams. This parameter is essential for real time applications. Another aspect addresses if we need the original data in the retrieval process or not. Here we distinguish also between non-oblivious and oblivious (blind) watermarking schemes which influences the complexity.
- Capacity describes how many information bits can be embedded. It addresses also the possibility of embedding multiple watermarks in one document in parallel.
- The verification procedure describes whether the verification can be performed by a secret verification based on a private-key cryptosystem or by a public verification based on a public-key cryptosystem.

The optimization of all parameters is mutually competitive and cannot be clearly done at the same time.

4 Confidentiality

Confidentiality can be achieved by means of cipher systems. These systems are used to keep information secret from unauthorized entities.

A cipher system consists of a set of functions, parameterized by a key K1 of the key space (encryption functions), and a set of functions, parameterized by a key K2 of the key space (decryption functions). The functions have the property, that for each key K1 exists a key K2 such that encryption function and decryption function are inverse to each other.

The data to be encrypted (plaintext) are transformed by the encryption function parameterized by key K1. The result of this transformation is called ciphertext or cipher. The plaintext can be recovered by a decryption function parameterized by key K2.

Private-key and some public-key cryptosystems can be used for cipher systems. In addition, there exist so-called session-key systems (also known as hybrid cryptosystems), which employ both types of cryptosystems. Because of the importance of session-key schemes a more detailed discussion is given in the following section. The second section describes partial encryption.

4.1 Session-Key Scheme

In consideration of performance[1] large amounts of data are enciphered by a **session-key scheme**. This scheme applies both a private-key and a public-key cryptosystem to an encryption scheme (see figure 1, x‖y defines the

[1] Example [22]: In hardware, DES is about 1000 times and, in software, about 100 times faster than RSA.

concatenation of x and y, // symbolizes the communication channel).

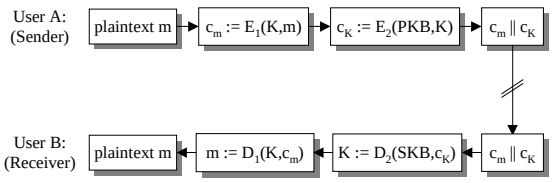


Figure 1: Session-key scheme

Plaintext m shall be encrypted by use of a session key, which is used for the secret key of a private-key cryptosystem. This key is generated in form of a random number by the originator of m during the beginning of each communication (session). The key is only valid within one session.

User A (Sender) encrypts the plaintext m with the encryption function E_1 parameterized by key K. To transmit this key to the recipient in a secure way, a public-key cryptosystem is used: session key K is encrypted with encryption function E_2, parameterized by the public key PKB of the receiver, user B. Then the ciphertexts c_K and c_m are transmitted to B.

In a first step user B (receiver) recovers the session key K by decrypting the key-ciphertext c_K: he computes K by using the function D_2, which is parameterized by his private key SKB corresponding to PKB. Then he computes the plaintext m from the encrypted data c_m by use of function D_1 of the private-key cryptosystem, parameterized by key K.

Based on the combination of private-key and public-key cryptosystems described above, the key exchange problem of secret keys with respect to private-key cryptosystems can be solved. Given that the public key has been exchanged authentically, it ensures that only the legal owner of private key SKB is able to recover the secret key K used for encryption. A possible solution for this problem will be given in section 9. Besides that it ensures that only the legal owner of private key SKB is able to recover secret key K used for encryption.

Some examples of private-key cryptosystems which are used for cipher systems are DES [24], triple-DES [1] and IDEA [21]. Examples of public-key cryptosystems, used for encryption schemes, are RSA [33] and ElGamal [8]. Commonly used combinations for session-keys schemes are DES with RSA or IDEA with RSA.

These mechanisms provide no protection after deciphering, for example to check if the data is presented in an unchanged form or who is the owner of the data to ensure copyright protections.

4.2 Partial Encryption

Within streaming applications usually a huge amount of data needs to be transmitted from a sender to a receiver in a time-critical and confident manner. Even the described session-key scheme fails to support the necessary level on performance. Another problem deriving from a huge amount of data is that media data can be changed due to transmission errors, higher compression rates or scaling operations during transmission or life time. Using common encryption methods the decryption of a ciphertext block may fail, because these methods have the property that the original plaintext cannot be recovered if one bit of the ciphertext block is altered.

A general solution of these problems is partial encryption: instead of encrypting the whole amount of data only special parts of the entire data are enciphered. If the selection is well chosen, a sound confidentiality of the whole data can be achieved.

Considering the results from performance measures in secure video systems, several methods for partial encryption of video data have been proposed in the last few years [20]. The basic idea of the approaches is to encrypt only relevant information, for example motion vectors, coefficients or header information. MPEG-1/MPEG-2 and H.261/H.263 are widespread compression standards used in most of today's video conferencing applications. They are well suited for partial encryption because on the one hand they make use of DCT, which has a high potential for dividing data in more relevant less relevant parts (entropy of the coefficients). On the other hand, large amounts of video data are encoded by reference to preceding or succeeding blocks (intracoded blocks), where only the referenced blocks have to be protected.

There are several sophisticated approaches for applying partial encryption to non-scalable standard-based hybrid video coding schemes like MPEG video. Base layer encryption does not require content parsing and, therefore, has a much lower overall computational complexity than partial MPEG encryption. Note that for base layer encryption the amount of encrypted data has to be determined a priori whereas partial MPEG encryption allows different security levels even if a video has already been encoded.

5 Data Integrity

The integrity of data can be checked by means of one-way hash functions (short: hash functions). Furthermore, some mechanisms presented in section 6 can be applied for detecting alteration of data. These mechanisms cannot prevent data manipulations, but they make these manipulations detectable. Therefore they are called detection mechanisms. The protected data still remain in plaintext.

Synonyms for hash functions are manipulation detection code (MDC), message digest, digital finger print, cryptographic checksum or message integrity code (MIC).

Hash functions possess the characteristic described in section 3.1: the image H(m) can be computed easily, but that it is computationally infeasible to find any preimage m such that m = H(m). A hash function H maps strings of arbitrary length to strings of a maximum or fixed length. Regarding binary strings used as input, H can be defined as follows: $H: \{0,1\}^* \rightarrow \{0,1\}^n$, where n typically assigns one of the values 64, 128 or 160. A hash function reduces the data m to its so-called hash value h := H(m).

Since there exist infinitely many strings of arbitrary length, but only finitely many strings with a length $\leq n$, it is obvious that so-called collisions exist, where different input values are mapped to the same hash value. However, hash functions must have the property of collision resistance: it must be hard to find two different pre-images m_1 and m_2 which are mapped to the same hash value $H(m_1) = H(m_2)$.
Figure 2 illustrates the use of a one-way hash function: The originator of data m computes the hash value of m (precisely: of a copy of m) by h:=H(m). Then he appends h to the data m and sends the concatenation m||h to the receiver.

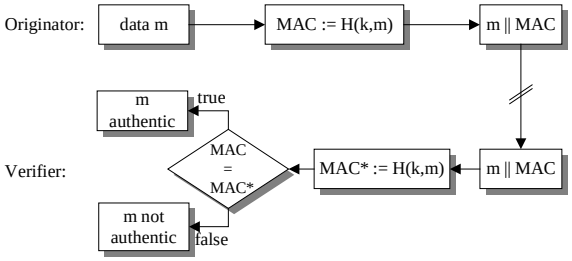


Figure 2: One-way hash function H

To verify data integrity of m (and h), the received hash value h is compared with the newly computed hash value h*=H(m). If h equals h*, the data (and also the hash value) are considered to be unchanged. This is due to the fact that the modification of even one bit in m leads to a different hash value H(m). In addition to the above explained collision resistance property, hash functions must fulfil the following criterion: whenever one input bit is changed, every bit of its hash value will change with probability of 1/2 (avalanche effect).
Hash functions are public, i.e. no secret information is used for the computation of a hash value. Thus, everyone knowing the function may compute the hash value and thereby check the integrity of the data.
Regarding multimedia applications, media data can be changed by compression or scaling without content manipulation. Therefore, hash functions are not very appropriate if they are applied to media data directly. To solve the problem hash functions should be applied to data concerning the semantic of the media stream. These data are called feature codes, which represent the content of the media. For image data e.g. DCT coefficients [22] or edges [6] can be used. The issue "content extraction" is discussed in more detail in section 6.3.
Some examples of hash functions are MD5 [32], RIPEMD-128 [7], RIPEMD-160 [7] and SHA-1 [25].

6 Data Origin Authenticity

The following mechanisms assure data origin authenticity:

- message authentication codes (MAC),
- digital signatures (especially content-based digital signatures),
- fragile digital watermarkings, and
- robust digital watermarkings.

Additionally, the first three mechanisms ensure also data integrity. Similar to the mechanisms for data integrity all four mechanisms are detection mechanisms and the protected data remains in plaintext.
The following three sections explain the main cryptographic mechanisms, namely MACs and digital signatures, as well as content-based signatures for media data. In the fourth section fragile digital watermarkings are discussed. Section 6.5 introduces robust digital watermarkings for copyright protection, owner or customer authentication.

6.1 Message Authentication Code

A message authentication code (MAC) is a one-way hash function h = H(k,m), which is parameterized by a secret key k. The security of a MAC depends on the length of the generated hash value as well as on the quality of the used key k. Only those entities that know the secret key k may calculate the MAC.
The mechanism works as follows (see figure 3):
The originator who wants to protect the data m calculates a checksum of m using a one-way hash function and the key k, i.e. he computes MAC := H(k,m). Anyone who owns key k can check the data m for authenticity. For this the verifier computes a checksum MAC* := H(k,m). If this value corresponds to the original MAC, the data m (and also the MAC) are authentic. Otherwise either m or the MAC has been changed in the time period between the generation of the MAC and its verification process.

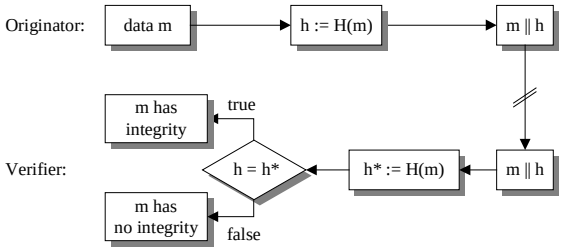


Figure 3: Message authentication code (MAC)

It is important to note that for this mechanism to work at least two parties, namely the originator and the verifier, need to hold the same key k. Thus, a MAC can not be used to prove anything (e.g. transmission or authenticity) to a third party.
A simple hash function commonly used to compute a MAC is based on a block cipher operating in the cipher-block-chaining mode (CBC-based MAC, see figure 4). Data m is divided into n blocks of the same length, determined by the domain of the block cipher (for example 64-bit blocks): $m = m_1||m_2||...||m_n$. If necessary, the last block m_n is padded with a number of padding bits to extend it to the required length. Each block m_i is linked in some way to the previously generated ciphertext block c_{i-1} (i>1) and encrypted with the encryption function E parameterized by a secret key k. The last ciphertext block c_n forms the resulting MAC (sometimes the MAC is defined by a part of this ciphertext block).

If the key is publicly available, the MAC can be taken as a manipulation detection code.

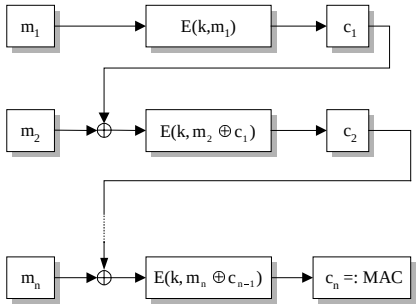


Figure 4: CBC-based MAC

6.2 Digital Signatures

The idea and the term "digital signature" were introduced by Diffie and Hellman. In [5] they suggest the following: The digital signature of an entity A (the signer) to data m shall depend on the content of m and additionally, on some secret information only known to the signer. Each user shall be able to verify the authenticity of the signature created by A (verification), by using a publicly available information of A. Since only A possesses the secret information, only he is able to create the signature to m by using the signing function S. Therefore, unlike the MAC, the digital signature may be used to prove some fact (origin, authenticity) to a third party.

The functions used for generating a digital signature are called trapdoor one-way functions These functions are one-way functions in the following sense: given a pre-image x it is easy to calculate the image f(x), but it is computationally infeasible to find a pre-image x for any given f(x). However, if some additional information y (called the trapdoor information) is known, it is easy to compute x.

Public-key cryptosystems can be used to generate and verify digital signatures. The private key SK of a user represents the secret information, and the public key PK the publicly available information.

Sometimes a MAC generated with a private-key cryptosystem is called "digital signature". But this does not have one of the most important properties of a signature, namely that it can only be generated by one entity.

Some examples of a public-key cryptosystem which can be used for digital signatures are RSA [33], DSS [28], ElGamal [8], GMR [12] and Fiat-Shamir [9].

The document m to be signed may not exceed a certain size, which is determined by the domain of the employed digital signature scheme. For example, some functions used in a digital signature scheme operate on the finite set of integers $\mathbb{Z}_n \rightarrow \mathbb{Z}_n$ where $n = p \cdot q$ or $GF(p) \rightarrow GF(p)$ where p and q prime.

Thus, for signing and verifying data m outside the range of the signature function there are two possibilities. One is to split the data m into blocks $m_1,...,m_k$ with e.g. $m_i < n$ and sign each block separately. The other, commonly used possibility is to use a hash function to reduce m to a value $H(m) < n$ which can then be signed. This increases both the security and the performance. For example, it is no longer possible to change the order of the signed blocks (and thereby the signed data). Thus, the signature is not calculated from the data itself, but from the hash value of the data.

Hash functions used in digital signature schemes are for example MD5 [32], RIPE-MD 128 [7], RIPE-MD 160 [7] and SHA-1 [25].

For protecting the authenticity of data by digital signatures the following steps are performed (see figure 5). The description given here is limited to a simple scheme of a digital signature (e.g. RSA [33]).

Signer A wants to transmit data m and its signature to a verifier. For this A computes the hash value h of m (precisely: of a copy of m) by means of a hash function h := H(m). Then A calculates the value s := S(SKA,h) by applying the signing function S to H(m) and a secret value only known to him (his private key SKA). Finally A transmits m and the corresponding digital signature s to the verifier.

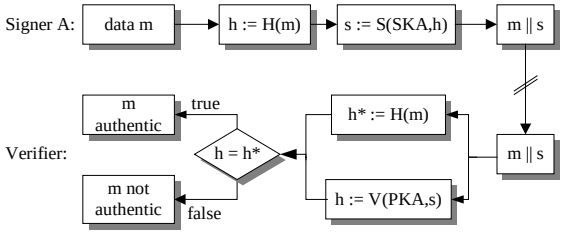


Figure 5: The principle of a digital signature

The verifier needs to know the public key PKA of A, the hash function H and the verification function V. First he computes a hash value h* := H(m) of the received data m. Then he transforms the received signature using the verification function and the signer's public key, i.e. he calculates h = V(PKA,s). Finally, he compares the values h and h*. If h = h*, A's signature is correct, meaning that neither the data nor the signature have been altered after their generation. Since A is the only one being in possession of the private key SKA, only A can compute the correct signature s to m. If $h \neq h^*$, the signature is considered as false and the data as not authentic. This can be caused for example by the modification of the data m or the signature s in the time between the signing and verifying process, or by a public key not corresponding to the private key used for the signature generation.

Besides this relatively simple possibilities for computing and verifying signatures (signature with appendix [18]) there are further, more complex methods, which concern the signature's format (like signature giving message recovery or signature giving limited message recovery [15]).

If data needs to be transmitted confidentially as well as authentically, the sender signs the data with his private key and then encrypts m together with the signature using the recipient's public key.

6.3 Content-based Digital Signatures

Regarding multimedia data, the described digital signatures could be used e.g. for image/video authentication to ensure trustworthiness by means of public-key cryptosystems. However, applying digital signatures directly to digital image data is vulnerable to image processing techniques like conversion, compression or scaling. The image material is changed irreversible without content modifications. Although the content of the image has not been changed and the viewers still have the same image impression, the signatures verification would fail. Manipulations can be differed from content-preserving and content-changing manipulations, see tables 1 and 2.

Content-preserving manipulations	Content-changing manipulations
Transmission errors–Noise Data storage errors Compression and quantization Brightness reduction Resolution reduction Scaling Color convertions γ-distortion Changes of hue and saturation	Removing image objects (persons, objects, etc.) Moving of image elements, changing their positions Adding new objects Changes of image characteristics (color, textures, structure, impression, etc.) Changes of the image background (change of the day time or location (forest, ocean)) Changes of light conditions (shadow manipulations, etc.)

Table 1: Content-preserving and content-changing manipulations

Content-preserving image effects	Content-changing image effects
Loss of details and depth of focus Loss of color resolution, color shifting Whole image effected (except of transmission error rates)	Mostly no loss of details and depth of focus Changes influences usually only image parts All changes manipulate the image content

Table 2: Effects of content-preserving and content-changing manipulations

Digital signatures should be applied to the feature codes of the media data (see figure 6). Such feature codes have to be used, which are not altered by the allowed operations such as scaling and conversion of media formats. Because feature codes should represent the content of the media these mechanisms are called content-based authentication codes or content-based digital signatures.
The main concept for image authentication is to extract the image characteristics of human perception, called content. Digital signatures are expected to survive only acceptable transcoding or compression and reject other manipulations. Very important is that content-based digital signatures cannot prevent forgery, but can be used to determine whether an image/ video is authentic or not.

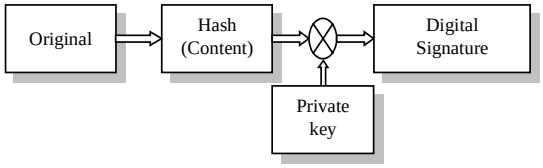


Figure 6: Content-based digital signatures

Content-based signatures can be classified into the content, the image attributes, which is used as input for the digital signature algorithm. The content extraction is called feature extraction. First approaches can be found in [2], which are based on histogram techniques.
In order to judge the usability of feature codes, their manipulation weakness and robustness against most critical content-preserving alterations has to be examined. The most content-preserving alterations are scaling and quantization as they have rather powerful effects on the data and they are used rather frequently.
Today several main approaches for feature code extraction are defined. One approach based on DCT-coefficient characteristics was introduced by [22, 23, 36], another method based on intensity/color/luminance histograms or textures information was introduced by [2, 34, 39], and a additional technique uses edges [6].

6.4 Fragile Digital Watermarking

Media integrity by use of digital watermarking is different from the introduced cryptographic mechanisms of hash functions, message authentication codes, digital signatures, and content-based digital signatures, where the check value is appended to the data. Watermarking uses redundant information of media data to slightly modify the media and embed integrity information. The integrity verification data is embedded in the media rather than appended to it. Possessing the appropriate secret key K it is possible to verify the watermark and to evaluate whether the data was altered – particularly tampered – or not by checking the embedded information. The use of public-key cryptosystems is in the moment unknown for fragile watermarking.
Several techniques and concepts have been introduced for image and audio data as fragile digital watermarks using private-key cryptosystems. The existing approaches have different strategies of tamper detection. Some approaches are very sensible to changes like in check values, others try to recognize only content changes [11, 38].
The latter approaches are usually called content-fragile watermarks. The problem to embed the content as a watermark is that watermarking techniques usually cannot embed more than 10 to 100 bytes. Therefore, it is impossible to embed the content with a data rate higher than 1 kByte. The solution for content-fragile watermarkings combines a robust watermarking technique (see section 6.5) and the content characteristic for integrity detection. The main idea is to initialize a robust watermarking pattern with the content of the media. E.g. in [6] edge characteristic of images are used. If the copyright holder

wants to know if there is an image content violation, he can search for the content dependent pattern in the found dependent on the actual edge characteristic of this image and his secret key. If he is not able find the watermark, the image seems to be manipulated.

6.5 Robust Digital Watermarking

A robust mark is designed to resist attacks that attempt to remove or destroy the mark. The intention is to embed owner, producer or customer identification into the media data to ensure copyrights using a private-key cryptosystem. Secure public-key techniques are also not known today like for fragile watermarks.

The robust watermark should remain present even after media processing or attacks, even if the content is manipulated. Today a great variety of approaches can be found in research and industry. A number of techniques require the original image in the detection process. Watermarking methods which do not require the original image in the detection process are called blind schemes. The algorithm can be classified into spatial domain, transform domain and morphological transformation approaches. Spatial domain techniques embed the watermark information directly into the pixel values, mostly by adding a modulated signal to the brightness and/or one of the color bands. Transform domain approaches work in the frequency domain or in transformations according a wavelet base. Morphological transformations work on vectored domains and are sometimes similar to the spatial domain approaches. Additionally, these techniques use the semantic of the data.

Altogether several attacks exist to remove the watermark and to destroy the owner identification. A well known attack is the Stirmark tool [30]. The tool performs random geometric distortions on digital image data. The existing schemes today have several problems during the detection process to synchronize the watermark information when the media data is altered. A lot of efforts is spend towards devising an efficient and robust watermarking method. But none of the existing techniques seems to be robust against all possible attacks and more research is necessary.

7 Entity Authenticity

As described in the previous paragraph, data authenticity can be checked by digital signatures and especially for media data by content-based digital signatures. Additionally, it is often necessary to ensure the authenticity of entities, e.g. for guaranteeing that the communicating parties (this may be persons as well as devices) are indeed the ones they claim to be. Schemes enabling such a proof are called authentication protocols. The data which is transmitted between the parties during the protocol may contain additional text fields. These fields may be used to exchange secret keys for a further confidential communication.

In the following the simplest version of an authentication protocol is presented: the challenge-response protocol. This protocol can be implemented on the basis of a private-key or a public-key cryptosystem (see figures 7, 8 and 9) [19].

Basically such a protocol works as follows: The verifier sends to the claimant a randomly generated number, the so-called challenge. The claimant returns a response to the verifier which consists of a ciphertext generated by using the challenge. For each authentication a new question is generated, thus this kind of authentication is called dynamic authentication.

Authentication is subdivided into unilateral and mutual authentication. Within the unilateral authentication an entity proves to another entity its authenticity, within the mutual authentication both entities prove their authenticity mutually.

Within a **challenge-response protocol based on a private-key cryptosystem** the two entities use the same encryption/decryption algorithm, E respectively D parameterized by a secret key K. In the following the unilateral authentication according to ISO 9798-2 is described (see figure 7).

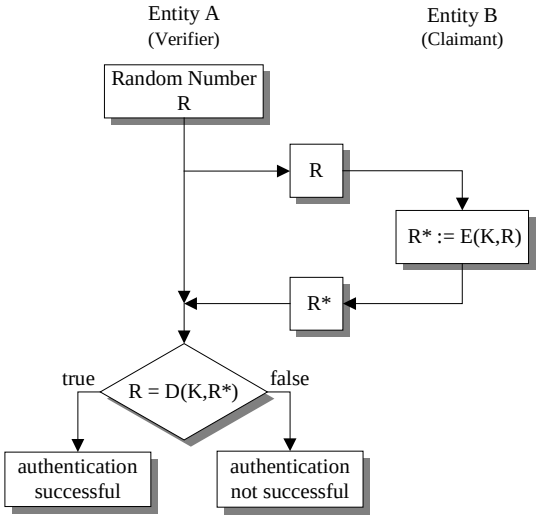


Figure 7: Unilateral authentication using a private-key cryptosystem

Entity A (verifier) wants to check the identity of entity B (claimant). For this, A generates a random number R (challenge), and transmits it to B. Entity B encrypts this random number by means of an encryption function E parameterized by key K. Then he sends the resulting cipher R* (response) to A. Entity A decrypts the received cipher by use of D and key K. Then he checks if the calculated value corresponds to the random number R. If so, claimant B is considered to be authentic.

Since each entity possesses the same key, high security requirements result on the storage of the key. The need of user A and user B to hold the same key may be overcome by the so-called derived key concept: individual keys, which are derived from master keys and some additional information, are used within the challenge-response protocol. Let us assume the master key MK is stored by entity B. Entity A possesses an individual key IK, which can be calculated by B using MK and data provided by entity A. For this, A transmits unique data

describing his identity (IDA) to B. IDA is used as an argument of the calculation of the derived key: IK = f(MK,IDA). Finally both A and B share a common secret key, which may be used within a challenge-response protocol.

If two entities want to authenticate themselves mutually, there exist two possibilities. The straight forward solution is to process the presented unilateral authentication twice with reversed roles of claimant an verifier in the second run. In order to simplify this protocol and to reduce the transaction time, the following authentication protocol is used for mutual authentication (see figure 8):

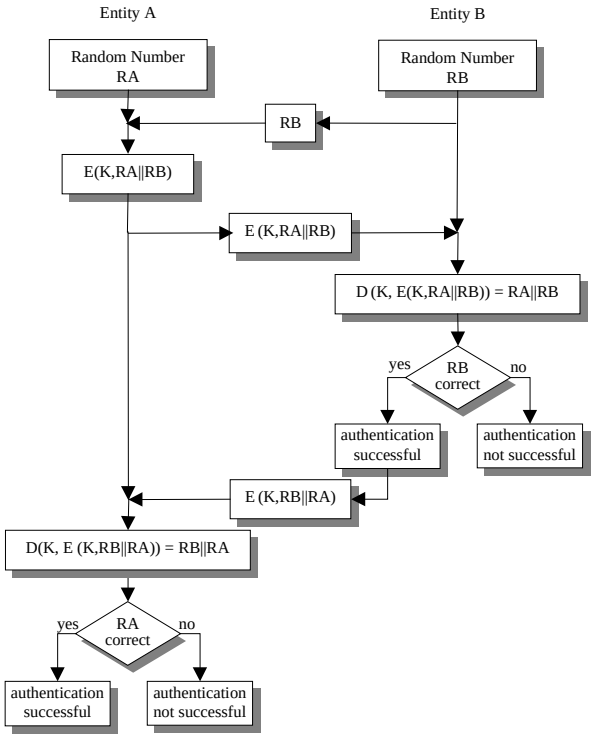


Figure 8: Mutual authentication using a private-key cryptosystem

Both entities A and B generate a random number RA and RB, respectively, and B sends its random number to A. A encrypts the concatenation RA||RB and transmits the cipher E(K,RA||RB) to B. Entity B decrypts the cipher and checks if the resulting second integer corresponds to the random number RB generated by himself. If so, B encrypts the concatenation RB||RA and sends the cipher E(K,RB||RA) to A. Entity A decrypts the cipher and performs the equivalent check. If both checks succeed, A has proven his authenticity to B and vice versa.

Since the transmitted data are depending on each other and thus no instruction can be inserted unnoticed during the protocol, the security of the authentication protocol increases.

Private-key cryptosystems, which are used for cipher systems, are e.g. DES [24], triple-DES [1] and IDEA [21].

Challenge-response protocols based on a public-key cryptosystem use the fact that digital signature are appropriate for authentication protocols. Here, two different keys are used: the public key and the private key of the claimant. The unilateral authentication is performed as follows (see figure 9):

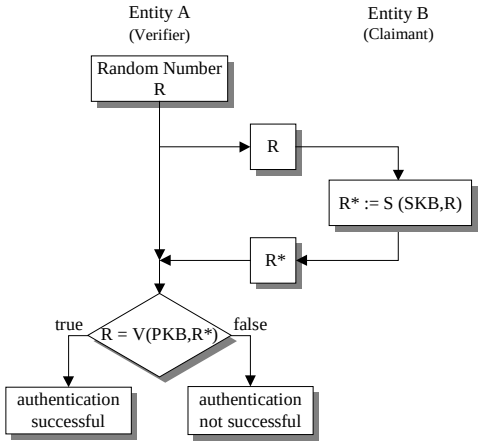


Figure 9: Unilateral authentication using a public-key cryptosystem

Entity A wants to verify the identity of entity B. First A obtains B's public key PKB, e.g. provided by public-key directory. Then A generates a random number R and transmits R to B. Entity B signs R by means of the signature function S and his private key SK. Subsequently, he transmits the result R* to A. By the use of the verification function V and B's public key PK, A verifies the received signature, by checking if R corresponds to the value calculated by him. If so, B is considered authentic.

Besides the simple authentication protocols described in this paper there exist more complicated protocols which are discussed in the standard ISO/IEC 9798 [19]. Here five methods are defined: the unilateral one pass authentication, the unilateral two pass authentication, the mutual two pass authentication, the mutual three pass authentication, and the mutual two pass parallel authentication.

Some examples of public-key cryptosystems, which are used for digital signatures, are RSA [33], DSS [28], ElGamal [8], GMR [12] and Fiat-Shamir [9].

It is important to note that the above described authentication protocols are not secure in general. If both A and B are able to start the protocol, and additionally, the received random number is accepted as a challenge without any check, then the following attack, the so-called replay attack, may be performed: Verifier A transmits a random number R1 to the claimant, which is intercepted by some adversary X. In the role of the claimant, X sends R1 to A by starting a second protocol run. Then entity A as claimant encrypts the random number R1 and transmits the cipher R1* to X as the verifier of protocol run 2. This terminates the second protocol run, and adversary X can use R1* to send, again adopting the role of the claimant of the first run, R1* to verifier A. A will then consider the communication as authentic.

In order to prevent this (and other possible) attacks, the unique identification number of the verifier and/or

claimant are added to the transferred data [19]. Using timestamps instead of random numbers disables replay attacks as described above, but this will raise the problem that A and B have to be equipped with synchronized clocks.

8 Non-Repudiation

Within legal facilities digital signatures in their own are not sufficient to link data and actions to their originators. The two following examples may clarify this:

- A sender may disavow that he signed a particular message, e.g. by publishing his private key anonymously, and then claiming the key has been lost or stolen. Thus, he may also declare that the signature of the message has been forged.
- A sender may claim that messages, which were already signed by him before the compromising of his private key, are forged. To achieve this, he simply attaches an earlier timestamp to already signed messages and signs them again. Now he may claim that the signatures have been forged.

Here, security infrastructures and security techniques may be used to provide some evidence that will be accepted by courts. So-called non-repudiation mechanisms [14], which are based on private-key cryptosystems (message authentication code) or public-key cryptosystems (digital signatures), are supporting such security techniques. They comprise non-repudiation certificates, non-repudiation tokens and protocols. Trusted Third Parties (TTP) supply notary services, timestamping services and evidence recording. By means of these mechanisms it can be proven to involved parties and third parties whether or not a particular event occurred or a particular action happened. The event or action may be generating a message, sending a message, receiving a message or transmitting a message. Therefore, these mechanisms are subdivided into:

- non-repudiation of origin,
- non-repudiation of delivery,
- non-repudiation of submission, and
- non-repudiation of transport.

In the following we will give an example of non-repudiation of origin by use of arbitrated digital signatures (see figure 10).

Entity A wants to transmit data to entity B, whereby A must not be able to repudiate being the originator of the data. Sender A possesses an identity string IDA, which uniquely describes his identity. First A signs the data m by using his private key SKA. Then he signs the concatenation IDA||m||s_m, and transmits it together with its signature s to a trustworthy third party, the arbiter Z. Arbiter Z checks IDA and verifies the signature s of the data IDA||m||s_m generated by A. If all checks are successful, the arbiter Z attaches a timestamp T to the data IDA||m||s_m and signs these sequence, too. Now, he transmits the signed data to entity B. Receiver B verifies the signature of Z, checks IDA for correctness and finally verifies the signature s_m of A. If all checks are correct, A can not deny to be the originator of the data.

9 Public-Key Infrastructure

The use of public-key cryptosystems raises the following problems:

- By means of session-key schemes the encrypted session key (and thus the plaintext) may be recovered only with the private key of the recipient (so-called addressed confidentiality). However it cannot be ascertain whether or not the public key, which is used for the encryption of the session key, actually belongs to a particular person (or device).
- By use of digital signatures and signature-based authentication protocols it can be checked whether the signature to particular data was generated by a specific key by verifying the digital signature. Thus, the authenticity of a message or communication can be proven. However it is not provable whether or not the used keys actually belong to a certain person.

Obviously, an authentic link between the public key and its owner is needed. Such a link is provided by so-called public-key certificates [16, 17]. For the issuing of certificates a trustworthy authority, a so-called trust center (TC), is needed. Trust centers authenticate the link of

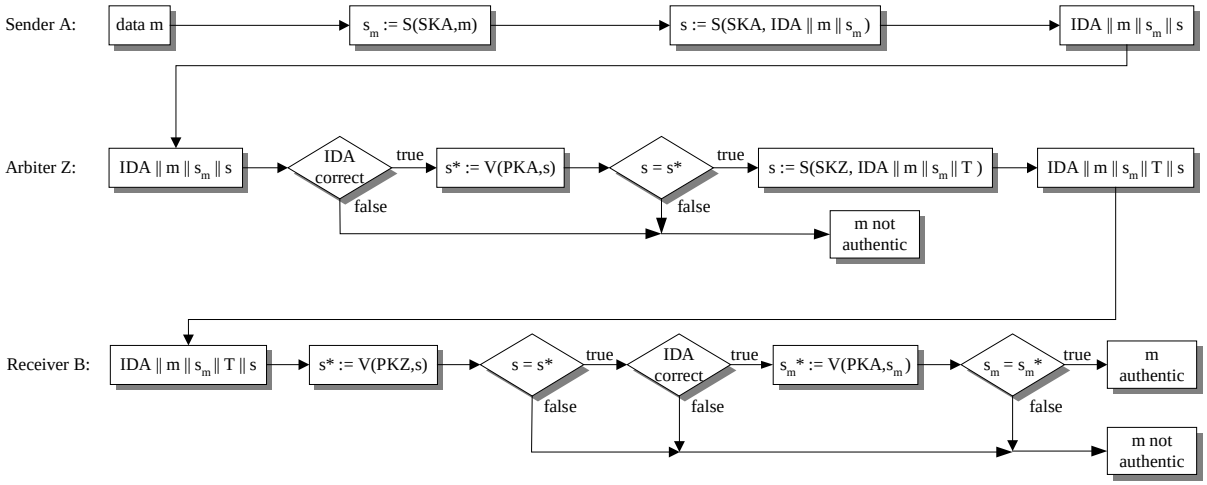


Figure 10: Arbitrated digital signature

users to their public keys, and can provide further services like non-repudiation, revocation handling, time-stamping, auditing and directory service.
Within a trust center these services are provided by special components. Each trust center, and even its components, comply with a so-called security policy. This policy regulates the generation and distribution of certificates, and how to ensure the availability of the services.

10 Conclusion

Whether or not the presented security mechanisms can be used easily for multimedia systems and on which features the mechanisms need to be based on must be examined for each kind of multimedia data and multimedia applications separately. All together the main difference to non-media data is, that cryptographic mechanisms usually have to be applied to special parts of the media data:

- For providing confidentiality: Instead of encrypting the whole data only special parts of the entire data are encrypted (partial encryption), which ensures a high performance. If the selection is well chosen, a sound confidentiality of the whole data can be achieved.
- For providing authenticity: Message authentication codes and digital signature schemes have to be applied to the content, the feature code, of media data, which is robust to allowed media operations.
- For providing data originality: New techniques have to be developed which can guarantee that data are presented in an unchanged form and not in a copy, and therefore support copyright protection.

In order to contribute non-repudiation services for cryptographic mechanisms as well as for watermarking techniques a proper security infrastructure has to be established and proper security policies must be defined.

11 References

[1] ANSI X9.17 (Revised): American National Standard for Financial Institution Key Management (Wholesale). American Bankers Association, 1985.

[2] Chang, Shih-Fu, Schneider, Marc: A Robust Content-Based Digital Signature for Image Authentication. Proc. of the International Conference on Image Processing, Lausanne, Switzerland, Sep. 1996.

[3] Craver, S.; Memon, N.; Yeo, B.; Yeung, M.: Can Invisible Watermarks Resolve Rightful Ownerships? Technical Report RC 20509, IBM Research Division, Jul. 1996.

[4] Department of Defense: Department of Defense Trusted Computer System Evaluation Criteria (Orange Book). DOD 5200.28-STD, Dec. 1985.

[5] Diffie, Whitfield; Hellman, Martin E.: New Directions in Cryptography. IEEE Transactions on Information Theory, Vol.22, Nr.6, 11/1976, pp.644-654.

[6] Dittmann, Jana, Steinmetz, Arnd, Steinmetz, Ralf: Content-Based Digital Signature for Motion Pictures Authentication and Content-Fragile Watermarking. Proc. of ICMCS'99, Florence, Italy, 1999.

[7] Dobbertin, Hans; Bosselaers, Antoon; Preneel, Bart: RIPEMD-160: A strengthened version of RIPEMD. Fast Software Encryption, Cambridge Workshop 1996, Md. 1039, Berlin: Springer 1996, pp.71-82.

[8] ElGamal, Taher: A Public-Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. IEEE Transactions on Information Theory, Vol.31, Nr.4, Jul 1985, pp.469-472.

[9] Fiat, Amos; Shamir, Adi: How to prove yourself: Practical solutions to identification and signature problems. Advances in Cryptology – Crypto'86 Proceedings, LNCS 263, Springer, pp.186-194.

[10] Fridrich, J.; Goljan, M.: Protection of Digital images Using Self-Embedding. Symposium on Content Security and Data Hiding in Digital Media, New Jersey Institute of Technology, May 14, 1999.

[11] Fridrich, Jiri: Applications of Data Hiding in Digital Images. Tutorial for the ISPACS'98 Conference in Melbourne, Australia, Nov. 4-6, 1998.

[12] Goldwasser, Shafi; Micali, Silvio; Rivest, Ronald L.: A 'Paradoxical' Solution to the Signature Problem. 25th Symposium on Foundations of Computer Science (FOCS), 1984, pp.441-448.

[13] Information Technology Security Evaluation Criteria (ITSEC): Provisional Harmonised Criteria. Version 1.2, Jun. 1991.

[14] ISO/IEC 13888: Information technology – Security techniques – Non-repudiation. Part 1: General (IS 1997). Part 2: Using private-key techniques (DIS 1997). Part 3: Using public-key techniques (IS 1997).

[15] ISO/IEC 14888: 1998 Information technology – Security techniques – Digital Signatures with appendix.

[16] ISO/IEC 9594-8 | ITU-T Recommendation X.509: Information technology - Open Systems Interconnection - The Directory. Part 8: Authentication Framework, 1993.

[17] ISO/IEC 9594-8 | ITU-T Recommendation X.509: Final Text of Draft Amendments DAM 1 to ITU-T Recommendation X.509 (1993) | ISO/IEC 9594-8 on Certificate Extensions: ISO/IEC JTC 1/SC 21/WG 4 and ITU-T Q15/7. Dec 1996.

[18] ISO/IEC 9796: 1991 Information technology – Security techniques – Digital signature scheme giving message recovery.

[19] ISO/IEC 9798: Information technology – Security techniques – Entity authentication. Part 1: General (IS 1997). Part 2: Mechanisms using encipherment

algorithms (IS 1994). Part 3: Mechanisms using a public-key algorithm (IS 1993).

[20] Kunkelmann, Thomas: Sicherheit für Videodaten. Vieweg Verlag, 1998.

[21] Lai, Xuejia; Massey, James: A proposal for a New Block Encryption Standard (IDEA). Advances in Cryptology – Eurocrypt'90 Proceedings, Berlin: Springer 1991, pp.389-404.

[22] Lin, C.-Y., Chang, Shih-Fu: A Robust Image Authentication Method Surviving JPEG Lossy Compression. SPIE Storage and Retrieval for Image and Video Databases, San Jose, CA, USA, Jan. 1998.

[23] Lin, C.-Y.; Chang, Shih-Fu: Issues and Solutions for Authenticating MPEG Video. SPIE Storage and Retrieval for Image and Video Databases, San Jose, CA, USA, Jan. 1999.

[24] National Bureau of Standards: Data Encryption Standard (DES). FIPS PUB 46-1, Jan. 1988.

[25] National Bureau of Standards: Secure Hash Standard (SHS-1). FIPS PUB 180-1, 17.4.1995.

[26] National Computer Security Center: Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria. NCSC-TG-021, Version 1, Apr. 1991.

[27] National Computer Security Center: Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria (Red Book). NCSC-TG-005, Version 1, Jul. 1987.

[28] National Institute of Standards and Technology: Digital Signature Standard (DSS). NIST FIPS PUB 186, May 1994.

[29] NATO: NATO Trusted Computer System Evaluation Criteria (Blue Book). NATO AC/35-D/1027, 1987.

[30] Petitcolas, Fabien; Anderson, Ross: Weaknesses of Copyright Marking Systems. Multimedia and Security Workshop at the Sixth ACM International Multimedia Conference, Sep. 12-16 1998, Bristol, England; Workshop notes published by GMD – Forschungszentrum Informationstechnik GmbH, GMD Report 41, 1998, pp.55-61.

[31] Quisquater, J.-J.; Macq, B.; Joye, M.; Degand, N.; Bernard, A.: Practical Solution to Authentication of Images with a Secure Camera. SPIE Storage and Retrieval for Image and Video Databases, San Jose, CA, USA, Feb. 1997.

[32] Rivest, Ronald L.: The MD5 Message Digest Algorithm. RFC 1321, Apr. 1992.

[33] Rivest, Ronald L.; Shamir, Adi; Adleman, Leonard A.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, Vol.21, Nr.2, Feb. 1978, pp.120-126.

[34] Schneider, Marc; Chang, Shih-Fu: Digital Watermarking and Image Authentication. 1996, http://www.ctr.columbia.edu/~mars/papers/reports/water/doc.html.

[35] Schneier, Bruce: Applied Cryptography. John Wiley & Sons 1996, p.469.

[36] Storck, D.: A New Approach to Integrity of Digital Images. Proc. of IFIP World Conference- Mobile Communication, Canberra, Australia, 1996.

[37] Voloshynovskiy, S.; Herrigel, A.; Jordan, F.; Baumgärtner, N.; Pun, T.: A Noise Removal Attack for Watermarked Images. Workshop Multimedia and Security at ACM Multimedia 1999, Orlando, Florida, 1999.

[38] Wolfgang, R.; Delp, E.: Fragile watermarking using the VW2D watermark. Proceedings of the IS&T/SPIE Conference on Security and Watermarking of Multimedia Contents, San Jose, California, Jan. 1999, pp.204-213.

[39] Zhong, D.; Chang, Shih-Fu: Video Object Model and Segmentation for Content-Based Video Indexing. IEEE International Conference on Circuits and Systems, Hong Kong, Jun. 1997.

Introduction to the Taxonomy of Multiple Cryptography

Petra Wohlmacher

University of Klagenfurt
Department of Informatics – System Security
Klagenfurt, Austria

petra.wohlmacher@uni-klu.ac.at

ABSTRACT

Today, it is a common approach to use modern cryptographic mechanisms in order to increase the security of IT-systems and their applications. Almost every researcher and developer is familiar with several kinds of different cryptographic mechanisms dealing with just two entities, one representing the "actor" and the other denoting the "reactor". However, a large number of applications requires security mechanisms which address more than two entities.

This paper gives an introduction into multiple cryptography, e.g. multiple-key ciphers, multiple-key-parameter ciphers, multiple encipherment, multiple-key digital signatures, multiple-key-parameter digital signatures, and multiple signing. Additionally, some applications are presented where this type of cryptographic mechanisms solves specific security problems if more than two entities are involved.

KEYWORDS

Multiple cryptography, multiple cipher, multiple-key cipher, multiple-key-parameter cipher, multiple encipherment, multiple digital signature, multiple-key digital signature, multiple-key-parameter digital signature, multiple signing, multisignature, secret sharing scheme, RSA scheme, DES, triple-DES, asymmetric cryptosystem, symmetric cryptosystems, society and group oriented cryptography, services-on-demand.

1 Motivation

Today, modern cryptographic mechanisms are increasingly used for enhancing the security of IT-systems and their applications. Large security infrastructures like public-key infrastructures are developed and currently going to be established for meeting security requirements [PKIX99]. For example, the following security requirements are met by the security measures listed below:

- Confidentiality: Cipher systems are used to keep information secret from unauthorized entities.
- Data integrity: The alteration of data can be detected by means of one-way hash functions, message authentication codes, and digital signatures.
- Data origin authenticity: Message authentication codes and digital signatures enable the proof of origin (and integrity) of data.

In a conventional cipher system a plaintext is encrypted by only one entity using one single key. The ciphertext as the result is decrypted by a single entity by means of one key. Additionally, in conventional signature schemes a digital signature is generated by an entity using one key for signing. The signature is verified by a single entity using one key for the verification process. Furthermore, multiple cryptography provides security mechanisms, where more than two entities and also more than one key or key pair, respectively, is involved.

The given paper is organized in two major sections. Section 2 introduces multiple cryptography. Initially, we describe the family of multiple ciphers like multiple-key ciphers, multiple-key-parameter ciphers, and also multiple encipherment. Then, we present the family of multiple digital signatures like multiple-key signatures, multiple-key-parameter signatures, and multiple signing. Finally, section 3 gives an overview of how multiple cryptography can be used in different applications, where security measures have to be involved.

2 Introduction to Multiple Cryptography

Multiple cryptography comprises a family of different kinds of cryptographic functions, which are basically dealing with more than one key or key-parameter within an atomic operation like enciphering, deciphering, signing or verifying. For instance, the number of keys is increased to more than two. The large family of multiple cryptography contains multiple-key ciphers, multiple-key-parameter ciphers, multiple encipherment, multiple-key digital signatures, multiple-key-parameter digital signatures and multiple signing.

2.1 Multiple Cipher

The term "multiple cipher" includes three kinds of techniques using more than one key or one pair of keys:

multiple-key cipher, multiple-key-parameter cipher, and multiple encipherment. It is important to mention, that multiple-key encipherment is neither similar to multiple-key-parameter encipherment nor to multiple encipherment.

In the following paragraphs, we will provide definitions for multiple-key ciphers, multiple-key-parameter ciphers, and multiple ciphering. For this purpose we distinguish between the use in symmetric cryptosystems, particularly block ciphers, and the use in asymmetric cryptosystems. Within a symmetric cryptosystem each key k and each key-parameter k_i, respectively, has to be kept secret. Consequently, these keys are named secret keys. Within an asymmetric cryptosystem some keys and key parameters are secret (sk resp. sk_i). These keys are called private keys. The remaining keys may be made public and are, therefore, named public keys pk resp. public-key parameters pk_i. Additionally, the keys have the property that, given a public key pk or even a public-key parameter pk_i, it is computationally infeasible to find the corresponding private key sk or a private-key parameter sk_j, respectively. In other words, even with the most powerful computers it is impossible to deduce sk or sk_j from pk or pk_i, respectively, during a given period of time. Multiple ciphers also have the property, that other key parameters or even one of the proper keys cannot be computed based on the knowledge of one single key parameter or key.

2.1.1 Multiple-Key Cipher

Term and definition of multiple-key cipher have been introduced by John M. Carroll [Carr84]. In the following section, we describe multiple-key ciphers more generally and even more detailed than Carroll and even Boyd [Boyd89] did.

In a multiple-key cipher using symmetric cryptosystems, the encryption function E is defined by $E: K_1 \times \ldots \times K_r \times M \to C$, where $K_1 \times \ldots \times K_r$ denotes the key space, M the message or plaintext space, and C the space of ciphertexts. Thus, the plaintext m is transformed by

$$c = E(k_1, \ldots, k_r, m).$$

The decryption $D: K_1^* \times \ldots \times K_t^* \times C \to M$ is performed by

$$m = D(k_1^*, \ldots, k_t^*, c),$$

where $D = E^{-1}$, with $r,\ t \in IN+1$ and both values do not equal 1. (Notation: IN denotes the set {0,1,2,...} and $IN + i$ describes the set {i,i+1,i+2,...}.)

The encryption function E of a multiple-key cipher using an asymmetric cryptosystem is defined by $E: PK_1 \times \ldots \times PK_r \times M \to C$, where

$$c := E(pk_1, \ldots, pk_r, m)$$

and $PK := PK_1 \times \ldots \times PK_r$ denotes the space of the public keys. The decryption is computed by

$$m = D(sk_1, \ldots, sk_t, c),$$

where $D = E^{-1}$ and $SK := SK_1 \times \ldots \times SK_t$ defines the space of private keys, $r,\ t \in IN+1$ and both values do not equal 1.

Remark: The single-key cipher, where $r = 1$ (also $t = 1$, respectively), represents the conventional encipherment.

Two examples for a multiple-key cipher using symmetric cryptosystems are given by the Henry multiple-rotor electric coding machine (ECM) of World War II (for details see [Carr84]), and the cipher machine Enigma [Kahn67].

An example for a multiple-key cipher using an asymmetric cryptosystem is given by Colin Boyd in [Boyd88], where he generalized the RSA scheme [RiSA78]. The public-key and private-key parameters are chosen to satisfy the congruence

$$pk_1 \cdot pk_2 \cdot \ldots \cdot pk_r \cdot sk_1 \cdot sk_2 \cdot \ldots \cdot sk_t \equiv 1 \mod \varphi(n),$$

where n is a product of two large primes. Therefore, the multiple-key encryption by means of the RSA scheme is defined by $E(pk_1, \ldots, pk_r, m) = m^{pk_1 \cdot \ldots \cdot pk_r} = c \mod n$. The multiple-key decryption is performed by $D(sk_1, \ldots, sk_t, c) = c^{sk_1 \cdot \ldots \cdot sk_t} = m \mod n$.

2.1.2 Multiple-Key-Parameter Cipher

In a multiple-key-parameter cipher using symmetric cryptosystems, the encryption function E is defined by $E: K \times M \to C$, and a function $f: K_1 \times \ldots \times K_r \to K$, where K_i denotes the space of each key parameter k_i ($i = 1, \ldots, r$, where $r \in IN+2$). Thus, the plaintext m is transformed by

$$c := E(k, m) = E(f(k_1, \ldots, k_r), m),$$

where $k := f(k_1, \ldots, k_r)$ represents the proper secret key of the cipher. The decryption is performed by

$$D(k, c) = D(f^*(k_1^*, \ldots, k_t^*), c),$$

where $D = E^{-1}$ and $f^*: K_1^* \times \ldots \times K_t^* \to K$.

By using asymmetric cryptosystems, a multiple-key-parameter encryption is defined by

$$c := E(pk, m) = E(f_E(k_1, \ldots, k_r), m),$$

where $f_E: K_1 \times \ldots \times K_r \to PK$, and the decryption by

$$D(sk, c) = D(f_D(k_1^*, \ldots, k_t^*), c),$$

where $f_D: K_1^* \times \ldots \times K_t^* \to SK$, $r,\ t \in IN+1$ and both values do not equal 1. The public key $pk \in PK$ and the private key $sk \in SK$ are the proper keys of the cipher.

Remarks:

- The image of f (f^*) can be pre-computed, because f (f^*) is not directly an inherent part of the cipher.
- The single-key cipher, where $r = 1$ (resp. also $t = 1$), $f = f^* = id$, and $K = K_1 = K_1^*$ (resp. $PK = K_1$ and $SK = K_1^*$), defines the conventional encipherment.

The function f (f^*) is often called secret sharing scheme. Examples for secret sharing schemes used for multiple-key-parameter ciphers are split-knowledge schemes or secret-splitting schemes [Feis70, MeOV97], threshold schemes (introduced by Adi Shamir in 1979 [Sham79], see also [Simm92]), generalized secret sharing schemes [MeOV97], and key agreement schemes [Blom83, DiHe76]. The process that makes a shared secret key available to two or more parties, is also called key establishment protocol.

Due to the multiplicative property, the RSA scheme can be used for multiple-key ciphering as well as for multiple-key-parameter ciphering, where the modulus n is fix. Thus, the multiple-key-parameter encryption by means of the RSA scheme is defined by

$$E(pk, m) = E(f_E(pk_1, \ldots, pk_r), m)$$
$$= E(pk_1 \cdot \ldots \cdot pk_r, m) = m^{pk_1 \cdot \ldots \cdot pk_r} = m^{pk} = c \mod n,$$

where f_E specifies the multiplication. Multiple-key-parameter decryption is performed by

$$D(sk, c) = D(f_D(sk_1, \ldots, sk_t), c) = D(sk_1 \cdot \ldots \cdot sk_t, c)$$

$$= c^{sk_1 \cdot \ldots \cdot sk_t} = c^{sk} = m \mod n,$$

where f_D also defines the multiplication. The common method of using the RSA scheme in practice is multiple-key enciphering.

2.1.3 Multiple Encipherment

In multiple encryption using symmetric cryptosystems, the plaintext m is enciphered more than once by cascading r identical encryption functions E:

$$c := E(k_r, E(k_{r-1}, \ldots, E(k_1, m)\ldots)).$$

Thus, E is defined by $E: (K \times M)^r \to C$, where $r \in IN + 2$. The keys $k_1, \ldots, k_r$ are inserted sequentially. The corresponding multiple decryption D is determined by

$$m = D(k_1, D(k_2, \ldots D(k_r, c)\ldots)),$$

where $D = E^{-1}$, and applying all keys in the reverse order like for encryption.

Multiple encipherment using asymmetric cryptosystems is equivalently defined by the encryption function

$$c := E(pk_r, E(pk_{r-1}, \ldots E(pk_1, m)\ldots)),$$

where $E: (PK \times M)^r \to C$, and the decryption function

$$m = D(sk_1, D(sk_2, \ldots D(sk_r, c)\ldots)),$$

where $D: (SK \times C)^r \to M$ with $D = E^{-1}$ and $r \in IN + 2$. Each public key pk_i corresponds to a private key sk_i.

Remarks:

- It is a major problem within multiple encryption, if the key space with the operation encryption is a group. In this case, there is always a key k_{r+1} leading to $E(k_r, E(k_{r-1}, \ldots E(k_1, m)\ldots)) = E(k_{r+1}, m)$. For security reasons, e.g. if this property reduces security, it has to be decided whether multiple encryption algorithms should be used in practice or not.
- The single encipherment, where $r = 1$, represents a conventional cipher.

An example for multiple encryption is double encryption using symmetric cryptosystems, defined by $c := E(k_2, E(k_1, m))$, where E denotes a block cipher, e.g. DES [NBS_77]. Another example is a variant of the block-cipher mode CBC [DaPr89].

The RSA scheme can also be used for multiple encryption. In general, any cipher with a commutative property, e.g. multiplicative or additive, may be used for multiple encryption. Moreover, there are some restrictions if the underlying plaintext space and ciphertext space are based on Galois Fields. These restrictions lead to the so called reblocking problem [MeOV97]. Multiple encipherment can easily be performed unrestrictedly, e.g. if the modulus n is fixed.

2.1.4 Combining Multiple Encipherment and Decipherment

Multiple encrypting functions can be combined with their corresponding decipher functions for computing ciphertexts. By using symmetric cryptosystems, such a resulting encryption function can be defined by

$$c := G_r(k_r, G_{r-1}(k_{r-1}, \ldots G_1(k_1, m)\ldots)).$$

The corresponding decryption function is performed by

$$m = F_1(k_1, F_2(k_2, \ldots F_r(k_r, c)\ldots)),$$

where $F_i = G_i^{-1}$ with $i = 1, \ldots, r$ and $r \in IN + 2$.

The major reason for combining multiple encipherment and decipherment is to increase security by extending the key length of a well known algorithm without developing a new one.

For combining multiple encryption and decryption, various techniques are available. A well known algorithm using symmetric cryptosystems is triple-DES [Tuch79, ANSI85, ISO_87]. Denoting the function used for enciphering by DES, the triple-DES encryption is defined by

$$DES(m) = DES(k_3, DES^{-1}(k_2, DES(k_1, m))).$$

The special case $k_1 = k_3$ is often called two-key triple-DES (in general: two-key triple encryption). Because DES is not a group [Camp93], it can be used for the combination of enciphering and deciphering, and, therefore, increasing the security of the cipher by extending the key length.

Note: In literature, the combined method often belongs to multiple encryption [MeOV97]. In this paper we will distinguish between these two kinds of techniques and use our definitions in the corresponding conceptual framework.
The combination of asymmetric cryptosystems can be defined similar to symmetric cryptosystems. Therefore, it will not be discussed in this section.

2.1.5 Coherence between the Families of Multiple Ciphers

In multiple-key ciphers as well as in multiple encipherments the inputs to each operation are keys. This is different to multiple-key-parameter cipher e.g. secret sharing schemes, secret splitting schemes or key agreement schemes. In multiple-key-parameter cipher, the inputs are key parameters and not necessarily elements of the key space of the proper scheme except the resulting value which is calculated by all input values.
If different entities are involved in a multiple-key cipher or multiple enciphering, the following problem arises: except for the first entity, each entity performing the encryption next in line is not able to recognize the plaintext, because the entity first in line encrypted the plaintext. This problem may be rather difficult to solve, e.g. one approach consists of a high degree of trust among the parties involved.

2.2 Multiple Digital Signature

Usually, digital signatures are realized through asymmetric cryptosystems. There are alternative techniques which are using symmetric cryptosystems, but these techniques are not very practical and versatile. Therefore, the following explanations address only asymmetric cryptosystems.
Since digital signature schemes need more extensive mathematical descriptions than ciphers, digital signatures are in general defined in a first step. The main components of a digital signature scheme are named as follows (for easier understanding, the definitions, which are taken partly and in pattern from [MeVO97], are simplified and generalized in a practical way – e.g. random elements within a scheme are disregarded).
The significant sets are defined as:

- M denotes the set of messages – the message space,
- M^* describes a set of elements, which can be signed – the signing space,
- SIG is the set of elements containing the signatures – the signature space,
- PK defines a set of public keys – the space of public keys,
- SK denotes the set of private keys corresponding to the public keys in PK – the space of private keys,
- H is the set of one-way functions,
- R is the set of redundancy functions.

The significant functions are defined as:

- $h \in H$ is a contracting function, which maps an element of M to an element of M^*. It is used to increase both the security and the performance of the digital signature scheme – the hash function,
- $r \in R$ is a one-to-one and onto function mapping $m \in M$ to M^* by inserting additional information like specific bits in m – the redundancy function,
- S is a function, transforming $(sk, m^*) \in (SK, M^*)$ to $s \in SIG$ – the signing transformation,
- V is a predicate, where $(pk, m^*, s) \rightarrow \{true, false\}$ and V corresponds to S – the verification transformation. The signature s of a message m is valid if and only if $V(pk, m^*, s) = true$, otherwise $V(pk, m^*, s) = false$ and the signature has to be rejected.

Each set and function is publicly known with the exception of set SK.

Digital signature schemes are classified into two main kinds of schemes [ISO_91, ISO_98]:

- Digital signature schemes with appendix. Here, the hash-function h contracts m to m*, which is transformed to signature s. The signature represents additional data (the appendix) to the original message m, where m (and h) is required as input to the verification transformation V. This type of scheme is most commonly used.
- Digital signature schemes giving message recovery. Initially, redundancy is inserted in the original message m. The result, m*, is transformed to signature s. Applying the verification transformation V the value m* is computed out of s. Therefore, no additional data is needed. Since m* is related to a redundancy scheme, the correctness of the recovered m* can be ascertained. Then, by removing the redundancy information, the original data m is obtained.

In the following section an introduction to "multisignatures" is given. Herein, signatures depend on more than one entity and provide more than one key and key parameter, respectively.
"Digital multisignatures" were presented by Colin Boyd for the first time [Boyd86]. He distinguished between threshold schemes (which can be generalized to other secret sharing schemes) and namely adapting methods. By means of threshold schemes all signers generate the signing key by inserting their secrets into the scheme in a well-defined order. This technique will be considered in the following section by the term "multiple-key-parameter signature". Boyd also gave an example for an adapting method by extending the RSA scheme. In the example, he considered digital signatures giving message recovery. We will describe this method in the context of multiple-key digital signatures.
By the term "multiple digital signature" we summarize three kinds of techniques using more than one private key or private-key parameter for the signing process and more than one public key or public-key parameter for the associated verifying process (remark: within one atomic operation, it is also allowed to apply one key if and only

if the opposite operation needs more than one key or key parameter). Equal to multiple-key ciphers, multiple digital signatures are distinguished into multiple-key digital signature, multiple-key-parameter digital signature, and multiple signing. Similar to section 2.1, it is important to mention that these three techniques are not similar to each other.
In the following we discuss digital signature schemes given message recovery.

2.2.1 Multiple-Key Digital Signature

In a multiple-key digital signature, the signing function S is defined by

$$S: SK_1 \times \ldots \times SK_t \times M^* \rightarrow SIG,$$

where t secret keys are needed for the signing transformation. The verification predicate V is performed by

$$V: PK_1 \times \ldots \times PK_r \times M^* \times SIG \rightarrow \{true, false\},$$

where $r,\ t \in IN+1$ and both values do not equal 1.

Remark: The single-key signature, where $r=1$ (and $t=1$, respectively), represents a conventional signature scheme.

A very smart example for multiple-key signatures giving message recovery is represented by the RSA scheme using one public key pk and two private keys sk_1, sk_2. The resulting scheme is also called double signature scheme [Boyd86]. The two private keys are selected randomly and the public key is then chosen to fulfill the property $pk \cdot sk_1 \cdot sk_2 \equiv 1 \mod \varphi(n)$. The first signature for redundancy message m^* is performed by $s_1 := m^{*\,sk_1} \text{ MOD } n$. The subsequent signer is able to check the signature by verifying if $s_1^{sk_2 \cdot pk} = m^{*\,sk_1 \cdot sk_2 \cdot pk} \text{ MOD } n$ is related to a redundancy scheme of the message space. If successful, he recovers the message m from m^* and gets the opportunity to recognize what he will sign. Within a next step he generates his signature by $s := s_1^{sk_2} \text{ MOD } n$. Afterwards, the signature of both entities can be verified by applying the public key pk: $\tilde{m} := s^{pk} \text{ MOD } n$ and checking if $\tilde{m}$ is related to a redundancy scheme. If the verification process results in *true*, the signature is accepted as having been created by both entities.
As mentioned earlier, the RSA scheme possesses the multiplicative property. Therefore, signatures can be forged by $m_1^{*\,x} \cdot m_2^{*\,x} = (m_1^* \cdot m_2^*)^x$. As a result, the signature for $m_1^* \cdot m_2^*$ can be obtained from those for m_1^* and m_2^*. One way to avoid this attack, is to use digital signatures with appendix, e.g. by applying a hash function to each message and subsequently signing the hash result. This improvement is not described in detail in this paper, since the main idea does not differ from the one already discussed.
Another problem within the RSA scheme is its extension to a usage of more than three keys. This problem was mentioned in section 2.1.5 and is also relevant for both digital signatures given message recovery and digital signatures with appendix. Thus, non of the subsequent signers is able to verify any signature except the last signer assuming that he has access to all necessary public keys. One solution to this problem is the usage of trustees.

2.2.2 Multiple-Key-Parameter Digital Signature

In multiple-key-parameter digital signatures, a signature is generated if and only if the signing key is computed by means of several secret-key parameters, and verified, respectively, if and only if the verifying key is reconstructed by means of the corresponding public-key parameters. Therefore, the signing function S is performed by

$$s := S(sk, m^*) = S(f_S(k_1^*, \ldots, k_t^*), m^*),$$

where $f_S: K_1^* \times \ldots \times K_t^* \rightarrow SK$. The verification predicate V is defined by

$$V(pk, m^*, s) = V(f_V(k_1, \ldots, k_r), m^*, s)$$

mapped to the set $\{true, false\}$, where
$f_V: K_1 \times \ldots \times K_r \rightarrow PK$ and $r,\ t \in IN+1$, both values do not equal 1. The public key $pk \in PK$ and the private key $sk \in SK$ represent the proper verification key and signing key, respectively, of the multiple-key-parameter signature.
The disadvantage of multiple-key-parameter signatures is that all entities involved in the verifying process need to meet each other for checking the authenticity of the signature.

Remarks:

- f is often called secret sharing scheme (for details see section 2.1.2). The image of f may also be precomputed, since f is not an inherent part of the signature scheme.
- A multiple-key signature is a special multiple-key-parameter signature.
- $r=1,\ t=1,\ f=id,\ PK=K_1$ and $SK=K_1^*$ represent a conventional digital signature scheme.

Because of the multiplicative property, the RSA scheme can be used both for multiple-key signatures schemes and multiple-key-parameter signatures schemes, assuming modulus n is fix. Providing digital signature given message recovery, the multiple-key-parameter signature is defined as follows:
the signing function by

$$S(sk, m^*) = S(f_S(sk_1, \ldots, sk_t), m^*) = S(sk_1 \cdot \ldots \cdot sk_t, m^*)$$
$$= m^{*\,sk_1 \cdot \ldots \cdot sk_t} = m^{*\,sk} = s \text{ MOD } n,$$

the verification predicate by

$$V(pk, m^*, s) = V(f_V(pk_1, \ldots, pk_r), m^*, s)$$

$$= V(pk_1 \cdot \ldots \cdot pk_r, m^*, s) = V(m^* = s^{pk_1 \cdot \ldots \cdot pk_r} \text{ MOD } n),$$

where f_S and f_V specify the multiplication.

2.2.3 Multiple Signing

In case of multiple signing, the message m is signed more than once by cascading t identical signing functions parameterized by different private keys. Each public key pk_i corresponds to a private key sk_i.

Applying signing with given message recovery, each generated signature is transformed by the signature function of the next signer i. Assuming $SIG_i \subseteq M^*_{i+1}$ (otherwise we have to define a corresponding mapping), the signing process S and the resulting signature s are defined by

$$s := S(sk_t, S(sk_{t-1}, \ldots S(sk_1, m^*)\ldots)),$$

where $s_1 := S(sk_1, m^*)$, $s_i := S(sk_i, s_{i-1})$ and $t \in IN + 2$. The corresponding multiple verifying process is performed by t verifications

$$V(pk_t, s_{t-1}, s_t) \in \{true, false\},$$
$$V(pk_{t-1}, s_{t-2}, s_{t-1}) \in \{true, false\},$$
$$\ldots,$$
$$V(pk_2, s_1, s_2) \in \{true, false\},$$
$$V(pk_1, m^*, s_1) \in \{true, false\},$$

where all verifying keys are applied in reverse order as their corresponding signing key. The verification is successful, if all t verification processes result in *true*. Here, only the last verifier is able to verify the signature of the message m^*.

Remark: The single signing, where $t = 1$, represents a conventional signature scheme.

With respect to the reblocking problem, the obvious way to perform this type of multiple signing by use of the RSA scheme is to fix the modulus n.

Using signing with appendix, each signature produced is concatenated to the original message m, and the concatenation is signed by the next signatory. Therefore, the recursively defined signing process computes the signature s by

$$s_1 := S(sk_1, m^*),$$
$$s_i := S(sk_i, (m \| s_1 \| \ldots \| s_{i-1})^*), \text{ where } i > 2,$$
$$s := S(sk_t, (m \| s_1 \| \ldots \| s_{t-1})^*)$$

where $\|$ denotes the concatenation and $t \in IN + 2$. The corresponding verification is recursively performed by t verification steps:

$$V(pk_t, (m \| s_1 \| \ldots \| s_{t-1})^*, s) \in \{true, false\},$$
$$V(pk_{t-1}, (m \| s_1 \| \ldots \| s_{t-2})^*, s_{t-1}) \in \{true, false\},$$
$$\ldots,$$
$$V(pk_2, (m \| s_1)^*, s_2) \in \{true, false\},$$
$$V(pk_1, m^*, s_1) \in \{true, false\}.$$

Here, the RSA scheme might be used without restrictions.

3 Applications by Use of Multiple Cryptography

In day-to-day life, several situations emerge, where even a group of entities instead of one entity needs to be addressed. These groups are representing e.g. governmental institutions like a senate, a council, and a jury, or municipal organizations like a fire department and a police department. Additionally, groups can consist of an enterprise like a board of directors and procurators, or business partners. In the past, there was only the world of paper documents, and often there was a great necessity for security measures like the importance of keeping information confidential and/or enabling data authenticity also for a group of entities. Today, the world of paper documents changes more and more to a paperless world – the world of electronic documents. Due to this fact, the security measures applied must be adapted also to those, which are concerning groups of entities. Based on the spreading of IT-Systems also new working areas have been developed like computer supported cooperative work (CSCW) and services-on-demand (SoD). This development leads to various applications of multiple cryptography. Cryptography, which is dealing with groups, is named "society and group oriented cryptography". This term was introduced by Yvo Desmedt in 1987 [Desm88].

Keys and key parameters of those algorithms which are used for supporting security measures may remain with a single entity or even be distributed among a set of entities. The latter case splits up the knowledge of keys and key parameters between several entities.

By distributing keys or key parameters to different entities, key recovery can be supported. Usually, key recovery is not applied to digital signatures, because the verification parameters are usually public. If these parameters get lost, the document can be signed again by its originator. Nevertheless, key recovery is very suitable for ciphers. If deciphering key(s) or key parameter(s) get lost, it is infeasible to compute the plaintext from the ciphertext. By means of key recovery, these keys can be reconstructed and, therefore, the ciphertext can be decrypted.

Depending on the application, there are different kinds of groups and, hence, different kinds of security requirements. Groups can be classified in different ways. Firstly, groups can be distinguished in groups with anonymous members, e.g. system administrators, postmasters, or known members, e.g. pop groups. Secondly,

groups can be defined by their degree of responsibility, e.g. procurators, supervisors, or departmental chiefs. Thirdly, groups can be differentiated by their privileges, e.g. by their right to access applications or databases.
Some examples for applications are given in the following subsections.

3.1 Applications by Use of Multiple Ciphers

Multiple ciphers are suitable for information, which originates from a group (or a single entity) and needs to be kept secret from others with the exception of a dedicated entity or group. By means of data encryption the information is only granted to dedicated entities.
The obvious method for granting access to information separately is to encrypt data for each entity individually. The problem coming along with this method is that each encryption requires a separate key and, therefore, the number of keys increases according to the number of entities. Multiple ciphers can solve this key management problem, e.g., by making use of special mathematical properties of algorithms.
The different kinds of multiple ciphers are exemplarily used as follows:

- Multiple-key cipher: By using multiple-key ciphers encryption keys are distributed among the members of a group, which is enciphering data. Particularly, encryption keys of an asymmetric cryptosystem, which are usually public, may remain secret and, hence, are only known by the encoders. Decryption keys are distributed to the group members, who are supposed to get knowledge of the information by deciphering the ciphertext. Multiple-key ciphers are particularly suitable for selective distribution, where information is required to be made available to different groups of entities. Existing applications are for instance:
  - information in companies which is restricted to different departments,
  - services-on-demand like video-on-demand (VoD) and audio-on-demand (AoD) where information is limited to several customers,
  - applications within the area of data protection, where data privacy is protected by involving both the owner of the data and data protection commissioners, and
  - E-Commerce.
- Multiple-key-parameter cipher: The idea of this type of multiple cipher is to distribute parameters of the enciphering key and deciphering key, and to apply them within the context of key recovery. Like in case of multiple-key cipher, these parameters (and the resulting key) may remain secret if required by the application.
- Multiple encipherment: By means of multiple encipherment the order of applying keys can be fixed. This is of particular interest for applications where an order of actions needs to be arranged.
- Combined multiple en-/decipherment: As already mentioned in section 2.1.4, this kind of multiple cipher is commonly used for strengthening the security of an existing algorithm by extending the key length. Thus, it is not necessary to develop a new algorithm. Here, the different keys usually belong to one entity.

Depending on the application, combinations of the different methods may be found in practice.

3.2 Applications by Use of Multiple Digital Signatures

The main idea of digital signatures is to support authenticity of electronic documents like hand-written signatures do with paper documents. Accordingly, this feature needs to be transferred to electronic documents.
In many applications a document needs to be signed by more than one entity. Some typical documents are for example:

- documents, which need signatures of different business partners,
- cheques for electronic funds transfer, where two or more entities of a company are required for authorization,
- contracts, where always at least two signers exist,
- countersigning, advocacy, and attestations with respect to the whole document or even parts of it.

Even the verification process may be performed by more than one entity, e.g. by a cooperation of witnesses or notaries. Additionally, in some circumstances the individual verifiers may be particularly selected.
The easiest way to support more than one signatory is to hand it over to the application itself, e.g. by attaching the signatures to the document by use of tags or pointers for each signature (e.g., the language XML provides multiple signatures by using pointers [Brow99]). Accordingly, also the verification process needs to be controlled by the application. A disadvantage of this method is the time a complete signing and verifying procedure takes. In contrast, some kinds of multiple signatures may not take this quantity of time, because they make use of special characteristics during the signing or verification process, like specific mathematical properties of algorithms.
The distinctive multiple digital signatures might be applied as listed below:

- Multiple-key digital signature: Here, different signing keys and verifying keys are distributed to their related entity.
- Multiple-key-parameter digital signature: The parameters of the main signing key and verifying key, can be distributed to different entities. Depending on the application, key parameters may remain secret. Similar to multiple-key-parameter cipher this kind of multiple signature can be applied to key recovery.
- Multiple signing: Multiple signing is suitable for applications where the sequence of signatories is important. The easiest way to achieve this is to use signatures with appendix. However, the disadvantage is, that the amount of data increases with the number of appendices. As mentioned before, the

use of signatures given message recovery must be handled carefully, because it may cause some difficulties (e.g. reblocking problem) during the recovery process.

Even combinations of the different methods may be found.

4 Final Remarks

If an application requires both confidentiality and authenticity for a group of entities, multiple ciphers and multiple digital signatures can be combined in different ways. Two examples for these applications are videoconferences and voting schemes for electronic voting. Often a trusted party is needed to set up the system and to generate and distribute the keys or key parameters to their distinguished owner. Additionally, trusted parties can be involved for improving the performance of a scheme by reducing the amount of communication ways, e.g. the number of signing processes [ABSW99]. Hence, this is of great interest, because it may decrease costs in some circumstances.

Additionally, we want to emphasize some important requirements on multiple cryptography. Still, not every implementation of a multiple cipher or multiple signature scheme is able to satisfy all of them.

- An overall requirement is flexibility: It must be easy to leave or join a group without the necessity of changing keys.
- Using multiple-key ciphers or multiple encipherment, each entity involved in the enciphering process should know the plaintext.
- If the signing process is done subsequently e.g. by multiple signing, each signatory should be able to verify the signature of the previous signer. This property may be less suited for applications where anonymity is required, e.g. in blind signatures [Chau83].
- If the signing process is performed by applying a multiple-key-parameter signature scheme, all signatories should be able to verify the correctness of the resulting signing key.

In our opinion, it would be appropriate to generate a catalogue containing different cryptographic mechanisms assigned to different applications.

Finally, three important aspects should be mentioned. Firstly, it has to be specified how the authenticity of keys and key parameters is guaranteed. Secondly, each entity needs to know the relevant information which is required for a positive result of each process. Finally, with respect to multiple digital signatures it has to be clarified, how the jurisdiction of each country is able to handle multiple signatures.

5 References

[ABSW99] Asokan, N.; Baum-Waidner, Birgit; Schunter, Matthias; Waidner, Michael: Optimistische Mehrparteien-Vertragsunterzeichnung. In: Verläßliche Informationssysteme. Eds.: R. Baumgart, K. Rannenberg, D. Wähner, G. Weck. Braunschweig / Wiesbaden: Vieweg 1999, pp. 49-66.

[ANSI85] ANSI X9.17(Revised): American National Standard for Financial Institution Key Management (Wholesale). American Bankers Association 1985.

[Blom83] Blom, Rolf: Non-Public Key Distribution. In: Advances in Cryptology – Crypto´82. Eds.: D. Chaum, R.L. Rivest, A.T. Sherman. Plenum 1983, pp. 231-236.

[Boyd86] Boyd, Colin: Digital Multisignatures. In: Cryptography and Coding. Eds.: H. Beker, F. Piper. Clarendon 1986, pp. 241-246.

[Boyd88] Boyd, Colin: Some applications of Multiple Key Ciphers. In: Advances in Cryptology – Eurocrypt´88. Ed.: C.G. Günther. Springer 1988, pp. 455-467.

[Boyd90] Boyd, Colin: A New Multiple Key Cipher and an Improved Voting Scheme. In: Advances in Cryptology – Eurocrypt´89. Eds.: J.-J. Quisquater, J. Vandewalle. Springer 1990, pp. 617-625.

[Brow99] Brown, Richard: Digital Signatures for XML. Proposed Internet Standard RFC, draft-brown-xml-dsig-00.txt, Jan 1999, 42 pages.

[Carr84] Carroll, John M.: The Resurrection of Multiple-Key Ciphers. Cryptologia, 8 (1984) 3, pp. 262-265.

[CaWi93] Campbell, Keith W.; Wiener, Michael J.: DES is not a Group. In: Advances in Cryptology – Crypto´92. Ed.: E.F. Brickell. Springer 1993, pp. 512-520.

[Chau83] Chaum, David L.: Blind Signatures for untraceable Payments. In: Advances in Cryptology – Crypto´82. Eds.: D. Chaum, R.L. Rivest, A.T. Sherman. Plenum 1983, pp. 199-203.

[DaPr89] Davies, Donald W.; Price, Wyn L.: Security for Computer Networks. John Wiley & Sons 1989.

[Desm88] Desmedt, Yvo: Society and Group Oriented Cryptography: A new Concept. In: Advances in Cryptology – Crypto´87. Ed.: C. Pomerance. Springer 1988, pp. 120-127.

[DiHe76] Diffie, Whitfield; Hellman, Martin E.: New Directions in Cryptography. IEEE Transactions on Information Theory, 22 (1976) 6, pp. 644-654.

[Feis70] Feistel, Horst: Cryptographic Coding for Data-Bank Privacy. RC 2827, Yorktown Heights, NY: IBM Research, Mar 1970.

[ISO_87] ISO 8732: Banking-key Management (Wholesale). London: Association for Payment clearing Services, Dec 1987.

[ISO_91] ISO/IEC 9796:1991 Information technology – Security techniques – Digital signature scheme giving message recovery.

[ISO_98] ISO/IEC 14888:1998 Information technology – Security techniques – Digital Signatures with appendix.

[Kahn67] Kahn, David: The Codebreakers. New York: Macmillan Publishing Company 1967.

[Koch96] Kocher, Paul C.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and other Systems. In: Advances in Cryptology – Crypto´96. Ed.: N. Koblitz. Berlin: Springer 1996, pp. 104-113.

[MeOV97] Menezes, Alfred J.; van Oorschot, Paul C.; Vanstone, Scott A.: Handbook of applied cryptography. CRC 1997.

[NBS_77] National Bureau of Standards: Data Encryption Standard. Washington D.C.: FIPS PUB 46, 15.1.1977.

[PKIX99] Public-Key Infrastructure (X.509) (pkix), http://www.ietf.org/html.charters/pkix-charter.html

[RiSA78] Rivest, Ronald L.; Shamir, Adi; Adleman, Leonard A.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21 (1978) 2, pp. 120-126.

[Sham79] Shamir, Adi: How to share a secret. Communications of the ACM, 22 (1979), pp. 612-613.

[Simm92] Simmons, Gustavus J.: An Introduction to Shared Secret and/or Shared Control Schemes and Their Application. In: Contemporary Cryptology. The science of Information Integrity. Ed.: G.J. Simmons. New York: IEEE 1992, pp. 441-497.

[Tuch79] Tuchman, Walter L.: Hellman presents no shortcut solutions to DES. Spectrum, 16 (1979) 7, pp. 40-41.

The "German Digital Signature Act" in the Context of Implementing the "EU Directive for Electronic Signatures"

Klaus Keus

Bundesamt für Sicherheit in der Informationstechnik – BSI
Bonn, Germany

keus@bsi.de[1]

ABSTRACT

Currently there exist a lot of world wide activities creating legal frameworks for the application of electronic and digital signatures. As Germany as one of the major leadership nations has implemented the digital signature act and its signature ordinance two years ago a lot of other EU member states will follow up by introducing the European harmonised "European Union Directive for Electronic Signatures" [IUKDG_97] [SigV_97] [EC_RL_99]. European and international agreement of certificates and electronic signatures need a common approach and understanding, based upon common conditions and requirements. Even if this directive is agreed after the final approval by the EU-Parliament in fall 1999 the EU member states will have and they will need 18 months for implementing this directive into national law.

For Germany this means that the implemented and working scheme based on the German digital signature act dated August 1997 has to be analysed in respect to the adoption of the directive. It has to be checked if the current act has to be replaced by the directive, if it may exists as a separate solution or if a combined wider scheme respecting the act in accordance with the directive is applicable. Latter would offer a broader approach and would cover a market range respecting solutions for each detail required application in the sense of scaled and hierarchical structured offering.

In this paper the main elements of the German digital signature act and the EU directive will be compared and a proposal for a possible national implementation of the directive will be given.

1 The German digital signature act: its focus and its objectives

Main objectives of the "German digital signature act" are to discover the fake of digital signatures or modified contents of signed data or at least the discovery of these fogeries.

To follow these objectives under the perspective of future legal liability for electronic applications, for authentication or for the application of electronic storing of proof related data and information a complete secure scheme is required which will ensure to recognise and to prove authenticity of the originator and the integrity of signed data.

The act should arrange the structure and the creation of the required infrastructure and should define a framework for a broad practical oriented implementation. Moreover in special it builds the conditions to respect the individual rights for electronic commerce.

The digital signature act rules the security of digital signatures, it does not cover determinations when a digital signature has to be applied. These legal prescriptions are in the responsibility of the specific areas of legislation. The creation of the required conditions and circumstances are under work currently and will be adopted during the next phase. They are intended to be finished in fall/winter 1999/2000 [GAFPR_99].

The digital signature act may be seen as a reference regulation which is based upon the German decision to prefer a more technical oriented based approach.

Major framework conditions are the requirements for the based infrastructure as e.g. for the involved parties (CAs, RAs, users), description of the global and the detail processes and for the scheme itself. In addition to furthermore requirements for the technology all these different conditions define the intended security level.

[1] This paper reflects explicitly only the personal view of the author.

In respect to the CAs there exists a set of rules for its licensing, its procedures and its internal processes, personal and organisational conditions. To cover the more technical oriented issues a complete catalogue of requirements for security functionality, the development of the technical components and its evaluation and confirmation was built.
Major motivation is the installation of a scheme which respects an appropriate and general security level. This is important insofar as the German approach does not cover a multilevel scale for digital signatures except those which follow the digital signature act as described in § 1(1) and others, covered by § 1(2).
Foreign certificates and signatures from other EU member states or such from members of the EEA will be recognised only if these cover the same level of assurance as the German ones. Other foreign certificates need to be based upon bilateral or multilateral agreements.

2 The EU Directive for Electronic Signatures

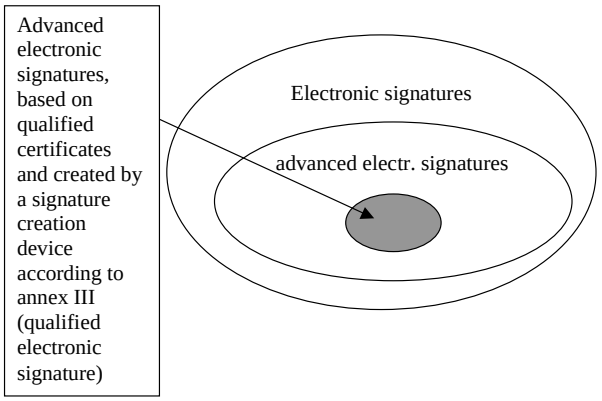
To exclude these kind of problems - based upon the order of the European Parliament - the EC in corporation with the EU members states has drafted a first directive for electronic signatures which was approved by the member states in spring this year. During a period of the following 18 months after approval by the parliament in fall 1999 the EU member states will adopt this directive into national legislation to establish the base for a European wide common approach for electronic signatures using this defined harmonised framework.
In the following some major central issues of this directive will be explained in more detail.

2.1 Approach and Application Area

The original approach of this European directive in spring 1998 was based on a so called "Liability-Approach" which is quite different from the so called "Licensing-Approach" used in the German act. The licensing approach is based on a well defined "minimum security level". Background for the former decision by the EC to follow the liability approach was the aim prevent a more technical oriented approach, to exclude so many technical details and requirements as possible. In the beginning the EC intended to prefer a technical neutral solution to keep open for future technical solutions. But during the ongoing discussion the following consequence was the change and move from the liability approach towards a combined model, representing both views. Even the former intention to express any requirements by liability and guarantee conditions was changed insofar that these conditions were mixed and enhanced by more technical oriented requirements defined in mandatory and optional annexes.
The main objectives of the directive are the promotion of electronic signatures and its legal recognition. It reduces itself on the determination of legal framework conditions to ensure a well functioning of the European market for electronic trade and commerce.
Individual national or even European joint rights, prescriptions for the application in the area of contractual legislation or formal requirements for the civil authorities are explicitly excluded in the directive. The directive still keeps a wide room for additional special conditions to satisfy the specific conditions for these areas.

2.2 Kinds of electronic signatures

Major goal of the EU directive is the implementation of a European wide joint security standard for so called "enhanced electronic signatures" in combination with "qualified certificates". This requires the installation of a comparable security level, which is defined by technical, organisational and procedural conditions in respect to the infrastructure as well as for the technical components.
Additional simpler solutions defined as "electronic signatures" are enlarging the range for the lower assurance level. This kind of electronic signatures may be seen as a more general approach without the intention to cover all the security issues as integrity, authentication and non-repudiation in common and are predominant focused to cover the more specific low level e-commerce application area.
The combination of additional requirements defined in the directive as an enhancement for the advanced electronic signatures will end up in "qualified electronic signatures" as a useful combination.



2.3 Technical Requirements

The technical requirements are defined not by the directive itself, they are determined by the annexes I-IV. Annexes I-III are mandatory, Annex IV is optional and has to be seen as a strong recommendation.
Annex I defines the minimal and not concluding requirements for so called "qualified certificates" and its structure. A certificate is defined as an electronic confirmation which links a person to its signature verification data and which is used to express the persons identity. In accordance with this definition a "qualified certificate" is defined as a certificate covering the requirements of Annex I and which is offered by such a "Certification Service Provider (CSP)" which follows the requirements of Annex II.
In Annex II the organisational, technical, personnel and legal minimal requirements for those CSPs are deter-

mined which will issued qualified certificates for public applications. A CSP is a legal or natural person which issues certificates or offers other kind of services around electronic signatures as time stamping services or others.
Annex III defines the requirements for the creation, storing and application of the "signature creation data". Signature creation data means unique data, codes or private cryptographic keys used by the signer during the process of creating an electronic signature.
Requirements for the signature verification devices, signature verification data itself and its application during the process of verification are defined in annex IV as optional recommendations.

2.4 Licensing of CSP's

In contradiction to the pre-licensing of CSP's required by the German act the directive does not require such kind of preconditions e.g. based upon an accreditation. Even such CSP's offering qualified certificates have to follow a steering and control mechanism performed by a national civil authority after they have issued the first certificate. These mechanisms are not described yet and it is up to a European expert group to define these conditions which have to be recognised by the EU member states during the process of adopting the directive. An optional but not mandatory accreditation is not excluded insofar that this kind of additional feature of the CSP may be used to express its additional security and quality offer and level and to separate itself from the mass of CSP's .

2.5 Liability and legal Effectiveness

The question dealing with liability is one of the most corner stones of the directive. Liability and questions around guarantee are covering such important issues as user protection in the sense that the CSP is liable in a wide sense for all such mishandling and damage based on the application of the certificate. This is expressed by the fact that the burden of proof is up to the CSP in a case at the court.
The legal effectiveness of digital signatures are well organised and ruled. Of course there is more legal effectiveness for such "advanced electronic signatures" than for simple electronic signatures, but even the latter have to be respected at the court and may not be rejected because of the fact that these are simple ones only or that these are electronic ones. The advanced ones have the status of fulfilling the formal requirements. It is the responsibility and duty of the EU member states to ensure that advanced electronic signatures will get the same legal status as a manual hand written signature and that these signatures will be accepted at the court as a proof.

2.6 Recognition of foreign Certificates and Signatures

The directive explicitly respects the recognition of foreign certificates and signatures. There exists three different situations which are explained by the table below.

2.7 Standardisation

In special in respect to the above mentioned international regulations and recognition of certificates and electronic signatures, standardisation is an important general issue and it is defined in the directive as one framework condition. Such kind of interoperability and standardisation activities should be started as soon as possible and should be created and implemented to ensure the global application and use of electronic signatures.
Preferred is the use and application of existing standards, only in those cases where required standards are still missing, the EC will launch such kind of projects covering the creation of filling the gap between existing solutions and required conditions. This e.g. is done by the EC during a project called EESSI (European Electronic Signatures Standardisation Initiative), performed by ICTSB (Information and Communication Technologies Standards Board) [EESSI_99] or by BSI focused on the creation of "Interoperability Guidance for digital signatures" [SigI98].

3 Proposals for further Ongoing, Recommendations for a national Implementation

During the process of adopting the EU directive in the German legislation, Germany has the chance to look at some specific issues in the digital signature law a second time. At least in respect to the technical and legal requirements and consequences some new constellations need to be discussed. One major issue of course is the question dealing with the legal equivalence which is intended to be solved during the next follow up phase at the end of this year or during the spring next year. Currently these activities are going on performed under the leadership of the Ministry of Justice [GAFPR_99].

Foreign Certificate = European qualified certificate, if	I	- foreign CSP fulfils Annex II - foreign CSP fulfils Annex I - foreign CSP is accredited by another EU-MS
	II	- foreign CSP is cross-certified by another European CSP - foreign certificate fulfils Annex I - foreign CSP fulfils Annex II - European CSP fulfils Annex II
	III	Foreign CSP is recognised in the context of an agreement between the EU and Third Countries or based on international organisations

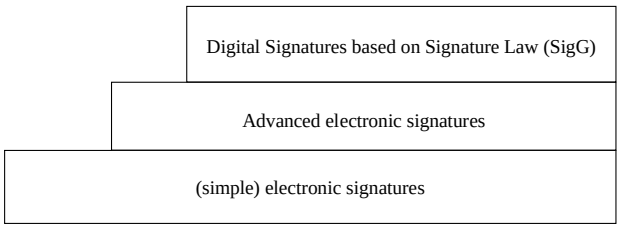
Based on two years of experience the level of assurance defined in the German law is applicable because of the easy fact that the German scheme is well working. But useful improvements are welcome as the respect of the international harmonised evaluation criteria (Common Criteria, CC) [CC_98], which allows more flexibility or the use of some more specific criteria for special components (e.g. comparable to FIPS 140-1)[FIPS-140] in addition to the ITSEC.
A more general question is dealing with the adoption of the directive in the existing German scheme. From the perspective of the author there exist two solutions:

- Integration into the existing scheme or
- Existence of an additional new second scheme.

In respect to the existing solutions combined with the proposed solutions defined by the directive we may have five different levels, defined as:

1. Electronic signatures based on the directive
2. digital signatures based on § 1 (2) Signature Law
3. Advanced signatures based on EU directive
4. Advanced electronic signatures
5. Digital signatures based on the Signature Law.

Based on this offer you may talk about a scheme including a "scalable security scheme for electronic signatures" which is able to cover all the different market requirements. The major problem of course is the transparency for the user including all the different legal consequences. Hence the author suggests to skip solution (2) and (4) because these solutions may be covered in an adequate way by the next higher one. In respect to this suggestion there still will exist 3 levels insofar as:



It should be respected that these solutions may be exist as separate ones but all of them should built in a hierarchical structure e.g. as basic-, medium- and high-end-solutions. Based on a concept as explained before Germany should be able to offer a nice and useful strategy for the migration from one level into the next higher one if requested by the application or by the user.

Some additional specific questions and problems will raise up with the liability of CSP's and furthermore open issues deal with their duty of insurance. All of these have to be solved and are still open yet. The question dealing with the underlying security level will raise up during the process of implementing the directive as expressed above. Solutions covering this problem and offering solutions are under discussion in Germany.

4 References

EC_RL_99 Gemeinsamer Standpunkt (EG) Nr. /1999 des Rates im Hinblick auf den Erlaß der Richtlinie 1999/ EG des europäischen Parlaments und des Rates über die gemeinschaftliche Rahmenbedingungen für elektronische Signaturen, Stand 8. Juni 1999, Quelle: www.accurata.se/QC/index.html (engl. Fassung)

CC_98 Common Criteria, Version 2.0, 22. May 1998, Quelle: www.crsc.nist.gov/cc/ccv20/ccv2list.htm

EESSI_99 European Electronic Signature Standardization Initiative (EESSI): Final Draft of the EESSI Expert Team Report, Stand 18. Juni, 1999; Quelle: www.ict.etsi.org/eessi/

FIPS140-1 Federal Information Processing Standards Publication 140-1, Stand 11. Januar 1994, Quelle: www.nist.gov

GAFPR_99 Gesetz zur Anpassung der Formvorschriften des Privatrechts an den modernen Rechtsgeschäftsverkehr (ENTWURF), BMJ IB1-3414/2, Stand 19.05.1999

IUKDG_97 Gesetz zur Regelung der Rahmenbedingungen für Informations- und Kommunikationsdienste (Informations- und Kommunikationsdienste-Gesetz – IuKDG) , Bundesgesetzblatt 1869, Teil I G5702, 1997, S. 1870 ff, Quelle: http://www.iid.de/iukdg

SigI98 Schnittstellenspezifikation zur Entwicklung interoperabler Verfahren und Komponenten nach SigG/SigV", Stand Juli 1999, http://www.bsi.bund.de

SigV_97 Verordnung zur digitalen Signatur (Signaturverordnung - SigV), Entwurf, Stand: 1. November 1997, Quelle: http://www.iid.de/iukdg

Global Authentication Framework Preserving Privacy in Multimedia Mobile Environments

Klara Nahrstedt and Hung-Shiun Alex Chen[1]

University of Illinois at Urbana-Champaign

{klara,chen5}@cs.uiuc.edu

1 Introduction

A wide variety of multimedia services is now available on the Internet and it is envisaged that a mobile user will be able to access these services regardless of his/her location. However, lack of global authentication mechanisms across different administrative domains results in either inadequate authentication, making both multimedia servers and clients vulnerable to intrusion attacks, or inconvenience to users. Furthermore, in a multimedia mobile environment it is often desire-able to provide *anonymity* and/or preserve confidentiality about users movements, locations and activities. Hence, a multimedia mobile environment must provide a flexible, scalable, cross-domain authentication system to secure proper service access and privacy of a mobile user.

One possible solution to the problem of providing privacy in a mobile environment is to assign traveling aliases [1] when the user travels to a foreign domain. Only the mobile entity and its home domain know the mapping between the traveling alias and the real identity. However, in the absence of a secure channel, the user can change its alias only when he/she returns to the home domain, and this can lead to trace-ability during a long term usage of the same alias. The alternative, short term aliases [3], requires synchronized and secure protocol between the mobile user and the home domain server which may not be always possible. Overall, most authentication protocols [4,5,6] require the home authentication authority (authentication servers) to be contacted during the execution of the authentication protocols across multiple domains. This approach brings out also the authentication server chain availability problem because it means that all servers in the authentication chain must be available at the same time.

Therefore, there is a need [6] for a flexible global authentication system with (1) a scheme for expressing trust relations, and various levels of trust, (2) suitable protocol for propagating information about trust relationships, (3) methods for evaluating trust-related information, (4) enforcement of trust relations, and (5) suitable protocol for preserving privacy. In this paper, we present a *global authentication framework* for preserving privacy, which is based on the integration of the distributed trust system with a token-based authentication protocol to achieve this goal.

2 Global Authentication Problem

Traditionally, the basic idea behind global authentication is the generalization of the inter-domain authentication mechanism through an authentication chain. It means a client *C* has to provide an authentication chain, $\{E_i, i = 0,\ldots,n+1\}$, to a server *S* to prove its identity. Each E_i is an entity in the system and E_{i+1} should be able to trust the entity E_i to assure the authenticity of E_{i-1}.This implies that two adjacent entities in an authentication chain need to have prior knowledge of each other. Typically, E_i is an authentication server (AS) for domain *i*, and domains *i* and *j* must arrange for prior cross-registration of their ASs. Usually, it is feasible to have a complete cross-registration for at most two domains, however cross-registration of a large number of domains is neither feasible nor efficient. Therefore, the questions are "Given the name of a server S, how can a client C construct an authentication chain that will be acceptable to the server, and given an authentication chain from the client, how can a server verify that the chain is acceptable?"

We solve these questions by designing a *token-based authentication protocol* which assumes:(1) the identity of the client is of no importance in decision making whether to provide a requested multimedia service, and (2) once the token is validated, the token issuer can be trusted to pay/transfer funds/take any pre-defined responsibility on behalf of the user for the requested service. The token-based authentication protocol works together with a distributed trust management to validate the tokens across multiple domains.

3 Integration of Trust Management with Token-based Authentication

Notation and Definitions: The distributed trust management model is based on the *trust relationship* [2] which properties are: (1) it is always between exactly two entities, (2) it is unidirectional, and (3) it is conditionally transitive. There exist two types of trust relations, the ***direct trust relationship*** where entity E_i trusts

[1] This work was supported by the National Science Foundation Career grant under contract number NSF CCR 96-23867 and by the Research Board of the University of Illinois at Urbana-Champaign.

entity E_j, and ***recommender trust relationship*** where E_i trusts E_j to give recommendations about other entities trustworthiness. Entities, which are able to execute the trust relations, are call *agents*. Given two agents *x* and *y*, we denote the direct trust relationship as $T^d_x(y)$ and the recommender trust relationship as $T^r_x(y)$. The relation *T* is normalized over (0,1), e.g., $T^d_x(y)=1$ means that *x* has complete trust in *y*. Agents use *trust categories* to express trust towards other agents depending upon which particular characteristics of that entity is under consideration at that moment[1]. More formally, $T^d_x(y,\alpha_x,c)$ represents a direct trust of x in y in situation α_x for category c. The agent's *x* estimate, how important a situation α_x is, is a normalized value over (0,1) and it is represented by $I_x(\alpha_x)$. The *importance* of a situation is useful to the agent *x* when determining the amount of situational trust in the agent *y*. Related to the importance of a situation are cost and benefits associated to a situation. The *cost* of a situation $C_x(\alpha_x)$ is measured in terms of problems associated with incompetent or malevolent behavior on the part of another agent in the relationship. The agent can only estimate the potential cost of a situation based on past experiences of a similar situation. The expected *benefits* of a situation $B_x(\alpha_x,c)$ decide whether to cooperate with another agent.

Rules for Interaction for Direct Trust Relationship: Using the notation above, we can reason about the direct trust relationship and derive rules of interaction as follows:

- The trust of an agent *x* to an agent *y* in a situation α_x and category *c* is related to the amount of trust of the agent *x* in category *c* and the importance of the situation α_x to the trusting agent: $T^d_x(y,\alpha_x,c) = T^d_x(y,c)*(1- I_x(\alpha_x))$

  The value $T^d_x(y,c)$ is related to the competence of that agent as observed by the agent itself or by other recommenders.
- In order to cooperate with the agent y, the trust x has in y for that particular situation has to be above a certain threshold which is a function of the cost and benefits:

  If $T^d_x(y,\alpha_x,c) \geq RiskThreshold \Rightarrow WillCooperate$,

  where $RiskThreshold = (C_x(\alpha_x)*\beta)/ B_x(\alpha_x,c)$;

  $C_x(\alpha_x)*\beta$ represents an expected earning as a percentage of the potential cost with β being the percentage, and $T^d_x(y,\alpha_x,c)*B_x(\alpha_x,c)$ reflects the estimated earning. If the estimated earning is higher than the expected earning, then the agent *x* is willing to cooperate with another agent *y*.

Recommendation Protocol for Recommender Trust Relationship: For provision of a recommender trust relationship, the distributed trust model relies on a recommendation protocol which specifies exchange of request messages and trust-related recommendation messages between two agents: the *recommender* and the *requester* of a recommendation. The recommendation message contains a *recommendation path* consisting of an ordered sequence of recommender IDs through which the recommendation message must be passed from the recommender to the requester. There is an expiration date associated with the request message and the recommendation message. In the case of expiration of request messages, old messages are discarded. In case of recommendation messages, the expiration date indicates the recommendation validity period after which the message should be discarded. During the exchange of recommendation messages, each recommender *x* specifies its trust values T^r_x [2] about the other recommenders R_i in the recommendation path and the overall trust value *tv* of a server *S* for a single recommendation path *p* is:
$tv_p(S)=T^r_x(R_1)* T^r_x(R_2)*\ldots* T^r_x(R_n)* T^r_x(S)$; where $T^r_x(S)$ is the recommendation trust value of the server S given in the recommendation, $T^r_x(R_i)$ is the recommender trust value of the recommenders *i* in the recommendation path. A requester may request multiple recommendations for a single target S and thus recommendations tv_{pi} must be then combined into a single value, for example, an average value $tv(S)=(\Sigma_{i=1..n} tv_{pi}(S))/n$.
In our framework, the recommendation protocol is assisted by the directory service which stores recommendation paths information and trust values from past experiences with other agents to speed up the connectivity between the client and server across multiple domains. Especially, the directory service provides information for recommendation path construction. The path construction algorithm seeks N paths where the confidence of each path is higher than a threshold $\sigma= (C_x(\alpha_x)*\beta)/((1- I_x(\alpha_x))*B_x(\alpha_x))$.
In case of failing to find N recommendation paths with confidence over σ, the agent starts to collect information about trust values, cost and benefits or recommendations to evaluate the trust value of another agent.

Token-based Authentication Protocol: Each client receives a token(s) from a token issuer before leaving for a foreign/remote domain and the client uses the token(s) to

[1] For example, we trust a certificate authority to certify public keys ("certify public key" is a trust category), but do not attest to the key-holder's credit status.

[2] Note that the individual recommender trust value $T^r_x(R_i)$ can be quantified using the same rules of interaction as shown for the direct trust relationship.

request a service from a remote multimedia service provider. The token issuer will be authenticated by the multimedia service provider, using the token-based authentication protocol, since the token issuer may not be globally and unconditionally trusted as follows: The service provider gets the name of the token issuer from the token and estimates the trust values for decision making. The token protocol can use either the rules of interaction for direct trust relationship or the recommendation protocol for getting recommender trust value over recommendation path. We will outline the case of using the direct trust relationship rules as follows:

- $T^d_x(y,c)$: direct trust relationship between the service provider x and the token issuer y in the category "c: token issuing".
- $I_x(\alpha_x)$: importance value of the situation (α_x) for the service provider x. This value depends on the monetary value of the service, i.e., the higher the service cost, the higher the importance of the situation. For example, to get a movie preview should be cheaper than to playback the whole movie.
- $C_x(\alpha_x)$: potential cost to the service provider in case of untrustworthy behavior of a token issuer in situation α_x .
- $B_x(\alpha_x)$: potential benefit to the service provider in case of a trustworthy behavior of a token issuer in situation α_x.
- β: percentage which the service provider expects to earn when cost is $C_x(\alpha_x)$.

Using the rules of interaction above, the service provider can decide the validity of the presented token. If the token cannot be accepted due to a low trust value T for the token issuer, the service provider has three options:

- rejection: the service provider rejects the tokens and refuses to provide services to the mobile user;
- discount rate: the service provider can offer discount rate to clients with poorly-rated tokens and their token issuers[3]. The discount rate may be computed as $\text{DiscountRate} = (T^d_x(y,\alpha_x,c)* B_x(\alpha_x))/ (C_x(\alpha_x)*\beta)$. Note that the discount rate lies within (0,1) interval because the token can be only honored when
- $T^d_x(y,\alpha_x,c)* B_x(\alpha_x) \le C_x(\alpha_x)*\beta$.
- liability insurance: the service provider can rely on the liability insurance of the token issuer to compensate a lack of confidence in the token issuer. In this case, the risk of a malfeasant behavior on the part of the token issuer is partially shifted from the service provider to the insurance provider. The insurance provider, based on the level of risk, determines insurance policy and premiums for insurance. The electronic insurance policy is an agreement certified by signatures of the insurance provider and the token issuer.

In summary, the token provides a hiding mechanism to preserve the user's privacy and instead of user's authentication, the token issuer is examined and authenticated. To avoid the traditional problems of static authentication service, availability, and knowledge of user's home location, we are using the distributed trust management to verify either the direct trust value between the service provider and the token issuer or the recommender trust value about the token issuer.

4 Conclusion

We have outlined an integrated approach of the distributed trust management framework with the token-based authentication protocol as a possible solution for the global authentication when requiring user's privacy and anonymity within the multimedia mobile environment. Especially, this approach allows us to provide a global authentication and user privacy when operating across multiple foreign administrative domains.

5 References

[1] R. Molva et al., "Authentication of Mobile Users", IEEE Network, 8(2):26-34, March/April 1994.

[2] F. Abdul-Rahman and S. Hailes, "A Distributed Trust Model", ACM New Security Paradigms Workshop'97, Cumbria, UK, September 1997.

[3] ETSI (European Telecommunications Standard Institute), "GSM-Security Related Network Functions", ETSI Standard GSM 03.20, October 1993.

[4] D. Samfat, R. Molva, N. Asokan, "Untraceability in Mobile Networks", Proceedings of the ACM International Conference on Mobile Computing and Networking, Berkeley, November 1995.

[5] N. Asokan, "Anonymity in a Mobile Computing Environment", Proceedings of Workshop on Mobile Computing Systems and Applications", Santa Cruz, CA, December 1994.

[6] N. Asokan, "Security Issues in Mobile Computing", Research Proposal April 1995.

[7] B. Patel and J. Crowcroft, "Ticket based Service Access for Mobile Users", Proceedings of Third ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'97), Budapest, Hungary, September 1997.

[3] Since the Internet network is becoming more decentralized and more distributed, thousands of tokens will exist and be traded, hence appropriate discount rates will be applied to attract business.

Effective Models of Real Data to Enhance Digital Watermarking Methods

David Hilton

Signum Technologies Limited
Witney, Oxford

signum@signumtech.com

ABSTRACT

Many papers on watermarking have aspired to too much generality and failed to address the exact nature of the data with which they deal. Two particular deficiencies result, the first is that the quality of the data is adversely affected, the second is that the detection of the watermark is not as efficient as it might be. Image data and audio data are frequently presented as being almost interchangeable, this presentation considers how the situation might be improved.

1 Introduction

There is now a substantial body of published work on methods of watermarking both for images and audio files and some work on the relative merits of the proposed systems. However, from the practical commercial view thas been little detailed discussion of the nature of the data that is used and the types of transformation that is likely to have to undergo in the commercial world.

This presentation is intended to raise questions from the practical viewpoint, discussing the variety of contexts in which watermarking is used and, in passing, indicating the reasons for the choice of method adopted by my own company. The issues from our viewpoint have tended to focus more on quality than is perhaps reflected in current literature, and more on the need to detect from dislocated and screened data. However, the generally discussed issues of robustness, security, and payload are all germane to our activities.

One possible scenario is that, rather like JPEG in its initial stages, watermarking may be rejected because it introduces obvious artefacts which outweigh the benefits. One of the worst scenarios is the introduction of a method that copes satisfactorily for 99% of situations but produces the occasional unsatisfactory result. This situation militates against the use of batch processing which is a major part of workflow issues. Many users wish to watermark as the image goes out to a customer and either the quality assurance has already been carried out or it is at a final stage where a rejection will cause great inconvenience.

There are references in the literature to simple and elegant solutions to watermarking, but parallels with problems of physics indicate that the production of a reductionist theory is only the first step in the solution of a practical problem. In hydrodynamic problems, for instance, one can write down Navier Stokes equation quite simply, but in any practical problem it is often necessary to make gross simplifications, neglecting several terms and then dealing with complex boundary conditions.

The effort spent on dealing satisfactorily with all the configurations that an image or audio file can present is as great as that spent in designing the original method.

Many of the methods adduced for watermarking will have only marginal differences in efficacy and those margins may well be altered by the manner in which real contexts are considered.

This presentation considers the contexts in which watermarking is commercially used and then considers the implications of these contexts on quality, detectability, robustness and security. An attempt is made to share some of the experience of watermarking as seen from a commercial viewpoint.

2 Watermarking Contexts

Several significant users have already adopted watermarking of still images in a wide variety of contexts. The straightforward use for published images from photographers or photographic libraries is steadily developing. This usage includes photographers who publish their work in hard copy where the watermark has to be detected following the scanning process.

These images are usually subjected to some compression, typically JPEG at a ratio of about 10:1, and of course will be subjected to the half toning process at a range of resolutions. An additional complication is that photographers need to detect a watermark when part of an image has been used in the assembly of some sort of collage or general cut and paste action. Detection from an irregularly shaped fragment is an essential requirement.

Whilst these images demand a high quality there are users in the medical or forensic world, for instance, who use 12 bit images for higher quality viewing and where the watermark has to be of an even less disruptive type. The level of alteration has to be no greater than the lowest quantum of change (but not, of course, simply an LSB adjustment.)

At the other extreme there are users who demand extreme robustness but are less demanding about quality. An example is the photo ID usage where a small head and shoulders photograph is compressed as much as 40:1, printed on a plastic card, covered with a laminated sheet and then required to be detected by scanning with a modest quality scanner. Clearly if such high compression is used the quality has to be less of an issue.

In the audio context a similar range of qualities needs to be considered. At the one end are those musical clips that are compressed by the MP3 algorithm, frequently at fairly low quality to cope with low bandwidth. At the other are the highest quality sound recordings whose quality has to satisfy the most acute observers, as with images the fear is that pirates will cut and paste small sections of some musical offering.

It is clearly advantageous if the same basic method is applicable to all of these situations. Consideration is given below to the way in which these various contexts should influence the choice of algorithm.

3 Quality

The point that emerges most strongly as the divide between customer requirement and theoretical models is the measure of satisfactory quality. This situation mirrors that which occurred when JPEG first came into common usage. By careful choice of quantisation coefficients high quality compressed images could be produced. These could be proved to be a good representation of the original by the use of such global measurements as sum of least square differences from the original, or other more sophisticated measurements.

Unfortunately, these measurements of excellence were not what customers required. A single visible artefact in a sensitive part of an image was enough to render that image unusable and it was the likelihood of such an artefact which was the appropriate measure.

Likewise with watermarking. a globally calculated parameter expressing quality is only of glancing significance. The required statistic is the probability of producing at any point an artefact that may be clearly perceptible. Most of the measures of quality put forward are summations over chunks of an image and these have little relevance. As mentioned in the introduction the use of batch processing for high quality images is commonplace. Such a system fails if there is even a 1% chance of failure.

The implication for watermarking methodology is that the strength of embedded signal at any point should be governed as far as possible by the nature of the image or the audio values rather than by some precalculated geometrical pattern. Now the use of any frequency domain watermark involves the addition of a pregenerated periodic function, be it a simple cosine function or a more sophisticated orthogonal function.

There will also be several such functions combined which will tend to produce regions where there is a requirement for a heavier embedded signal, and even if some sort of template is applied to augment or decrement heaviness in sensitive areas the signal is dictated by the imposed frequency and not by the image. In JPEG there was always a problem when a frequency ran from a noisy part into a smooth area and this is the case with some frequency domain watermarks.

The same problem applies when a precalculated 'noise' is added to an image. There tend to be points where the noise imposes a heavier signal than the image would warrant.

In the case of a music file, the addition of any frequency in the perceivable range is risky because of the chance of the existence of a close frequency which would produce beats. A prior assessment of a whole section is a necessity for high quality, modifying the watermark according to the frequencies already present.

We do have a great deal of information about the human audio perception as a guide to the quality we seek. In most of the audio literature we hear about the perceptive coding removing masked frequencies. There is not much discussion of the other parameters of perception which we know about from music theory and might well have a bearing on the type of signal that we could embed imperceptibly or even on the manipulation that might be imperceptibly used to destroy a watermark.

We know, for instance, that even for a musician of the quality of Bach the alteration of frequencies to produce an equal temperament scale is not an unacceptable manipulation. However, we know that a transposition of Beethoven's ninth from the key of D minor would be perceptible. These facts suggest the limits within which a purely musical file could have its histogram changed in important ways without damaging quality.

The alternative to the imposition of predefined patterns is the use of direct data addition where greater control can be exerted, particularly if only one signal, representing one bit is added at any point. This is the method used by the Signum method and also by the MIT patchwork technique.

4 Detectability

A fear amongst photographers is that parts of images will be used for collages without proper attribution, and hence a watermark must survive fragmentation and dislocation. In the cases of photos used for ID purposes the head may well be cut out from the background. In any such cases there is a need to assess which watermarking technique will best survive. There seems a prima facie case for supposing that a watermark depending on a pattern which necessarily must extend across a range of pixels will be weakened far more than a watermark which simply accumulates its evidence from any available pixels.

An analysis of how cross correlation is affected by such dislocation would be useful. If direct addition signal is added bit by bit to each pixel it is, of course, necessary to ensure that the whole signal occurs within any small area that may be selected.

A second practical detection problem that occurs with any method that introduces periodicity as part of a wa-

termark is that periodicity may already be present. In the case of image usage periodicity is present in a huge percentage of cases for two reasons. Firstly the quantisation process in the commonly used JPEG compression introduces periodicity which may give a false signal in watermark retrieval. Secondly, whenever images are printed with a conventional screening process, the half toning introduces a regular periodicity.

I have not yet seen in the analysis of watermarking methods any sort of modelling of data to represent these processes and which will give an indication of the degree to which detection is thereby attenuated.

The use of sophisticated predictors has proved invaluable in data compression. Such predictors rely on assumptions about the nature of data and in the same way knowledge of the probable form of data greatly improves the detection performance of watermarks.

If we take a very simple case where each pixel has either a positive or negative signal added we can attempt to assess the probable contribution of any pixel to our knowledge of the watermark and we instantly realise the necessity for a reasonable predictive method.

Suppose for a selected pixel we consider the 4 neighbouring pixels and try to calculate the probability that, given the values of the pixel in question and its 4 nearest neighbours there has been a positive embedded signal added. We need to estimate the correlation between neighbouring pixels and whether or not there is any benefit in extending to more remote pixels, and to what extent we can assume that the area we are considering is representative of the whole image. The analysis is tedious but important.

It would be interesting to see an analysis which revealed which watermarking method derived optimal information from any given pixel.

5 Robustness

In practical situations robustness must depend upon the watermark being embedded sufficiently comprehensively as to be inseparable from the data. However, we cannot produce a detection system that will automatically detect watermarks after any possible image manipulation. It seems almost in reality that we must distinguish between two types of detection.

Firstly there is the automatic detection provided by a publicly available detector where the watermark is used largely for publicity purposes. This detection does not require the presence of the original image and can typically cope with simple resizing and small rotation. This automatic detection can be foiled relatively simply by geometrical manipulation. If we wish to make the watermark more widely detectable we have to add signals that are invariant under a wider range of transformations and the level of data addition required tends to produce unacceptably perceptible artefacts. If there is sufficient redundant data to embed a signal of high symmetry then undoubtedly a compression scheme could remove the signal.

Secondly there is what might be termed forensic detection where an image provider is prepared to spend an hour or two proving ownership of an image. This forensic search may involve the use of image manipulation tools to resize the suspect image and to undo some of the manipulation which has been applied. It may involve the use of the original image in some way. There can be no real hope that this process can be rendered automatic because the number of possible image manipulations that still leave an image as a clear derivative of some original is infinite. The holy grail of automatic detection is not worth the search.

6 Security

There appears to be a difference in the way that security is regarded by practitioners and the way its dealt with in literature. There tends to be a feeling by image users that 80% success rate in watermark detection is a satisfactory deterrent. There seems to be little concern that hackers will undertake elaborate decryption processes to undo watermarks, although clearly if a method has very low security it will not be acceptable.

Certainly there is an awareness that if the quality of an image is too far damaged then watermarks will not be used and there will be no security.

It has also become apparent from customers that the preference of security experts that the security should not rest in the nature of the algorithm but rather in control of the keys is not a preference that they share. There are users who wish to watermark digital files without revealing the fact to the public at large.

7 Conclusion

Advances in watermarking will be greater if there is increased collaboration between commercial practitioners and academics, or, in the case of some large organisations, better co-operation between those who design the algorithms and those who deal with the customers. There is always a danger that people who deal simply with the method of handling data but never with the data content will produce less than ideal solutions.

Methods for Tamper Detection in Digital Images

Jiri Fridrich

State University of New York in Binghamton
and Mission Research Corporation

fridrich@binghamton.edu

ABSTRACT

The purpose of this paper is to present a comprehensive overview of current steganographic techniques for tamper detection and authentication of visual information. Fragile, semi-fragile, robust watermarks, and self-embedding are discussed as a means for detecting both malicious and inadvertent changes to digital imagery. Some attacks and security gaps are discussed.

KEYWORDS

Authentication, tamper detection, fragile watermarks, semi-fragile watermarks, self-embedding, hybrid watermarks, feature authentication.

1 Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

In the past, several techniques [1–14] and concepts based on data hiding or steganography have been introduced as a means for tamper detection in digital images and for image authentication – fragile watermarks, semi-fragile watermarks, robust watermarks, and self-embedding. The visual redundancy of typical images enables us to insert imperceptible additional information and make the images capable of authenticating themselves without accessing the originals. The goal is to prevent the possibility of creating a forgery that goes undetected. An example application would be a secure digital camera equipped with a watermarking chip that authenticates every image it takes before storing it on the flash card. The embedded information could be uniquely tied to the camera's serial number thus creating a link between the images and the hardware that took them. Such smart images may play an important role in detecting digital forgeries or establishing the origin of digital images.

Fragile watermarks [1–4,10,13] are designed to detect every possible change in pixel values. They can be designed to provide a very high probability of tamper detection while making it practically impossible to create a forgery. However, since images are highly redundant, and their visual content is generally not modified under small perturbations, it may not be desirable to have this kind of sensitivity at least in some applications. Semi-fragile watermarks [5,10] are moderately robust and thus provide a "softer" evaluation criterion. The value identifying the presence of the watermark (a correlation in most cases) can serve as a measure of tampering. A natural extension of the concept of a semi-fragile watermark is the robust spread spectrum watermark on medium sized blocks [7–9]. If an image feature comparable in size to the watermarking block is removed or added, the watermark in that block will no longer be present. On the other hand, typical image processing operations, such as filtering, gamma correction, or lossy compression will decrease the evidence for watermark presence more or less uniformly over all blocks. Consequently, one can distinguish malicious changes from innocent image processing operations. Such techniques [7–9,13,14] could be termed authentication of the visual content. They can be combined with fragile watermarks if the fragile watermark is inserted after the robust one. This hybrid watermark [13] combines the accuracy and precise localization of the fragile watermark with the robustness of the robust watermark.

The last category of image authentication techniques is called self-embedding. The image is embedded into itself in such a manner that it is later possible to not only detect tampered or cropped out portions of the image, but also to recover the original content. Due to very large payload requirements of self-embedding techniques, it is not possible to have a good reconstruction quality, watermark invisibility, and robustness at the same time. Currently developed techniques [11,12] can

be classified as fragile or semi-fragile watermarks with very high quality of the reconstructed image.

In this paper, we present the technical details of the above mentioned steganographic techniques for authentication and tamper detection in digital images. We compare their performance, security, and outline research directions in this field. In section 2, we start with fragile watermarks, and continue with semi-fragile watermarks in Section 3. Robust watermarks and hybrid watermarks for tamper detection are covered in Section 4. In Section 5, we describe algorithms for self-embedding techniques and conclude the paper in Section 6.

2 Fragile watermarks

If the inserted watermark is fragile so that any manipulation of pixels will disturb its integrity, one can readily detect the tampered areas by checking for presence of this fragile watermark. A very simple scheme is obtained by encrypting the seven most significant bit planes and hashing the result. This hash can then be inserted into the least significant bit plane of the image. With high probability, any change made to any bit plane will be detected. The localization properties of this simple scheme can be improved if it is applied to image blocks rather than the whole image.

One of the first fragile watermarking techniques proposed for detection of image tampering was based on inserting check-sums of gray levels determined from the seven most significant bits into the least significant bits (LSB) of pseudo-randomly selected pixels [1]. In this paper, we are going to describe one possible implementation of this idea. First, we choose a large number N that will be used for calculating the check sums. Its size directly influences the probability of making a change that might go undetected. The image is then divided into 8×8 blocks, and in each block, a different pseudo-random walk through all 64 pixels is generated. Let us denote the pixels as $p_1, p_2, \ldots, p_{64}$. We also generate 64 integers $a_1, a_2, \ldots, a_{64}$ comparable in size to N. The check sum S is calculated as

$$S = \sum_{i=1}^{64} a_i g(p_i) \bmod N ,$$

where $g(p_i)$ is the gray level of the pixel $p(i)$. It is then expressed in a binary form, encrypted, and embedded in the LSBs of the image block. Swapping pixels within one block will change the value of S because the two pixels will have different coefficients a_i. The random walk p_i and the coefficients a_i can be block dependent (using a secret key), thus making it impossible to swap entire blocks without making undetected changes. One weakness of this scheme is that it is possible to swap identically positioned blocks in two authenticated images, unless one does not make the watermark dependent on the image or at least the image order. This could be achieved for example using the robust bit extraction algorithm proposed in [16]. Another alternative to thwart this "collage" attack is to use randomly placed pixels rather than publicly known 8×8 blocks. This may, however, somewhat negatively influence the ability to localize changes.

Yeung and Wong [2,3] proposed the following method for authentication of digital images. The process of image authentication starts with a secret key that is used to generate a key dependent binary valued function f, f: {0, 1, …, 255} → {0,1}, that maps integers from 0 to 255 to either 1 or 0. For color images, three such functions, f_R, f_G, f_B, one for each color channel, are generated. These binary functions are used to encode a binary logo L. The gray scales are perturbed to satisfy the following expression

$$L(i,j) = f_g(g(i,j)) \text{ for each pixel } (i,j).$$

For an RGB image, the three color channels are perturbed to obtain

$$L(i,j) = f_R(R(i,j)) \oplus f_G(G(i,j)) \oplus f_B(B(i,j)) \text{ for each pixel } (i,j),$$

where ⊕ denotes the excluded OR. Error diffusion is further employed to preserve the original colors. The image authenticity is easily verified by checking the relationship $L(i,j) = f_g(g(i,j))$ for each pixel (i,j).

There are some obvious advantages of this approach. First, the logo itself can carry some useful visual information about the image or its creator. It can also represent a particular authentication device or software. Second, by comparing the original logo with the recovered one, one can visually inspect the integrity of the image. Third, the authentication watermark is embedded not only in the LSBs of the image but somewhat deeper (± 5 gay scales). This makes it more secure and harder to remove. Fourth, the method is fast, simple, and amenable to fast and cheap hardware implementation. This makes it very appealing for still image authentication in digital cameras.

This method, however, has a serious security gap if the same logo and key are reused for multiple images. Given two images I_1 and I_2 with gray levels $g^{(1)}$ and $g^{(2)}$ watermarked with the same key and logo L, we have

$$f_g(g^{(1)}(i,j)) = L(i,j) = f_g(g^{(2)}(i,j)) \text{ for all } (i,j).$$

The last equation constitutes $M \times N$ equations for 256 unknowns f_g. As reported in [17] only two images are needed on average to recover over 90% of the binary function f_g. Once the binary function is estimated, the logo can be easily derived. Actually, if the logo is a real image rather than a randomized picture, we can use this additional information to recover the rest of the binary function f_g. Although the situation becomes more complicated for color images, the method appears to have a serious security gap. Making the embedded information depend on the image index would not be too practical because one would have to search for the right index, which may in turn increase the complexity of the algorithm. Embedding the index in a robust manner in the

image using a secret (camera) key may alleviate this situation. As another approach, we can use the robust bit extraction algorithm [16] and make the logo and / or the binary function(s) a nontrivial function of the secret key and the image itself.

3 Semi-fragile watermarks

Another class of authentication watermarks is formed by semi-robust watermarks. Such watermarks are marginally robust and are less sensitive to pixel modifications. Thus, it is possible to use them for quantifying the degree of tamper and distinguish simple LSB shuffling from malicious changes, such as feature adding and removal. Van Schyndel et al. [4] modify the LSB of pixels by adding extended m-sequences to rows of pixels. For an $N \times N$ image, a sequence of length N is randomly shifted and added to the image rows. The phase of the sequence carries the watermark information. A simple cross-correlation is used to test for the presence of the watermark. Wolfgang and Delp [5,10] extended van Schyndel's work and improved the localization properties and robustness. They use bipolar m-sequences of –1's and 1's arranged into blocks and add them to corresponding image blocks. If $X(b)$ denotes the gray levels of the original image block b, the watermarked block $Y(b)$ is calculated as

$$Y(b) = X(b) + W(b).$$

The verification process used to test an image Z to see if the watermark is in the image is:

$$\delta(b) = Y(b) \cdot W(b) - Z(b) \cdot W(b).$$

A threshold test is then performed on the test statistic δ. If $\delta < T$, where T is a user-defined threshold, $Z(b)$ is considered genuine. Large values of T allow the toleration of changes to the marked image block $Y(b)$. If $Z(b) = Y(b)$, then $\delta = 0$.

Zhu et al. [6] propose two techniques based on spatial and frequency masking. Their watermark is guaranteed to be perceptually invisible, yet it can detect errors up to one half of the maximal allowable change in each pixel or frequency bin depending on whether frequency or spatial masking is used. The image is divided into blocks and in each block a secret random signature (a pseudo-random sequence uniformly distributed in [0,1]) is multiplied by the masking values of that block. The resulting signal depends on the image block and is added to the original block quantized using the same masking values. Errors smaller than one half of the maximal allowable change are readily detected by this scheme.

The authors apply this technique to small 8×8 pixel blocks. The block is DCT transformed, and the frequency masking values $M(i,j)$ for each frequency bin $P(i,j)$ are calculated using a frequency masking model. The values $M(i,j)$ are the maximal changes that do not introduce perceptible distortions. The DCT coefficients are modified to $P_S(i,j)$ according to the following expression

$$P_S(i,j) = M(i,j) \{ \lfloor P(i,j) / M(i,j) \rfloor + r(i,j) \operatorname{sign}(P(i,j)) \},$$

where $r(i,j)$ is a key-dependent noise signal in the interval (0,1), and $\lfloor x \rfloor$ rounds x towards zero. Since $|P(i,j) - P_S(i,j)| \le M(i,j)$, the modifications to DCT coefficients are imperceptible.

For a test image block with DCT coefficients $P_S'(i,j)$, the masking values $M'(i,j)$ are calculated. The error at (i,j) is estimated by the following equation

$$e' = P_S' - M' \{ r \operatorname{sign}(P_S') + \lfloor P_S' / M' - (r - 1/2) \operatorname{sign}(P_S') \rfloor \},$$

where all the values are evaluated at the same frequency bin (i,j). The authors show that if the true error e at (i,j) is smaller in absolute value than $M(i,j)/2$, and if $M'(i,j) = M(i,j)$, the estimated error $e' = e$. It is further shown that the error estimates are fairly accurate for small distortions, such as high quality JPEG compression.

4 Authenticating the visual content

Fridrich [7,8] describes a technique in which an image is divided into medium-size blocks and a robust spread-spectrum watermark is inserted into each block. If watermarks are present in all blocks with high probability, one can be fairly confident that the image has not been tampered with in any significant manner (such as adding or removing features comparable in size to the block). If the watermark correlation is lower uniformly over all image blocks, one can deduce that some image processing operation was most likely applied. If one or more blocks show very low evidence for watermark presence while other blocks exhibit values well above the threshold, one can estimate the probability of tampering and with a high probability decide whether or not the image has been tampered with.

Kundur and Hatzinakos [9], propose a wavelet-based telltale image authentication. Because the watermark is localized both spatially and in the frequency domain, it provides spatial and frequency domain information on how the signal was modified. For example, if certain frequencies in the image block have been untouched, they will be authenticated as credible. The image is first decomposed using the Haar transformation to the L-th level into the high frequency components $f_{k,l}(m,n)$ and the lowest resolution level $f_{a,L}$. The secret key is used to generate the subset of wavelet coefficients that will be modified. A special quantization function Q is used to assign binary values to wavelet coefficients f

$$Q_{\Delta,l}(f) = 0 \text{ if } \lfloor f/(\Delta 2^l) \rfloor \text{ is even,}$$
$$Q_{\Delta,l}(f) = 1 \text{ if } \lfloor f/(\Delta 2^l) \rfloor \text{ is odd}$$

at the quantization level l. If a wavelet coefficient $f_{k,l}(m,n)$ is chosen for watermark embedding, it is modified so that

$$Q_{\Delta,s}(f_{k,l}(m,n)) = w(i) \text{ XOR } qkey(m,n),$$

where $w(i)$ is the i-th watermark bit and $qkey$ is a bit generated from the image and a secret key. The construction of the quantization function Q guarantees that one will never have to modify the coefficient at the level l by more than $\pm\Delta 2^l$. The watermark is extracted by evaluating the expression

$$w(i) = Q_{\Delta,s}(f'_{k,l}(m,n)) \text{ XOR } qkey(m,n),$$

where f' is the wavelet coefficient of the potentially tampered image. The extent of tampering is evaluated using the number of correctly recovered watermark bits $w(i)$. The authors also provide an estimate of the probability that a random modification of the wavelet coefficients will go undetected. This probability is shown to decrease exponentially with the number of modified coefficients.

Schneider and Chang [14] propose a content-based signature for robust feature authentication. First an image is processed and a set of features is extracted. The result is hashed and encrypted using a public key. The encrypted information is finally embedded in the image. To authenticate an image, the embedded information is first extracted, decrypted, and compared to the hash of extracted features. Since the hash function is sensitive to every bit of its input, the feature extraction and normalization needs to be robust to achieve insensitivity to small modifications while being able to detect –large changes. The authors use image histograms on small blocks as features. Another possibility would be to use special "robust" hash functions that are not sensitive to every input bit. The robust bit extraction algorithm [16] mentioned previously is one possible approach.

It appears that no single scheme can have both precise localization properties without being too sensitive. Indeed these two requirements are in conflict. On the other hand, it should be possible to combine a robust watermark or a feature authentication watermark with a fragile one, if the fragile watermark is embedded as the second one. The fragile watermark is usually very weak and should not influence the robust one in any significant manner. This hybrid watermark [13] can, therefore, enjoy the good properties of both watermarks. If a subtle change is made to a highly localized group of pixels, such as changing the eye color in a portrait photograph, the fragile watermark can be used to precisely localize the change. On the other hand, a simple lossy compression or applying a filter to the image will be indicated as non-malicious tamper because the robust watermark will survive.

5 Self-embedding

The idea of self-embedding the image into itself enables not only detection of areas that have been tampered or damaged, but also recovering the missing information. The self-embedded information can be in a fragile or in a semi-fragile form. Thus, self-embedding is a means both for protecting the image content and for authentication. Because the image or its approximation needs to be embedded into itself without introducing visible artifacts, the embedded information cannot be robust. There is an obvious trade-off between the robustness of the self-embedded image and its visual quality.

Fridrich and Goljan [11] describe a self-embedding technique in which the image is first divided into blocks of 8×8 pixels. Setting the LSB of each pixel to zero, a DCT is calculated for each block. The DCT matrix is then quantized with the quantization matrix corresponding to a 50% quality JPEG. The resulting quantized matrix is then encoded using 64 bits and the code is inserted into the LSBs of a distant 8×8 block. The watermarking process on average modifies 50% of pixels by one gray level. The quality of the reconstructed image is somewhat worse than 50% quality JPEG. If two LSBs are used for inserting the code (encoded quantized DCT coefficients) 128 bits can be used instead of 64 bits. For most blocks, this enables encoding almost all quantized DCT coefficients. Thus, the quality of the reconstructed image is roughly equivalent to a 50% JPEG compression.

The same authors [12] introduce another algorithm for self-embedding that is based on differential encoding. The differences between neighboring pixels are adjusted so that they mimic a decreased color depth approximation of the same image shifted by a third of the image dimension in a random direction. This provides a weak robustness to small noise adding while maintaining the quality of the embedded image quite reasonable (16 color approximation of the original). The embedded image gradually degrades with noise adding. The visual quality of the recovered image is still acceptable after adding noise of amplitude of 2 gray levels. However, the embedded information is lost after a 65% quality JPEG compression (the default setting in many commercial software products).

6 Conclusion

In this paper, we provide a comprehensive overview of steganographic techniques for tamper detection and authentication of digital images. The techniques are divided into several categories according to their ability to identify changes. Fragile watermarks can detect changes to every pixel and provide accurate information about the image integrity. However, it is not possible to distinguish small, innocuous changes due to common image processing operation from malicious changes, such as feature removal or addition. Semi-fragile watermarks are more robust and allow "authentication with a degree". It is possible to set a threshold in those techniques so that images after high quality JPEG compression, or contrast/brightness adjustment will still be considered authentic to a high degree. In the third category, we put techniques that attempt to authenticate image features. Such techniques are even more robust and enable robust distinction between innocuous and malicious modifications at the expense of losing the sensitivity to small

changes and sometimes the ability to localize modifications. However, it is possible to combine those watermarks with fragile watermarks if the fragile watermark is inserted as the second one. Such hybrid watermarks provide a much wider spectrum of protection against unauthorized modifications. The last category of data embedding techniques for tamper detection is called self-embedding. In those techniques, the image is embedded into itself in a judicious manner so that it is actually possible to later recover areas that have been cropped out, missing features, or identify newly added features. Self-embedding watermarks are fragile or semi-fragile and their disadvantage is that they cannot be combined with lossy compression.

One common problem of all watermarking techniques for tamper detection or authentication is that the watermark has to be image and key dependent in a non-trivial non-invertible manner to prevent creating forgeries that will go undetected. If the same key is used for authentication of multiple images, many block-based techniques will be vulnerable to a collage attack in which blocks from different images are combined. This is a serious issue if the authentication algorithm is expected to be implemented in digital cameras or surveillance video-cameras. It appears that "robust hash functions" also called robust bit extraction is the only practical way to solve this problem.

7 Acknowledgments

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under a grant number F30602-98-C-0009. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, or the U. S. Government.

8 References

[1] S. Walton, "Information Authentication for a Slippery New Age", *Dr. Dobbs Journal*, vol. 20, no. 4, pp. 18–26, Apr 1995.

[2] M. Yeung, and F. Mintzer, "An Invisible Watermarking Technique for Image Verification", *Proc. ICIP'97*, Santa Barbara, California, 1997.

[3] P. Wong, "A Watermark for Image Integrity and Ownership Verification", *Proc. IS&T PIC*, Portland, Oregon, 1998.

[4] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A Digital Watermark", *Proc. of the IEEE Int. Conf. on Image Processing*, vol. 2, pp. 86–90, Austin, Texas, Nov 1994.

[5] R. B. Wolfgang and E. J. Delp, "A Watermark for Digital Images", *Proc. IEEE Int. Conf. on Image Processing*, vol. 3, pp. 219–222, 1996.

[6] B. Zhu, M. D. Swanson, and A. Tewfik, "Transparent Robust Authentication and Distortion Measurement Technique for Images", preprint, 1997.

[7] J. Fridrich, *"Image Watermarking for Tamper Detection"*, Proc. ICIP '98, *Chicago, Oct 1998.*

[8] J. Fridrich, "Methods for Detecting Changes in Digital images", ISPACS99, Melbourne, November 4th–6th, 1998.

[9] D. Kundur and D. Hatzinakos, "Towards a Telltale Watermarking Technique for Tamper Proofing", *Proc. ICIP*, Chicago, Illinois, Oct 4–7, 1998, vol 2.

[10] R. B. Wolfgang and E. J. Delp, "Fragile Watermarking Using the VW2D Watermark", *Proc. SPIE, Security and Watermarking of Multimedia Contents*, San Jose, California, Jan 25–27, 1999, pp. 204–213.

[11] J. Fridrich and M. Goljan, "Protection of Digital images Using Self-Embedding", *Symposium on Content Security and Data Hiding in Digital Media*, New Jersey Institute of Technology, May 14, 1999.

[12] J. Fridrich and M. Goljan, "Images with Self-Correcting Capabilities", *ICIP'99*, Kobe, Japan, October 25–28,1999.

[13] J. Fridrich, "A Hybrid Watermark for Tamper Detection in Digital Images", *ISSPA'99 Conf.*, Brisbane, Australia, August 22–25, 1999.

[14] M. Schneider and S.-F. Chang, "A Content-Based Approach to Image Signature Generation and Authentication", *Proc. ICIP '96* vol. III, pp. 227–230, 1996.

[15] M. Holliman, N. Memon, and M. M. Yeung, "On the Need for Image Dependent Keys for Watermarking", *Proc. Content Security and Data Hiding in Digital Media*, Newark, NJ, May 14, 1999.

[16] J. Fridrich, "Robust Bit Extraction From Images", *ICMCS'99*, Florence, Italy, June 7–11, 1999.

[17] J. Fridrich and N. Memon, "Attack on a Fragile Watermarking Scheme", in preparation for *Electronic Imaging 2000, Security and Watermarking of Multimedia Contents*, San Jose, California, January, 2000.

A Review of Fragile Image Watermarks

Eugene T. Lin and Edward J. Delp

Video and Image Processing Laboratory (VIPER)
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, USA

ABSTRACT

Many image watermarks have been proposed to protect intellectual property in an age where digital images may be easily modified and perfectly reproduced. In a fragile marking system, a signal (watermark) is embedded within an image such that subsequent alterations to the watermarked image can be detected with high probability. The insertion of the watermark is perceptually invisible under normal human observation. These types of marks have found applicability in image authentication systems.

In this paper we discuss fragile marking systems and their desirable features, common methods of attack, and survey some recent marking systems.

KEYWORDS

Fragile watermarking, image authentication.

1 Introduction

The age of digital multimedia has brought many advantages in the creation and distribution of image content but the ease of copying and editing also facilitates unauthorized use, misappropriation, and misrepresentation. Content providers are naturally concerned about these issues and watermarking, which is the act of embedding another signal (the watermark) into an image, have been proposed to protect an owners rights [1].

Many types of watermarks have been developed for a variety of applications. Watermarks may be visible or invisible, where a visible mark is easily detected by observation while an invisible mark is designed to be transparent to the observer and detected using signal processing techniques [1]. The process of embedding the watermark requires modifying the original image and in essence the watermarking process inserts a controlled amount of "distortion" in the image. The recovery of this distortion allows the one to identify the owner of the image. Invisible or transparent marks use the properties of the human visual system to minimise the perceptual distortion in the watermarked image [1] [2]. In the class of transparent watermarks one may further categorise techniques as robust or fragile. A robust mark is designed to resist attacks that attempt to remove or destroy the mark. Such attacks include lossy compression, filtering, and geometric scaling. A fragile mark is designed to detect slight changes to the watermarked image with high probability. The main application of fragile watermarks is in content authentication. Most of the work, as reported in the literature, in watermarking is in the area of robust techniques. Many important applications could benefit from the use of fragile watermarks.

This work was supported by a grant from Texas Instruments and an equipment grant from Intel. Address all correspondence to E. J. Delp, ace@ecn.purdue.edu, http://www.ece.purdue.edu/~ace, or +1 765 494 1740.

2 Fragile Marking Applications

A fragile watermark is a mark that is readily altered or destroyed when the host image is modified through a linear or nonlinear transformation [3]. Fragile marks are not suited for enforcing copyright ownership of digital images; an attacker would attempt to destroy the embedded mark and fragile marks are, by definition, easily destroyed. The sensitivity of fragile marks to modification leads to their use in image authentication. That is, it may be of interest for parties to verify that an image has not been edited, damaged, or altered since it was marked.

Image authentication systems have applicability in law, commerce, defense, and journalism. Since digital images are easy to modify, a secure authentication system is useful in showing that no tampering has occurred during situations where the credibility of an image may be questioned. Common examples are the marking of images in a database to detect tampering [4][5], the use in a "trustworthy camera" so news agencies can ensure an image is not fabricated or edited to falsify events [6], and the marking of images in commerce so a buyer can be assured that the images bought are authentic upon receipt [7]. Other situations include images used in courtroom evidence, journalistic photography, or images involved in espionage.

Another method to verify the authenticity of a digital work is the use of a signature system [8]. In a signature system, a digest of the data to be authenticated is ob-

tained by the use of cryptographic hash functions [8][9]. The digest is then cryptographically signed to produce the signature that is bound to the original data. Later, a recipient verifies the signature by examining the digest of the (possibly modified) data and using a verification algorithm determines if the data is authentic. While the purpose of fragile watermarking and digital signature systems are similar, watermarking systems offer several advantages compared to signature systems [10] at the expense of requiring some modification (watermark insertion) of the image data. Since a watermark is embedded directly in the image data, no additional information is necessary for authenticity verification. (This is unlike digital signatures since the signature itself must be bound to the transmitted data.) Therefore the critical information needed in the authenticity testing process is discreetly hidden and more difficult to remove than a digital signature. Also, digital signature systems view an image as an arbitrary bit stream and do not exploit its unique structure. Therefore a signature system may be able to detect that an image had been modified but cannot characterise the alterations. Many watermarking systems can determine which areas of a marked image have been altered and which areas have not, as well as estimate the nature of the alterations.

2.1 Image Authentication Framework

The framework for embedding and detecting a fragile mark is similar to that of any watermarking system. An owner (or an independent third party authority) embeds the mark into an original image (see Figure 1).

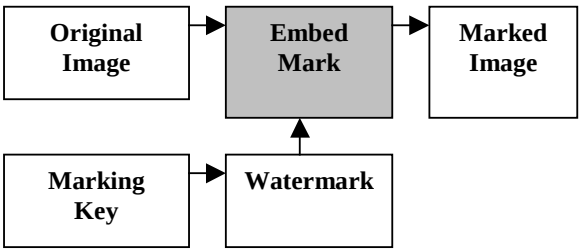


Figure 1: Watermark Embedding

The marking key is used to generate the watermark and is typically an identifier assigned to the owner or image. The original image is kept secret or may not even be available in some applications such as digital camera. The marked image may be transmitted, presented, or distributed. The marked image is perceptually identical to the original image under normal observation. See Figure 2 and Figure 3 for an example of original and marked images using the fragile marking technique described in [9][11].

When a user receives an image, they use the detector to evaluate the authenticity of the received image (see Figure 4). The detection process also requires knowledge of "side information." This side information may be the marking key, the watermark, the original image, or other information. The detector is usually based on statistical detection theory whereby a test statistic is generated and from that test statistic the image is determined to be authentic. If it is not authentic then it would be desirable for the detector to determine where the image has been modified.

Figure 2: Original Image

Figure 3: Watermarked Image

The side information used by the detector is very important in the overall use of a fragile watermark. Techniques that require that the detector have the original image are known as private watermarks while techniques that do require the detector to have the original image are known as public watermarks. To be effective a fragile watermarking system must be a public technique. In many applications the original image may never be available since it might have been watermarked immediately upon creation.

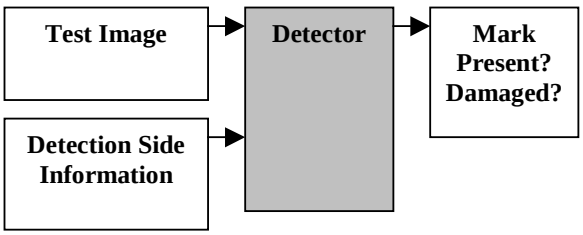


Figure 4: Watermark Detection

In database applications the owner or authority who marks the images is often the party interested in verifying that they have not been altered a subsequent time. For example, in a medical database it is important that any modifications to images be detected. In other applications, such as commerce, the verifying parties are dis-

tinct from the marking entity. In these cases, it is desirable to choose a system where the marking and detection information are distinct. In such a system, the ability to determine the authenticity of images does not also grant the ability to mark images. The vast majority of fragile systems described in the current literature do not implement this approach.

3 Features of Fragile Marking Systems

We now present desirable features of fragile marking systems, noting that the relative importance of these features will depend on the application. Applications may have requirements other than the ones mentioned. In addition to the features described below and the desired properties we previously mentioned, other properties can be found in [4][12][13]:

1. Detect tampering. A fragile marking system should detect (with high probability) any tampering in a marked image. This is the most fundamental property of a fragile mark and is a requirement to reliably test image authenticity. In many applications it is also desirable to provide an indication of how much alteration or damage has occurred and where it is located (see Feature 4 below).

2. Perceptual Transparency. An embedded watermark should not be visible under normal observation or interfere with the functionality of the image [1]. In most cases this refers to preserving the aesthetic qualities of an image, however if an application also performs other operations on marked images (such as feature extraction) then these operations must not be affected. Unfortunately there is not a lot of information how the "noise" introduced by marking process affects other image processing operations [14]. This is an open research problem. Also, transparency may be a subjective issue in certain applications and finding measures, which correlate well with perceived image quality, may be difficult.

3. Detection should not require the original image. This was discussed in detail in Section 3. As mentioned above the original image may nor exist or the owner may have good reason not to trust a third party with the original (since the party could then place their own mark on the original and claim it as their own.)

4. Detector should be able to locate and characterise alterations made to a marked image. This includes the ability to locate spatial regions within an altered image which are authentic or corrupt. The detector should also be able to estimate what kind of modification had occurred.

5. The watermark detectable after image cropping. In some applications, the ability for the mark to be detected after cropping may be desirable. For example, a party may be interested in portions (faces, people, etc.) of a larger, marked image. In other applications, this feature is not desired since cropping is treated as a modification.

6. The watermarks generated by different marking keys should be "orthogonal" during watermark detection. The mark embedded in an image generated by using a particular marking key must be detected only by providing the corresponding detection side information to the detector. All other side information provided to the detector should fail to detect the mark.

7. The marking key spaces should be large. This is to accommodate many users and to hinder the exhaustive search for a particular marking key even if hostile parties are somehow able to obtain both an unmarked and marked versions of a particular image.

8. The marking key should be difficult to deduce from the detection side information. This is particularly important in systems that have distinct marking and detection keys. Usually in such systems the marking key is kept private and the corresponding detection side information may be provided to other parties. If the other parties can deduce the marking key from the detection information then they may be able embed the owner's mark in images that the owner never intended to mark.

9. The insertion of a mark by unauthorised parties should be difficult. A particular attack mentioned in [4] is the removal of the watermark from a marked image and subsequently inserting it into another image.

10. The watermark should be capable of being embedded in the compressed domain. This is not the same as saying the watermark should survive compression, which can be viewed as an attack. The ability to insert the mark in the compressed domain has significant advantage in many applications.

4 Attacks on Fragile Marks

One must be mindful of potential attacks by malicious parties during the design and evaluation of marking systems. It may be practically impossible to design a system impervious to all forms of attack, and new methods to defeat marking systems will be invented in time. But certainly knowledge of common attack modes is a requirement for the design of improved systems.

The first type of attack is blind modification of a marked image (that is, arbitrarily changing the image assuming no mark is present). This form of attack should be readily recognized by any fragile mark, yet we mention it because it may be the most common type of attack that a marking system is to defeat. Variations of this attack include cropping and localized replacement (such as substituting one person's face with another.) The latter type of modification is a significant reason why an application may want to be able to indicate the damaged regions within an altered image.

Another type of attack is to attempt to modify the marked image itself without affecting the embedded mark or creating a new mark that the detector accepts as authentic. Some weak fragile marks easily detect random changes to an image but may fail to detect a carefully constructed modification. An example is a fragile mark embedded in the least-significant bit plane of an image. An attempt to modify the image without realizing that a mark is expressed in the LSB is very likely to disturb the mark and be detected. However, an attacker that may attempt to modify the image without disturbing

any LSBs or substitute a new set of LSBs on a modified image that the detector classifies as authentic.

Attacks may also involve using a known valid mark from a marked image as the mark for another, arbitrary image [4]. The mark-transfer attack is easier if it is possible to deduce how a mark is inserted. This type of attack can also be performed on the same image; the mark is first removed, then the image is modified, and finally the mark is re-inserted.

An attacker may be interested in completely removing the mark and leaving no remnants of its existence (perhaps so they can deny ever bearing witness to an image which has their mark embedded in it). To do so, an attacker may attempt adding random noise to the image, using techniques designed to destroy marks (such as Stirmark [15]), or using statistical analysis or collusion to estimate the original image.

An attacker may also attempt the deduction of the marking key used to generate the mark. The marking key is intimately associated with an embedded mark, so if it is possible to isolate the mark the attacker can then study it in an attempt to deduce the key (or reduce the search space for the marking key). Once the key is deduced, the attacker can then forge the mark into any arbitrary image.

There are also known attacks that involve the authentication model itself and not so much on the specific mark in an image. Attacks on authentication systems over insecure channels are also discussed in [8] and similar vulnerabilities can apply to watermarking systems.

5 Examples of Fragile Marking Systems

We now survey some fragile marking systems described in the literature. We can classify the techniques as ones which work directly in the spatial domain or in the transform (DCT, wavelet) domains.

5.1 Spatial Domain Marks

Early Early fragile watermarking systems embedded the mark directly in the spatial domain of an image, such as techniques described in Walton [16] and van Schyndel et al. [17]. These techniques embed the mark in the least significant bit plane for perceptual transparency. Their significant disadvantages include the ease of bypassing the security they provide [3][18] and the inability to lossy compress the image without damaging the mark.

Wolfgang and Delp [11] extended van Schyndel's work to improve robustness and localization in their VW2D technique. The mark is embedded by adding a bipolar M-sequence in the spatial domain. Detection is via a modified correlation detector. For localization, a blocking structure is used during embedding and detection. This mark has been compared to other approaches using hash functions [9].

P. Wong describes another fragile marking technique in [19], which obtains a digest using a hash function. The image, image dimensions, and marking key are hashed during embedding and used to modify the least-significant bit plane of the original image. This is done in such a way that when the correct detection side information and unaltered marked image are provided to the detector, a bi-level image chosen by the owner (such as a company logo or insignia), is observed. This technique has localization properties and can identify regions of modified pixels within a marked image.

The technique of Yeung and Mintzer [3], whose security is examined in [10], is also one where the correct detection information results in a bi-level image. However, the embedding technique is more extensive than inserting a binary value into the least-significant bit plane. The marking key is used to generate several pseudo-random look-up tables (one for each channel or color component) that control how subsequent modification of the pixel data will occur. Then, after the insertion process is completed, a modified error diffusion process can be used to spread the effects of altering the pixels, making the mark more difficult to see. As discussed in [10], the security of the technique depends on the difficulty of inferring the look-up tables. The search space for the table entries can be drastically reduced if knowledge of the bi-level watermark image is known. A modification (position-dependent lookup tables) is proposed in [10] to dramatically increase the search space.

5.2 Transform Domain Marks

Various transformations, such as the discrete cosine transform (DCT) and wavelet transforms, are widely used for lossy image compression and much is known of how the actual transform coefficients may be altered (quantized) to minimize perceptual distortion [1]. There is also a great deal of interest in transform embedding for robust image marking systems to make embedded marks more resilient to attacks.

There are advantages for fragile marking systems to use the transform domain as well. Many fragile marking systems are adapted from lossy compression systems (such as JPEG), which have the benefit that mark in embedded in the compressed representation. The properties of a transform can be used to characterize how an image has been damaged or altered. Also, applications may require a mark to possess some robustness to certain types of modification (such as brightness changes) yet be able to detect other modifications (e.g. local pixel replacemnet).

Wu and Liu [13] describe a technique based on a modified JPEG encoder. The watermark is inserted by changing the quantized DCT coefficients before entropy coding. A special lookup table of binary values (whose design is constrained to ensure mark invisibility) is used to partition the space of all possible DCT coefficient values into two sets. The two sets are then used to modify the image coefficients to encode a bi-level image (such as a logo.) To reduce the blocking effects of altering coefficients, it is suggested that the DC coefficient and any coefficients with low energy be not marked.

Kundur and Hatzinakos [12] and Xie and Arce [20] describe techniques based on the wavelet transform. Kundur embeds a mark by modifying the quantization process of Haar wavelet transform coefficients while Xie se-

lectively inserts watermark bits by processing the image after it is in a compressed form using the SPIHT algorithm 0. A wavelet decomposition of an image contains both frequency and spatial information about the image hence watermarks embedded in the wavelet domain have the advantage of being able to locate and characterize tampering of a marked image.

6 Conclusion

Fragile watermarking is the embedding of a signal (the watermark) into an image so that modifications to the resulting marked image can be detected with high probability. A fragile marking system is useful in a variety of image authentication applications. We feel that fragile watermarking has been somewhat ignored by the watermarking community in favor of robust techniques. There are many open research problems that need to be addressed in fragile watermarks such as the development of techniques that allow the detection of authenticity without permitting mark embedding. Many important applications can benefit from the use of fragile techniques.

7 References

[1] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual Watermarks for Digital Images and Video", *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1108-1126, July 1999.

[2] M. Swanson, M. Kobayashi, A. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1064-1087, June 1998.

[3] M. Yeung and F. Mintzer, "Invisible watermarking for image verification," *Journal of Electronic Imaging*, vol. 7, no. 3, pp. 578-591, July 1998.

[4] F. Mintzer, G. Braudaway, and M. Yeung, "Effective and ineffective digital watermarks," *Proceedings of the IEEE International Conference on Image Processing*, pp. 9-12, Santa Barbara, California, October 1997.

[5] F. Mintzer, G. Braudaway, and A. Bell, "Opportunities for watermarking standards," *Communications of the ACM*, vol. 41, no. 7, pp. 57-64, July 1998.

[6] G. Friedman, "The trustworthy digital camera: Restoring credibility to the photographic image," *IEEE Transactions on Consumer Electronics*, vol. 39, pp. 905-910, November 1993.

[7] P. W. Wong, "A public key watermark for image verification and authentication," *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, pp. 455-459, Chicago, Illinois, October 1998.

[8] D. Stinson, *Cryptography Theory and Practice*, CRC Press, Boca Raton, 1995.

[9] R. Wolfgang and E. Delp, "Fragile watermarking using the VW2D watermark," *Proceedings of the IS&T/SPIE Conference on Security and Watermarking of Multimedia Contents*, pp. 204-213, San Jose, California, January 1999.

[10] N. Memon, S. Shende, and P. Wong, "On the security of the Yueng-Mintzer Authentication Watermark," *Final Program and Proceedings of the IS&T PICS 99*, pp. 301-306, Savanna, Georgia, April 1999.

[11] R. Wolfgang and E. Delp, "A watermark for digital images," *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, pp. 219-222, 1996.

[12] D. Kundur and D. Hatzinakos, "Towards a telltale watermarking technique for tamper-proofing," *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, pp. 409-413, Chicago, Illinois, October 1998.

[13] M. Wu and B. Liu, "Watermarking for image authentication," *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, pp. 437-441, Chicago, Illinois, October 1998.

[14] S. Pankanti and M. Yeung, "Verification watermarks on fingerprint recognition and retrieval," *Proceedings of the IS&T/SPIE Conference on Security and Watermarking of Multimedia Contents*, pp. 66-78, San Jose, California, January 1999.

[15] Stirmark software: `http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark`, 1997.

[16] S. Walton, "Information authentication for a slippery new age," *Dr. Dobbs Journal*, vol. 20, no. 4, pp. 18-26, April 1995.

[17] R. van Schyndel, A. Tirkel, and C. Osborne, "A digital watermark," *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, pp. 86-90, Austin, Texas, November 1994.

[18] J. Fridrich, "Image watermarking for tamper detection," *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, pp. 404-408, Chicago, Illinois, October 1998.

[19] P. Wong, "A watermark for image integrity and ownership verification," *Final Program and Proceedings of the IS&T PICS 99*, pp. 374-379, Savanna, Georgia, April 1999.

[20] L. Xie and G. Arce, "Joint wavelet compression and authentication watermarking," *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, pp. 427-431, Chicago, Illinois, October 1998.

[21] A. Said and W. Pearlman, "A new fast and efficient image codec based on set partitioning and hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-250, June 1996.

Public Watermarking Surviving General Scaling and Cropping: An Application for Print-and-Scan Process

Ching-Yung Lin

Department of Electrical Engineering
Columbia University
New York, NY 10027, USA

cylin@ctr.columbia.edu

ABSTRACT

Scaling and cropping are very common in today's image processing software. When an image is printed-and-scanned, the final image is generally a cropped version of the rotated, scaled original image, with additional noises. The cropped image usually does not have the same aspect ratio as the original, so the DFT coefficients of the cropped image and the original would be quite different. In this paper, we propose an algorithm embedding spread spectrum watermarks in the DFT magnitudes of the log-log map magnitudes in the Fourier domain. These watermarks would be resistant to any aspect ratio of scaling and cropping and pixel value distortions as in the print-and-scan process.

KEYWORDS

Public Watermarking, Rotation, Scaling, Cropping, DFT, Scanning.

1 Introduction

Watermarking methods embed information in a multimedia object in which the modification should be imperceptible. The embedded information can be used as a proof of ownership or as a kind of secret data transmission. Several types of watermarking systems have been proposed since 1990 [1]. Among them, public watermarking is considered to have a broader application value, because it can detect the watermarks without the original object. There are two types of public watermarking systems: one-bit watermark and multiple-bit watermark [2]. The one-bit public watermarking system (also referred to as *semi-private watermarking*) detects the existence of a specific identification watermark in the multimedia content. It usually serves as evidence of ownership. The multiple-bit public watermarking system (or *blind watermarking*) extracts the embedded information of the watermark. It is usually used for data hiding or ownership declaration.

Today, the print-and-scan process is commonly used for image reproduction and distribution. It is popular to transform images between the electronic digital format and the printed pictures. Therefore, for copyright protection, an effective watermarking system should be able to detect or extract watermarks, regardless the media format of the image. However, while most of previous research on watermarking focused on the electronic digital image, it was not clear how the print-and-scan process affects the image, nor how a watermarking system can survive it.

The rescanned image is generally distorted in both the pixel values and the geometric boundary. The distortion of pixel values is caused by (1) the luminance, contrast, gamma correction and chromnance variations, and (2) the blurring of adjacent pixels. These distortions are the typical effects of the printer and scanner.

The distortion of the geometric boundary in the print-and-scan process is caused by rotation, scaling, and cropping (RSC). We would like to point out that the geometric distortion in the scanning process could not be adequately modeled by the well-known rotation, scaling, and translation (RST) effects, because of the design of today's Graphic User Interface (GUI) for the scanning process as in Fig 1. The RST is usually used to model the geometric distortion on the image of an observed object. It has been widely used in pattern recognition. In those cases, the capturing window of camera is usually predetermined, *i.e.* the size of the captured image is usually determined by the system. However, in the scanning process, the scanned image may only cover a part of the original picture and may have an arbitrary cropped image size. Cropping introduces large changes to the image. Detailed discussion of the modeling of the Print-and-Scan process can be found in [3].

Some watermarking methods have been proposed to solve the related problems. O'Ruanaidh and Pun [4] first suggested a watermarking method based on the log-polar map of the Fourier coefficients (also known as the Fourier-Mellin Transform). They proposed that the Discrete Fourier Transform (DFT) magnitudes of the Fourier-

Mellin coefficients can be used to embed the watermark, because their well-known shifting property to RST distortions. However, the Fourier-Mellin transform can only deal with uniform scaling (*i.e.*, the same scaling factor in both horizontal and vertical direction), as well as cropping which keeps the original aspect ratio. Therefore, if the image is cropped with arbitrary aspect ratio as in the print-and-scan process, the Fourier-Mellin-based methods would become invalid. A detailed discussion of the Fourier-Mellin-based watermarking method can be found in [5], where we proposed an RST resilient watermarking method and solved many implementation difficulties. Another method proposed by Pereira *et.al.* [6] embeds a registration pattern as well as a watermark into an image. This is an effective solution but can have the problems of reducing the fidelity and tampering the watermark.

Another problem of the previous proposed methods is that their theorems are derived in the continuous Fourier domain, but implemented in the DFT domain. Although DFT coefficients are the sampling values of the continuous Fourier coefficients, their properties are not simply the same, because the sampling rate, the aliasing effect, and the discontinuity on the boundary of periods affect DFT coefficients. The DFT coefficients are, in fact, sampled from a repeated discrete image. Their sampling positions in the continuous Fourier domain are determined by the repetition period, which is, in many cases, the size of image or the smallest radix-2 size larger than the size of image. It may be noticed that, without special considerations, DFT-based robust watermarking methods can automatically survive scaling with any aspect ratio and cropping with a fixed aspect ratio of the original image. This phenomenon comes from the fact that the DFT coefficients are at the same sampling positions in the continuous Fourier domain after these manipulations [3], if the sizes of DFT points are always the sizes of image (*e.g.*, using 256x256 DFT for 256x256 images, and 128x128 DFT for their down-sampled 128x128 versions). However, if the image is cropped with an arbitrary aspect ratio, its DFT coefficients will be quite different because they are sampled at different positions in the Fourier domain. The changes of the DFT coefficients on the cropped image are similar to the changes of the continuous Fourier coefficients of the scaled and translated original image. We will discuss them further in Section 2.

Acknowledging the RSC effects during the print-and-scan processes, in this paper, we propose an algorithm embedding spread spectrum watermarks in the DFT magnitude of the log-log map coefficients on the DFT domain. We will show in Section 2 that scaling and cropping an image with arbitrary aspect ratio results in a simple two-dimensional translation in the log-log map of DFT coefficient magnitudes. Therefore, the DFT magnitudes of this map will be invariant after scaling and cropping. Because, so far, we could not find a reliable transformation which simultaneously provides applicable properties to RSC, a drawback of the proposed system is that it is not rotation invariant and one would have to test the scanned image several times within the possible range of rotation. In practice, this may not be a serious problem because the rotation angle of scanned image would not be too much, while the range of cropping and scaling are usually unbounded. The proposed algorithm can also be applied to the images edited by general image software, in which scaling and cropping are more common than rotation. We will explain the watermarking method in detail in Section 3, and show some preliminary experimental results in Section 4.

Figure 1: The control windows of scanning process. The scanned image only includes the cropped area.

2 Modeling of the Print-and-Scan Process

When a user scans a picture, at the first step, he/she has to place the picture on the flatbed of the scanner. This may introduce a small orientation, if the picture is not well placed. Then, the scanner scans the whole flatbed to get a previewed low-resolution image. After this process, the user subjectively selects a cropping window to decide an appropriate range of the picture. Then, the scanner scans the picture again with a higher resolution to get a scanned image. The scanned image is usually a different size because the resolution in the scanner and the printer are generally different. Usually, it includes only a part of the original image. In our tests, the image is not generally rotated because users usually place the picture or document along the corner of the flatbed. Even if the picture is not well placed, the rotation angle is generally within a small degree, *e.g.*, ±3 degrees.

Assume we have a continuous finite support image,

$$x(t_1,t_2)=\begin{cases} x_0(t_1,t_2), & t_1\in[-\frac{T_1}{2},\frac{T_1}{2}],t_2\in[-\frac{T_2}{2},\frac{T_2}{2}] \\ 0, & \textit{elsewhere} \end{cases} \quad (1)$$

If this image is scaled by λ_1 in the t_1-axis and λ_2 in the t_2-axis, then

$$x_S(t_1,t_2)=x(\tfrac{t_1}{\lambda_1},\tfrac{t_2}{\lambda_2})\xleftrightarrow{F}X(\lambda_1 f_1,\lambda_2 f_2)=X_S(f_1,f_2) \quad (2)$$

From Eq. (2), if we assume a transformation, *LL*, which maps the original Cartesian coordinate points to their log-log coordinate points [6], *s.t.*,

$$(LL \circ X)(f_1', f_2') = X(e^{f_1'}, e^{f_2'}) \quad (3)$$

then we can get,

$$(LL \circ X_S)(f_1', f_2') = (LL \circ X)(f_1' + \log\lambda_1, f_2' + \log\lambda_2) \quad (4)$$

If the image is translated, then we should change the X_S and X to their magnitudes $|X_S|$ and $|X|$ in Eq. (4).

For cropping, we can consider the cropped image, x_C, as a subtraction of the discarded area, $x_{\bar{C}}$, from the original image, *x*. Then, this equation,

$$X_C(f_1, f_2) = X(f_1, f_2) - X_{\bar{C}}(f_1, f_2) \quad (5)$$

represents the cropping effect in the continuous Fourier domain. If the discarded area is much smaller than the original image, then the Fourier coefficients of the discarded area, $X_{\bar{C}}$, can be considered as noises in Eq. (5).

Because we only have the discrete images before and after scanning, in practical cases, DFT is usually used as a sampling method on the frequency domain and takes advantage of FFT. The relationships of (continuous) Fourier transform (FT), Fourier Series (FS), and DFT are:

- The FS coefficients are the samples of the FT coefficients of a finite support continuous signal. They are calculated by repeating the signal in the time domain. Assume the repetition period is T. Once the signal becomes periodic in the time domain, its FT coefficients will have non-zero values only in the n/T positions. These values are the multiplication of the FS coefficients and a delta function. We should notice that the FS coefficients are always proportional to the FT values of the original non-periodic signal in the n/T positions.
- The DFT coefficients represent the FS coefficients of the discretized original signal. After the original signal is sampled, its FS coefficients would become periodic. The DFT coefficients are the FS coefficients in a period. Smaller sampling frequency in the time domain would introduce an aliasing effect in the frequency domain. That can be considered as additive noises to the DFT coefficients.

From the above descriptions, we know that the repetition period of the original signal decides the sampling positions of DFT coefficients in the frequency domain. In the scaling cases, if the repetition is always the same as the image size, then the FS of the original continuous image, $\tilde{X}$, and the scaled image, $\tilde{X}_S$, should be the same. That is,

$$\tilde{X}_S(n_1, n_2) = X_S(\tfrac{n_1}{T_{S1}}, \tfrac{n_2}{T_{S2}}) = X(\tfrac{n_1\lambda_1}{T_{S1}}, \tfrac{n_2\lambda_2}{T_{S2}}) = X(\tfrac{n_1}{T_1}, \tfrac{n_2}{T_2}) = \tilde{X}(n_1, n_2) \quad (6)$$

where T_{S1} and T_{S2} are the sizes of the scaled image. Adding the concern of discretization in the spatial domain, we can get the DFT coefficients in the scaled case, $\hat{X}_S$ as

$$\hat{X}_S(n_1, n_2) = \hat{X}(n_1, n_2) + N_{sampling} \quad (7)$$

where $\hat{X}$ is the DFT of original image. In Eq. (7), the sampling noises happen when images are down-sampled. If an image is cropped, then the changes of DFT coefficients are introduced from three factors: (1) *the change of image size*, (2) *the information loss of the discarded area*, and (3) *the translation of the origin point of the image*. Assume the size of cropped image is $\alpha_1 T_1$ x $\alpha_2 T_2$. If the size of DFT is the same as the size of the cropped image, then we can obtain the DFT coefficients after scaling and cropping,

$$|\hat{X}_{SC}(n_1, n_2)| \models |\hat{X}(\tfrac{n_1}{\alpha_1}, \tfrac{n_2}{\alpha_2}) + \hat{N}_{SC}(n_1, n_2)| \quad (8)$$

where

$$\hat{N}_{SC}(n_1, n_2) = -\hat{X}_{\bar{C}}(\tfrac{n_1}{\alpha_1}, \tfrac{n_2}{\alpha_2}) + N_{sampling} \quad (9)$$

In Eq. (9), if the cropped area include the entire original image, i.e., $\alpha_1, \alpha_2 >= 1$, then the effect of the discarded area can be ignored. If the cropping ratios are too small, then the power loss in the discarded area may not be just ignored as noises. In our experiments, the reliable minimum thresholds are at about 0.8, which may be small enough for most scanned images [3]. In Eq. (9), strictly speaking, there is no definition in $\hat{X}$ at the non-integer positions. But, since $\hat{X}$ are samples of *X*, we can set $\hat{X}(\tfrac{n_1}{\alpha_1}, \tfrac{n_2}{\alpha_2}) = X(\tfrac{n_1}{\alpha_1 T_1}, \tfrac{n_2}{\alpha_2 T_2})$ directly from the original Fourier coefficients. In practical applications, these values are generally obtained from interpolation.

In addition to using the same size DFT of the scaled and cropped image, some cases use the smallest radix-2 FFT that are larger than the image size. In that case, Eq. (8) and (9) are still applicable, but α_1 and α_2 should be replaced by other values. Detailed modeling description of cropping and scaling can be found in [3].

Comparing Eq. (2) and Eq. (8), we can find the changes of the DFT coefficients after scaling & cropping and those of the continuous Fourier coefficients after scaling are similar. Therefore, after scaling & cropping, as in Eq. (3) and Eq. (4), the log-log map of the DFT coefficient magnitudes will suffer simple shift. Then, the DFT magnitudes of the log-log map of DFT magnitude should be similar. We can use this property for watermarking.

3 Algorithm

The embedding algorithm is shown in Figure 2. At the first step, we scale the image to a fixed size of 256x256 pixels. As shown in Eq. (7), scaling with arbitrary aspect ratio does not affect the DFT coefficients, if all images are resized to a small standard size, it can reduce both the computational cost and implementation cost. We

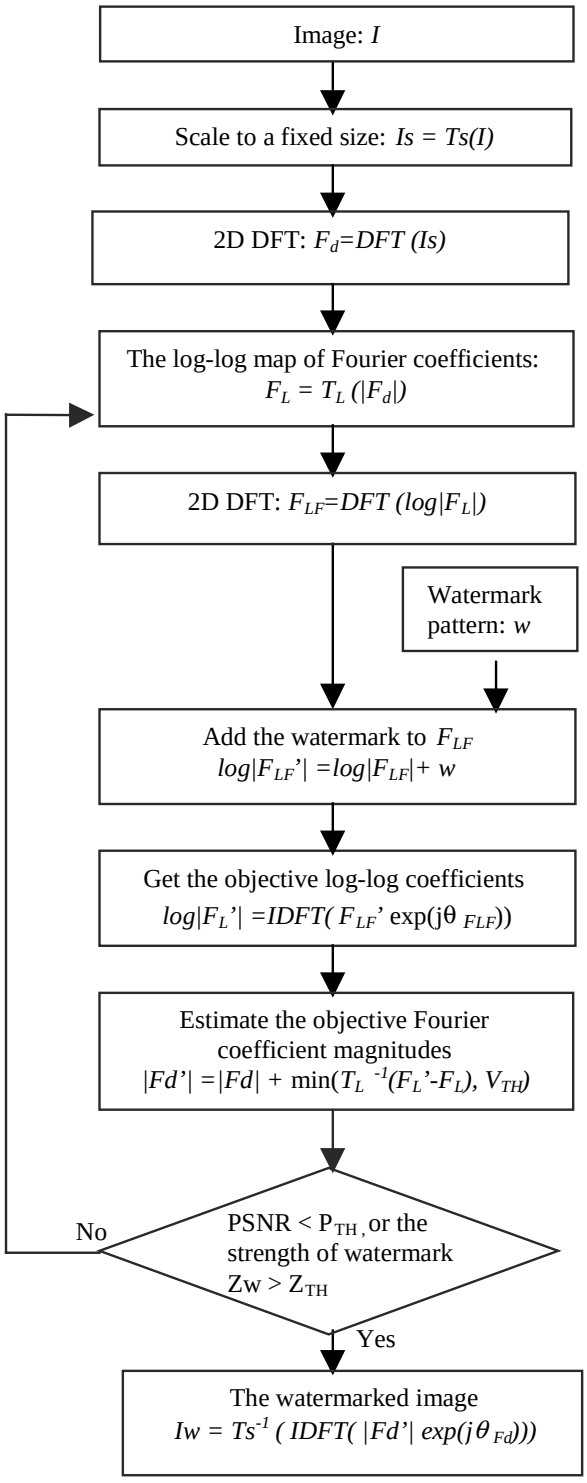


Figure 2: The embedding process

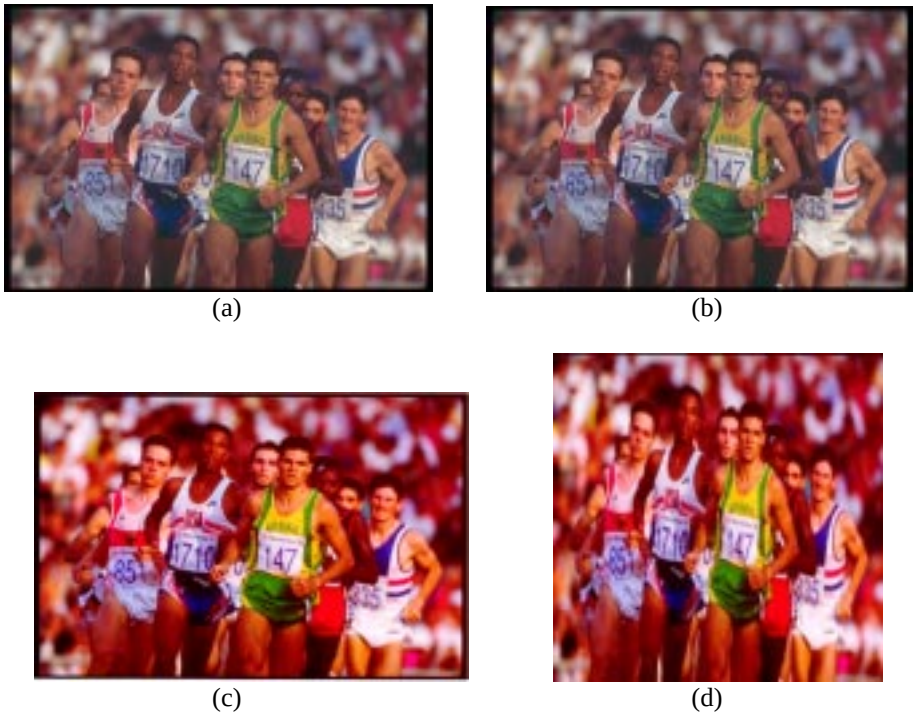
only use this standard size image for calculating the amount of additive watermarks on this scale. Then, these additive watermarks are resized to the original size and added to the original image. In this way, the watermarked image will not suffer the fidelity loss of down-sampling. Since the image size is generally larger than this standard size, the additive watermarks are usually scaled up, which will not introduce the sampling noise in Eq. (7).

The second step is to get the log-log map from the DFT magnitudes. We use the bilinear interpolation because it is easier to implement and can get reasonable results. We noticed that the system has to interpolate the DFT magnitudes, instead of interpolating the complex DFT coefficients and then get their magnitudes. Intuitively, the latter method seems to be more correct. However, because the image may be translated by cropping, the phase change of DFT coefficients will result in incorrect interpolated log-log map magnitudes [3].

We then use the spread spectrum method to embed watermarks [4][7]. Two kinds of watermarks are tested in our system. For the 1-bit watermark, we generate a user identification code and take its convolution with pseudo-noise patterns as an additive watermark on the log values of DFT magnitudes of the log-log map magnitudes. For the multiple-bit watermark, we use the same method as in [4], which uses the shift position of the fixed pseudo-noise pattern to represent the embedded information. The watermarks are only embedded in the mid-band areas, e.g., 33 <= n_1, n_2 <= 128 in the 256x256 DFT, to avoid significant effect on the fidelity and to be robust to other manipulations such as compression. We embed the same pattern in the four quadrants of the map to satisfy the real pixel value constraint, and to make consistent watermark detection if the scanned image is rotated for 90, 180, 270 degrees.

Because the log-log map coefficients and the DFT coefficients are not 1-to-1 mapping, the embedding process could not be done directly. We can use the iterative method to estimate the change in the log-log map, but manipulate the original DFT coefficients to make the mapping results as close as possible. The iteration process will be continued until either the strength of watermark or the fidelity loss is larger than a threshold. We use the Z-statistic as a measure of the strength of watermark [8].

The first few steps of watermark detection process are the same as in the embedding process. For the 1-bit watermark, the Z-statistic between the DFT magnitudes of log-log map and the known watermark pattern is calculated as an indication of the existence of watermark. In general, if Z>3, then it is considered as the existence of watermark with a false positive rate of 10^{-3}. For the multiple-bit watermarks, the embedded symbols are extracted by calculating the largest Z-statistic value of the position of shifted PN patterns.



(a) (b)

(c) (d)

Figure 3: Experimental results:
(a) original image [384x256];
(b) watermarked image, PSNR= 45.14dB, Z=16.26;
(c) print-and-scanned image [401x268], Z=6.37;
(d) after cropping, resizing and JPEG compression [300x300, CR=10:1], Z=3.16.

4 Experiments

In our experiments, we tested the watermarked images at multiple stages of manipulation, because that is usually the real case.

We tested the robustness of watermark on a color image randomly chosen from the Corel Stock Photo Library. The original image size is 384x256. After the embedding process, the PSNR of watermarked image is 45.14 dB and the original strength of watermark, Z, is 16.26. There is no visible degradation on the watermarked image. These images are shown in Fig. 3(a) and 3(b).

We first test the watermarked image with general cropping and resizing functions using the Paint Shop Pro. The results are shown here.

Manipulation (step by step)	Z value
(1) Cut off borders (cropping) to 365x248	7.60
(2) Resize it to the original size: 384x256.	7.67
(3) Crop the image again to 339x253 (84%x96%) of the original area ratio.	5.00
(4) Resize the image: 256x256	4.45
(5) JPEG compression (Paint Shop Prop default QF, CR=10.6:1), Z=3.88	3.88

We print the watermarked image on an inkjet paper using the EPSON Stylus Photo EX. The physical size of printed image is 13.5x9cm^2. Then, we use the HP Scanjet 4C to scan this picture with defaulted resolution. The scanner automatically adjusts the brightness, contrast, gamma correction, and all other settings. We then test several manipulations on the scanned image. The images after step (1) and step (4) are shown in Fig 3(c) and 3(d).

Manipulation (step by step)	Z value
(1) Print & Scan: Image Size: 401x268.	6.37
(2). Crop the image to 385x259	4.70
(3) Resize the image: 300x300	4.43
(4) JPEG compression (Paint Shop Prop default QF, CR=9.97:1).	3.16

We use another image, which is a parrot with trees as background, to test the multiple-bit watermarking in the proposed algorithm. An information of "SIGNAFY" represented by 14 3-bit symbols is embedded to the image. The original image size is 512x768. After watermarking, the PSNR is 46.43 dB. Then, this image is cropped to 486x740, resized to 512x512, and compressed by JPEG. We find that all the bits can be extracted correctly after these attacks.

5 Summary and Future Direction

In this paper, we proposed a public watermarking algorithm that is robust to the print-and-scan and general scaling and cropping processes. Preliminary experiments have shown the effectiveness of this algorithm. In the future, we will go on to a test of large image database with the proposed method, and look for a method which can deal with the rotation, scaling, and cropping simultaneously.

6 Acknowledgements

This work was performed at and funded by the NEC Research Institute and Signafy, Inc. The author would like to thank Yuiman Lui, Matt Miller, and Jeff Bloom for sharing ideas, and Dr. I.J. Cox, and Prof. S.-F. Chang for helpful discussions and reviewing this paper.

7 References

[1] M. Kutter and F. A. P. Petitcolas, "A Fair Benchmark for Image Watermarking Systems," *SPIE Security and Watermarking of Multimedia Content*, San Jose, CA, Jan 1999.

[2] J. Fridrich and M. Goljan, "Comparing Robustness of Watermarking Techniques," *SPIE Security and Watermarking of Multimedia Content*, San Jose, CA, Jan 1999.

[3] C.-Y. Lin, S.-F. Chang, and I. J. Cox, "Modeling Image Print and Scan Processes," submitted to *Intl. Symp. on Multimedia Information Processing (ISMIP 99)*, Taipei, Taiwan, Dec. 1999.

[4] J. O'Ruanaidh and T. Pun, "Rotation, Scale and Translation Invariant Spread Spectrum Digital Image Watermarking," *Signal Processing 66 (1998)*.

[5] C.-Y. Lin, M. Wu, M. L. Miller, I. J. Cox, J. Bloom and Y. M. Lui, "Geometric Distortion Resilient Public Watermarking for Images," submitted to *SPIE Security and Watermarking of Multimedia Content II*, San Jose, CA, Jan 2000.

[6] S. Pereira, J. O'Ruanaidh, F. Deguillaume, G. Csurka and T. Pun, "Template Based Recovery of Fourier-Based Watermarks Using Log-polar and Log-log Maps," *IEEE ICMCS 99*, Florence, Italy, June 1999.

[7] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. on Image Processing*, Dec. 1997.

[8] H. S. Stone, "Analysis of Attacks on Image Watermarks with Randomized Coefficients," *NECI Technical Report*, 1996.

Advanced Spread Spectrum Watermarking

Andrew Z. Tirkel
Scientific Technology
tirkel@c031.aone.net.au

Tom E.Hall
Department of Mathematics, Monash University
tom.hall@sci.monash.edu.au

ABSTRACT

This presentation outlines the connections between spread spectrum techniques and watermarking, from the original concept to the current state of the art.

KEYWORDS

Watermark, Spread Spectrum, Correlation, Multidimensional Arrays, Finite Fields.

1 Motivation

The concept of the spread spectrum watermark dates back to 1992, when a series of natural images were successfully watermarked in the pixel domain with a set of binary m-sequences carrying a brief ASCII message [10].

Since then, such watermarks have been embedded in a multitude of transform domains: DFT, DCT, wavelet etc. The embedding process has become more sophisticated, making use of the properties of the Human Visual System (HVS) [8] and the recovery process has been improved by advanced processing techniques such as linear and non-linear filtering [3] etc. Robustness against unintentional image distortions and compression as well as cryptographic attack has occupied significant research efforts. In these respects, the art of steganography has matured, as witnessed by a multitude of research papers, patents and products in the many facets of this discipline [6].

During this period, the theory and applications of spread spectrum have advanced significantly as well. The main motivation for this has been communications, radar/sonar and instrumentation technologies. The construction and analysis of pseudonoise sequences is the cornerstone of this discipline. Pseudonoise arrays designed specifically for image and multimedia watermarking have been comparatively rare. This presentation is concerned with the construction and properties of such arrays. Although the discussion is directed towards two dimensional arrays, it also pertains to 3d suitable for multimedia (with time acting as the third dimension) and higher dimensional image arrays constructed from abstract transforms. Typically, information is conveyed by choice of array(s), their combination, or by rotating the array from a reference construction. In this application, such an array is embedded in an image by additive (linear) or multiplicative (non-linear) means. Recovery is based on comparison with a template. Two main measures are known: LMS (least mean square error) and Maximum Likelihood. Applications of LMS in this context are in their infancy, whilst the latter technique has been implemented through the use of correlation.

1.1 Correlation

Currently, all spread spectrum watermarking techniques employ correlation as a measure of similarity between the embedded watermark and a template array. In this context, correlation is defined as the inner product of two matrices. This deterministic quantity is distinct from its statistical namesake. Correlation can be computed in a periodic or aperiodic manner. The former is accomplished by replicating one of the matrices so that all the terms in the inner product are always well defined. In one dimension, 3 replicas of a watermark are sufficient, whilst in 2 dimensions, 9 replicas are required. Symmetry may reduce these requirements. The above effect can also be achieved without replication, by invoking modulo arithmetic in indexing pixels. The effectiveness of correlative recovery is determined by a low probability of missed or false detection. This depends on: autocorrelation performance of the watermark array, crosscorrelation between different watermark arrays and the crosscorrelation between the watermark and the image.

1.2 Autocorrelation

This can be described by a Merit Factor (*MF*):

$$MF = \frac{(\sum_i a_i a_i^*)^2}{\sum_{\tau \neq 0} \left| \sum_i a_i \, a_{i+\tau}^* \right|^2}$$

For aperiodic correlation, the Barker sequence of length *13* has a *MF* of about *14*, whilst in *2* dimensions, a *13%13* Barker Array has an MF exceeding *6*. Currently, these are the best performers for aperiodic correlation. By contrast, if periodic correlation is considered, numerous arrays of various sizes exhibit *MF* values of hundreds or thousands. This illustrates an advantage of using periodic correlation.

1.3 Cross Correlation Between Arrays

This is important where more than one watermark array is to be deployed. Different arrays can signify different origins of the source material or simply increase the information carrying capacity of the watermark. Multiple arrays can be superimposed on one image, to further increase capacity and robustness against attack.

1.4 Cross Correlation Between Array and Image

This causes an undesirable background correlation capable of producing false or missed detection. It can be interpreted as a complex statistical problem which plagues every correlative recovery scheme. It is related to the inherent symmetries (higher order moments) of the image distribution. The choice of transform domain may have significant effects on the magnitude of this crosscorrelation. Preconditioning the image by histogram equalization, or using deliberate quantization can reduce this effect. Alternatively, if the unwatermarked original image is available, this crosscorrelation can be subtracted out. If the original is not available, an estimate of it can be obtained by applying a non-linear filter, such as the median filter to the watermarked image [3]. This is about 80% effective at removing the background correlation. A similar improvement can be achieved by using spatial high pass filtering (Laplace) on the correlation result. This works because the autocorrelation response is an impulse, whilst the crosscorrelation with the image contains predominantly low frequencies. Matched filtering does not appear to have been tried in this context.

1.5 Dimensionality

Traditionally, two dimensional arrays have been constructed from one dimensional sequences by the techniques described below. In this analysis, it is assumed that the parent pseudonoise sequence has off-peak autocorrelation values of -1 (otherwise, a simple conversion applies).

1.6 Folding

This requires the sequence length l to have factors with gcd=1. M-sequences and some generalized chirp sequences have this property. The sizes and aspect ratios of the resulting arrays are severely limited by this constraint. The Merit Factor of such arrays is the same as that of the parent sequence and can therefore be arbitrarily high. However, the crosscorrelation between arrays is also the same as that between the parent sequences and therefore subject to the Sidelnikov bound of $l^{1/2}$.

1.7 Product

This construction requires two pseudonoise sequences: one employed for column construction and one for row generation. Various sizes of arrays generated by these means have been analyzed by [1]. Two types of product can be involved.

(a) Periodic

Lüke et al [4] showed that two given sequences with good autocorrelation can be multiplied to form a two-dimensional array, whose autocorrelation is a product of the individual autocorrelations. They also gave generalizations to n-dimensional arrays. When this product is applied to a sequence (array) with autocorrelation values (a,b,c) by a sequence (array) with a perfect periodic autocorrelation with values (d,0) the product array has the autocorrelation values (ad, bd, cd, 0). This maintains the normalized off-peak correlation values, apart from introducing an extra set of 0 entries. When c=0, the autocorrelation remains three-valued. In this manner, new sequences and arrays with useful autocorrelation values can be constructed from known ones.

(b) Kroneker

An alternative method of forming the product of two sequences is described in [4]. This involves a sequence $\boldsymbol{s}$ and a two-dimensional array $\boldsymbol{A}$ as starting points. A product of $\boldsymbol{s}$ and $\boldsymbol{A}$ formed in the manner of Kroneker matrix multiplication results in a new two-dimensional array, whose autocorrelation is a product of those of its constituents. The array $\boldsymbol{A}$ is assumed to have perfect periodic autocorrelation in one dimension and perfect ***aperiodic*** autocorrelation in the other dimension. Only one (binary) array, is known to possess this property:

$$\boldsymbol{A} = \begin{pmatrix} \mathbf{01} \\ \mathbf{11} \end{pmatrix}$$

1.8 Rotation

This method uses cyclic shifts (rotations) of a pseudonoise sequence as columns of an array. This is similar to the method of constructing pseudonoise sequences by appending rotations of the complex roots of unity. There are at least three methods of arranging such shifts to produce arrays with good autocorrelation:

(a) Quadratic Shifts (Chirp-Like)

This construction [2], produces arrays of size $p \% p$ (p prime) with three valued autocorrelation. Where p is of the form $(4k+3)$ or (2^n-1) these values are: p^2 for 0 shift, $-p$ for purely vertical shifts and $+1$ elsewhere. For large p, the Merit Factor approaches p. There are $p-1$ such arrays, all with three valued cross-correlation: $p+2, +1, -p$. The $+1$'s occur once per column (i.e. p times), when a discriminant is 0, the $p+2$'s occur for $p(p-1)/2$ shifts for which the discriminant is not a square (modulo p), whilst the $-p$'s occur for the remaining $p(p-1)/2$ shifts, for which the discriminant is a square.

For primes not of the form above, the autocorrelation values are: $p(p-1)$ for 0 shift, $-p$ for purely vertical shifts and 0 elsewhere. The cross-correlation values are: $p, 0, -p$.

The same construction can be employed to produce some arrays of size $n \% p$, with $n > p$, provided that there exist pseudonoise sequences of length n. The auto and cross-correlation properties of such arrays is similar to the $p \% p$ case, although it degrades with increasing n and is more complex to analyze. Where $gcd(n,p)=1$, $n \% p$ arrays can be unfolded to produce sequences with the same auto and cross-correlation.

(b) Primitive Root Shifts

This construction (adapted by the authors from [7]) produces arrays of size $p \times (p-1)$ with three valued autocorrelation. Where p is of the form $(4k+3)$ or (2^n-1) these values are: $p(p-1)$ for 0 shift, $-p+1$ for purely vertical and purely horizontal shifts and $+2$ elsewhere. For large p, the Merit Factor approaches $p/2$. The number of such arrays is the number of primitive roots of p, but their cross-correlation is unknown, although its upper bound is not as tight as that for arrays constructed by method (a).

(c) Galois Field Shifts (Logarithmic)

This construction (adapted by the authors from [7]) produces arrays of size $(p^m-1) \times (p^m-1)$. The autocorrelation is four valued: $(p^m-2) \times (p^m-1)$ for 0 shift, $-p^m+2$ for purely vertical and purely horizontal shifts, $-p^m+3$ for shifts along a leading diagonal and $+3$ elsewhere. There are $\phi(p^m-1)$ such arrays, where ϕ is the Euler Totient function. The cross-correlation between them is unknown. Although this construction is inferior in performance, it does provide arrays with useful sizes. For example, for $p=3$ and $m=2$, an 8×8 array results; Fig 1(a). This array (and its relatives) are the only ones known to be commensurate with JPEG compression blocks! Black denotes the absence of a watermark, whilst red, blue and green denote the three rotation angles 0°, 120° and 240° respectively.

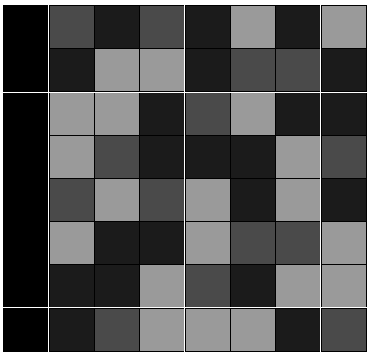


Figure 1(a)

The autocorrelation of the above array is shown below.

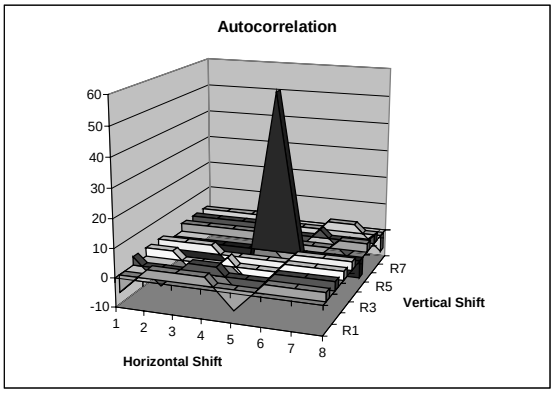


Figure 1(b)

1.9 Computer Search

This is the only method which does not rely on one-dimensional sequences. However, the search space becomes prohibitively large for any practical arrays. Also, it is unknown whether multiple arrays with suitable autocorrelation exist for any particular size, and if so, if any constraints can be placed on their crosscorrelation.

1.10 Alphabet

The sequences and arrays described above are defined over a finite (closed) alphabet, which permits unambiguous computation of correlation. The most common alphabet satisfying these requirements is the set of roots of unity. The advantages of such an alphabet for watermarking of images are:

(a) equal weighting of all entries

(b) finite arithmetic: multiplicative rings, Galois Field

The disadvantage for watermarking of images is that only the binary version {-1,1} is simple to use. Otherwise, complex roots of unity require the construction of an abstract image space, where circular rotations can be defined. Colour space is a natural candidate, although greyscale equivalents can also be generated, by considering the image as a multichannel entity. In the case of audio, a complex alphabet is easy to implement by a sinusoidal subcarrier, whose phases are quantized to the roots of unity required to carry the watermark [12].

Where these advantages are not required, sequences and arrays can be constructed over an infinite alphabet of real numbers defined over an interval. A method of constructing such sequences and arrays has recently been developed by Dr. Jiri Fridrich and his group at SUNY. Such watermarks can produce excellent autocorrelation Merit Factors, but it is difficult to constrain their cross-correlation.

A hybrid technique may be possible, in a similar manner to Quadrature Amplitude Modulation in communications. It may also be possible to employ higher dimensional constructions such as quaternions, octions etc.

All image and multimedia data is quantized. The alphabet chosen must be commensurate with the quantization levels (grid). It is possible to turn this restriction into an advantage by reserving certain quantized states (constellation) for the watermark only [11]. This can reduce the cross-correlation with the image, even when the image is corrupted.

1.11 Embedding Method

1. Information content in watermark

This has been implemented in the cyclic rotations of the watermark array, the choice of array, or combination of arrays.

2. Watermark in Image

(a) Additive

This (linear) technique is obvious. Modifications of it have included adaptation to the Human Visual System, where weighting of the added watermark is dependent on masking in the spatial or frequency domain.

(b) Multiplicative

The first example of a multiplicative watermark [11] used addition of angles in colour space and in an abstract two-channel space for greyscale images. In this context, the multiplication involved is like that of complex numbers on the unit circle. Multiplication over purely real numbers has not yet been implemented as an embedding method.

A multiplicative watermark in an abstract space can offer security against cryptographic attack. For example, if an attacker obtains a watermarked image as well as the original (unwatermarked) version he cannot determine the watermarking method easily, by subtraction or log conversion followed by subtraction. An intimate knowledge of the embedding space and method is required for cracking such a watermark.

1.12 Image/Watermark Registration (Synchronization)

Many techniques of maintaining synchronization in the presence of image distortion or deliberate attack have been described in the literature. Most relate to the embedding technique. As an alternative, it is possible to employ separate watermarks, whose only purpose is to carry synchronization information [9]. Perfect maps and log spiral patterns appear to be good candidates for this task.

1.13 Watermark Security (Robustness)

(a) Unintentional distortions

Significant advances in the understanding of and countermeasures against distortions have been achieved in the past few years. Such distortions include cropping, rotation, skew, noise, lossy/lossless compression etc. Typically, these countermeasures have been designed for a specific image or multimedia format: JPEG, MPEG etc. [8]. Some watermarks have been integrated with the compression process! Typically, multimedia watermarks are separate from such operations at the physical network layer, such as the communication channel, or encryption, which is usually reserved for higher layers of the OSI model. Major advances have also occurred in the watermarking of audio for the CD/DAT format. Comparatively less effort has been directed at watermarking non-multimedia audio data, such as telephone audio. This may be because of the difficulty in devising watermarks which can survive the increasingly savage compression techniques such as ADPCM or CELP. It is likely that spread spectrum techniques are useless in this application.

(b) Cryptographic attack

Owing to the unpredictable nature of cryptographic attack, countermeasures against it need to be more sophisticated. Such countermeasures can be incorporated in the embedding technique (e.g. non-linear embedding) or in the watermark itself. Some arrays are inherently more secure than others. An intuitive measure of this can be attributed to the minimum fraction of the array which must be known accurately before the remainder is predictable with certainty. This is similar to unicity distance in cryptography and is formally known as array complexity. The complexity of sequences has been analyzed [5] but the translation of this knowledge to arrays requires multidimensional thinking.

1.14 Non-Multimedia Watermarks

Spread spectrum techniques were first applied to communications and radar/sonar. Other disciplines to benefit from this were auditory testing, the electroretinogram, ultrasonic imaging and concert hall acoustics [7]. Sequences and arrays have been developed specifically for these applications. Another area where low autocorrelaton sequences and arrays are a central theme is that of magnetic order in solid state materials.

2 Conclusions

The spread spectrum watermark has evolved considerably since its inception 7 years ago. Whilst numerous application issues have been addressed, there exist many unsolved problems, particularly concerning the design and analysis of arrays suitable for watermarking.

3 Acknowledgements

The authors express their gratitude to their colleagues: Associate Professor Charles Osborne, Ron van Schyndel and Dr. Imants Svalbe for their inspiration, ingenious ideas and tireless work throughout this ongoing research project.

4 References

[1] Calabro D., Wolf J.K. "On the Synthesis of Two-Dimensional Arrays with Desirable Correlation Properties". Information and Control, 11, p.537-560. 1968.

[2] Hall T.E., Osborne C.F., Tirkel A.Z. "Some Binary Arrays and their Auto and Cross-Correlation". Submitted.

[3] Langelaar G.C., Lagendijk R.L., Biemond J. "Removing Spatial Spread Spectrum Watermark" Eusipco'98, Rhodes Greece, 8-11 September 1998, p.2281-2284.

[4] Lüke H-D., Bömer L., M.Antweiler M.F.M. "Perfect Binary Arrays." Signal Processing 17 (1989) Elsevier Science Publishing, p.69-80.

[5] Massey J.L., Serconek S. "Linear Complexity of Periodic Sequences: A General Theory" CRYPTO 1996: p.358-371.

[6] Petticolas F Information hiding & digital watermarking: an annotated bibliography http://www.cl.cam.ac.uk/~fapp2/steganography/bibliography/

[7] Schroeder M.R."Number Theory in Science and Communication" 3rd Edition. Springer 1997. ISSN 0720-678X.

[8] Swanson D.M, Kobayashi M., Tewfik A.H. "Multimedia data Embedding and Watermarking Tech-

nologies" Proc. IEEE vol 86, No. 6, p.1064-87. June 1998.

[9] Tirkel A.Z., Osborne C.F., Hall T.E."Image and Watermark Registration", Signal Processing Journal vol 66, No 3, p.373-383, May 1998.

[10] Tirkel A.Z., Rankin G.A., van Schyndel R.M., Ho W.J., Mee N.R.A., Osborne C.F. "Electronic Water Mark". DICTA-93 Macquarie University, Sydney, December 1993. p.666-672.

[11] van Schyndel R., Tirkel, A.Z., Svalbe, I.D. "A Multiplicative Color Watermark", *IEEE-EURASIP Workshop on Non-Linear Signal and Imaging Processing*, Antalya, Turkey, 1999.

[12] van Schyndel R., Tirkel, A.Z., Svalbe, I.D. "Delay Recovery from a Non-Linear Polynomial Response System". IEEE International Workshop on Intelligent Signal Processing and Communication Systems, Melbourne, November 1998, Vol. 1, pp.294-298.

Improved Digital Watermarking Through Diversity and Attack Characterization

Deepa Kundur

Department of Electrical & Computer Engineering
University of Toronto
10 King's College Road
Toronto, Ontario
Canada M5S 3G4
Phone: (416) 946-5181
Fax: (416) 971-3020

deepa@comm.toronto.edu

ABSTRACT

In this work, we propose and evaluate the use of novel communication theory tool-sets to improve the performance of digital watermarking algorithms. The emphasis is on the application of basic channel estimation and communication diversity principles to the problem of robust data hiding in multimedia signals. An analytic framework is presented from which we derive general insights into strategies for effective watermarking.

KEYWORDS

Digital watermarking, content-based security, attack characterization, multimedia security, copy protection.

1 Introduction

Much of the previous work in digital watermarking has addressed the problem in a practical manner by presenting novel algorithms for a variety of applications. Improved performance was often gained through the use of tool-sets employed in communication theory.

In this work, we propose an analytic framework applicable to most general multimedia signals to understand effective embedding and extraction strategies for robust watermarking. This work is a preliminary investigation and is not intended, in its present form, to represent a complete methodology.

A great deal of the work on robust digital watermarking is based on spread spectrum (SS) principles [1-5]. In SS watermarking the embedded signal is generally a low energy pseudo-randomly generated white noise sequence. It is *detected* by correlating the known watermark sequence with either the extracted watermark or a transformed version of the watermarked signal itself (if the original *host* signal is not available for extraction). If the correlation factor is above a given threshold then the watermark is detected. The anti-jamming properties of SS signaling makes it attractive for application in watermarking since a low energy and hence imperceptible watermark, robust to narrow band interference, can be embedded [4].

However, SS approaches have the following limitations:

- SS allows detection of a known watermark, but the fundamentally large bandwidth requirement does not facilitate the *extraction* of a long bit sequence or logo from an audio signal or an image.
- SS approaches are specifically vulnerable to the "near-far" problem [6]. For watermarking this implies that if the energy of the watermark is reduced due to fading-like distortions on the watermark, any residual correlation between the host signal and watermark can result in unreliable detection.
- Most SS approaches are not adaptive. That is, they neither take into account spatial non-stationarity of the host signal and attack interference nor readily incorporate adaptive techniques to estimate the statistical variations.
- The correlator receiver structures used for watermark detection are not effective in the presence of fading. Although SS systems in general try to exploit spreading to average the fading, the techniques are not designed to maximize performance.

We also consider a communication paradigm to watermarking; communicating the watermark is analogous to transmission of the signal through an associated *watermark channel*. However, we hypothesize that common multimedia signal distortions including cropping, filtering, and perceptual coding are not accurately modeled as narrow band interference which is a common assumption in SS approaches. Instead, we believe that such signal modifications have the effect of fading on the embedded watermark. As a result, the watermark can be made more robust by employing effective diversity techniques and channel estimation. Previous work has demonstrated through practical implementation and simulations the improved performance of taking such an approach [7].

This paper provides analysis to demonstrate the advantages of incorporating these new principles for digital watermarking, and outlines approaches to improve algorithm performance to specific watermark attacks. It should be emphasized that the ideas presented in this work are meant to be employed within existing watermarking techniques, and are not intended to replace well-established watermarking strategies such as SS watermarking and modulation.

2 Context and Scope

We incorporate diversity and channel estimation into our analysis framework through the use of watermark repetition and attack characterization. In particular, we assume that the watermark is embedded many times throughout the host signal. Each repetition is assumed to be separately extracted. Attack characterization is the process of measuring the reliability of each extracted watermark repetition. We do not specify how the characterization is performed as this is an implementation issue, but assume a reliability factor is available. To perform analysis we limit the scope of our framework to the broad class of watermarking systems with the following basic characteristics:

1. The watermark w is binary and of length N_w bits.
2. The watermark information is repeatedly embedded $M \geq 1$ times within the host signal.
3. The embedding process occurs in the *watermark domain*. Specifically, an invertible transformation T_w is applied to the host signal to produce coefficients in which the watermark bits are repeatedly inserted.
4. Each repetition of the watermark is embedded in a localized region of coefficients in the watermark domain, so that most traditional distortions on multimedia signals such as perceptual coding and filtering will have a similar degree of distortion on all embedded watermark bits of a given repetition.
5. Each embedded watermark repetition is extracted separately.
6. The watermarked signal may undergo distortions that affect the integrity of the embedded watermark information. We assume that there exists a method of attack characterization such that each extracted watermark repetition has a known associated reliability. In our analysis, we make use of the probability of bit error measure.

Many proposed watermarking algorithms [2,8-13] (this is by no means an exhaustive list) are encompassed by this class of techniques or can be easily modified to fit this category.

The specific details of the data embedding and extraction processes are not relevant to our work. Although we restrict the watermark to be a bit sequence and the reliability measure to be the bit error rate, we believe the spirit of the results discussed in the paper holds for non-binary watermarks with a different reliability measure such as the signal-to-noise ratio.

3 Modeling and Estimation

3.1 Parallel BSC Model

Given the characterization presented in the previous section, we can extract the individual watermark repetitions to produce M estimates of the watermark, $\hat{w}_1$, $\hat{w}_2$, ..., $\hat{w}_M$. As discussed in the previous section each estimate $\hat{w}_k$ has an associated probability of bit error p_{Ek}. We assume that some sort of characterization is performed so that a good estimate of p_{Ek} is available. Such a technique is presented, implemented and tested in [7].

Our framework is analogous to transmitting the watermark simultaneously along M independent binary symmetric channels (BSC) as shown in Figure 1. The error probabilities $0 \leq p_{Ek} \leq 0.5$ are assumed to be known and independent of one another. If $p_{Ek} > 0.5$, then the output is complemented and $1 - p_{Ek}$ is used as the probability of error parameter value.

This type of localized characterization of the distortion in the watermark domain allows better modeling of non-stationary fading-like distortions. The new perspective provides insights on effective strategies for watermarking. Most of the theoretical work in the area so far has considered the attacks on the watermark to be stationary [2]. This basic assumption precludes

the benefits that diversity can provide, and limits understanding into the advantages of using one watermarking domain over another.

3.2 Linear Watermark Estimation

To estimate the embedded watermark w, we choose to linearly weight and add the extracted repetitions so that the overall estimate of the ith watermark bit is given by

$$\hat{w}(i) = \text{round}\left[\sum_{k=1}^{M} \alpha_k \hat{w}_k(i)\right],$$

for $i = 1, 2, \ldots, N_w$, where round[·] is the integer round operator. It is shown in [14] that

$$\alpha_k = \frac{\log\left(\frac{1-p_{Ek}}{p_{Ek}}\right)}{\sum_{j=1}^{M} \log\left(\frac{1-p_{Ej}}{p_{Ej}}\right)},$$

minimizes the bit error rate of the overall extracted watermark estimate $\hat{w}(i)$.

This linear estimation procedure is by no means the only alternative for combining the various extracted repetitions, but it is computationally simple, and it has been successfully implemented and tested in [7]. The following section summarizes theoretical observations concerning the analysis of such watermark recovery.

4 Error Statistic Bound

We provide a sketch of the analytic work initially presented in [7,14-16] and attempt to present a more intuitive perspective of the theory in the subsequent section.

Consider the bit $e_k(i)$ defined as

$$e_k(i) \triangleq w(i) \oplus \hat{w}_k(i) = \begin{cases} 1 & \text{if bit error in } \hat{w}_k(i) \\ 0 & \text{otherwise} \end{cases},$$

where $\oplus$ is the exclusive-OR operator. Similarly, we let

$$e(i) \triangleq w(i) \oplus \hat{w}(i) = \begin{cases} 1 & \text{if bit error in } \hat{w}(i) \\ 0 & \text{otherwise} \end{cases}.$$

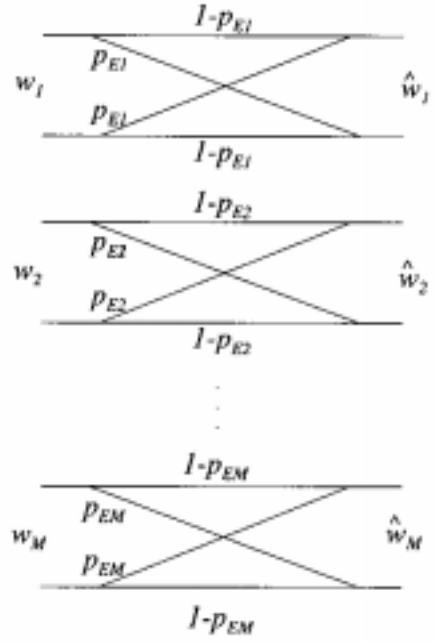


Figure 1: Parallel BSC model of the watermarking channel. Each repetition of the watermark is considered to undergo transmission through an independent BSC.

It follows that [14]

$$e(i) = \text{round}\left[\sum_{k=1}^{M} \alpha_k e_k(i)\right],$$

which relates the bit errors of the individual extracted repetitions to the bit error of the overall watermark estimate. For perfect watermark extraction, we would like $e(i) = 0$ for $i = 1, 2, \ldots, N_w$. The bit error $e(i)$ is a statistical quantity dependent on the probabilities $p_{E1}, p_{E2}, \ldots, p_{EM}$. This follows from the BSC model of the watermark channel in which $\hat{w}_k(i)$ (or equivalently $e_k(i) = w(i) \oplus \hat{w}_k(i) = 1$) occurs with probability p_{Ek}. The greater the degree of distortion on the region of the watermark domain in which the kth watermark repetition is embedded, the larger the value of p_{Ek} where $0 \le p_{Ek} \le 0.5$.

Analysis of the characteristics of $e(i)$ is not straightforward due to the presence of the integer round operator. Alternatively, we can consider the argument of this operator which is given by

$$E' \triangleq \sum_{k=1}^{M} \alpha_k e_k(i).$$

A bit error occurs in $\hat{w}(i)$ if $E' > 0.5$. We can analyze the mean value of E', denoted $\mathcal{E}\{E'\}$, to assess the reliability of the watermark channel. Although this is not a precise measure of the error rate of the system since a smaller $\mathcal{E}\{E'\}$ does not necessarily guarantee a lower overall bit error rate, it does provide some useful insight into the watermarking problem.

The following *error statistic bound* is established in [16],

$$\mathcal{E}\{E'\} \le \frac{\bar{p}_E}{1-\bar{p}_E}\left[1 - \frac{D(q_a\|q_b)}{\log\left(\frac{1-\bar{p}_E}{\bar{p}_E}\right)}\right]$$

where

$$\bar{p}_E = \frac{1}{M}\sum_{k=1}^{M} p_{Ek},$$

is the average bit error rate, and the quantity $D(q_a\|q_b)$ is the relative entropy given by [17]

$$D(q_a\|q_b) = \sum_{k=1}^{M} q_a(k)\log\left(\frac{q_a(k)}{q_b(k)}\right),$$

where the arguments are $q_a(k) = p_{Ek}/(M\bar{p}_E)$, and $q_b(k) = (1-p_{Ek})/(M(1-\bar{p}_E))$.

We can see that q_a and q_b are probability-like distributions since their elements are nonnegative and sum to one. It is discussed in [14,16] the error bound is tight for small $\bar{p}_E$ and p_{Ek} close to a constant. The equality of the error bound holds if and only if $p_{Ek}=0$ for all k.

A smaller value for the bound on $\mathcal{E}\{E'\}$ implies that, for the most part, we can guarantee better accuracy of the extracted watermark, and, hence, greater robustness. In the next section, we intuitively discuss ways of diminishing the bound on the error statistic, which provides practical strategies for more effective watermarking.

6 Implications and Design Insights

From our analysis we find that the following possible tactics may be incorporated into a watermarking scheme to lower the value of the error statistic bound on $\mathcal{E}\{E'\}$ and, hence, improve the robustness of the watermarking system in some way:

1. Reduce the value of the average bit error rate. Reducing the value of $\bar{p}_E$ decreases the term $(\bar{p}_E/(1-\bar{p}_E))$ and increases the denominator term $\log((1-p_E)/\bar{p}_E)$ which both serve to lower the overall bound.

Many proposed watermarking methods attempt to gain performance by diminishing this average bit error rate. Signal processing strategies to imperceptibly embed a higher energy and, hence, on average more robust watermark are commonly employed. The deficiency of most watermarking methods is that they solely rely on embedding a stronger watermark using sophisticated human perceptual mathematical models for improved performance. Our next two theoretical observations shed light on a different strategy to increase robustness.

2. Embed the watermark such that the distributions q_a and q_b are dissimilar for a large class of distortions. For a fixed value of $\bar{p}_E$, we may reduce the performance bound by increasing the value of $D(q_a\|q_b)$. The relative entropy is a measure of the distance between its two argument distributions [17]. Roughly, we can see that $D(q_a\|q_b)$ is relatively large when $q_a(k) = p_{Ek}/(M\bar{p}_E)$ and its corresponding $q_b(k) = (1-p_{Ek})/(M(1-\bar{p}_E))$ are *dissimilar*.

Assuming a fixed average probability of bit error, this requires that p_{Ek} vary in amplitude for different values of k, implying that we should embed the watermark in a domain for which the degree of distortion varies in each localized region containing a repetition of the watermark. As a result, the amplitude of p_{Ek} will be different for distinct values of k. This can be achieved by inserting the watermark in a domain which distributes the distortion more to certain coefficients, leaving the others less affected.

3. Localize the distortions on the watermarked signal. It is shown in [15,16] that the existence of $p_{Ek} = 0$ for at least one $k \in \{1, 2, ..., M\}$ implies that $E' = 0$. Thus, if there exists a set of localized coefficients containing one complete repetition of the watermark which are unmodified by the distortion, then perfect watermark recovery is possible, as long as all the values of p_{Ek} are known. This translates to embedding the watermark in a domain which

completely localizes the distortion to a finite and relatively small percent of the coefficients.

Both **2.** and **3.** relate the accuracy of the extracted watermark to the watermark domain in which the hidden data is embedded. By using diversity and attack characterization, it is possible to improve the effectiveness of the watermark to a specific class of distortions by inserting the mark in signal coefficients which localize these distortions. For example, to design a watermark robust against cropping, it would be wise to embed the mark in the spatial domain, which completely localizes the manipulation. Although a portion of the watermark is clipped out, the repetitions in the remaining signal are still accessible. Similarly, for robustness against filtering, the watermark should be embedded in the discrete Fourier domain which localizes the associated degradations. Mild linear filtering will affect some Fourier coefficients more than others. To make the watermark robust to both, a compromise would be to use the discrete wavelet domain for hiding the data.

More specific work on incorporation of particular wavelets to be robust against perceptual coding is presented in [16]. It is demonstrated that use of different domains for watermarking and perceptual coding improves the robustness of the embedded watermark. This work is in direct conflict with well-established principles which suggest the same domain is superior [18].

7 Final Remarks

In this paper, we take a different perspective on the problem of digital watermarking. We hypothesize that common watermark attacks are non-stationary and model the associated watermark channel as a set of parallel BSCs. We incorporate notions of diversity and channel estimation into our framework. Preliminary analysis provides new insights into the problem of robust data hiding.

Specifically we demonstrate how it is not only necessary to gain performance improvement by maximizing watermark signal energy, but it is better to embed the mark in a domain which localizes the distortions to a relatively small fraction of the coefficients. By assuming a localized non-uniform degradation model for the watermark we gain insight into appropriate domains in which to robustly hide data.

Future work involves using more sophisticated methods of estimating the overall embedded watermark from the various extracted repetitions by implying nonlinear order statistics [19]. We predict that analysis of such a system will lead to further performance improving approaches.

8 Acknowledgments

Part of this work was conducted by the author as part of her doctoral thesis under the supervision of Professor Dimitrios Hatzinakos.

9 References

1. R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," *Proc. IEEE Int. Conference on Image Processing*, vol. 2, pp. 86-90, 1994.
2. I. J. Cox, J. Killian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," Tech. Rep. 95-10, NEC Research Institute, 1995.
3. R. B. Wolfgang and E. J. Delp, "A watermark for digital images," *Proc. IEEE Int. Conference on Image Processing*, vol. 3, pp. 219-222, 1996.
4. J. R. Smith and B. O. Comiskey, "Modulation and information hiding in images," *Proc. First Int. Workshop on Information Hiding* (R. Anderson, ed.), no. 1174 in Lecture Notes in Computer Science, pp. 207-226, May/June 1996.
5. X.-G. Xia, C. G. Boncelet, and G. R. Arce, "A multiresolution watermark for digital images," *Proc. IEEE Int. Conference on Image Processing*, vol. 1, pp. 548-551, 1997.
6. P. G. Flikkema, "Spread-spectrum techniques for wireless communications," *IEEE Signal Processing Magazine*, vol. 14, pp. 26-36, May 1997.
7. D. Kundur and D. Hatzinakos, "Improved robust watermarking through attack characterization," *Optics Express focus issue on Digital Watermarking*, vol. 3, no. 12, pp. 485-490, Dec. 7, 1998.
8. E. Koch and J. Zhao, "Towards robust and hidden image copyright labeling," *Proc. Workshop on Nonlinear Signal and Image Processing* (I. Pitas, ed.), pp. 452-455, June 1995.
9. J. Ohnishi and K. Matsui, "Embedding a seal into a picture under orthogonal wavelet transform," *Proc. Int. Conference on Multimedia Computing and Systems*, pp. 512-521, June 1996.
10. C. I. Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models," *IEEE Journal in Selected Areas in Communications*, vol. 16, pp. 525-539, May 1998.

11. G. W. Braudaway, "Protecting publicly-available images with an invisible image watermark," *Proc. IEEE Int. Conference on Image Processing*, vol. 1, pp. 524-527, 1997.
12. J. J. K. O'Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking," *Proc. IEEE Int. Conference on Image Processing*, vol. 1, pp. 536-539, 1997.
13. S. D. Servetto, C. I. Podilchuk, and K. Ramchandran, "Capacity issues in digital image watermarking," *Proc. IEEE Int. Conference on Image Processing, 1998.*
14. D. Kundur and D. Hatzinakos, "Attack characterization for effective watermarking," to appear in *Proc. Int. Conf. on Image Processing*, Kobe, Japan, October 1999.
15. D. Kundur and D. Hatzinakos, "Mismatching perceptual models for effective watermarking in the presence of compression," to appear in *Proc. SPIE -- Multimedia Systems and Applications II*, vol. 3845, Boston, Massachusetts, September 1999.
16. D. Kundur, "Mulitresolution digital watermarking: Algorithms and implications for multimedia signals," Ph.D. Thesis, University of Toronto, 1999.
17. T. Cover and J. Thomas, *Elements of Information Theory*. Toronto: John Wiley & Sons, Inc., 1991.
18. R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "The effect of matching watermark and compression transforms in compressed color images," *Proc. IEEE Int. Conference on Image Processing*, vol. 1, 1998.
19. G. R. Arce, Personal Communication, August 9, 1999.

Synchronization Recovery in Image Watermarking

Masoud Alghoniemy and Ahmed H. Tewfik

Electrical Engineering Department
University of Minnesota
Minneapolis, MN 55455

{masoud,tewfik}@ece.umn.edu

ABSTRACT

In this paper we report a novel method to estimate the scaling factor of a previously scaled watermarked image and the angle by which the image has been rotated. Scaling and rotation performed on a watermarked image, as part of the attacks the image may undergo, can very easily confuse the decoder unless it rescales and/or rotates the image back to its original size/orientation. To be able do this, the decoder needs to know by how much the image has been scaled and rotated, i.e., needs to know both the scaling factor and the rotation angle. In our approach, we compute the Edge Standard Deviation Ratio (ESDR) which gives us an accurate estimate for the scaling factor as well as we estimate the angle of rotation. The ESDR is computed from the wavelet maxima which have been calculated form the non-orthogonal dyadic wavelet transform. Angles of rotation also have been estimated from the wavelet maxima as well. Our method has proved its robustness to wide rotation and scale ranges.

KEYWORDS

Image watermarking, StirMark, attacks, wavelet transform.

1 Motivation

In watermarking applications, robustness of the inserted watermark to scaling and rotation operations is very essential. This is due to the fact that changing the image size or its orientation, even by slight amount, could reduce the receiver ability to correctly retrieve the watermark back. This can be compared to loosing synchronization in a communication system. In [1] we showed that we still can retrieve the inserted watermark even if the image was scaled by a factor γ or rotated by an angle θ. Previously, the decoder did not know the scaling factor or the rotation angle and performed exhaustive search to find γ and θ with the aid of a training sequence. In this paper, we estimate γ by computing the ESDR from the wavelet maxima calculated from the non-orthogonal dyadic wavelet transform and estimate θ by computing the angles of the wavelet maxima in a predetermined region of the image. The key idea is trying to find a suitable feature by which the receiver can estimate γ and θ. In principle, this can be done if it reflects the variations which occurred by scaling and rotating the image. We found that wavelet maxima satisfy this requirement. Wavelet maxima is computed by decomposing the image into its multiresolution levels using the non-orthogonal dyadic wavelet transform, and then spatially correlating these levels [2]. By doing so, edges which have the same spatial locations are magnified and edges which are not spatially aligned are diminished. This is a powerful way to find edges at different scales which is helpful in our analysis as explained later. One may ask, why we do not use the orthogonal wavelet transform instead of the non-orthogonal one?. The answer comes in two parts. First, the non-orthogonal dyadic wavelet transform is a shift-invariant representation which means that if the signal is shifted, say by δ,then the wavelet coefficients, and hence wavelet maxima, will also be shifted by δ. This is very important since other attacks may include shifts to the image which implies that our approach is more likely to survive shift attacks than others. Second, the spatial correlation between the multiresolution levels in case of the orthogonal wavelet transform is minimized due to the orthogonality constraint.

On the other hand, the non-orthogonal wavelet transform provides higher correlation coefficients between the various levels. This property makes it more convenient to use the non-orthogonal wavelet transform since the higher the correlation coefficient between scales is, the more robust the edge estimate is. Since wavelet maxima represent edges in the image, they should have been scaled by γ when the image is scaled by the same factor and they also should be rotated by θ if the original image has been rotated by the same angle. This can be used

This work was supported by the AFRL under grant AF/F30602-98-C-0176

to estimate γ by comparing the standard deviation of the wavelet maxima locations from their center of gravity before and after scaling. It can also be used to estimate θ by comparing the angles of the wavelet maxima locations before and after rotation. In the next section, we give a brief review for the non-orthogonal dyadic wavelet transform and wavelet maxima computations. In section 4 we give a description to edge standard deviation calculations and angle estimation as well as the results we obtained.

2 Wavelet Maxima

In this section we will review the non-orthogonal wavelet transform and the wavelet maxima implementation. The main difference between the orthogonal and the non-orthogonal discrete wavelet transform is that in the latter the time variable is not sampled to make the wavelet coefficients shift invariant. Let us consider the 1-D case for illustration as it can be generalized for the 2-D case using separable bases. Let $h[n]$ be the discrete time filter corresponding to the scaling function, $g[n]$ corresponding to the wavelet function and the discrete time signal at resolution $j = 0$ is $a_0[n]$. The dyadic wavelet transform of $a_0[n]$ can be represented as a convolutional process with a cascade filter bank. The coarse approximation in the
decomposition stage at resolution $j \geq 0$ is

$$a_{j+1}[n] = a_j * \overline{h_j[n]} \tag{1}$$

and the details

$$d_{j+1}[n] = a_j * \overline{g_j[n]} \tag{2}$$

where $*$ is a convolution process. $h_j[n]$ is the filter obtained by inserting $2^j - 1$ zeros between each sample of $h[n]$, and $\overline{h_j[n]} = h_j[-n]$.

It is clear that there is no down sampling after each decomposition stage which clarify the redundancy introduced by the non-orthogonal wavelet transform.

Let $|Wf(s,x)|$ be the modulus wavelet transform of a function $f(x)$. Then the modulus maximum at any point (s_0, x_0) is such that

$$|Wf(s_0, x)| \leq |Wf(s_0, x_0)| \tag{3}$$

where s is the scale parameter and x is the spatial/delay parameter [2]. For each scale, points satisfy (3) are connected together defining the wavelet maxima contour for this scale. As mentioned earlier, the 2-D wavelet transform can be computed for images using separable bases, i.e., performing the previous algorithm for rows and then for columns. Fig. 1. shows the first four levels of the modulus maxima of the F16 image which were computed with [3].

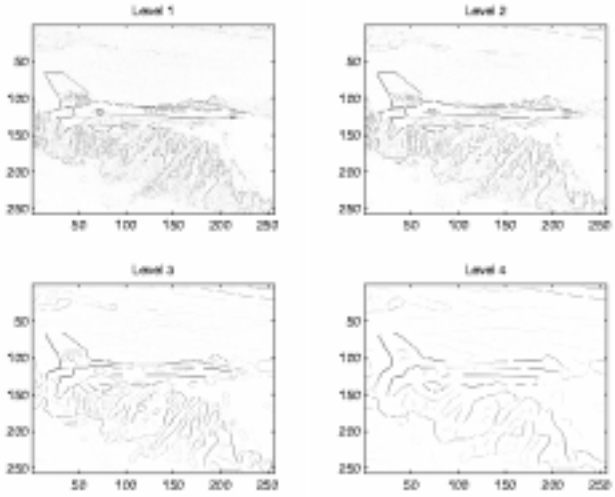


Figure 1: Modulus maxima for the first 4 levels

It should be noted that low levels preserve high frequency informations while higher levels tend to preserve coarse informations. This is well understood from the filter bank formulations (1), (2).
Since high frequency edge informations are not robust to low pass filtering which may occur to the watermarked image, we compute wavelet maxima by spatially correlating wavelet maxima corresponding to high levels, levels 3 and 4, to increase the robustness against low pass filtering.

3 Rotation and Scaling Parameters Estimation

The computed wavelet maxima in the previous section is used to estimate the scaling factor as well as the angle of rotation for the attacked image as explained in the next section.

3.1 Scaling Factor Estimation

The scaling factor is estimated by comparing the deviation of the maxima from the center of gravity (CG) of the wavelet maxima before and after the attack has been performed. The choice of CG as a reference point has its significance as explained later. Let the wavelet maxima locations, computed in the previous section, be (x_i, y_i) then edge standard deviation can be formulated as

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - x_0)^2 + (y_i - y_0)^2} \tag{4}$$

where (x_0, y_0) is the coordinate of the center of gravity of wavelet maxima

$$x_0 = \frac{1}{N}\sum_i x_i\,, \qquad y_0 = \frac{1}{N}\sum_i y_i \tag{5}$$

and N is the total number of wavelet maxima as shown in Fig. 2, σ is a measure of how much edges deviate

from the CG of the image. The estimated scaling factor, γ is thus

$$\gamma = \frac{\sigma_s}{\sigma_0} \tag{6}$$

where σ_s and σ_0 are the edge standard deviations of the scaled and the original image respectively. Note that, for practical implementations, wavelet maxima is first normalized such that the maximum value is unity and wavelet maxima locations are determined according to a threshold. To be more specific, (x_i, y_i) is declared a wavelet maxima location if

$$WM(x_i, y_i) \geq T \tag{7}$$

where $WM(x_i, y_i)$ is the normalized wavelet maxima strength at (x_i, y_i) and T is a predetermined threshold. Typically $0.2 \leq T \leq 0.8$ to avoid considering spurious edges.

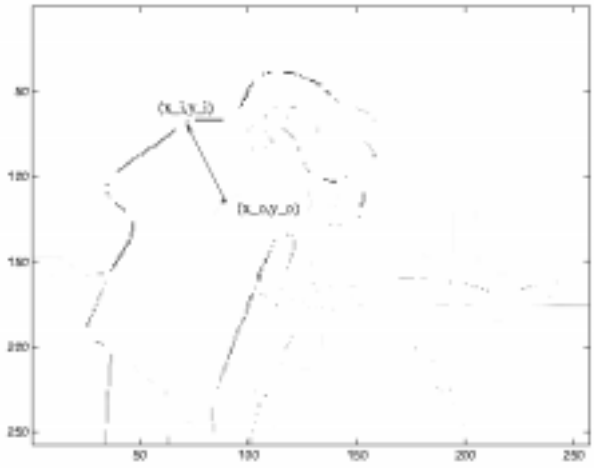


Figure 2: ESD computation for the cameraman image, (x_0, y_0) is the CG

Fig. 3 and Fig. 4 show the estimated scale factor for different scales for the F16 and Cameraman images respectively.

In our experiments we used the StirMark program to scale the image [4].

It is clear that our estimate is very accurate. For a certain scale, the scaling factor estimate is the mean of all estimates for different threshold values. This estimate is used to rescale the image back to is original size and extract the watermark as explained in [1].

Robustness to Non-uniform Cropping

By non-uniform cropping we mean removing unequal parts from the image, This is illustrated in Fig. 5 which shows the patterns of the non uniform cropping that we performed on the Cameraman and Lena images of size 256×256. The numbers on pattern B indicate blocks of size 8×8 which means that we cropped from Lena

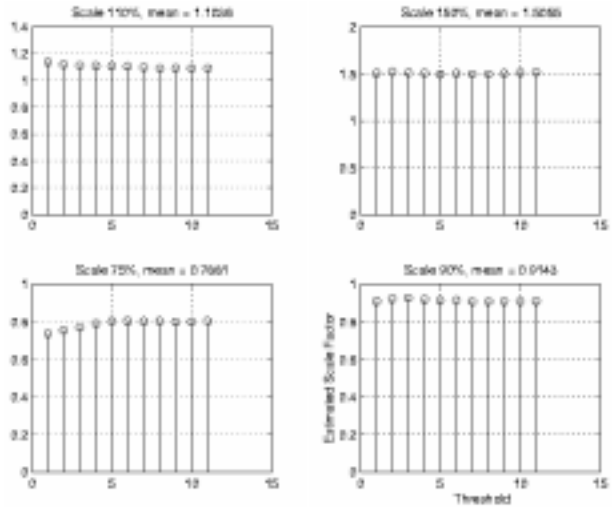


Figure 3: Estimated scaling factor for F16 image

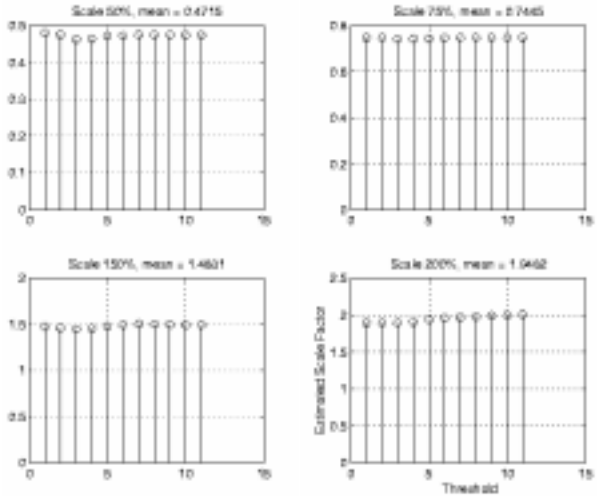


Figure 4: Estimated scaling factor for cameraman image

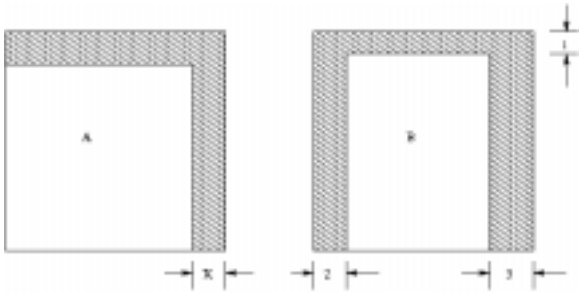


Figure 5: Cropping patterns.
A for Cameraman, B for Lena.

image from the right 24 pixels, from the top 8 pixels and from the left 16 pixels. Hence the size of the cropped image is 248×217.

The center of gravity of the original Lena image was $(x_0, y_0) = (186, 122.7)$ and after cropping was (190.6, 109.9).

Although the center of gravity has changed, the ESD is almost the same as the original image as shown in Fig. 6.

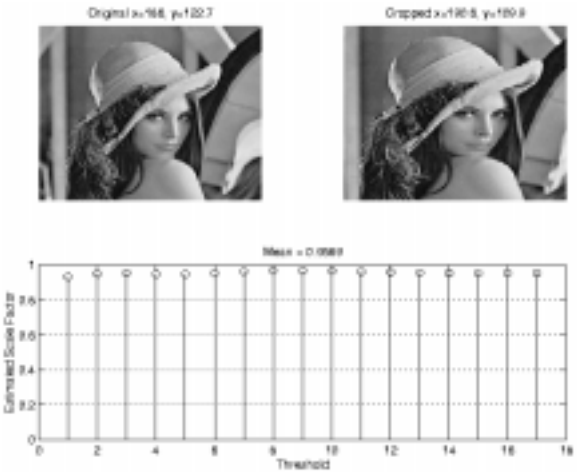


Figure 6: Lena before and after cropping

The same process was done for the Cameraman image with pattern A in Fig. 6 with X=3 and X=2 respectively. The original center of gravity was (86.9,113.7) and after cropping with X=2 the image size was 240 × 240 and the center of gravity was (85.4, 97.4). The corresponding size for the cropped cameraman image with X=3 is 232 × 232 and the resulting center of gravity was (85.6, 89.4). As before, the ESD from the center of gravity is the same as the original image as indicated in Fig. 7.

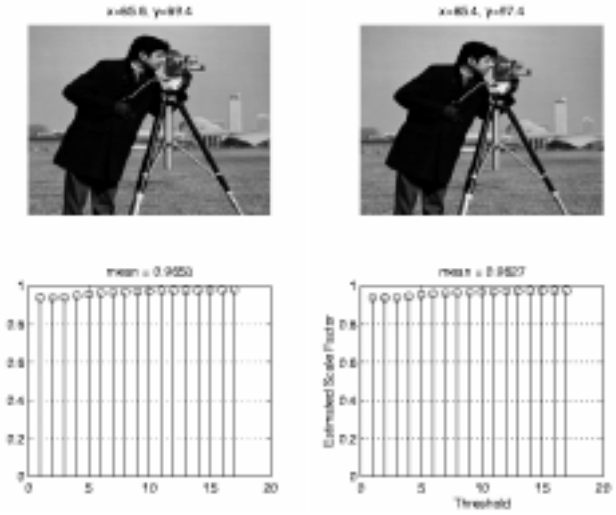


Figure 7: Cameraman after cropping, Left "X=3". Right "X=2"

3.2 Angle Estimation

To estimate the angle of rotation, we use the wavelet maxima as our tool as we did in the scaling factor estimation. As we mentioned before, wavelet maxima reflect any variation happens to the image, we use this to estimate the angle of rotation, θ. By comparing the angles of the wavelet maxima locations, say in the first quadrant, before and after rotation, the difference should be equal to the angle by which the image has been rotated, θ. It should be noted that in angle computations we use the center of the image as our reference point. Fig. 8 demonstrates angles computation for the Cameraman image.

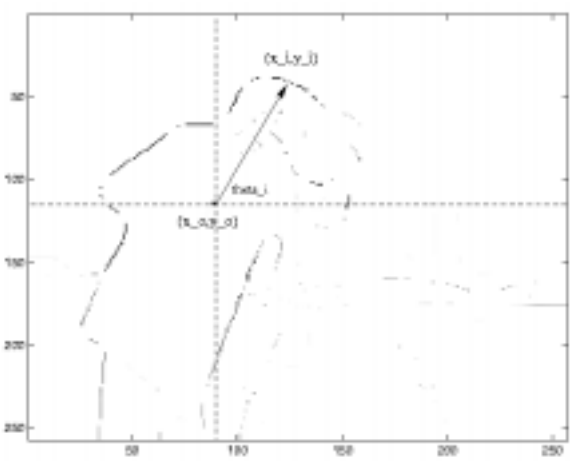


Figure 8: Angle computation for the cameraman image, (x_0, y_0) is the center of the image

The estimated angle of rotation, θ, can be set to be equal to the difference between the angles of wavelet maxima in the first quadrant region, $\Re$, before and after rotation.

$$\theta = \frac{1}{N}\left(\sum_{i}^{N}\theta_i^o - \sum_{i}^{N}\theta_i^r\right) \tag{8}$$

where N is the total number of wavelet maxima in $\Re$, θ_i^0 is the angle of the wavelet maxima at location (x_i, y_i) of the original image and θ_i^r is the corresponding angle for the rotated image. The transmitter and the receiver can agree on a predetermined value of $\sum_{i}^{N}\theta_i^0$ so that the receiver can estimate the angle of rotation. As in the scaling factor estimation case, we declare a wavelet maxima location according to (7). The final estimate is the average of all estimated angles for the different threshold values. Fig. 9 and Fig. 10 show the estimated angles of rotations for the F16 and the cameraman image respectively.

4 Conclusion

We presented a novel approach to determine by how much an image has been scaled and rotated, if any. This is very useful in watermarking and data hiding applications since by scaling and/or rotating the image, the hidden information will be out of synchronization. Once we

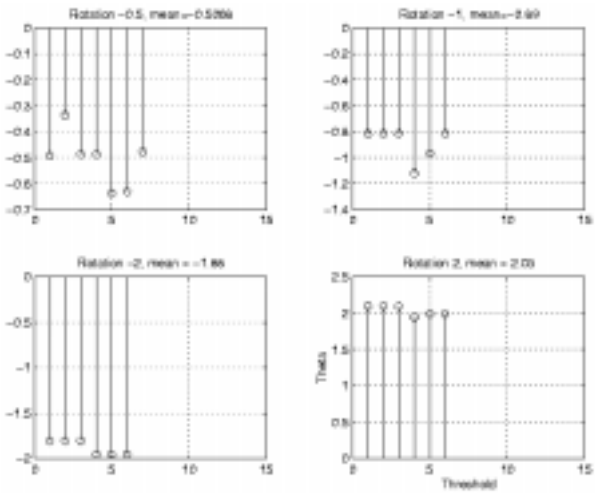


Figure 9: Estimated rotation angle for F16 image

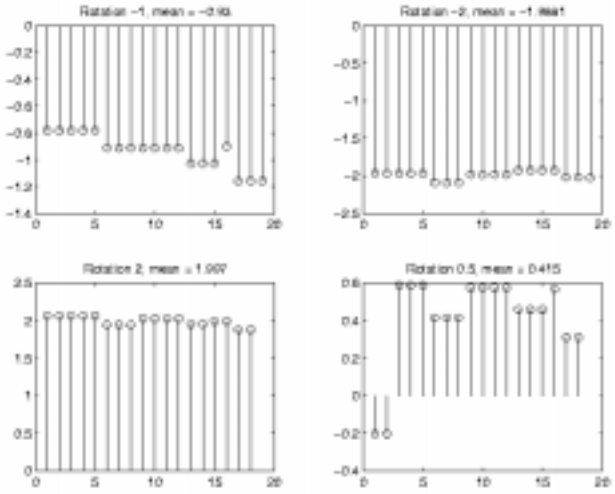


Figure 10: Estimated rotation angle for cameraman image

know the scaling factor and the angle of rotation we can rescale and rotate the image back to its original size, i.e., recover the synchronization and retrieve the watermark. The scaling factor and the rotation angle are estimated from the wavelet maxima of the image. Our approach is robust to a wide range of scales and rotations and can be implemented in real time

5 References

[1] Masoud Alghoniemy and Ahmed H. Tewfik, "Progressive Quantized Projection Watermarking Scheme''. T*o be published in the ACM'99 Multimedia Proceedings.*

[2] S. Mallat, *A wavelet Tour of Signal Processing*. Chestnut Hill, MA: Academic Press 1998.

[3] W. Hwang, S. Mallat, and S. Zhong, *XWAVE* ftp://cs.nyu.edu/pub/wave/software/.

[4] Fabien A. P. Petitcolas and Markus G. Kuhn, http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/.

Improving DFT Watermarking Robustness through Optimum Detection and Synchronisation

Alessandro Piva, Franco Bartolini,
Vito Cappellini, Alessia De Rosa, Monica Orlandi

Dip. Ingegneria Elettronica
Università di Firenze, Italy

{piva,barto,cappell,derosa}@lci.die.unifi.it

Mauro Barni

Dip. Ingegneria dell'Informazione
Università di Siena, Italy

barni@dii.unisi.it

ABSTRACT

In this paper, a new watermarking system for copyright protection of digital images is presented. The method operates in the frequency domain, by embedding a pseudo-random sequence of real numbers in a selected set of DFT coefficients of the image. Moreover, a synchronisation pattern is introduced into the watermarked image, to cope with geometrical attacks. After embedding, the watermark is hidden by exploiting the masking characteristics of the Human Visual System. An optimum criterion to verify if a given code is present is derived based on statistical decision theory, allowing a robust watermark detection without resorting to the original uncorrupted image.

KEYWORDS

Watermarking, Copyright Protection, Optimum Detection.

1 Introduction

A possible solution against copyright violation of multimedia documents consists of digital watermarking. A digital watermark is a signal permanently embedded into digital data that can be detected or extracted later to make an assertion about the data. Here, a new watermarking algorithm for digital images operating in the frequency domain is presented: the method embeds a pseudo-random sequence of real numbers in a selected set of DFT coefficients of the image. Moreover, a synchronisation pattern is embedded into the watermarked image, to cope with geometrical attacks, like resizing and rotation. After embedding, the watermark is adapted to the image by exploiting the masking characteristics of the Human Visual System, thus ensuring the watermark invisibility. An optimum decoder has been derived based on statistical decision theory, so that robust watermark detection without resorting to the original uncorrupted image is achieved.

2 Watermark casting

In the following, the main steps of the watermark casting process, shown in Figure 1, are described.

2.1 Luminance extraction

The original color image I, is decomposed in the three color bands R, G, B, which are used to extract the luminance, where the watermark will be embedded.

2.2 Expansion to 1024x1024 pixels

This step allows the system to be robust against cropping. Let us demonstrate the effects of cropping on a 1D signal. If cropping is performed on the 1D signal, its temporal duration is reduced from N to M samples, where $M < N$, so that in the frequency domain the sampling step changes from $\Delta f = 1/N$ to $\Delta f' = 1/M > \Delta f$ (normalised frequency are considered). In such a case, in detection it is not possible to recover the modified DFT coefficients, since, because of the lack of the original image, we do not know the original sampling step. To cope with this problem, the image is always extended to the same size by means of zero padding; in this way, the sampling step in the DFT domain is always the same, and resampling effects are avoided [1]. The extended size has to be chosen in such a way that:

- the number of rows and columns is a power of two, in order to use the FFT algorithm;
- all the images are extended to the same size.

According to these constraints, the luminance Y is extended to a reference size of 1024x1024 pixels, which has been considered a good trade-off between computational complexity, and the possibility to encompass a large part of the images usually available on Internet.

2.3 Pseudo-random generator

For the watermark sequence production, a pseudo random generator has been created by combining four Linear Congruential Generators (LCG), obtaining a combined generator producing a sequence of real number with period 2^{121}, and uniform distribution in the interval [0,1]. The sequence is then scaled and translated in [-1,1], in order to obtain a null mean. The pseudo-random sequence is identified by the seed given to the generator. In this system, the seed of the composed generator (CLCG), is a sequence of four integers, each of them being the seed for a single LCG:

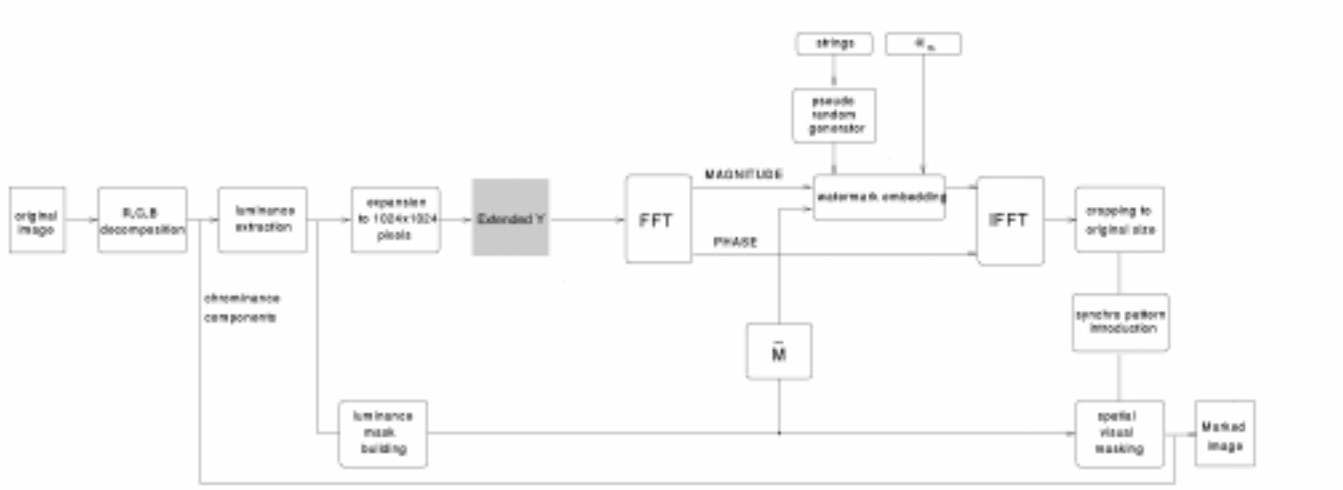


Figure 1: The watermark casting process

$S = \{s_1, s_2, s_3, s_4\}$

The seed *S* is obtained by properly coding two strings, a character string, C_L, maximum 16 characters long, and a number string, C_N maximum 8 digits long.

2.4 Watermark embedding

Watermark casting consists in the modification of a subset of DFT coefficients of the extended luminance *Y*. The FFT is applied to the luminance *Y*: two 1024x1024 matrices are obtained, representing the magnitude and the phase of the DFT coefficients. Watermark casting interests the magnitude only: since a spatial translation corresponds to a shift of the phase of the DFT, whereas the magnitude is unaltered, by marking the DFT magnitude spectrum of the image, robustness against image translations is automatically achieved.

In order to build a blind system, which do not require the original image in detection, the position and the number of coefficients to be modified is fixed a priori. In particular, 66.384 elements belonging to the medium range of the spectrum are chosen, in order to achieve a compromise between robustness and invisibility. The modification of the coefficients has to respect the well known property of symmetry of the Discrete Fourier Transform: for this reason, the symmetric coefficients are modified in the same manner.

The watermark embedding rule is the following:

$$y_i' = y_i + \alpha m_i y_i$$

where y'_i represents the watermarked DFT magnitude coefficient, y_i the corresponding original, m_i is a sample of the watermark sequence, and α is the watermark energy. The modified DFT matrix is inverted, and the image is cropped to the original size, obtaining a watermarked luminance *Y'*.

2.5 Synchronisation pattern introduction

The system described so far is robust against translation and cropping, but it is weak against resizing or rotation. In fact, these attacks modify the DFT spectrum in such a way that we are not more able to find the watermarked coefficients. To cope with such attacks, we need to find the original size and position of the image, without the use of the original image. To do so, we add pixel by pixel a reference pattern to the image. The pattern should be resistant to cropping, and it should not degrade the image to which is added. To satisfy these requirements, a periodic pattern has been chosen:

$$f(x, y) = A\,rect\left(\frac{1}{T}x\right) rect\left(\frac{1}{T}y\right)$$

The pattern is added to the watermarked luminance with an amplitude pattern *A* fixed to 2. In the frequency domain, four peaks, with position depending on *T*, are correspondingly added to the spectrum of the watermarked luminance. The *T* value has to be chosen such that the frequency pulses are far from the DC component (so that the peaks can be distinguished from the image spectrum), and are not in the high frequency region (so that the peaks are not removed if low pass filtering is applied): a good compromise has been found by fixing *T* between 4 and 8 pixels.

When the watermarked and synchronised image is processed by means of a rotation or a resizing, a spectrum analysis will allow to reveal these peaks. Depending on their new position, the estimation of the geometric attack the image has undergone will be obtained, allowing to reverse the attacks.

2.6 Spatial visual masking

The exploitation of the characteristics of the Human Visual System is a very important and delicate task for implementing effective image watermarking tools [2]. Such characteristics are implicitly exploited by our embedding algorithm: to each sinusoid which is present in the image (masking signal), another sinusoid (disturb/watermark) is added, having amplitude proportional to that of the masking signal itself. Visual masking in the frequency domain is, thus, achieved. However such an approach lacks in spatial localisation; in fact, the disturbing signal, inserted in the DFT domain, is spread over the whole image, also where the masking signal is not present: in these areas (e.g. uniform areas) masking is ineffective. The approach we propose is to apply a spatial visual masking process, based on a masking im-

age *M* giving a measure of the insensitivity to noise of the original image *I* pixel by pixel. The mask *M* is a grey level image having the same size of the original image, appearing brighter in the regions where the human eye is more sensitive to noise, and darker where the watermark can be embedded with an higher energy without a visual quality degradation. It is obtained by processing the original luminance *Y* according to many possible approaches [3].

Thus, given the original luminance *Y*, the watermarked luminance *Y'* and the mask *M*, another watermarked luminance Y_M is constructed by means of the masking process, realised in the spatial domain pixel by pixel:

$$Y_M = MY + (1-M)Y'$$

It is worth noting that where $M = 0$ (that is in the darker regions of the mask), the watermark energy is higher (equal to α) and $Y_M = Y'$; whereas where $M = 1$ (that is in the brighter regions), the watermark energy is null and $Y_M = Y$. Finally, to obtain the watermarked colour image, the watermarked luminance Y_M is combined to the original colour components, obtaining the watermarked RGB bands.

Some considerations are needed regarding the amount of energy that is inserted in the image. It is evident that, by masking the watermark, its energy depends also on the masking image. In particular given a mean value of α, defined as α_m, we want to embed into the image, the following value α has to be used in the embedding rule:

$$\alpha = \frac{\alpha_m}{1-M}$$

where $\alpha > \alpha_m$, so that the final watermarked luminance *Y'* will appear visibly degraded in some regions. The final step will consist in spatially hiding the watermark where the watermark will result more visible, by means of a combination between the original and the watermarked luminance.

3 Watermark decoding

In watermark detection step, the system is asked to decide if a given mark, provided by the user, is present into an image or not (*detectable* technique), without resorting to the original image. The decoder can decide the presence or the absence of a mark by comparing a decoding function against a predefined threshold. With regard to error detection probability, the decoder is optimum according to the Neyman-Pearson decision criterion: fixed a maximum value for the false alarm detection probability (in our case it is equal to 10^{-6}), the decoder minimises the missing watermark detection probability. In the following, the main steps of the detection process, shown in Figure 2, are described.

3.1 Watermarked luminance extraction

The watermarked color image I_M, is decomposed in the three color bands R_M, G_M, B_M, which are used to extract the luminance Y_M, where the watermark will be looked for.

3.2 Expansion to 1024x1024 pixels

The watermarked luminance Y_M is extended by means of zero padding, in order to obtain an image having a reference size of 1024x1024 pixel, that is the same size used in watermark casting. To this image, the FFT is applied. As we will see, this step is required for the synchronisation pattern detection. After synchronisation, a new expansion and a new FFT computation will be required, in order to carry out the watermark detection process.

3.3 Synchronisation pattern detection

The FFT of the watermarked luminance is analysed in order to extract the peaks corresponding to the synchronisation pattern. Their position will reveal if on the image a resizing or a rotation has been carried out, and, in this case, also the extent of these modifications. This is possible since a rotation or a resizing of the image corresponds to an equal rotation or resizing of its FFT magnitude.

Let us note with *PR* the reference pattern, that is the synchronisation signal introduced in watermark casting, and with *PS* the synchronisation pattern, that is the distorted synchronisation signal extracted in this step; the comparison between the two signals allows to find the transformation T_E linking the two patterns: $T_E\{PR\} = PS$

It is possible to demonstrate that a generic geometric transformation can be represented by means of an equivalent transformation, consisting of a rotation by an angle α, a resizing with scaling factors Δ_X (in the horizontal axis) and Δ_Y (in the vertical axis), and a new rotation by an angle β. Thus, T_E can be represented as the product of the matrices corresponding to each of the three simple geometric operations:

$$T_E = \begin{pmatrix} \cos\beta & -\operatorname{sen}\beta \\ \operatorname{sen}\beta & \cos\beta \end{pmatrix} \begin{pmatrix} \Delta_x^{-1} & 0 \\ 0 & \Delta_y^{-1} \end{pmatrix} \begin{pmatrix} \cos\alpha & -\operatorname{sen}\alpha \\ \operatorname{sen}\alpha & \cos\alpha \end{pmatrix}$$

If f_{X_A}, f_{Y_A}, f_{X_B}, f_{Y_B}, represent the frequency coordinates of the main harmonics of the reference pattern *PR*, and $f_{X'_A}$, $f_{Y'_A}$, $f_{X'_B}$, $f_{Y'_B}$, those of the extracted synchronisation pattern *PS*, the link between the two patterns is given by:

$$T_E \begin{pmatrix} f_{x_A} & f_{x_B} \\ f_{y_A} & f_{y_B} \end{pmatrix} = \begin{pmatrix} f_{x'_A} & f_{x'_B} \\ f_{y'_A} & f_{y'_B} \end{pmatrix}$$

where f_{X_A}, f_{Y_A}, f_{X_B}, f_{Y_B} depend on the period *T*. When the analysis of the FFT spectrum of the attacked image is carried out, the values $f_{X'_A}$, $f_{Y'_A}$, $f_{X'_B}$, $f_{Y'_B}$ are obtained; by resolving the previous system, the values of the parameters α, Δ_X, Δ_Y and β are computed. Next step is given by the application of the inverse geometrical transformations to the watermarked luminance in order to obtain an image without geometrical attacks, where the watermark can be more easily looked for.

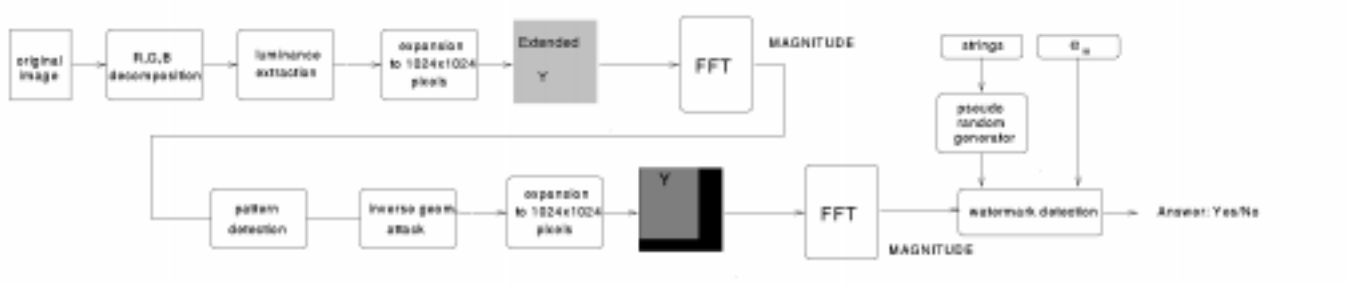


Figure 2: The watermark detection process

3.4 Expansion to 1024x1024 pixels

The new watermarked luminance obtained after the synchronisation process is extended by means of zero padding, in order to obtain an image having a reference size of 1024x1024 pixel, that is the same size used in watermark casting. To this image, the FFT is applied in order to carry out the watermark detection process.

3.5 Threshold-based watermark detection

The problem of watermark detection is the following: given a possibly watermarked image luminance Y_M, we want to know if a watermark m^* is present in Y_M or not. Since the position of the watermarked FFT coefficients is known, these coefficients can be selected obtaining a vector of elements. Thus, the input parameters are:

- The vector $\mathbf{y'} = \{y'_i\}_{i=0,1,\ldots N-1}$ of the watermarked FFT coefficient's amplitudes;
- The watermark sequence $\mathbf{m^*} = \{m^*_i\}_{i=0,1,\ldots N-1}$ generated using as seed the two strings we are looking for,
- The mean watermark energy, α_m used in watermark embedding, considered as a fixed parameter,

where N = 66384 is the number of watermarked coefficients and the length of the watermark sequence.

An optimum criterion to verify if a given code is present in an image is derived based on statistical decision theory [4]. Two hypotheses are defined: the image contains the watermark we are looking for (hypotheses H1) or the image does not contain this mark (hypotheses H0). Relying on Bayes theory of hypothesis testing, the optimum criterion to test H1 versus H0 is minimum Bayes risk; the test function results to be the likelihood ratio function L that has to be compared to a threshold:

- if $L > \lambda$, the system decides the watermark **m*** is present;
- if $L < \lambda$, the system decides the watermark **m*** is absent.

To choose a proper threshold, we have chosen to fix a constraint on the maximum false positive probability and we have referred to the Neyman-Pearson criterion [5] to design the optimum decoder, obtaining:

$$L(y) = \sum_{i=0}^{N-1}\left[-\beta_i \ln\left(1+\alpha_m m_i^*\right)\right] + \sum_{i=0}^{N-1}\left[-\left(\frac{y_i}{\alpha_i\left(1+\alpha_m m_i^*\right)}\right)+\left(\frac{y_i}{\alpha_i}\right)^{\beta_i}\right]$$

and

$$\lambda = 3.3\sqrt{2\sum_{i=0}^{N-1}\left[\frac{\left[\left(1+\alpha_m m_i^*\right)^{\beta_i}-1\right]}{\left(1+\alpha_m m_i^*\right)^{\beta_i}}\right]^2} + \sum_{i=0}^{N-1}\left\{\frac{\left[\left(1+\alpha_m m_i^*\right)^{\beta_i}-1\right]}{\left(1+\alpha_m m_i^*\right)^{\beta_i}}\right\} - \sum_{i=0}^{N-1}\left[\beta_i \ln\left(1+\alpha_m m_i^*\right)\right]$$

where $\mathbf{m^*} = \{m^*_i\}_{i=0,1,\ldots N-1}$ is the watermark, α_m the mean watermark energy, α_i and β_i the statistic parameters describing the probability density function shape of the magnitude of the watermarked DFT coefficient, y_i.

As a matter of fact, the function $L(y)$ depends on the knowledge of the probability density function of y_i, so that a procedure to estimate a posteriori the pdf of the DFT coefficients has been derived. By relying on the analysis carried out on a large set of images, we have obtained that the magnitude of a generic DFT coefficient y_i follows a *Weibull* distribution:

$$f(y_i) = \frac{\beta}{\alpha}\left(\frac{y_i}{\alpha}\right)^{\beta-1}\exp\left\{-\left(\frac{y_i}{\alpha}\right)^{\beta}\right\}$$

The estimation of the parameters of the Weibull distribution is done on the watermarked image, by means of the Maximum Likelihood criterion. At this aim, we suppose the DFT coefficients of the watermarked image belonging to small sub-regions of the spectrum are characterised by the same statistic parameters. The region of the DFT spectrum, where the watermark is embedded, is divided into 16 sub-regions: in each of the 16 groups of coefficients, the parameters α and β are estimated. See [6] for more details.

In summary, the detection process can be decomposed in the following steps:

- generation of the watermark **m***;
- estimation of the parameters α, β into the 16 regions composing the watermarked area of the spectrum;
- computation of $L(y)$ and λ;
- comparison between $L(y)$ and λ;
- decision.

The decoder can detect the watermark presence also in highly degraded images. In particular, the system is robust to sequences of different attacks, such as rotation, resizing, and JPEG compression, or such as cropping, resizing and median filtering.

4 Experimental results

To evaluate the robustness of the proposed system, a large set of tests has been performed by applying the benchmark *StirMark* [7,8] to a set of standard images. Given a watermarked image, StirMark applies several image transformations with various parameters. Then the output images can be tested with watermark detection to evaluate its robustness: a value equal to 1 is given when the watermark is detected, and a value 0 when it is not revealed. The attacks are then grouped in 8 subsets, in such a way that a percentage of survival to each group of attacks is obtained, as described in the following table.

Signal enhancement	1,00
Compression	0,99
Scaling	0,90
Cropping	0,83
Shearing	1,00
Rotation	0,91
Other geometrical transform.	0,72
Random geometric distortion	0,00

Table 1: Experimental results obtained with Stirmark [7,8]

The results are quite interesting, since the watermark has been removed only in a few attacks. In particular, the system is not robust to the image flipping, to an image scaling of 50%, to a large cropping, to a rotation of 30° or 45°, and to the default StirMark attack. As a final result, the benchmark has given a value of 0.79, one of the highest values between the watermarking systems that have been evaluated at this moment.

5 Conclusions

In this paper, a new watermarking system for copyright protection of digital images has been presented. The method operates in the frequency domain, by embedding a pseudo-random sequence of real numbers in a selected set of DFT coefficients of the image. After embedding, the watermark is hidden by exploiting the masking characteristics of the Human Visual System. Moreover, a synchronisation pattern is introduced into the watermarked image, to cope with geometrical attacks. An optimum criterion to verify if a given code is present is derived based on statistical decision theory. The proposed system has revealed to be robust against a large set of attacks a counterfeiter can use to remove the watermark, as described by the experimental results obtained by applying the benchmark Stirmark [7,8]. The main factors contributing to the robustness of the system, are:

- the expansion of the image to a fixed size before watermark casting, to obtain robustness against image cropping and translation;
- the visual masking process, to increase the watermark invisibility;
- the insertion of a synchronisation pattern, to obtain robustness against image resizing and rotation;
- the optimum detection process, which increased the detection performance with respect to the previous correlation detector, also presented in [9].

6 Acknowledgements

This work was partially supported by the Italian National Research Council (*CNR*) in the framework of the "Progetto Finalizzato Beni Culturali" and by MURST.
Thanks to Fabien Petitcolas for carrying out the evaluation of our system with his benchmark Stirmark.

7 References

[1] A. Piva, M. Barni and F. Bartolini, "Copyright protection of digital images by means of frequency domain watermarking", in Mathematics of Data/Image Coding, Compression, and Encryption, Schmalz Ed., Proc. of SPIE Vol. 3456, pp. 25-35, San Diego, California, July 21-22, 1998.

[2] A. H. Tewfik and M. Swanson, "Data hiding for multimedia personalization, interaction, and protection", IEEE Signal Processing Magazine, vol. 14, no. 4, pp. 41-44, July 1997.

[3] F. Bartolini, M. Barni, V. Cappellini, A. Piva, "Mask building for perceptually hiding frequency embedded watermarks", *Proceedings of* 5th *IEEE International Conference on Image Processing ICIP'98*, Chicago, Illinois, USA, October 4-7, 1998, Vol I, pp. 450-454.

[4] L. Scharf, *Statistical Signal Processing: detection, estimation, and time series analysis*, Add. Wesley, 1991.

[5] J.V. Di Franco and W.L. Rubin, *Radar Detection*, Artech House, Inc., Dedham, Ma., 1980.

[6] A. De Rosa , M. Barni, F. Bartolini, V. Cappellini , and A. Piva, "Optimum decoding of non-additive full frame DFT watermarks", *Third International Workshop on Information Hiding*, 29 September-1 October, 1999, Dresden, Germany.

[7] F. Petitcolas, R. Anderson, M. Kuhn, "Attacks on copyright marking systems", in D. Aucsmith (Ed), Information Hiding, 2nd Int. Workshop, Portland, Oregon, USA, April 15-17, 1998, Proceedings, LNCS 1525, Springer-Verlag, pp. 219-239.

[8] F. Petitcolas and R. Anderson, "Evaluation of copyright marking systems". Proc. IEEE Multimedia Systems (ICMCS'99), June 7-11, 1999, Florence, Italy, Vol. 1, pp. 574-579.

[9] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A DCT-domain system for robust image watermarking," *Signal Processing*, Vol. 66, No. 3, May 1998, pp. 357-372.

A Noise Removal Attack for Watermarked Images

Sviatoslav Voloshynovskiy*#, Alexander Herrigel#, Frederic Jordan#,
Nazanin Baumgärtner#, Thierry Pun*

*CUI – University of Geneva
Department of Computer Science
Geneva, Switzerland

DCT – Digital Copyright Technologies
Research & Technology
Zurich, Switzerland

ABSTRACT

This paper[1] presents a new attack for watermarked still pictures or images. In contrast to the Stirmark benchmark, this attack does not severely reduce the quality of the image and is not based on a large number of image processing operations. The approach maintains the commercial value of the image after the attack has been completed. We demonstrate the effectiveness of the attack showing test cases of professional images[2], which have been watermarked with different commercial systems. The attack is based on a stochastic approach, which identifies the local image properties of the imaging, including flat, textured and edge regions investigating stationary generalized Gaussian and non stationary Gaussian distribution models. We show by experiment that any watermarked scheme, which is not based on the computation of the derived Noise Visibility Function (NVF) may be broken by the described attack.

1 Introduction

Copyright infringements of digital images can be detected by digital watermarks, embedded by special software programs. Visible and invisible watermarks may be applied for copyright protection. Visible watermarks direct the observer to the fact that the image is copyright protected (example: IBM watermark project for the Vatican). The practical usage of this watermark technology is, however, quite limited. The quality of still pictures is substantially changed and the applied protection procedure is not robust, because the visible watermark can be removed by image processing programs such as PhotoShop or others. The invisible watermark utilizes the inability of the human vision system to perceive small differences in optical data. These slight differences are exploited by special software programs for embedding copyright information directly into the images. Invisible watermarks have the advantage that they can not be identified and destroyed easily if the watermark process is robust against specific image transformations such as lossy compression, change of contrast, vector quantization, rotation, scaling, translation, cropping, change of aspect ratio, color editing, morphing etc. .

We use the expression Cover-Image and Stego-Image to explain the Watermark Embedding and Detection Process (WEDP). The WEDP embeds or detects owner authentication data in a specific digital image. The owner authentication data is embedded such that the commercial usability of the digital image is not affected. For this purpose, a cryptographic key is applied to embed encoded owner authentication data, called the watermark, into the Cover-Image CI, resulting in a Stego-Image SI. The watermark data can then be extracted from the Stego-Image if the correct cryptographic key is applied. Due to system security reported in Crawler [1], we assume that the CI is not applied in the watermark detection process. In addition, we assume a watermarking scheme, which supports the following features (the DCT technology satisfies for example this set of requirements needed for a secure and robust WEDP):

- The watermark embedding process depends on the original data.
- The watermark embedding process supports a time dependent identification.
- The watermark embedding process is content adaptive to prevent any visible artifacts, which may be generated by the embedding process.
- No original data is needed for the watermark detection.

[1] This work has been funded by the Swiss National Science Foundation under the SPP program (Grant. 5003-45334) and by the European Commission (ESPRIT-OMI Project No. 25530: Jedi-Fire).

[2] Since the images applied in the Stirmark benchmark do not always shre the same properties as commercial images, we have used in our tests professional images from Fratelli Alinary, which have been given to us for testing and benchmarking purposes. We would like to thank Fratelli Alinari, especially Mr. Andrea de Polo, for providing these test images. Fratelli Alinari is the copyright holder of these images. These images may not be used for any business purpose without the written permission of Fratelli Alinari, http://www.alinari.com.

- The watermark embedding is based on a scheme, which supports secure verification in a legal dispute who has really embedded the watermark.
- The WEDP supports secure means to uniquely identify the copyright owner if multiple watermarks have been embedded in the same image.

In contrast to steganography watermarking is constrained with additional robustness requirements. If we consider the watermark problem decomposed into a encoder, decoder and a transmission channel, then, in contrast to steganography, the pirate may intercept the transmission channel to modify, delete, or replace the encoded watermark. This means that the watermark should not only be unnoticeable but also robust, meaning that the interception of the pirate should be prevented by detecting the watermark after the applied image modification of the pirate. Since the modified image should be close to the watermarked image to preserve the commercial value of the image content, the number of the different attacks of the pirate is limited [2] and also constrained by possible visible artifacts. In contrast to the existing information theory based approaches, these attacks are very specific and can not be modeled by random Gaussian noise only. Furthermore, the distribution of an image as not purely stationary Gaussian [18], since the image regions of interest for watermarking may have very different local features. In addition, the channel capacity is not uniform since the image contents of every window constitute a channel capacity, which is closely linked to the local image characteristics, which are not uniform over the whole image and dependent from visible artifacts.

It has been recently shown that the proposed watermarking algorithms require adequate tools (UnZign [3], Stirmark [4]) to investigate their robustness to different kinds of attacks and to guide the development of the adequate means for improvements.

This paper presents a new stochastic formulation of the watermark removal problem considering an embedded watermark as additive noise with a Gaussian distribution. The watermark removal problem is reduced to the classical denoising problem. The approach *of the Maximum A Posteriori Probability (MAP)* estimate is presented and two types of image *prior* models are described. The first image model is based on a non-stationary Gaussian distribution. The second one is based on stationary Generalized Gaussian distribution with shape parameter and variance to be hyperparameters of the model.

The relationship between the obtained stochastic model of image discontinuities (edge and textured regions) is investigated along with the human perceptual models based on *Noise Visibility Function (NVF).* An Extension of the proposed approach to the general case is presented.

It is shown by experimental results that any scheme, which is not based on the computation of the NVF may be broken by the derived attack. Tests with professional images are presented applying the new approach in combination with commercial available watermark product solutions.

2 State-of-the-art Approaches

In recent publications several authors have proven that content adaptive schemes are a key issue for the watermark embedding process. Some proposals are based on the utilization of luminance sensitivity function of the human visual system (HVS) [5]. Since the derived luminance functions are based on a crude estimation of the image contrast the luminance based embedding is not efficient against different compression or denoising attacks. Other approaches exploit transfer modulation features of HVS in the transform domain to solve the compromise between the robustness of the watermark and its visibility [6]. These approaches embed the watermark in a predetermined middle band of frequencies in the Fourier domain with the same strength assuming that the image spectra have isotropic character. This assumption leads to some visible artifacts in images specially in the flat regions, because of anisotropic properties of image spectra. A similar method using blocks in DCT (discrete cosine transform) domain was proposed in [7, 8]. In the context of image compression using perceptually based quantizers, this concept was further developed in [9], which adjust the watermark for each DCT block. Since the original image is required to extract the watermark, the practical applications of this approach is very limited. It was proven that the usage of the cover image will result in watermark schemes, which may be broken [1]. Other DCT approaches use luminance and texture masking [10]. Some approaches are based on the image compression techniques [11] and exploits 3 basic conclusions: (1) all regions of high activity are highly insensitive to distortion; (2) the edges are more sensitive to distortion than highly textured areas; (3) darker and brighter regions of the image are less sensitive to noise. The typical examples of this approach are [12, 13]. The developed methods consist of a set of empirical procedures aimed to satisfy the above requirements. The computational complexity and the absence of closed form expressions for the perceptual mask complicate the analysis of the received results. However, experiments performed in these papers show high robustness of these approaches. A very similar method was proposed in [14], where edge detectors are used to overcome the problem of visibility of the watermark around the edges.

3 Problem Formulation

Consider the classical problem of non-adaptive watermark embedding, i.e. embedding the watermark regardless of the image content. In the most general case it can be defined according to the model:

$$y = x + n$$

where x is the cover image, y is the Stego-Image, $(y \in \Re^N, N = M \times M)$, and n is associated with the noise-like encoded watermark. The encoding is based on a spread spectrum type of technique [15]. Our goal is to find an estimate $\hat{n}$ of the watermark n and an esti-

mate $\hat{x}$ of the cover image x to compute an estimation of the watermark as

$$\hat{n} = y - \hat{x},$$

where $\hat{n}$ and $\hat{x}$ denote the estimates of the watermark and the estimate of the cover image. The decision about the presence/absence of the watermark in a given image is then made by a robust detector. This detector must consider the prior statistics of the watermark and the possible errors of its estimation due to the decomposition residual coefficients of the cover image and the possibly applied attack. This generalized idea has found practical applications for watermarking [5] and steganography [16]. The key moments of the above approach are the design of the corresponding estimator and the robust detector. The problem of estimation of the cover image from its noisy version is known as image denoising or image smoothing.

Our stochastic approach is based on two image models. An image is assumed to be a random process. We consider a stationary and non-stationary process to model the cover image. The stationary process is characterized by the constant parameters for the whole image and the non-stationary has spatially varying parameters. To estimate the parameters a maximum likelihood estimate is used in the specified neighborhood set. We assume that image is either a non-stationary Gaussian process or a stationary Generalized Gaussian process.

4 The Stochastic Model

4.1 Watermark Estimation

A probabilistic model of the watermark and the cover image is needed to consider the watermark problem from a statistic perspective. If the watermark has the distribution $p_n(n)$ and the cover image the distribution $p_x(x)$, then a MAP estimation of the watermark could is given by:

$$\hat{n} = \arg\max_{\tilde{n} \in \Re^N} L(\tilde{n}|y)$$

where $L(\tilde{n}\,|\,y)$ is the log function of a posteriori distribution

$$L(\tilde{n}|y) = \ln p_x(y|\tilde{n}) + \ln p_n(\tilde{n}).$$

The estimate $\hat{n}$ gives then

$$\tilde{x} = \arg\max_{\tilde{x} \in \Re^N} \{\ln p_n(y|\tilde{x}) + \ln p_x(\tilde{x})\}.$$

The solution for the last three equations are equivalent to each other. The last formulation is the typical image denoising problem, which will be further considered. A solution for this problem needs accurate stochastic models for the watermark $p_n(n)$ and the cover image $p_x(x)$. Under the assumption that the watermark is coded applying any spread spectrum technique, it is possible to model it as Gaussian random variable. Let samples $n_{i,j} (1 \le i, j \le M)$ be defined on vertices of a $M \times M$ grid, and let each sample $n_{i,j}$ takes a value in $\Re$. The following equation holds if the samples are independent identically distributed (i.i.d.):

$$p_n(y|x) = \frac{e^{\{-\frac{(y-x)^T (y-x)}{2\sigma_n^2}\}}}{\sqrt{(2\pi\sigma_n^2)^N}}$$

where σ_n^2 is the variance of the watermark. This assumption is reasonable, since the Gaussian distribution has the highest entropy among all other distributions. With respect to system security any applied spread spectrum encoding algorithm should approach this distribution in the limit. Thus, the watermark could be modeled as $n \sim N(0, \sigma_n^2)$. We derive in the following section now the corresponding stochastic model for the image.

4.2 Cover Image Estimation

If we apply the Markov Random Field (MRF) model [17], we can derive the following equation for the image distribution (MRF written as Gibbs distribution):

$$p(x) = \frac{e^{-\sum_{c \in A} V_c(x)}}{Z},$$

where Z is a normalization constant called the partition function, $V_c(\cdot)$ is a function of a local neighboring group c of points and A denotes the set of all possible such groups or cliques. We consider two particular cases of this model, i.e. the Gaussian and the Generalized Gaussian (GG) models. Assume that the cover image is a random process with non-stationary mean. Then applying the Auto Regressive (AR) model notations, we can derive for the cover image the following equation:

$$x = A \cdot x + \varepsilon = \overline{x} + \varepsilon,$$

where $\overline{x}$ is the non-stationary local mean and ε denotes the residual term due to the error of estimation. The particularities of the above model depend on the assumed stochastic properties of the residual term:

$$\varepsilon = x - \overline{x} = x - A \cdot x = (I - A) \cdot x = C \cdot x,$$

where $C = I - A$ and I is the identity matrix. If A is a low-pass filter, then C represents a high-pass filter (decomposition operator). We use here two different models for the residual term ε.

The first model is the non-stationary (inhomogeneous) Gaussian model and the second one is the stationary (homogeneous) Generalized Gaussian (GG) model. The choice of these two models is motivated by the wide application in image restoration and denoising [18, 19]. The today best wavelet compression algorithms are also based on these models [20, 21]. The main advantage of these models is that they take local features of the image into account. In the first case, this is done by introducing the non-stationary (spatially variant) variance using a

quadratic energy function, and in the second case, by using an energy function, which preserves the discontinuity of a stationary variance of the image. In other words, in the non-stationary Gaussian model, the data is assumed to be a locally i.i.d. random field with a Gaussian probability distribution function (pdf), while in the stationary GG model the data is assumed to be a globally i.i.d. random filed. The auto-covariance function in the non-stationary case is given by the following equation with $\{\sigma_{Xi}^2 | 1 \le i \le N\}$ as the local variances.

$$R_x(\sigma_{x1}^2,...,\sigma_{xN}^2) = \begin{pmatrix} \sigma_{x1}^2 00\ldots\ldots\ldots\ldots 0 \\ 0\sigma_{x2}^2 00\ldots\ldots\ldots 0 \\ 00\sigma_{x3}^2 00\ldots\ldots\ldots 0 \\ \vdots\cdots 0\ddots\cdots\ldots\cdots 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\sigma_{xN}^2 \end{pmatrix}$$

The auto-covariance function for the stationary model is given by:

$$R_x(\sigma_x^2) = \begin{pmatrix} \sigma_x^2 00\ldots\ldots\ldots\ldots 0 \\ 0\sigma_x^2 00\ldots\ldots\ldots 0 \\ 00\sigma_x^2 00\ldots\ldots\ldots 0 \\ \vdots\cdots 0\ddots\cdots\ldots\cdots 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\sigma_x^2 \end{pmatrix}$$

where σ_X^2 is the global image variance.

Considering the equation for the non-stationary Gaussian model results in the following equation if the model is combined with the auto-covariance function:

$$p_x(x) = \frac{e^{\{-0.5(Cx)^T R_x^{-1}(Cx)\}}}{(2\pi)^{\frac{N}{2}} |\det R_x|},$$

where $|\det R_x|$ denotes the matrix determinant, and ^{T} denotes the transposition. The stationary GG model is then given by

$$p_x(x) = \left(\frac{\gamma\eta(\gamma)}{2\Gamma(\frac{1}{\gamma})}\right)^{\frac{N}{2}} \cdot \frac{e^{\{-\eta(\gamma)(|C|^{\frac{\gamma}{2}})^T R_x^{-\frac{\gamma}{2}} |Cx|^{\frac{\gamma}{2}}\}}}{|\det R_x|^{\frac{1}{2}}},$$

$$\text{where } \eta(\gamma) = \sqrt{\Gamma(\frac{3}{\gamma}) \Big/ \Gamma(\frac{1}{\gamma})},$$

and $\Gamma(t) = \int_0^\infty e^{-u} u^{t-1} du$ is the gamma function, and the parameter γ is called the shape parameter. The last equation includes the Gaussian $(\gamma = 2)$ and the Laplacian $(\gamma = 1)$ models as special cases. For the real images the shape parameter is in the following

$$\text{range } 0.3 \le \gamma \le 1.$$

Having derived the stochastic models for the watermark and the cover image we can now investigate the problem of image estimation according to the MAP approach.

4.3 MAP based Image Denoising

For the MAP estimation, we have to solve the following optimization problem:

$$\hat{x} = \arg\min_{\tilde{x} \in \Re^N} \{\frac{1}{2\sigma_n^2} ||y - \hat{x}||^2 + \rho(r)\},$$

where $\rho(r) = [\eta(\gamma)|r|]^\gamma$, $r = \frac{x - \overline{x}}{\sigma_X} = \frac{Cx}{\sigma_X}$, and $||\cdot||$ denotes the matrix norm. $\rho(r)$ is the energy function for the GG model.

In the case of the non-stationary Gaussian model $\gamma = 2$ (i.e. convex function) and σ_X^2 is spatially varying. The advantage of this model is the existence of a closed form solution. The solution is given by an adaptive Wiener or a Lee filter [22]. In the case of stationary GG model the general closed form solution does not exist, since the penalty function could be non-convex for $\gamma \prec 1$. In practice, iterative probabilistic and deterministic optimization algorithms are applied for this problem. Examples are given in [17, 23-26]. A closed form solution in wavelet domain exists for $\gamma \prec 1$, called the soft-shrinkage [18, 27]. To obtain an adequate solution technique for the close form solution of the estimate, we map the constraints of the problem in a convex optimization function, which is based on the boundary conditions. This function is called the reweighted least squares (RLS) problem. The problem is, therefore, mapped to the following minimization problem:

$$\hat{x}^{k+1} = \arg\min_{\tilde{x} \in \Re^N} \{\frac{1}{2\sigma_n^2} ||\mathrm{y}\text{-}\tilde{\mathrm{x}}^{\mathrm{k}}||^2 + w^{k+1} ||\mathrm{r}^{\mathrm{k}}||^2\},$$

where $w^{k+1} = \frac{\rho'(r^k)}{r^k}$, $r^k = \frac{x^k - \overline{x}^k}{\sigma_x^k}$,

$\rho'(r) = \frac{\gamma[\eta(\gamma)]^\gamma r}{||r||^{2-\gamma}}$, and k is the number of iterations.

In this case, the penalty function is quadratic for a fixed weighting function w. Assuming w is constant for a particular iteration, we receive the general RLS solution in the following form:

$$\hat{x} = \frac{w\sigma_n^2 \overline{x}}{w\sigma_n^2 + \sigma_X^2} + \frac{\sigma_X^2 y}{w\sigma_n^2 + \sigma_X^2}.$$

This solution is similar to the closed form of Wiener filter solution [22]. The same RLS solution in the form of Lee filter [22] is given by:

$$\hat{x} = \overline{x} + \frac{\sigma_X^2 (y - \overline{x})}{w\sigma_n^2 + \sigma_X^2}$$

The principal difference between the classical Wiener or Lee filter is the presence of the weighting function w. This weighting function depends on the underlying assumptions about the statistics of the cover image.

We only consider the Lee version of the RSL solution, which coincides with the classical case of Gaussian prior of the cover image $w = 1$. It is important to note that the shrinkage solution of image denoising problem previously used only in the wavelet domain can easily be obtained from in the next close form solution:

$$\hat{x} = \overline{x} + \max\{0, (y - \hat{x}) - T\},$$

where $T = \frac{\sigma_n^2}{\sigma_x^\gamma} \gamma [\eta(\gamma)]^\gamma (y - \overline{x})^{\gamma - 1}$ is the threshold. In particular case of Laplacian image prior, $T = \frac{\sigma_n^2}{\sigma_x^\gamma} \sqrt{2}$. This coincides with the soft-threshold solution of the image denoising problem [18]. The properties of the image denoising algorithm are defined by the term:

$$b(w, \sigma_x, \sigma_n) = \frac{\sigma_X^2}{w\sigma_n^2 + \sigma_X^2},$$

in equations of the RLS solution. It is commonly known, that the local variance is a good indicator of the local image activity, i.e. when it is small, the image is flat, and a large enough variance indicates the presence of edges or highly textured areas. Therefore, the function $b(w, \sigma_x, \sigma_n)$ determines the level of image smoothing. For example, for flat regions $\sigma_x^2 \to 0$, and the estimated image equals local mean, while for edges or textured regions $\sigma_x^2 \gg \sigma_n, b \to 1$ and the image is practically left without any changes. Such a philosophy of the adaptive image filtering is very well matched with the texture masking property of the human visual system: the noise is more visible in flat areas and less visible in regions with edges and textures.

Based on the non-stationary Gaussian and stationary Generalized Gaussian models we propose the following texture masking function.

5 Texture Masking Function

We propose to relate the texture masking function to the noise visibility function (NVF) as:

$$NVF = 1 - b = \frac{w\sigma_n^2}{w\sigma_n^2 + \sigma_x^2},$$

which is just the inverted version of the function b. In the main application of the proposed NVF in context of watermarking, we assume that the noise (watermark) is an i.i.d. Gaussian process with unit variance, i.e. $N(0,1)$. This noise excite the perceptual model [26], in analogy with the AR image model. The NVF is the output of the perceptual model to a noise with the distribution $N(0,1)$. The developed perceptual model depends on the weighting w, which is determined by the other parameters of the RLS solution.

5.1 NVF Based on Non-Stationary Gaussian Model

The shape parameter γ is for the non-stationary Gaussian model 2 and the auto-covariance function is given by $R_x(\sigma_{x1}^2, ..., \sigma_{xN}^2)$. The weighting function w is then equal to 1 (see w in RLS close form solution) and the NVF is given by:

$$NVF(i, j) = \frac{1}{1 + \sigma_x^2(i, j)},$$

where $\sigma_x^2(i, j)$ denotes the local variance of the image in a window centered on the pixel with coordinates $(i, j), 1 \le i, j \le M$. The NVF is, therefore, inversely proportional to the local image energy defined by the local variance. The Maximum Likelihood (ML) estimate is applied for the computation of the local image variance. Assuming that the image is a locally i.i.d. Gaussian distributed random variable, the ML estimate is given by:

$$\sigma_X^2(i, j) = \frac{\sum_{k=-L}^{L} \sum_{l=-L}^{L} [x(i+k, j+l) - \overline{x}(i, j)]^2}{(2L+1)^2}$$

with $\overline{x}(i, j) = \frac{\sum_{k=-L}^{L} \sum_{l=-L}^{L} x(i+k, j+l)}{(2L+1)^2}$.

A window of size $(2L+1) \times (2L+1)$ is used for the computation.

5.2 NVF Based on Stationary GG Model

The following equations computes the NVF For the stationary GG model:

$$NVF(i, j) = \frac{w(i, j)}{w(i, j) + \sigma_x^2},$$

where $w(i, j) = \frac{\gamma [\eta(\gamma)]^\gamma}{||r(i, j)||^{2-\gamma}}$ and

$$r(i, j) = \frac{x(i, j) - \overline{x}(i, j)}{\sigma_x}.$$

The particularities of this model are determined by the shape parameter γ and the global image variance σ_x^2. We use a moment matching method reported in [21] to estimate the shape parameter. The analysis consists of the next stages. First, the image is decomposed according to the equation

$$\varepsilon = x - \overline{x} = x - A \cdot x = (I - A) \cdot x = C \cdot x ,$$

using equation $\overline{x}(i, j)$ from the last section as an estimate of the local mean. In the second stage, the moment matching method is applied to the residual image and the shape parameter and the variance are estimated.
The shape parameter for most of real images is in the range $0.3 \le \gamma \le 1$.

6 The Attack

Based on the presented results of the stochastic models, the attack has two phases. In phase 1, we assume that the embedded watermark has a Gaussian distribution and the image has a Generalized Gaussian distribution. Based on the moment matching technique, γ is estimated and then the identified Gaussian noise is removed. The Gaussian noise is very similar to the actual watermark, since many schemes are based on some type of spread-spectrum techniques. The noise removal is based on the local mean and the local variance from the non-stationary Gaussian model. In the second phase, the perceptual mask from the prefiltered image is calculated and additional noise is embedded in the image with respect to the perceptual components. The embedding depends on the stationary GG model presented in section 5.2.

7 Test Results

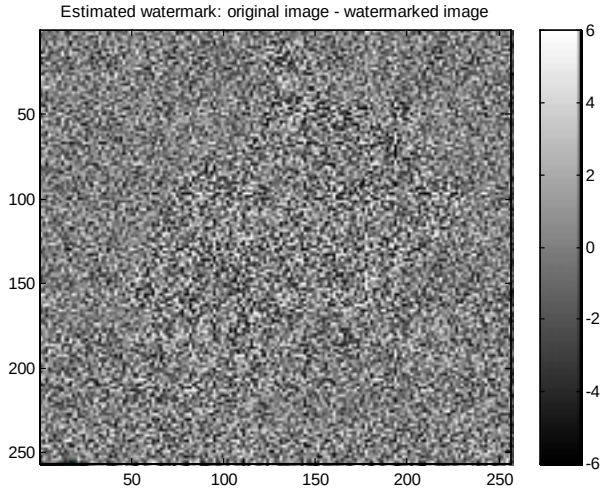
Several test results for Stirmark have been reported [4, 28]. The Stirmark benchmark tool has proven to be an effective tool for measuring the performance of the different watermark technologies with respect to specific image operations applied to destroy the embedded watermarks. Some of the applied operations (for example compression down to 10% quality), however, result in images, which have a visible quality difference compared to the original watermarked image. These attacks don't preserve, therefore, the commercial value of the image content. In addition, the PSNR is applied as an objective measurement criteria to limit the distortions generated by the tested watermark technologies. We have shown [29], however, that this measurement may not adequate, since the visible image quality may be quite different for very similar PSNRs. In addition, why should a pirate apply Stirmark if other attacks are available, which preserve the image quality and have a better performance during the execution (The Stirmark testbench generates more than 150 different files for destroying the watermark of a protected image)? For a fair comparison, we have collected the published test results and summarized them in the table below. Since not all watermark solutions have been consistently tested during the development of the Stirmark benchmark (different versions), we have consolidated the results with respect to the latest publications. We performed also the Stirmark benchmark against the DCT technology, which is based on the NVF approach.

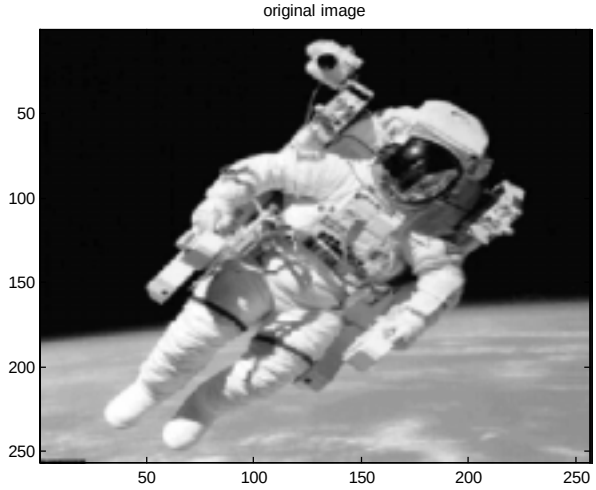
Stirmark Benchmark Results (12. 8. 1999)		
Position	Company	Stirmark Ratio (1 is maximum)
1	DCT	**0.828**
2	Digimarc	0.780
3	CUI[3]/DCT	0.700
3	Signum Technologies	0.700
4	Blue Spike/Dice	0.230
5	Alpha Tec	0.290
6	MediaSec	0.220
7	IP2	No results reported
7	Signafy	No results reported

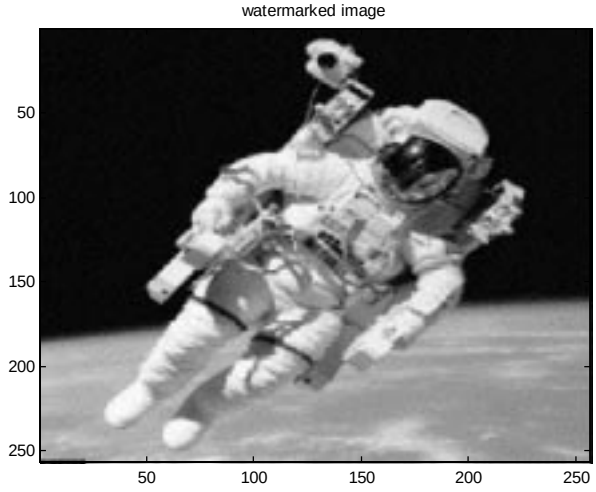
The following figures illustrate a specific attack we have run against the Digimarc system with the astro image. The figures illustrate the different phases applied and the remaining watermark after the processing.
In contrast to the DCT system, which is based on the noise visibility function computation, it was not possible for the Digimarc detector to detect the watermark at all, meaning, the decoder could not even detect that besides the payload a watermark from Digimarc was embedded. We have obtained similar results[4] with other systems such as SysCop.
We have tested our approach with several images such as astro, fisherman, boat, Lena and some professional images from Fratelli Alinari. The images from Fratelli Alinari are presented above. In contrast to the images applied in the Stirmark benchmark, these images have a larger region of flat area, which also influence the watermarking. As shown, the embedding should address the different regions, such as texture, flat area, and edges.
Running the different test cases we have noticed that the attack is effective in all cases.

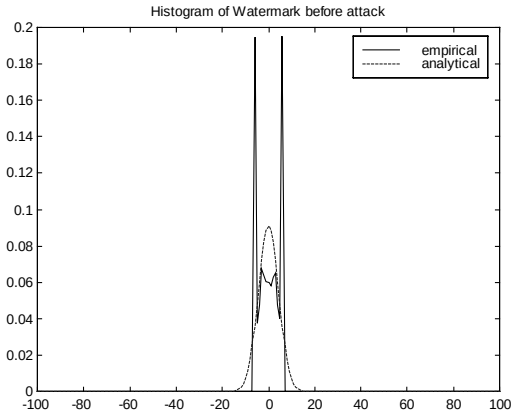
[3] This measurement was reported under University of Geneva in the Stirmark publications. DCT owns the IP and the implementations of the CUI technology.

[4] It should be noticed that the versions offered for free download may offer a reduced functionality as the real commercial solutions. It is, therefore, possible, that the reported results may be different, if the latest commercial solutions are applied. With respect to the ongoing Stirmark testing, we expect, however, that the detector functionality won't be different as the one supported by the downloaded versions.


Estimated watermark: original image - watermarked image
50
100
150
200
250
6
4
2
0
-2
-4
-6


original image
50
100
150
200
250


watermarked image
50
100
150
200
250


Histogram of Watermark before attack
empirical
analytical
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-100
-80
-60
-40
-20
0
20
40
60
80
100

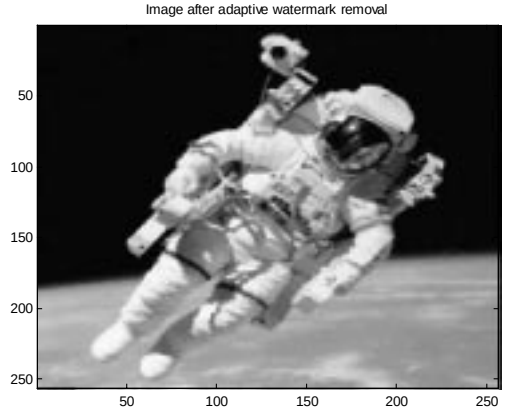
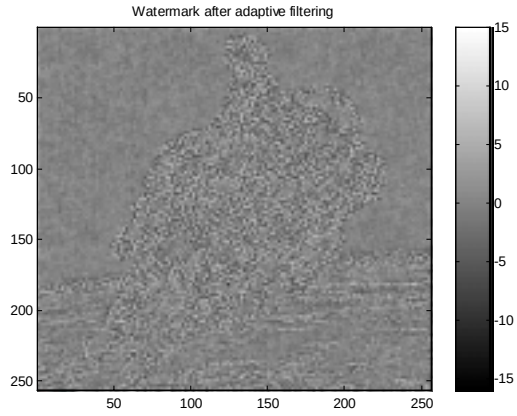
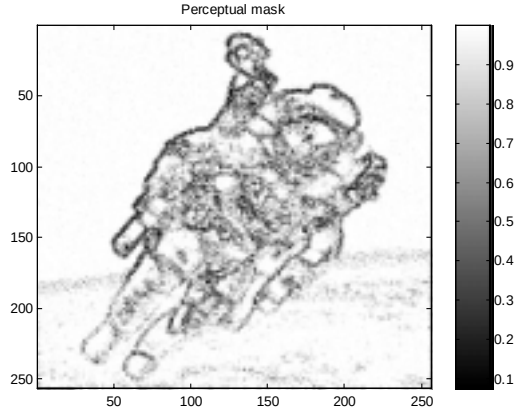
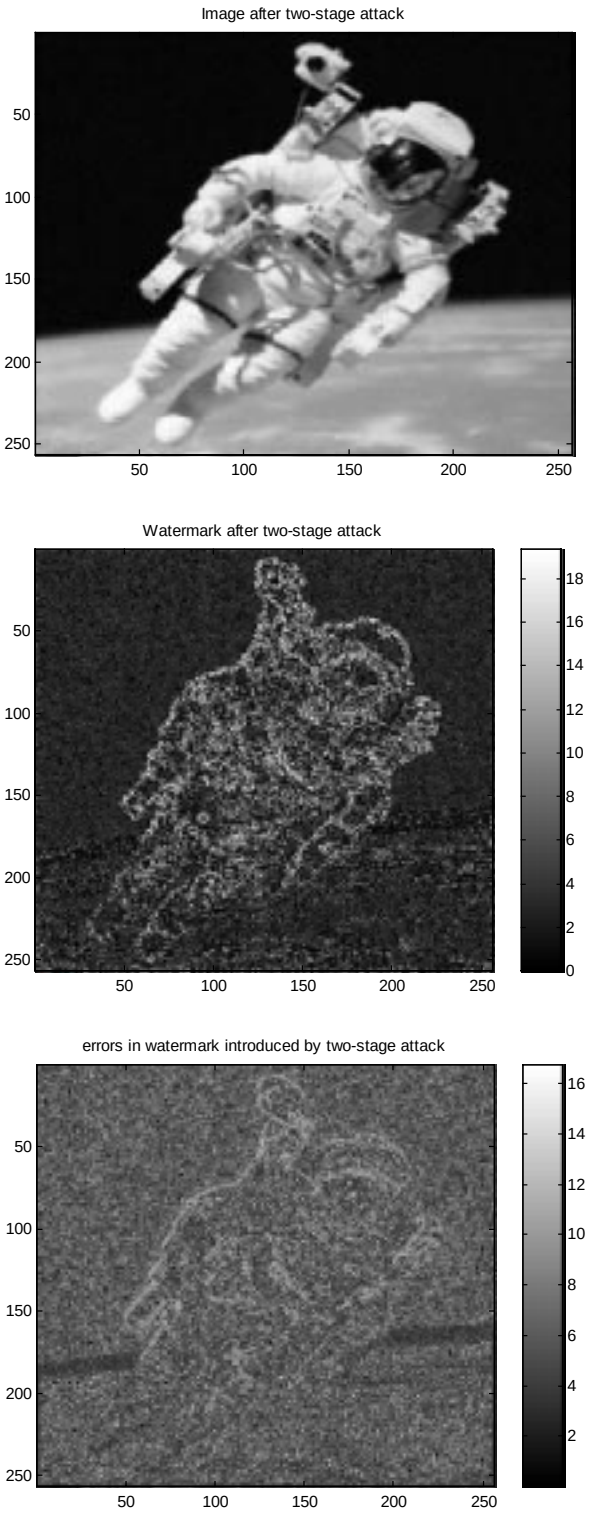
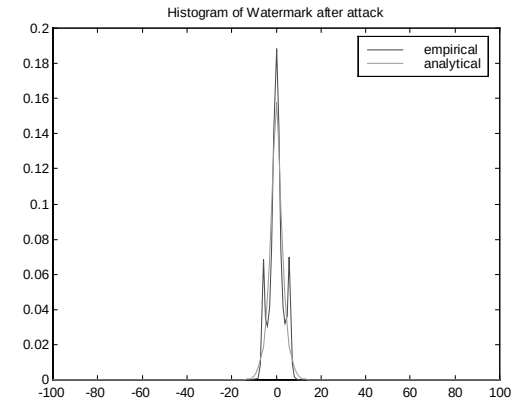


Image after adaptive watermark removal
50
100
150
200
250


Watermark after adaptive filtering
50
100
150
200
250
15
10
5
0
-5
-10
-15


Perceptual mask
50
100
150
200
250
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1





8 Conclusions

We have presented in this paper a new attack for watermarked images. Based on the stochastic modeling it is possible to model precisely the watermark as random noise. In contrast to the Stirmark benchmark, this attack does not severely reduce the quality of the image and is not based on a large number of image processing operations. The attack maintains the commercial value of the image after the attack has been completed. We demonstrate the effectiveness of the attack showing test cases of professional images, which have been watermarked with different commercial systems. And successful broken.

We show by experiment that any watermarked scheme, which is not based on the computation of the derived Noise Visibility Function (NVF) may be broken by the described attack.

9 References

[1] S. Craver, N. Memon, B. Yeo, and M. Yeung, Can invisible marks resolve rightful ownerships ?, IS&T/SPIE Electronic Imaging '97: Storage and Retrieval of Image and Video Databases, 1997.

[2] T. Mittelholzer, An Information-Theoretic Approach to Steganography and Watermarking, Digital Copyright technologies, Internal Report, October 7, 1998. http://www.altern.com/watermark/

[4] F. Petitcolas, R. Anderson, Proceedings of IEEE Multimedia Systems'99, vol. 1, pp. 574--579, 7-11 June 1999, Florence, Italy.

[5] M. Kutter, Watermarking Resisting to Translation, Rotation and Scaling, Proc. of SPIE, Boston, USA, November 1998.

[6] M. Kutter, F. Petitcolas, A fair benchmark for image watermarking systems, SPIE, Vol. 3657, San Jose, January 1999, pp. 226-239.

[7] J. Ruanaidh, T. Pun, Rotation, Scale and Translation Invariant Spread Spectrum Digital Image Watermarking, Signal Processing, May 1998, Vol. 66, No. 3, pp. 303-317.

[8] M. Swanson, B. Zhu, A. Twefik, Transparent Robust Image Watermarking, Proc. of 3rd IEEE International Conference on Image Processing ICIP96, 1996, Vol. 3, pp. 211-214.

[9] C. Podilchuk, W. Zeng, Image Adaptive Watermarking Using Visual Models, IEEE Journal on Selected Areas in Communication, May 1998, Vol. 16, No. 4, pp. 525-539.

[10] J. Huang, Y. Shi, Adaptive Image Watermarking Scheme Based on Visual Masking, Electronic Letters, April 1998, Vol. 34, No. 8, pp. 748-750.

[11] N. Jayant, J. Johnston, R. Safranek, Signal Compression Based on Models of Human Perception, Proc. of the IEEE, 1993, Vol. 81, No. 10, pp. 1385-1422.

[12] M. Kankanhalli, R. Ramakrishnan, Content Based Watermarking of Images, ACM Mulimedia´98, Bristol, UK, 1998, pp. 61-70.

[13] F. Bartolini, M. Barni, V. Cappelini, A. Piva, Mask Bilding for Perceptually Hiding Frequency Embedded Watermarks, Proc. of 5th IEEE International Conference on Image Processing ICIP98, Chicago, Illinois, USA, October 4-7, 1998, Vol. 1, pp. 450-454.

[14] J. F. Delaigle, C. De Vleeschouwer, B. Macq, Watermarking Algorithm Based on a Human Visual Model, Signal Processing, 1998, Vol. 66, pp. 319-335.

[15] I. Cox, J. Kilian, T. Leighton, T. Shamoon, Secure Spread Spectrum Watermarking for Multimedia, NEC Research Institute Tech Rep. 95-10, 1995.

[16] L. Marvel, C. Retter, C. Boncelet, Hiding Information in Images, Proc. of 5th IEEE International Conference on Image Processing ICIP98, Chicago, Illinois, USA, October 4-7, 1998, Vol. 1.

[17] S. Geman and D. Geman, Stochastic Relaxation, Gibbs Distributions and the Bayesian Restorations of Images\/}, IEEE Trans. on Pattern Analysis and Machine Intelligence, 1984, Vol. 14, No. 6, pp. 367-383.

[18] P. Moulin, J. Liu, Analysis of Multiresolution Image Denoising Schemes Using Generalized-Gaussian Priors,Proc. IEEE Sig. Proc. Symp. on Time-Frequency and Time-Scale Analysis, Pittsburgh, PA, October 1998.

[19] S. Chang, B. Yu, M. Vetterli, Spatially Adaptive Wavelet Thresholding with Content Modeling for Image Denoising, Proc. of 5th IEEE International Conference on Image Processing ICIP98, Chicago, Illinois, USA, October 4-7, 1998.

[20] S. LoPresto, K. Ramchandran, M. Orhard, Image Coding Based on Mixture Modeling of Wavelet

Coefficients and a Fast Estimation-Quantization Framework, Data Compression Conference 97, Snowbird, Utah, 1997, pp. 221-230.

[21] S. Mallat, A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, IEEE Trans. on Pattern Analysis and Machine Intelligence, 1989, Vol. 11, No. 7, pp. 674-693.

[22] J. S. Lim, Two-Dimensional Signal and Image Processing, Englewood Cliffs, NJ: Prentice-Hall, 1990.

[23] A. Blake, A. Zisserman, Visual Reconstruction, MA: The MIT Press, 1987.

[24] D. Deiger, F. Girosi, Parallel and Deterministic Algorithms from MRFs Surface Reconstruction, IEEE Trans. on Pattern Analysis and Machine Intelligence, 1991, Vol. 13, No. 6, pp. 401-412.

[25] P. Charbonnier, L. Blanc-Feraud, G. Aubert, M. Barlaud, Deterministic Edge-Preserving Regularization in Computed Images\/}, IEEE Trans. on Image Processing, 1997, Vol. 6, No. 2, pp. 298-311.

[26] A. Dalaney, Y. Bresler, Globally Convergent Edge-Preserving Regularization: An Application to Limited-Angle Tomography, IEEE Trans. on Image Processing, 1998, Vol. 7, No. 2, pp. 204-221.

[27] M. Nikolova, Estimees Locales Forment Homogenes, Comptes Rendus Ac. Sci. Paris, Serie I, 1997, Vol. 325, pp. 665-670.

[28] F. Petitcolas, R. Anderson, Weaknesses of copyright marking systems, Multimedia Security Workshop, ACM'98, Bristol, UK.

[29] S. Voloshynovskiy, A. Herrigel, N. Baumgärtner, T. Pun, A Stochastic Approach to Content Adaptive Digital Watermarking, Workshop on Information Hiding, 1999, Dresden, Germany.

Visual Optimization in Digital Image Watermarking

Wenjun Zeng

Sharp Laboratories of America
Camas, USA

zengw@sharplabs.com

ABSTRACT

One of the fundamental issues in digital watermarking is to find the best trade-off between imperceptibility and robustness to signal processing. This problem can be nicely solved by explicitly incorporating human perceptual models in the watermarking system to optimize the perceived quality of the watermarked image given a desired robustness degree. This paper discusses such a visual optimization process that exploits various properties of the human visual systems (HVS) to adaptively control the amount of watermark energy to be embedded into different transform coefficients/areas of the image. We will present several ways of incorporating properties of the HVS such as frequency sensitivity, luminance sensitivity, and contrast masking effect in a watermarking system. It will be shown that the derivation of an optimal watermark detector is more straightforward in some implementations than in others. We will also illustrate the advantage of exploiting neighborhood-masking effect in a watermarking system.

KEYWORDS

Digital watermarking, copyright protection, perceptual model, visual masking, perceptual watermarking, visual optimization, optimal watermark detection

1 Introduction

Due to the digital media revolution and the popularity of Internet commerce, the intellectual property right protection is becoming an increasingly important issue. Digital watermarking is an emerging technology that securely embeds invisible information such as ownership and copyright message into multimedia data to protect the intellectual property right of the content owners.

For most applications, two basic criteria used in evaluating a watermarking scheme are perceptual invisibility and robustness to intentional/unintentional attacks that tend to remove the watermarks. The watermarks inserted should be perceptual invisible, i.e., they should not interfere with the media content. This is the most basic but non-trivial requirement that all watermarking schemes should meet. For most applications, the watermarks should also be robust to some intentional/unintentional attacks. In other words, the watermarks should still be detectable even after common signal processing operations have been applied to the watermarked image. These two requirements, unfortunately, conflict with each other. If this issue is not carefully considered in the design of the watermarking system, two adverse effects may appear. Either the watermarked images will show visual artifacts, or it is likely the watermarks will not be detected after some common signal processing or intentional attacks are applied to the watermarked image. One of the fundamental issues in digital watermarking is thus to find the best trade-off between imperceptibility and robustness to signal processing. This problem can be nicely solved by incorporating explicit human perceptual models in the watermarking system. The perceptual models provide an upper bound on the amount of modification one can make to the content without incurring perceptual difference. Alternatively, given a robustness requirement, the visual quality of the watermarked image can be maximized. A watermarking system with such a visual optimization thus provides the maximal robustness to intentional or unintentional attacks, given a desired perceived quality.

The visual optimization process can exploit various properties of the human visual systems (HVS). For example, frequency sensitivity, local luminance sensitivity, and contrast masking effect of the HVS can be used to adaptively control the amount of watermark energy to be embedded into different transform coefficients/areas of the image. We will present several ways of incorporating properties of the HVS in a watermarking system. Some particular considerations for visual optimization for both DCT based and wavelet based watermarking systems will be discussed. We will show that the derivation of an optimal watermark detector is more straightforward in some implementations than in others. The advantage of exploiting neighborhood-masking effect will also be illustrated.

The rest of the paper is organized as follows. Section 2 summarises several properties of the HVS. Several ways of incorporating perceptual models for visual optimiza-

tion in a watermarking system are presented in Section 3. Section 4 provides some concluding remarks.

2 Properties of Human Visual Systems

Over the past three decades, there have been many efforts to develop models or metrics for image quality that incorporate properties of the HVS. These models have been used to access image visual quality [1], to help develop image compression systems that optimize the perceived quality of the compressed images [2][3][4][5][6], and recently to help design watermarking systems that ensure the imperceptibility of the embedded watermarks [7][8]. Most of the models incorporate frequency sensitivity, luminance sensitivity, and contrast masking properties of the HVS. These properties are summarized in the following.

Frequency sensitivity

Frequency sensitivity describes the human eye's sensitivity to sine wave gratings at various frequencies. It is usually described by the contrast sensitivity function (CSF) that characterizes the varying frequency sensitivity of the visual system to 2D spatial frequencies. In general, the CSF curve suggests that human eyes are less sensitive to high frequency errors than low frequency errors. Based on this model, given a fixed minimum viewing distance, it is possible to determine a static just noticeable difference (JND) for each frequency band. This static JND provides a basic visual model that depends only on viewing conditions and is independent of image content. This strategy has been widely used for Discrete Cosine Transform (DCT) and wavelet based compression systems where a quantization table tuned to the CSF curve is used for quantizing coefficients in different frequency bands. This static JND has also been used to control the amount of watermark energy to be inserted into each coefficient for a watermarking system [8][10]. The advantage of this technique, however, becomes less noticeable for lower resolution display and closer viewing distance, since the CSF curve tends to be flat under those viewing conditions.

Luminance sensitivity

Luminance sensitivity measures the effect of the detectability threshold of noise on a constant background. It typically involves a conversion to contrast, and is usually a nonlinear function of the local image characteristics. It basically suggests that the noise is more visible on a low intensity constant background than a high intensity constant background. For example, for an 8x8 block DCT based system, it can be achieved by using the DC coefficient from each DCT block to adjust the visibility in each block appropriately [2][3]. It is sometimes referred to as light adaptation process [9].

Contrast masking

Visual masking is a perceptual phenomenon where artifacts are locally masked by the image acting as a background signal. For example, in the wavelet transform domain, a larger coefficient can tolerate a larger distortion since the larger coefficient results in a large background signal that masks the visual distortion. The masking effect is the strongest when both the mask signal and the artifact are of the same spatial frequency, orientation and location. The masking effect can be roughly categorized as self-contrast masking and neighborhood masking. Self-contrast masking is referred to as the masking effect from a mask signal that is of exactly the same spatial frequency, orientation and location as the distortion signal. For example, the masking effect contributed from a coefficient is a self-contrast masking effect for a distortion signal that is introduced to that particular coefficient. Neighborhood masking is referred to as the masking effect contributed by spatially neighboring coefficients in the same band as well as in other bands. The neighborhood masking exploits the fact that more complex region can tolerate more distortion than a smooth region or a region containing a simple sharp edge.

There are some other masking effects that have been investigated in the vision science. These include, among others, noise masking that is explained by an increased variance in some internal decision variables, and entropy masking that arises when the mask is deterministic but unfamiliar [9].

3 Visual Optimization in Digital Watermarking

In this section, we will discuss how the perceptual models can be incorporated into a watermarking system. The goal here is to use the perceptual models as a guide to ensure that the watermarked image is visually optimised in the sense that, given a desired degree of robustness to signal processing, the perceived image quality is the best; preferably no artefact can be observed by ordinary users. The discussion is based on a typical class of perceptual watermarking system.

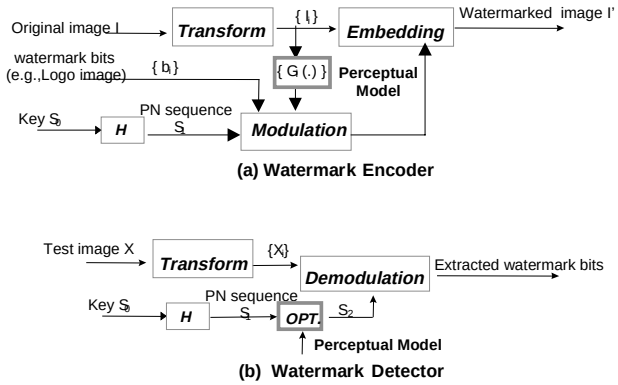


Figure 1: A general architecture for a typical class of watermarking systems.

3.1 A typical class of watermarking system

Fig. 1 shows a general architecture for a class of perceptual watermarking systems [10][11][12] At the watermark encoder, a private/public key S_0 (e.g. an owner name or ID) is first mapped, using a one-way deterministic function ***H***, to a single parameter which is then used as a seed to generate an *i.i.d.* pseudo random (PN)

sequence S_1 (or, equivalently $\{S_{1i}\}$). The nature of the key determines the security level of the system. A set of features $\{ I_i \}$ is first derived from the transformed (e.g., by DCT/Wavelet) version of the original image I. Then S_{1i} is modulated by some information bits to be embedded (e.g., a binary logo image), then multiplied by $G_i(I_i)$, where $G_i(.)$ could be a *function of* I_i and is controlled by a visual model, before being added to I_i. The encoding process can be formulated as

$$I'_i = I_i + G_i(I_i)\, b_i S_{1i} \qquad (1)$$

where b_i is the corresponding watermark bit to be embedded. Note that a single watermark bit may be spread over many feature points to increase the detection robustness.

In the watermark detector, the test image X is first transformed. Then the derived feature set $\{X_i\}$ is correlated with a pseudo random sequence S_2 which is closely related to S_1 and the perceptual model used. The correlator output q is compared to a threshold T to determine the extracted watermark bits. Choice of S_2 can be optimized to maximize the detector performance. In some scenarios where the original image is available and is allowed to use, it can be incorporated into the detector to enhance the detection performance.

For different applications of the watermark, the nature of the key S_0 and the embedded watermark information may be different. The detection mechanism and the extracted information may also have some slightly different desired properties. In this paper, we will focus on the visual optimization process of the system. Without loss of generality, we will assume that there is no other information bits except the key that is to be embedded. We will also assume that the original image is not available at the detector. Note that for some applications such as claiming rightful ownership, even though the original image may be available, it could not be used in the detection process for a valid claim [10][11].

For the scenario considered here, the detector will be applied to the test image to determine if it contains the unique watermark. In general, the secret key (e.g., a registered owner ID number) which corresponds to the watermark sequence is the mark that uniquely identifies someone who is associated with the test image. In this scenario, the correlator output q is compared to a threshold T to determine if the test image contains the watermark. Detection of the watermark is accomplished via the hypothesis testing:

H_0: $X_i = I_i + N_i$ not contain the claimed watermark

H_1: $X_i = I_i + G_i(I_i) S_{1i} + N_i$ contain the claimed watermark (2)

where N_i is noise, possibly resulted from some signal processing such as JPEG compression, etc..

Let $Y_i = X_i S_{2i}$. The correlating detector outputs the test statistic q

$$q = \frac{\sum_{i=1}^{n} Y_i}{V_y \sqrt{n}} = \frac{M_y \sqrt{n}}{V_y}$$

where n is the size of the feature set $\{X_i\}$, M_y and V_y^2 are the sample mean and sample variance of Y_i

$$V_y^2 = (\sum_{i=1}^{n} (Y_i - M_y)^2) / (n-1);$$

$$M_y = (\sum_{i=1}^{n} Y_i) / n$$

With some reasonable assumptions (including that $\{S_{2i}\}$ is zero mean and uncorrelated with the original image I), it can be shown [13] that under H_0, for large n, q is approximately a normal distribution with zero mean and unit variance, i.e., $q \sim N(0, 1)$. Let $E(.)$ denote the expectation operator. Under Hypothesis H_1 and for large n, it can also be shown that q follows a normal distribution $N(m,1)$ [10][11], where

$$m = \frac{(E(G_i(I_i) S_{1i} S_{2i}) + E(N_i S_{2i}))\sqrt{n}}{V_y}$$

By choosing a detection threshold T, one can quantify the false alarm detection probability, assuming hypothesis H_0 is true. Table 1 shows the false alarm detection probability with respect to the detection threshold T.

Threshold T	$P_{err}(q > T)$
3	0.0013
5	2.86E-7
6	9.86E-10
8	6.22E-16
10	7.62E-24
12	1.77E-33

Table 1: False alarm detection probability P_{err} for the proposed watermarking system in [10][11].

Optimal detection

It can be proved [10][11] that, if $G_i(.)$ is independent of I_i, then the choice of $S_{2i} = G_i S_{1i}$ is the optimal correlating signature which will results in the largest mean value m under H_1. On the other hand, if $G_i(.)$ is a function of I_i, and assume that $G_i(.)$ can be written as a product of two terms, i.e., $G_i(I_i) = U_i(I_i) W_i$ where W_i is independent of I_i, then a good choice of S_{2i} is $S_{1i} W_i$. It should be noted that setting S_{2i} to $G_i(I_i) S_{1i}$ is usually a very bad choice [10][11].

In Eq. (1), the amount of watermark embedded is controlled by G_i. This value has to be carefully chosen in order to guarantee imperceptibility of the watermarks. In the next three sub-sections, we will show how perceptual models can be used to determine an appropriate value of G_i.

3.2 Watermarking using JND in the DCT/Wavelet domain

One way to incorporate perceptual models in the watermarking system is to derive a JND for each DCT/Wavelet coefficient, and use this JND to control the amount of watermarks to be inserted into each coefficient. We will illustrate this general idea using the system proposed in [8]. Note that both block DCT based system and wavelet based system have been proposed in [8]. We will use the DCT based system as an example.

In the 8x8 block DCT-based perceptual watermarking scheme [8], a frequency threshold value is derived based on measurements of specific viewing conditions for each DCT basis function, which results in an image-independent 8x8 matrix of threshold values, denoted as $T_f(u,v)$, u,v=1, …, 8. One approach is to use $T_f(u,v)$ as G_i in Eq. (1). We will refer to this scheme as Scheme-1. On the other hand, one can use a more accurate perceptual model that also takes care of the luminance sensitivity and contrast masking effect of the human visual system to find the just noticeable difference for each coefficient. Luminance sensitivity is estimated as $T_l(u,v,b)=T_f(u,v)(X_{0,0,b}/X_{0,0})^a$, where $X_{0,0,b}$ is the DC coefficient for block *b*, $X_{0,0}$ is the DC coefficient corresponding to the mean luminance of the display, and *a* is a parameter which controls the degree of luminance sensitivity. A value of 0.649 is suggested for *a* in [2]. Then a contrast masking threshold, referred to as the JND, is derived as

$$T_c(u,v,b)=MAX[T_l(u,v,b), T_l(u,v,b)(|X(u,v,b)|/ T_l(u,v,b))^w],$$

where X(u,v,b) is the value of the coefficient, *w* is a number between zero and one [3]. The threshold simply implies that a larger coefficient can tolerate larger modification without incurring visual artifacts. Note that the JND here is coefficient-adaptive, unlike some others that are image-independent or region based. The JND $T_c(u,v,b)$ is then used as G_i in Eq. (1) to control the amount of watermark to be embedded into each coefficient. This perceptual watermarking scheme is referred to as IA-DCT (Image-adaptive DCT) scheme in [8]. It was shown in [11] that the IA-DCT scheme provides better performance than Scheme-1 where only frequency sensitivity is exploited.

Note that in the IA-DCT scheme, the feature set $\{I_i\}$ is the set of DCT coefficients (excluding DCs) which are larger than their corresponding $T_l(u,v,b)$. In other words, no watermark will be added to the coefficients that are too small. This generally avoids the risk of introducing high frequency noise to the watermarked image, and increases the detection performance when the watermarked image has been subject to signal processing [11].

Optimal detection

Fig. 2 shows typical distributions of the detector output *q* under different hypotheses using IA-DCT as the watermark encoding scheme. In this example, we set $T_l(u,v,b)= T_f(u,v)$, i.e., luminance sensitivity is not considered. B_i and $C_i(I_i)$, respectively, denote the T_l and $(|X(u,v,b)|/ T_l(u,v,b))^w$ components of T_c for each feature point. It can be seen that they are all normal-distribution-like. Fig. 2 also suggests that different choices of S_2 will have different detection performances. Note that the further apart the distributions under H_0 and H_1, the better performance the detector. Case 3 appears to be the best choice among the three, although it may *not* be the optimal solution. The optimal solution is not straightforward here. Our previous work in [11] does not provide the optimal solution, although it gives a good analysis for the optimal solution. For more details about the optimal detection strategy, see [11]. Note that the well formulated encoding process in Eq. (1) makes the analysis for optimal detection possible.

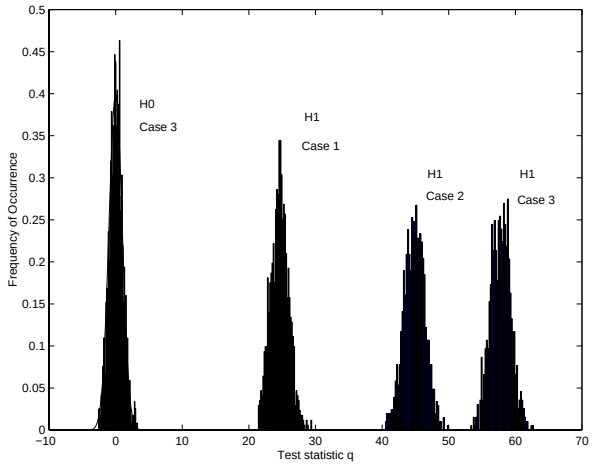


Figure 2: Distributions of output *q* for 512x512 "Lenna" image. Case 1: $S_{2i}=B_iC_i(I_i)S_{1i}$; Case 2: $S_{2i}=S_{1i}$; Case 3: $S_{2i}=B_iS_{1i}$. w=0.33.

3.3 Watermarking in a perceptually uniform domain

Another way of making use of perceptual models in a watermarking system is to first transform the image data to a domain that is perceptually uniform. In this perceptually uniform domain, each sample is perceptually no different from others. There is a common JND for all the samples, disregard its frequency, orientation, location and amplitude. As a result, this common JND can be used to control the amount of watermarks to be embedded in this domain.

Let *x* denote a DCT/wavelet transform coefficient, *f* denote a corresponding CSF value that is normalized to the range of [0,1], w^l denote the adjustment based on luminance sensitivity (corresponding to $(X_{0,0,b}/X_{0,0})^a$ in Section 3.2). Without loss of generality, assume x is non-negative. Then

$$y = x*f / w^l$$

is a CSF-and-luminance-compensated sample value. The self-contrast masking effect can usually be characterized by a power law [4], that is,

$$z = y^{\alpha} = (x*f / w^l)^{\alpha}$$

is a domain in which frequency sensitivity, luminance sensitivity and self-contrast masking effect have all been compensated. There is thus a common constant JND threshold in this domain that suggests the maximum amount of watermark energy that can be inserted to each sample value without incurring visual artifacts. To find out this JND threshold T_z, let us look at the first derivative of z with respect to x,

$$dz = f / w^l * \alpha(x * f / w^l)^{\alpha-1} dx \quad (3)$$

Recall that in the x domain, as discussed in Section 3.2, the JND for x is

$$T_c = T_f w^l (|x| / (T_f w^l))^w$$

If dx is replaced by T_c, then

$$T_z = f / w^l * \alpha(x * f / w^l)^{\alpha-1} * T_f * w^l * (x / (T_f w^l))^w$$

Let T_{min} denote the minimum value in the frequency threshold matrix. Then $f = T_{min}/T_f$. If α is chosen to be $1-w$, then

$$T_z = T_{min}/T_f * \alpha(x * T_{min}/T_f)^{\alpha-1} * T_f * (x/T_f)^w = \alpha * T_{min}^{\alpha}$$

Based on the above analysis, it is clear that the two encoding implementations, one in x domain and the other in the z domain, are equivalent to the first degree of approximation. We observe that samples in the z domain has a common JND threshold T_z. Therefore, we can use T_z to control the amount of watermark energy to be embedded into each sample, i.e., the encoding process is

$$z'_i = z_i + T_z S_{1i}$$

In this case, since T_z is a constant, the optimal choice of S_{2i} for detection is S_{1i} [10][11]. Therefore, by inserting and detecting watermarks in the z domain, the optimal detection can be derived straightforwardly, as opposed to the case in Section 3.2.

Now let us assume the embedding is performed in the x domain as described in Section 3.2. Eq. (3) also suggests that a modification of $T_c S_{1i}$ in the x domain is approximately equivalent to a modification of $T_z S_{1i}$ in the z domain. Note that in Section 3.2, although the choice of Case 3 provides very good detection performance, we are not sure if they are the optimal choice. Now we see the insertion of $T_c S_{1i}$ in the x domain is approximately equivalent to the insertion of $T_z S_{1i}$ in the z domain. The optimal detection in the z domain is thus to choose S_{2i} to be S_{1i}. In other words, although the watermarks are embedded in the x domain, we can find the approximate optimal detection in the z domain by first transforming the coefficients to the z domain, then using S_{1i} as the correlating sequence for optimal detection. Fig. 3 shows the distribution of the detector output using this strategy. Compared to Fig. 2, it is seen that Case 3 in Fig. 2 achieves performance that is very close to the optimal solution shown in Fig. 3.

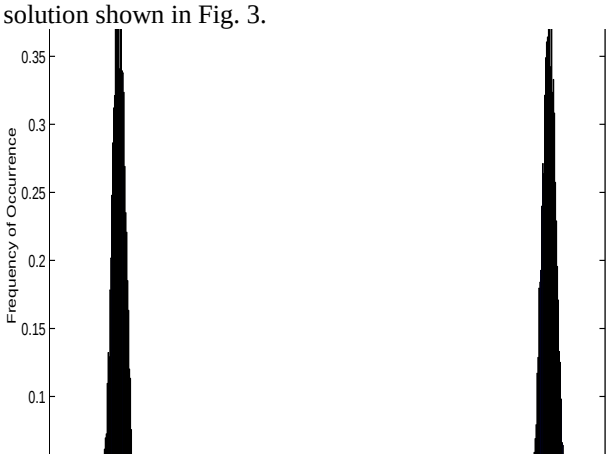


Figure 3: Distributions of output q for 512x512 "Lenna" image. The embedding is the same as in Figure 2 (in the x domain), whereas detection is performed in the z domain. α=0.67.

3.4 Exploiting neighborhood masking

The approach described in Section 3.3 exploits the self-contrast masking effect by applying a non-linear transducer function (e.g., power function) to the coefficients before watermark insertion and detection. However, there are several potential problems in this approach for wavelet or DCT based systems. The first is that it assumes the wavelet/DCT band structure and filters are a good match to the visual system's underlying channels, which is generally not true. For example, although, the wavelet structure is a much better model of the visual system than the DCT, it has a problem with the diagonal band due to the Cartesian separable approach [14]. In the visual system, frequencies at 45 degrees orientation have very little masking effect on those at –45 degrees, but the diagonal band in the wavelet has no way of distinguishing the two. This may give rise to artifacts perpendicular to the diagonal edge. Another diagonal related problem is that slanted edges cause high values in the horizontal and vertical bands, which causes high distortion for the bands at the edges, giving rise to horizontal and vertical artifacts along slanted edges. These overall problems may lead to over-masking at slanted edges.

To overcome the over-masking problem at slanted edges, other properties of the HVS have to be taken into account. One of the solutions is to exploit the masking capability of a complex region, while protecting regions with simple edge structures. More specifically, a masking weighting factor can be derived for each coefficient. The masking weighting factor can be derived based on neighborhood activities, e.g., as a function of the amplitudes of neighboring coefficients, as suggested in [5][14]. An advantage of this strategy is its ability to distinguish between large amplitude coefficients that lie in a region of simple edge structure and those in a complex region. This feature will assure the good visual quality of simple edges on a smooth background, which is often critical to the overall perceived visual quality,

especially for wavelet or DCT based watermarked images.

The principle discussed here in general can be applied to many transform based watermarking systems such as DCT, wavelet, and Cortex based systems. We will use the wavelet transform based system as an example to illustrate the main idea. We treat visual masking as a combination of two separate processes, i.e., self contrast masking and neighborhood masking [14]. The visual masking effect is therefore exploited for watermarking purpose in two steps. The first step is to apply a coefficient-wise non-linear transducer function $f(.)$ such as a power function to the original coefficient x_i, i.e., $x_i \rightarrow z_i = f(x_i)$. The approach described in Section 3.3 is a typical example. This step assumes each signal with which a coefficient is associated is lying on a common flat background. Under this assumption, $\{z_i\}$ are perceptually uniform. In a real image, however, this is usually not the case. Each signal is superimposed on other spatially neighboring signals. There is some masking effect contributed from spatially neighboring signals due to the phase uncertainty, receptive field sizes, as well as possible longer range effects [14]. To further exploit this neighborhood masking effect, the second step normalizes z_i by a masking weighting factor w_i which is a function of the amplitudes of the neighboring signals, i.e., $z_i \rightarrow p_i = z_i / w_i$, where w_i is a function $g(.)$ of the neighboring signals denoted in a vector form as $N_i(\{z_k\})$, i.e., $w_i = g(N_i(\{z_k\}))$. The neighboring coefficients could be in the same subband; they could also be coefficients around the same spatial location but in other frequency bands. As discussed above, the second step is especially important for wavelet/DCT based systems where over-masking may result from the first step.

An example is to use the non-linear transform

$$p_i = \gamma z_i / (1+\lambda \sum_{\{k \text{ near } i\}} |x_k|^{\beta} / |\phi_i|)$$

where $|\phi_i|$ denotes the size of the neighborhood, γ and λ are normalization factors, and the neighborhood contains coefficients in the same band that lie within an NxN window centered at the current coefficient. β is a positive value, and, together with N and λ, are used to control the degree of neighborhood masking. β and N play important roles in differentiating coefficients around simple edge from those in the complex area. N controls the degree of averaging; β controls the influence of the amplitude of each coefficient. Preferably β should be chosen as a value less than 1. An example value of β is 0.2. This helps to protect coefficients around simple sharp edges, since the coefficients around sharp edges usually have high values. A small value of β suppresses the contribution of large coefficients around sharp edges to the masking factor.

For watermarking purpose, now the p domain is considered perceptually uniform. A *common* constant JND T_p can be derived in this domain to control the amount of watermarks inserted into each p sample. Alternatively, the watermark can be inserted in the z domain using a JND of $T_p w_i$, or in the x domain using a JND of $sT_c w_i$, where s is a scaling factor.

Figs. 4 and 5 show the difference images between the original "Lenna" image and the watermarked images by exploiting self-contrast masking only and both self-contrast masking and neighborhood masking using a 4-level wavelet decomposition, respectively. For both schemes, the frequency sensitivity is also exploited as described in Section 3.3, and the detection is performed in the z domain using S_1 as the correlating sequence. It can be seen that there is more watermark energy in the complex hair region in Fig. 5 than in Fig. 4. On the other hand, the watermark energy embedded in some simple

Figure 4: Difference image between the original 512x512 "Lenna" image and the wavelet-based watermarked image exploiting self-contrast masking only.

Figure 5: Difference image between the original 512x512 "Lenna" image and the wavelet-based watermarked image exploiting both self-contrast and neighborhood masking.

edges (e.g., the top right corner of the image) is more conservative in Fig. 5 than in Fig. 4. In fact, some artifact can be observed at around the sharp edges in the top right corner of the watermarked image corresponding to Fig. 4, while the other watermarked image corresponding to Fig. 5 is perceptually lossless.

Table 2 summarises the detection performance of different watermarking schemes with regard to JPEG2000 compression [15]. The wavelet based scheme exploiting both self- and neigborhood masking performs slightly better than the other one that only exploits self masking effect. Both schemes, however, significantly outperform the 8x8 DCT based scheme described in Section 3.2. Interestingly, the watermarked images using the three schemes (self-masking (wavelet), both self- and neighborhood masking (wavelet) and self-masking (DCT), respectively), have PSNR of 38.5, 38.2 and 37.5 dB (with respect to the original image).

Watermarking Scheme	JPEG2000 compression (bpp)			
	No	0.5	0.25	0.125
Self-masking (wavelet)	75.2	31.7	18.9	11.6
Self+neighbor. Masking (wavelet)	74.2	32.6	20.3	12.7
Self-masking (DCT)	59.2	22.5	12.4	6.9

Table 2: Robustness of different watermarking schemes to JPEG2000 compression.

4 Conclusion

Visual optimisation is an important process in a watermarking system. This paper shows there are various properties of the HVS that can be exploited to achieve the best trade-off between the imperceptibility and robustness requirements. We discuss several different ways of incorporating perceptual models in the watermarking system. It is shown that by performing watermark insertion and detection in a perceptually uniform domain, optimal detection can be derived. The advantage of exploiting both self-contrast masking and neighborhood-masking is demonstrated. It is interesting to point out that a good compression system can also exploit the HVS properties to almost the same extent as a watermarking system can do [14]. In this regard, a watermarking system should exploit the HVS properties to the maximum extent so that the watermarks will be the most robust to the best compression schemes available.

5 Acknowledgement

The authors would like to thank Christine Podilchuk, Bede Liu, Scott Daly and Shawmin Lei for many discussions and collaboration that help to shape the work presented in this paper.

6 References

[1] S. Daly, "The visible difference predictor: an algorithm for the assessment of image fidelity," in *Digital Images and Human Vision*, A. B. Watson, Ed. Cambridge, MA: MIT Press, 1993, pp. 179-206.

[2] H. A. Peterson, A. J. Ahumada, Jr., and A. B. Watson, "Improved detection model for DCT coefficient quantization," *Proc. SPIE Conf. Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, pp.191-201, Feb. 1993.

[3] B. Watson, "DCT quantization matrices visually optimized for individual images," *Proc. SPIE Conf. Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, pp. 202-216, 1993.

[4] S. Daly, W. Zeng, J. Li, and S. Lei, "Visual masking in wavelet compression for JPEG2000", to appear in *IS&T/SPIE Conf. Image and Video Communications and Processing*, Jan. 2000.

[5] David Taubman, "High performance Scalable Image Compression with EBCOT", submitted to IEEE Transactions on Image Processing, March, 1999

[6] Watson, Yang, Solomon and Vilasenor, "Visibility of wavelet quantization noise", *IEEE Tran. Image Proc.*, vol. 6, No.8, pp. 1164-1175, 1997.

[7] M. Swanson, B. Zhu, and A. Tewfik, "Transparent robust image watermarking", In *Proc. Inter. Conf. Image Proc.*, vol. 3, pp. 211-214, Sept. 1996.

[8] Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models", *IEEE Journal on Selected Areas in Comm.*, special issue on Copyright and Privacy Protection, vol. 16, no. 4, pp. 525-539, May 1998. Partially presented *in IS&T/SPIE Electronic Imaging: Human Vision and Electronic Imaging*, vol. 3016, Feb. 1997.

[9] Watson, R. Borthwick, and M. Taylor, "Image quality and entropy masking", *Proc. SPIE Conf. Human Vision, Visual Processing, and Digital Display VI*, vol. 3016, pp.2-12, 1997.

[10] W. Zeng and B. Liu, "On resolving rightful ownerships of digital images using invisible watermarks", In *Proc. Inter. Conf. Image Proc.*, vol. 1, pp. 552-555, 1997.

[11] W. Zeng and B. Liu, "A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images," to appear in *IEEE Tran. Image Processing*, Nov. 1999.

[12] Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for images, audio and video", In *Proc. Inter. Conf. Image Proc.*, vol. 3, pp. 243-246, Sept. 1996.

[13] Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1991, pp. 269-270.

[14] W. Zeng, S. Daly and S. Lei, "Report on CE V1 (Exploitation of visual masking through control of individual block contributions)", ISO/IEC JTC1/SC29/WG 1 N1303, Vancouver, Canada, July 1999.

[15] "VM5.1 Software", ISO/IEC JTC1/SC29/WG 1 N1420, August 1999.

Preprocessed and Postprocessed Quantization Index Modulation Methods for Digital Watermarking

Brian Chen and Gregory W. Wornell

MIT
Cambridge, MA

bchen@mit.edu, gww@allegro.mit.edu

ABSTRACT

Data hiding and digital watermarking methods, which are methods for embedding one signal (the watermark) within another "host signal", have a number of multimedia applications such as copyright notification and enforcement, authentication, and covert communication. In each of these applications, the performance of an embedding method is measured in terms of its achievable trade-offs among the amount of information that can be embedded (rate), the amount of embedding-induced distortion to the host signal, and the robustness to intentional and unintentional attacks.

Quantization index modulation (QIM) methods, a class of embedding methods in which information is embedded by quantizing the host signal with a quantizer chosen from an ensemble of quantizers, have been shown to achieve very good rate-distortion-robustness trade-offs in a number of different contexts. For example, we have previously shown that for host-blind digital watermarking, where the host signal is not available during watermark decoding, QIM methods are provably better than previously proposed spread spectrum methods and low-bit modulation methods against squared-error distortion-constrained intentional attacks. Our new results show that against Gaussian noise attacks, which may be good models for some types of unintentional attacks, QIM methods exist that achieve the best possible rate-distortion-robustness trade-offs (i.e., capacity) asymptotically at high rates and achieve performance within a few dB of capacity at all finite rates. Furthermore, low-complexity realizations of QIM methods, such as so-called dither modulation, have also been shown to achieve favorable rate-distortion-robustness trade-offs.

We have developed certain types of pre- and post-processing techniques that improve the performance of QIM methods, including low-complexity realizations. One such example is so-called distortion-compensation, a post-quantization processing step in which some of the quantization error is removed. Such post-processing allows one to use coarser quantizers, which generally leads to greater robustness. However, the distortion compensation creates some interference during watermark decoding, thus offsetting part of this increased robustness. With the proper amount of distortion compensation, however, not only is the net effect on robustness positive, but also there exist distortion-compensated QIM systems that achieve capacity at all rates against Gaussian noise attacks. Experimental results also demonstrate that distortion compensation can improve performance against other types of attacks that are commonly encountered in practice.

Protocols for Watermark Verification

K. Gopalakrishnan
Department of Mathematics
East Carolina University
Greenville, NC 27858
gopal@cs.ecu.edu

Nasir Memon
Computer Science Department
Polytechnic University
Brooklyn, NY 11201
memon@poly.edu

Poorvi Vora
Imaging Technology Department
Hewlett-Packard Research
Palo Alto, CA 94304
pvora@hpl.hp.com

ABSTRACT

Digital watermarks have recently been proposed for the purpose of copy deterrence of multimedia content. Copy deterrence using digital watermarks is achieved by inserting a unique watermark into each copy of the image sold which could be used to trace unauthorized copies to the erring buyer. However, all the schemes proposed in the literature suffer from the following major limitation that is generally glossed over. The owner of the image reveals the embedded watermark to the buyer or to a "trusted" third party in the process of proving that a pirated copy originated from a specific buyer. Needless to say that the watermark could subsequently be removed from the watermarked image by the buyer or the "trusted" third party. This leads to the same fundamental problem viz., the buyer or the "trusted" third party could now sell illegal copies with complete impunity. While this limitation appears to be inherent, using appropriate cryptographic protocols can actually eliminate it. In this paper, we present a novel way of demonstrating the presence of a watermark in an image without revealing the watermark to the other party, thereby eliminating the possibility of subsequent watermark removal.

KEYWORDS

Digital Watermarking, Copyright Protection, Zero Knowledge Proofs, and Fingerprinting.

1 Introduction

Consider an application where multimedia content is electronically distributed over a network. In order to discourage unauthorised duplication and distribution, the content owner can embed a unique watermark (or a fingerprint) in each copy of the data that is distributed. If, at a later point of time, unauthorised copies of the data are found then the origin of the copy and hence the identity of the erring buyer could be determined by retrieving the unique watermark corresponding to each buyer. These types of schemes are sometimes called copy *deterrence watermarking schemes* or *digital fingerprinting schemes*. Here, we will focus our attention on digital image watermarking, although the same problems exist for other multimedia data such as video or audio.

A *watermark* is a signal added to the digital image that can later be extracted or detected to make an assertion about the image [4]. There are two types of watermarks *visible* and *invisible*. A visible watermark typically contains a conspicuously visible message or a company logo indicating the ownership of the image. On the other hand, invisible watermarks are unobtrusive modification to the image and the invisibly watermarked content appears visually very similar to the original. The existence of an invisible watermark can only be determined by using an appropriate watermark extraction or detection algorithm. Invisible watermarks are generally preferred as their unobtrusiveness makes them more desirable.

Watermarking techniques can also be classified as fragile and *robust*. Any form of image processing procedure easily corrupts fragile watermarks. Robust watermarks can resist common image manipulation procedures such as rotation, reflection, scaling, cropping, smoothing, contrast or brightness adjustment or lossy compression. Clearly, a watermark used for the purpose of copy deterrence needs to be a robust one.

Yet another classification of watermarking techniques is into *oblivious* and *non-oblivious* schemes. A non-oblivious scheme requires an original or reference image in the watermark detection procedure. On the other hand, an oblivious scheme does not require the use of an original or reference. Obviously, oblivious schemes are attractive in many applications.

In copy deterrence watermarking schemes, the watermarks used are generally invisible, robust and oblivious. Recall that copy deterrence using digital watermarks is achieved by inserting a unique watermark into each copy of the image sold which could be used to trace unauthorised copies to the erring buyer. In such a scenario, in order to indict the erring buyer, the seller has to demonstrate the presence of the unique watermark of the erring buyer on an unauthorised copy of the image and also evidence binding the specific watermark to the buyer.

The latter can be done by obtaining a digital certificate at the time of sale which, say, is in the form of the hash of the watermark, details of the terms of the sale, the identity of the buyer, all time stamped and signed by a trusted authority. To establish the former, it has been generally assumed in the literature that the seller reveals the embedded watermark to the buyer or to a "trusted" third party. Needless to say that once the watermark is revealed it could subsequently be removed from the watermarked image by the buyer (or by the "trusted" party) and now in fact the buyer (or the "trusted" party) can resell multiple copies of the image with complete impunity. While this limitation appears to be inherent, using appropriate tools from cryptography can actually eliminate it. In this paper, we present a novel way of demonstrating the presence of a watermark in an image without revealing the watermark to the other party. This prevents the adversary from removing the watermark subsequently.

The rest of the paper is organised as follows. In Section 2, we present the protocol that demonstrates the presence of a watermark without revealing it to the verifier. In Section 3, we discuss zero knowledge proofs and show that an earlier zero knowledge protocol based on the graph isomorphism problem for digital watermarking is seriously flawed. Finally, in Section 4, we provide some concluding remarks and suggest some open problems.

2 Protocol for Watermark Verification

The watermark verification protocol we propose works with linear and additive watermarking techniques where watermark detection is done by means of correlation. However, for ease of exposition, we present it in terms of the spread-spectrum watermarking technique proposed by Cox *et. al.* [1] that was demonstrated to be remarkably robust against malicious attacks aimed at its removal.

Before we present our protocol, we first briefly review this technique. Cox *et. al.* [1] embed a set of independent real numbers $W = \{w_1, w_2, \ldots w_n\}$ drawn from a zero mean, variance 1, Gaussian distribution into the n largest DCT AC coefficients of an image. Results reported using the largest 1000 AC coefficients show the technique to be remarkably robust against various image-processing operations, and also after printing and rescanning.

Specifically, they take the 2-dimensional DCT of an image X and the watermark W is inserted into the largest n AC coefficients $\{x_i, x_2, \ldots, x_n\}$ by a suitable insertion formula to yield modified coefficients $\{x'_i, x'_2, \ldots, x'_n\}$. For example, the insertion formula used could be

$$x'_1 = x_i(1 + \alpha w_i)$$

where α is a small constant.

An inverse 2D DCT is then taken, yielding the watermarked image X'. To determine if a given image Y contains the watermark W, the decoder first takes the 2-dimensional DCT of the image and extracts the largest n DCT coefficients $Y = \{y_1, y_2, \ldots y_n\}$. The confidence measure on the presence of the watermark W in Y is taken to be the correlation between W and Y. Note that this version of their technique is invisible, robust and oblivious.

Under our scenario of copy deterrence watermarking schemes using the spread-spectrum technique, the seller or the distributor inserts a unique watermark, that is distinct for each buyer, into the image before distributing it to the buyer. The seller also encrypts this watermark W using his public key of the well-known RSA public-key cryptosystem and obtains a time stamped digital certificate binding $E(W)$ to the specific buyer. At a later point of time, the seller encounters an image Y and contends that it is a pirated copy originating from a specific buyer. In order to establish it, the seller has to prove that the answer to the *watermarking decision problem* presented in Figure 1 is a resounding "yes".

Problem Instance: The digital image Y in dispute, seller's public-key and an encrypted spread-spectrum watermark $E(W)$.

Question: **Is the watermark W present in the digital image Y?**

Figure 1: The Watermarking Decision Problem

Note that the seller can of course do so, by simply disclosing the watermark W and the digital certificate that binds $E(W)$ to the buyer. The verifier can check the certificate and also check that $E(W)$ is indeed the encryption of W and then finally check that W is present in Y by using the watermark detection procedure of the spread-spectrum technique in the standard manner. But then, the verifier now knows the watermark W and hence can remove it from the image Y and then resell multiple copies of it with complete impunity. The basic problem is that the seller has lost his power of demonstrating that a disputed copy is a pirated copy the moment he has disclosed the unique watermark. However, there is no reason why the seller should prove that the answer to the watermarking decision problem is "yes" in the above manner. It is possible to prove that the answer is "yes" without revealing the watermark by using tools from cryptography.

Specifically, the seller can use the protocol presented in Figure 2 to prove that the answer to the watermarking decision problems is "yes" without revealing the watermark itself. Although most of the protocol is self explanatory, some additional clarification on step 4 of the protocol is in order. When $j=1$ the seller reveals Y' and r. The verifier checks that ε was indeed gener-

Input: The digital image Y in dispute, seller's public key and an encrypted spread-spectrum watermark $E(W)$.

Protocol: Repeat the following k times

1) The seller chooses a random number r and uses it to generate a sequence ε in "one-way manner". The seller then adds ε to Y to get an "image" $Y' = Y + \varepsilon$. The seller encrypts Y' and sends $E(Y')$ to the verifier.
2) The verifier chooses a random integer $j = 1$ or 2 and sends it to the seller.
3) If $j = 1$ the seller reveals r and as a consequence also reveals ε and Y'. The verifier checks $E(Y')$ is consistent i.e. $E(Y') = E(Y + \varepsilon)$ and ε is indeed random.
   If $j = 2$, seller demonstrates that Y' and W do correlate.
4) The verifier accepts the seller's proof if the computation of step 3 is verified in each of the k rounds.

Figure 2: A Protocol for the Watermarking Decision Problem.

ated by r in a manner that is "one-way" or difficult to invert. The verifier also checks that $E(Y') = E(Y + \varepsilon)$. This step ensures that the sequence ε added to Y is indeed random and does not correlate with W by design.

We now take a slight digression to explain the step corresponding to the case $j = 2$. Let $a = (a_1, a_2, \ldots, a_n)$ be a sequence and let $b = (b_1, b_2, \ldots, b_n)$ be another sequence. Whether these two sequences correlate or not is essentially determined by the value of the inner product

$$a \cdot b = (a_1 b_1 + a_2 b_2 + \ldots + a_n b_n).$$

If $E(a)$ and $E(b)$ are available to the verifier, then the seller could disclose the sequence

$$(a_1 b_1, a_2 b_2, \ldots, a_n b_n)$$

to the verifier. The verifier can simply add the elements of this sequence and thus determine whether or not the sequences a and b correlate. The verifier can be confident that the sequence given by the seller is not arbitrary by checking that

$$E(a_i b_i) = E(a_i) E(b_i) \quad \text{for} \quad i = 1, 2, \ldots, n.$$

This checking is possible as the verifier is in possession of both $E(a)$ and $E(b)$, has been given the plaintext values of $a_i b_i$ by the seller and the RSA cryptosystem has multiplicative homomorphic property.

Now, if $j = 2$, the seller discloses the sequence

$$(y'_1 w_1, y'_2 w_2, \ldots, y'_n w_n)$$

to the verifier. The verifier can then check the legitimacy of the sequence given to him, using the seller's public key, as he is already in possession of both $E(Y')$ and $E(W)$. He can then add up the elements of this sequence and use the result to check that Y' and W indeed do correlate.

As the verifier knows that Y' is derived from Y by insertion of ε, that Y' correlates with W and the random sequence ε does not correlate with W. These facts allow him to conclude that Y must correlate with W and therefore the pirated copy must have originated from the specific buyer.

Note that in each round of the protocol, the seller only proves one of two statements viz., Y' correlates with W or Y' was constructed as agreed upon. However, as he does not know which one of these two statements he will be asked to prove before he commits to $E(Y')$, he cannot choose ε by malicious design.

Several technicalities need to be addressed now. These include a formal proof that the protocol does not reveal absolutely any information about the watermark other than what could be gained from the input without participating in the protocol (that is, the protocol constitutes a zero knowledge proof as explained in the next section), the "random" nature of ε, a precise estimate of the confidence with which the verifier will be convinced that the image in dispute actually originated from the alleged buyer, and finer details of implementation using finite field arithmetic. We defer addressing these technicalities to the full version of this paper.

3 Zero Knowledge Proofs and Digital Watermarks

The protocol presented in the previous section closely follows a well-known tool in cryptography, called *zero knowledge proofs*. A refreshingly non-mathematical introduction to zero knowledge proofs is provided in [8]. Zero knowledge proof systems is an active area in cryptography and a formal and detailed introduction to it can be found in the texts [7, Ch. 13] and [6, Ch. 10]. Informally, a zero knowledge proof system allows one person, Peggy, to convince another person, Vic, of some fact without revealing any information about the proof. At the end of the protocol, Vic is "completely convinced" of the same fact, but does not gain any additional knowledge whatsoever.

In [3] Kinoshita Hirotsugu attempted to use the zero knowledge interactive proofs to assert ownership rights on an image. He used the well known zero knowledge interactive proof for the graph isomorphism problem presented in [2]. In this section, we briefly review his scheme and show that it is fundamentally flawed.

Kinoshita's scheme works in the spatial domain of the image. Essentially, he generates a graph with say n nodes, called the region graph G_r, from the most significant bits of the pixels of the image in a fixed manner. He then applies a permutation σ on n points to the region graph G_r to obtain an isomorphic graph called the concealed graph G_c. He then conceals the graph G_c in the least significant bits of the pixels. To assert ownership rights over the image, it is suggested that one could extract the region graph from the most significant bits and the concealed graph from the least significant bits and then the owner could demonstrate that these two graphs are isomorphic to each other without revealing the permutation σ using the zero knowledge interactive proof.

While the zero knowledge interactive proof for the graph isomorphism problem presented in [2] is perfect, the way it is used here is fundamentally flawed. The first problem is that this watermarking scheme is not robust. The concealed graph is encoded into the least significant bits of the pixels. The adversary can always modify the least significant bits thus preventing the real owner from proving his ownership of the image. More importantly, the adversary can construct the region graph G_r from the most significant bits of the pixels in exactly the same manner as the owner can. The adversary can then apply a permutation ρ, known only to him, to the region graph G_r and obtain an isomorphic graph $G_{c'}$. He can then embed $G_{c'}$ into the least significant bits of the image and can claim that the image actually belongs to him. Moreover he can prove so by using the very same zero knowledge interactive proof for graph isomorphism.

The above example shows that one has to be very careful in applying the subtle concept of zero knowledge interactive proofs to practical problems. The question that naturally arises is whether the protocol we have presented is zero knowledge. We believe this is true and hope to present a formal proof to this effect in the full version of this paper.

4 Concluding Remarks

In copy deterrence watermarking schemes, it is important to be able to demonstrate the presence of a watermark in an image without revealing the watermark. In this paper, we developed a novel way of demonstrating the presence of a watermark in an image without revealing information about the watermark that could lead to the possibility of the adversary removing the watermark and re-selling multiple copies of the image with impunity.

For the sake of brevity, we exclusively focused on the problem of demonstrating the presence of a watermark in an image without revealing it. Some other aspects of copy deterrence watermarking schemes such as preventing the ability of a malicious seller to frame the buyer are discussed in [5]. Indeed, the protocol presented here could be coupled with the buyer-seller protocol presented in [5] to form a more comprehensive solution to the problem of copy deterrence.

In addition to copy deterrence applications, the fundamental problem that we have pointed to in this paper also applies to watermarking for ownership assertion. Here, current techniques assume that the watermark needs to be revealed in order to assert ownership. This problem could also be addressed by a similar protocol. We hope to report on such a protocol in future work.

5 Acknowledgement

K. Gopalakrishnan was supported in part by the ECU Faculty Senate through a Research/Creative Activity Grant.

6 References

[1] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. Tech. Rep. 95-10, NEC Research Institute 1995.

[2] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems, *Journal of the Association for Computing Machinery*, 38, 691-729, 1991.

[3] Hirotsugu Kinoshita. An Image Digital Signature System with ZKIP for the Graph Isomorphism Problem, *Proceedings of the IEEE International Conference on Image Processing – ICIP'96*, VIII, 247-250.

[4] Nasir Memon and Ping Wah Wong. Protecting Digital Media Content: Watermarks for Copyrighting and Authentication, *Communications of the Association for Computing Machinery*, 35-42, July 1998.

[5] Nasir Memon and Ping Wah Wong. A Buyer-Seller Watermarking Protocol, *Proceedings of the Second Workshop on Multimedia Signal Processing – MMSP'98*.

[6] A.J. Menezes, P.C. van Oorschot and S.A. Vanstone. *Handbook of Applied Cryptography*, CRC Press, (1997).

[7] D. R. Stinson, *Cryptography – Theory and Practice*, CRC Press, (1995).

[8] J. J. Quisquater, L. Guillo and T. Berson, How to explain zero knowledge protocols to your children, *Advances in Cryptology – CRYPTO'89, Lecture Notes in Computer Science*, 435, 628-631.

Two High Capacity Methods for Embedding Public Watermarks into 3D Polygonal Models

Oliver Benedens

Department Security Technology for
Graphic and Communication Systems
Fraunhofer Institute for Computer Graphics
Darmstadt, Germany

benedens@igd.fhg.de

ABSTRACT

Two methods are presented which allow to embed public readable watermarks into three dimensional models consisting entirely of triangles. The algorithms alter only the geometry and they don't require the mesh to have special properties, most noticeably they don't require it to be 2-manifold. The mesh of the three dimensional model does not need to be closed or orientable. For one of the algorithms even no connectivity of triangular faces is required.

KEYWORDS

Public watermarks, 3D polygonal models, Labeling, Copyright protection.

1 Motivation

Current research on watermarking of 3D models focuses the topics object verification [3] and robust embedding of secret watermarks for the purpose of proving ownership and tracing of illegal copies. Methods for robust embedding strive to survive frequently applied geometric applications, e.g. affine transformations [4][5] or polygon reduction [1][2][6]. In this paper we propose two algorithms that complement a system for embedding secret robust watermarks by possibly providing the recipient of a 3D model with information about

- the place (web-site) where the material was fetched from or further related material could be downloaded
- license conditions
- additional information regarding the object itself: recommended textures and colors, information regarding object classification and context

All these information is stored in the model data and is readable by everyone, just functioning as a label. Compared to the alternative of adding the information in model-file specific sections, the information is more likely to survive everyday 3D file format conversions. In [4][5] several methods applicable for the purpose of embedding public readable watermarks with high capacity are presented. In comparison to these we try to lower the requirements for meshes that can be watermarked for being able to handle a wide range of triangular mesh data.

The algorithms described in this paper, namely the so called *vertex flood*- and *triangle flood* algorithm, do not rely on

- the mesh being a 2-manifold,
- the correctness of normals, the ability to determine in- and outward-normals,
- degeneracy-free meshes (can handle degenaracies such as three triangle edge points on a line or self intersecting faces).

In addition the proposed vertex flood *algorithm does not rely on topology at all*, in particular it does not require the triangular faces to be connected.

Further design issues were concerned with:

- achieving high capacity,
- scaleability: handle large meshes at applicable speeds (embedding, scanning and retrieval of watermarks),
- minimum impact of alterations with respect to visual and „constructive" quality (no modification of connectivity, no introduction of new degeneracy's).

2 Characteristics Shared by Both Algorithms

Both algorithms allow for embedding of one or several bit strings into a Model *M*. Let $V = \{v_1, .., v_n\}$, $v_i \in \Re^3, (1 \le i \le n)$ be the vertex set of the model, $F = \{f_1, .., f_m\}$ the triangle face set, which consists of triples $f_i = (f_{i_1}, f_{i_2}, f_{i_3})$ pointing at vertices $v_{f_{i_1}}, v_{f_{i_3}}, v_{f_{i_3}}$. Each bit string is prepended with a *leadin*- and appended with a *leadout*- sequence signaling the start and end of embedded data.

2.1 Selection of a Start Triangle for Embedding

In both algorithms one or several start triangles are chosen at which the embedding of information starts. In the retrieval process speed requirements may demand for fast and effective narrowing the candidates of possible start triangles.

In principle there are two ways for accomplishing this:

- choose a start triangle with topological properties shared only by a small subset of all faces, for example demand that a certain number of edges is incident to each of the vertices,
- choose a start triangle based on geometry properties, for example demand a certain triangle edge length ratio.

2.2 Embedding Multiple Watermarks

Both algorithms allow for embedding of multiple watermarks. However to make sure the watermarks don't interfere, the positions of previously embedded watermarks have to be known (we get the watermark positions simply by retrieving them). Both algorithms discussed can restrict their embedding to a local region, so for multiple watermarks simply non-overlapping regions could be chosen.

Due to point randomization's or CUT-operations, the proposed algorithms might get out of synchronization in retrieving process, that means part of retrieved bits may be correct, but it's position in the global bit string is unknown. For increasing robustness, the global bit string can be split into several smaller ones that are embedded as single watermarks including an index value determining the position of the fragment in the global bit string.

To increase robustness against individual loss of watermark bits, a global bit string may be embedded redundantly, possibly each time as a single independent watermark.

3 The Vertex Flood Algorithm

Basically the vertex flood algorithm modifies vertices, so their distance to the center of mass of a designated start triangle encode the watermark bits. Since the algorithm operates solely on vertices and does not take topological relationships into account, it does not require connectivity of the faces in the input mesh.

Let *com* be the center of mass of the start triangle with edge-points $S = \{s_1, s_2, s_3\}$. Next we populate the sets

$$M_k = \left\{ v \in V \setminus S \mid k \le \left\lfloor \frac{|v - com|}{W} \right\rfloor < k+1 \right\}$$

for $0 \le k \le N = \left\lfloor \frac{d_{MAX}}{W} \right\rfloor$.

d_{MAX} is the maximum allowed distance of a vertex *v* from *com* to be considered for embedding watermark bits. The width of each interval associated with a set is *W*.

Next we loop through the sets M_0 to M_N, skipping empty sets. The points of each set are modified in order to embed $m = 2^n$ bits. Each set covers an interval of length *W*. This interval is subdivided as follows:

$$\underbrace{\text{buf} \quad I_0 \quad \cdots \quad I_{m-1} \quad buf}_{\text{m+2 sub intervals, the buf named ones with length } W \cdot c_1, \text{ other intervals of of length } (W(1-2 \cdot c_1))/m}$$

(with a constant $0 < 2 \cdot c_1 < 1$).

For embedding the value *val* ($0 \le val \le m-1$), the distance of each vertex in the set to *com* is modified so it comes to lie in the middle of the sub-interval I_{val}. The vertex is moved in the direction *com-vertex*.

The purpose of the two intervals named with *buf* is to prevent modifications of vertices in one interval from causing effects on the embedded values in other intervals.

The last set M_N is not used for embedding information. Instead all the vertices assigned to it are moved so their distances come to lie in the middle of I_0. If a vertex distance is already above that value, the vertex is left untouched. Using this scheme, the minimum distance of two vertices in two subsequent sets is at least $W \cdot 2 \cdot c_1$.

In the retrieval process, the *mean* distance of all vertices contained in an interval can used for decoding the embedded bits or the sub-interval containing most of the vertices could be chosen (*majority voting*).

This way the robustness with respect to randomization of single vertices is increased.

The values d_{MAX} and *W* are derived from the start triangle. See section 6 for how these values were computed in experiments.

4 The Triangle Flood Algorithm

Basically the triangle flood algorithm generates a *unique* traversal path of triangles, starting from a designated start triangle. Vertices of triangles along the path are modified for two purposes: First to encode watermark bits in the height information, second to order the triangles in a specific way, yielding a *unique* traversal path.

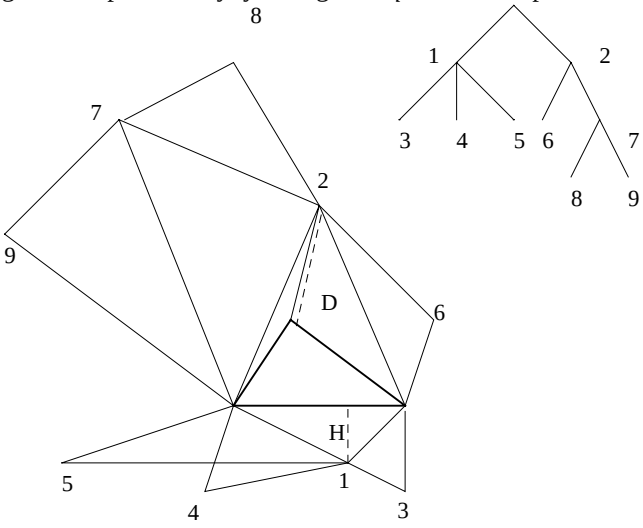


Figure 1: Generating a unique traversal path and embedding watermark bits through triangle height (H) or distance (D) modifications. The tree yields the path when traversed applying *breadth first search* (BFS).

The algorithm is now explained using the example shown in Figure 1. In this illustration all vertices (possibly) moved for the purpose of embedding watermark information and generating a unique traversal path are numbered.

4.1 Generating the Unique Traverse Path

Beginning with the start triangle (drawn with thick edges) all triangles edge-adjacent to it are collected. Next these triangles are sorted based on the distance of the edge shared and the vertex unshared with the start triangle (triangle height).

Denote the unshared vertices with $v_1,..,v_n$, the associated triangle heights with $h_1,..,h_n$ and the ordered sequence of heights (indices) with $I=\{i_1,..,i_n\}$.

We now require neighbored heights to have certain minimum distance d_{MIN}:

$$\forall k \in \{1,..,n-1\}: \left|h_{i_{k+1}} - h_{i_k}\right| \geq d_{MIN}.$$

The calculation of distance d_{MIN} is explained in section 4.2. We start from the middle $m=\lfloor n/2 \rfloor$ of the ordered sequence and proceed to both ends. If the distance for a height pair $h_{i\,k+1}$ and h_{i_k} is less than d_{MIN} and $k \geq m$, the vertex $v_{h_{i_{k+1}}}$ is modified to increase the triangle height to d_{MIN}. The vertex is moved along the line perpendicular to the base edge of the triangle and going through the vertex. If $k<m$, the vertex $v_{h_{i_k}}$ is modified. The heights are updated as we proceed through the sequence.

If any of the height/distance values happens to fall below d_{MIN} we use the following simple strategy: Move the i-th vertex in the sequence so it's height equals $i \cdot d_{MIN}$ $(1 \leq i \leq n)$.

The traverse path can be visualized using a tree structure. Each node in this tree represents a triangle. The root node is the start triangle. A node being a child of a parent node means the child node triangle is "visited by stepping over an edge" of the parent node triangle.

The sequence $v_1,..,v_n$ is now added to the tree as n child nodes of the parent node (from left to right).

In Figure 1 we encounter a frequent case that requires different treatment. The start triangle is edge adjacent to three triangles, but two of them have the same *unshared* vertex (the vertex numbered with 2).

We handle this case by treating these two triangles as one and measure the distance of the *unshared* vertex to the vertex of the start triangle adjacent to both triangles rather than the triangle height. The adjustment of a height is marked with "H", the distance adjustment with "D". The root node of the tree in Figure 1 therefore does only contain two child nodes: 1 and 2.

The algorithm continues by using the child nodes as start triangles for the next iteration, proceeding from left to right. A branch of the tree is discontinued if there are no more *usable* triangles edge adjacent to the triangles represented by the child node. This is the case when

- all the unshared vertices of edge adjacent triangles are already visited
- all unshared vertices have distance $\geq d_{MAX}$ to the center of mass the start triangle in the root-node

Vertices with distance close to d_{MAX} are handled the following way: If their distance d is $\geq d_{MAX} - d_{MIN}$ it is increased to $d_{MAX} + d_{MIN}$ and the vertices are not taken into consideration for embedding watermark bits.

When retrieving a watermark, traversing stops when

- the depth of the tree exceeds a maximum depth of $depth_{MAX}$
- a *leadout*-sequence (preceding the embedded bit string) is retrieved
- the first retrieved nodes don't contain a *leadin*-sequence} that was pre-pended to the bit string in embedding process
- all branches are discontinued

The algorithm can handle meshes that are not 2-manifold as is the case for the mesh in Figure 1 (faces with edge-points numbered with 5 and 4 share an edge with a third face).

4.2 Embedding Watermark Bits

After generating the unique traverse path, the vertices along the path are modified in order to embed the watermark bits. Some of these vertices have been modified before for the purpose of generating this path. We take care of not destroying the path while embedding watermark bits in the following way:

The distances of neighbored triangle heights in the ordered sequence are at least d_{MIN}. We describe now the process for embedding bits and how to calculate the value d_{MIN} to be used in unique generating traversal path.

Denote the number of bits to be coded per one height (distance) adjustment $m=2^n$. Subdivide the interval of possible height/distance values into subintervals each of length *w*, starting from 0. Name the interval *i* with *i mod m*.

To embed the value $val \in \{0,,,m-1\}$, the height/distance is adjusted by moving the unshared vertex so the value comes to lie in the middle of the nearest sub-interval labeled with *val*.

Because the maximum change height/distance change is $W \cdot \frac{m}{2}$, we choose $d_{MIN} = W \cdot (1+c_2) \cdot m$ $(c_2 \geq 1)$.

The values *W* and d_{MAX} are derived from the start triangle. See section 6 for how these values were computed in experiments.

5 Robustness

While the watermarks embedded by applying the algorithms described survive uniform scaling, rotation and translation, they are inherently not robust against complex geometry altering operations e.g. shearing, non-uniform scaling and topology altering operations, e.g. polygon-reduction. Further investigation of both algorithms with respect to their numerical stability is subject of future work.

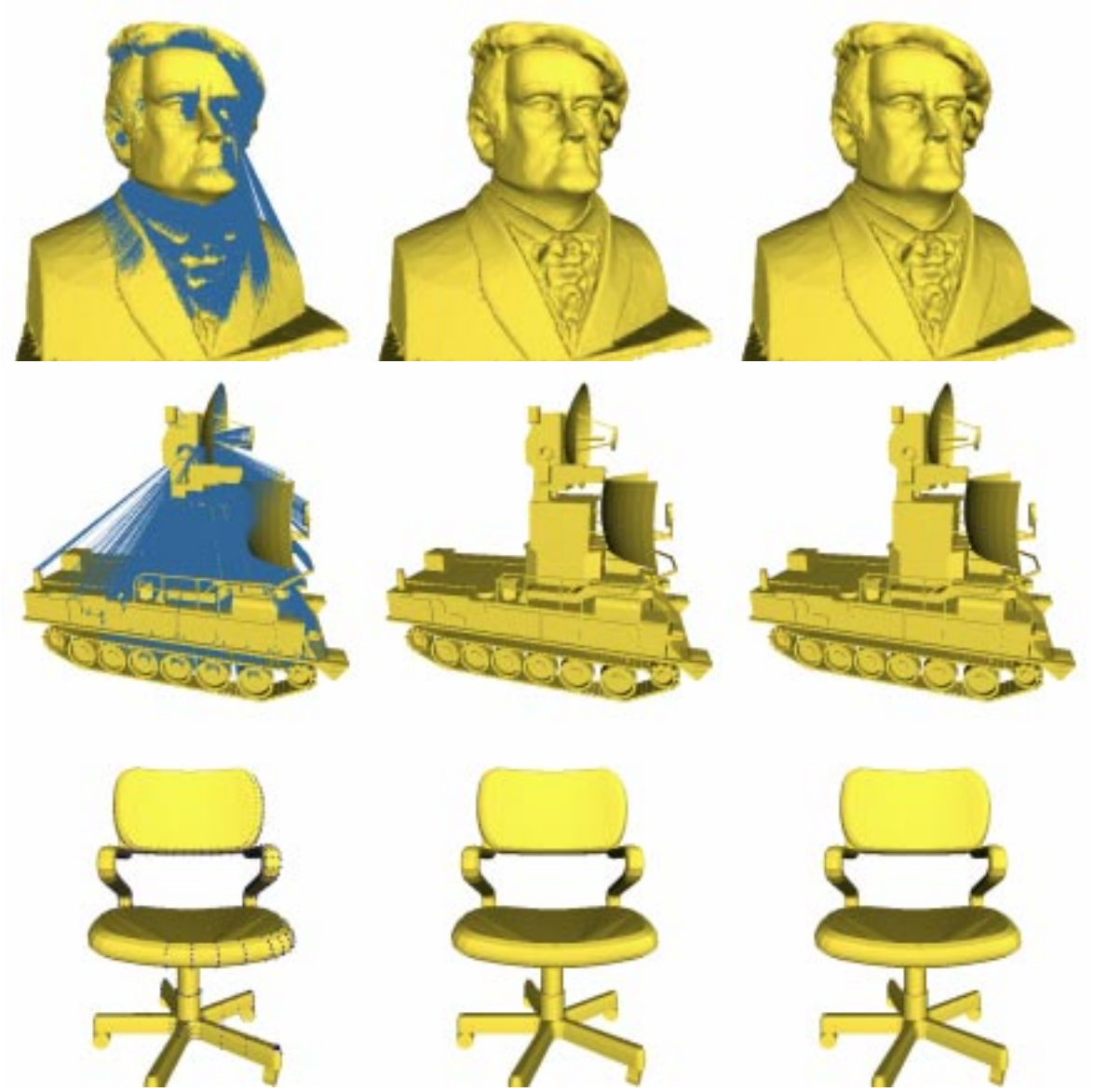


Figure 2: Applying the vertex flood algorithm to three different models (test cases 1-3). The left image shows the watermarked model. In case of the first two models a line is drawn from every vertex contributing to the watermark to center of mass of the start triangle. For the third model (chair) these vertices are marked with spheres. The middle images show the watermarked models without highlighting the watermark. The right image shows the original model.

6 Practical results

We applied the algorithms to following models:

No	Model	#vertices	#faces
1	wagner bust	30215	60426
2	tank	8659	18062
3	galleon	2372	4968
4	chair	1960	3832

The coordinates of models in model files was restricted to floating point precision (giving roughly a 6 decimal digit mantissa).

In all test cases the algorithm shared following parameters:

One single watermark was embedded. The watermark string was appended with an 8 bit leadin- and 8 bit leadout-sequence.

The number of embedded bits per vertex adjustment was 2,

the interval width W and the search range d_{MAX} were computed as follows:

$W = 0.01 \cdot mel$, $d_{MAX} = 1000 \cdot mel$

with $mel = \frac{1}{3}(e_1 + e_2 + e_3)$ and e_1, e_2, e_3 being the edge lengths of the start triangle. We used start-triangles close to a certain edge length ratio $1 : R_1 : R_2$ (we used 1:1.5:2 and 1:2:2) and close to the 25% quantil of ordered sequence of averaged edge lengths of all triangles. We used two C++ header files of the algorithms implementation as watermarks. The file sizes were 342 and 107 bytes. Each character in these files was embedded as one byte.

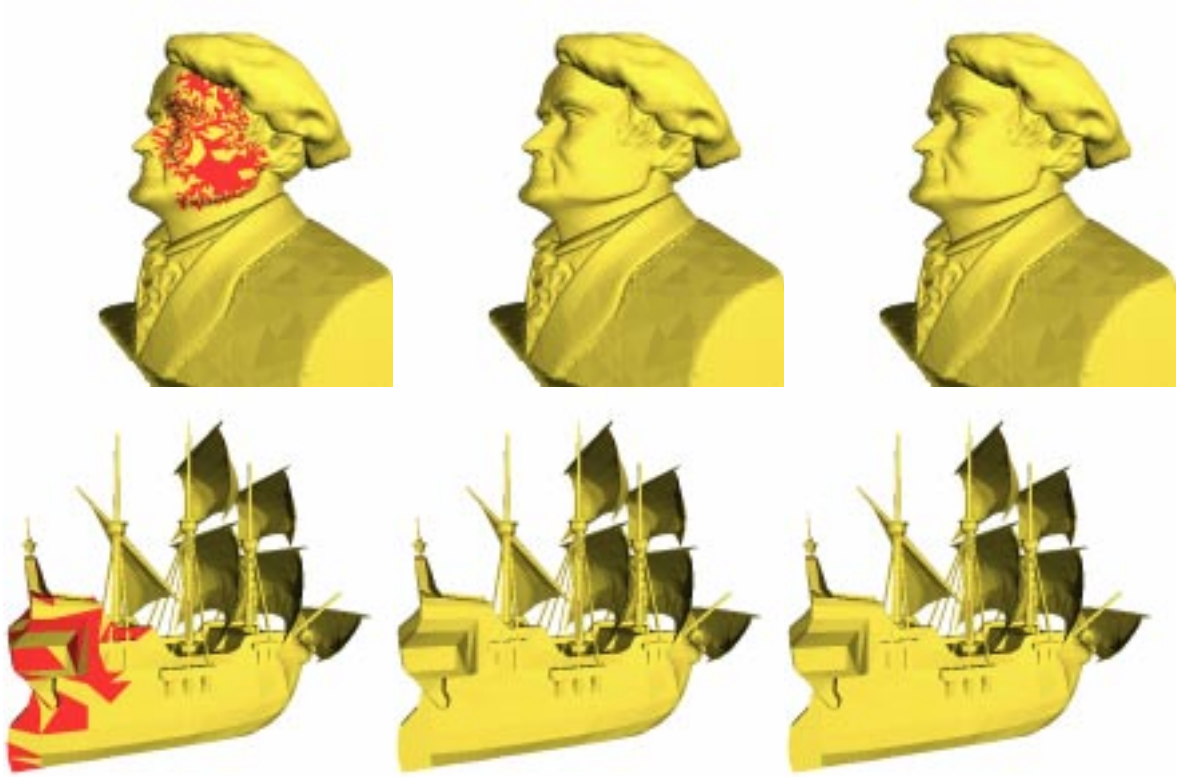


Figure 3: Applying the triangle flood algorithm to different models (test cases 4+5). The left images highlights the watermark by marking the faces with adjusted triangle heights red. Middle images show watermarked version without highlighting, the right images show the original model.

Test case	Watermark	Configuration	Remarks
1	342 Byte	model 1, vertex-flood	2752 bits embedded, 4478 were possible, 8628 vertices changed
2	342 Byte	model 2, vertex-flood	2752 bits embedded, 3570 were possible, 5077 vertices changed
3	107 Byte	model 3, vertex-flood	872 bits embedded, 1290 were possible, 1319 vertices changed
4	342 Byte	model 1, triangle-flood	2752 bits embedded, unique path contains 1376 triangles
5	107 Byte	model 4, triangle-flood	872 bits embedded, unique path contains 436 triangles

7 Conclusion

Both algorithms are suitable for embedding public watermarks with high capacity. The capacity can be further increased by raising the number of embedded bits per vertex adjustment. However due to the limited precision of floating point numbers representing vertex coordinates because of fixed mantissa length, the „applicable" interval length is lower bounded.

It should be stressed that the vertex algorithm can be applied to the general case of watermarking n-dimensional point clouds. Instead of deriving interval width and search range from a start triangle edge lengths, these values could be derived from e.g. the minimum distance of a designated start point to one of it's neighbors (of course these points are not allowed to be modified in the embedding process).

8 References

[1] Benedens, O.: Watermarking of 3D polygon based models with robustness against mesh simplification, Proceedings of SPIE: Security and Watermarking of Multimedia Contents, SPIE, 1999, 329-340.

[2] Benedens, O.: Geometry-Based Watermarking of 3D Models, IEEE Computer Graphics, Special Issue on Image Security, January/February 1999, 46-55.

[3] Boon-Lock, Y., Minerva, M.: Watermarking 3D Objects for Verification, IEEE Computer Graphics, Special Issue on Image Security, January/February 1999, 36-45.

[4] Ohbuchi, R., Masuda, H., and Aono, M.: Watermarking Three-Dimensional Polygonal Model", ACM Multimedia 97, ACM Press, 1997, 261-272.

[5] Ohbuchi, R., Masuda, H., and Aono, M.: Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications, IEEE Journal on selected areas in communications Volume 16, Number 4 (May 1998), 551-559.

[6] Praun, E., Hoppe, H., and Finkelstein, A.: Robust Mesh Watermarking, SIGGRAPH Proceedings, 1999, 69-76.

Content-Based Graph Authentication

Hong Heather Yu

Panasonic Information and Networking Technology Lab

heathery@research.panasonic.com

ABSTRACT

Document authentication techniques are used to ensure the integrity of a document, i.e., that the document has not been tampered with and that it originated with the presumed transmitter. Techniques for digital document, such as digital color/gray scale image and plain text, authentication using digital watermark have been studied by many researchers in the last ten years. Nowadays, more and more documents are using graphs in addition to images and text for system and idea illustration. In contrast to image, graph is more difficult to watermark based on adding noise. This is due to the binary nature of graph. In this paper, we propose two methodologies for binary graph authentication, one at object level and one at pixel level. Our goal is to achieve the authentication capability that alteration of the original document can be detected as well as localized. Both schemes utilize cryptographic hash function.

1 Introduction

The challenges of ensuring the confidentiality and integrity of messages have been long compelled the intellect. In recent years, advances in communication, networking, and multimedia information access technologies are making protection of digital copy of multimedia content a more and more important research area. Specifically, the objective of content-based authentication with digital watermarking is to identify the authenticity of digital media, to protect information from unauthorized alteration while encryption can be used to prevent information from unauthorized disclosure.

Various schemes of fragile and robust digital watermarking for image[1,2,3] and plain text[4,5,6,7] as well as audio[7,8] and video[9,10] have been proposed. However, little has been done on the protection of graph, an important part of information in documents. In this paper, we focus on *text documents*, i.e., writings that provide information, especially information of an official or original nature, which can be printed on paper. In specific, we present two new schemes for authentication of graph in text document. Roughly speaking, a document can consist plain text, image, and graph. We define a *graph* as a diagram symbolizing a system or illustrating a statement. In particular, we refer to only those graphs with binary pixel values (1bit/pixel) in the following discussion. The methodology is however applicable to those graphs with multiple-bits/pixel.

Let's define I to be the original document which will be authenticated by owner *O1* or owners *O1, O2… On* in the case of a contract. Denote the authenticated copy of document I as $\tilde{I}$. In correspondence, define G and $\tilde{G}$ to be the original and the authenticated copy of a graph respectively. Also denote R as an authorized receiver, whereas A is an attacker, i.e., unauthorized receiver. The following scenarios illustrate the application value and the objective of our work.

- $I_1 \in O1$, *O1* wants to check her document I_1 is authentic. The content of the document is genuine, especially the sensitive information, such as $1,000 or by June 01, 1999.
- $I_1 \in O1$, *O1* may need to send I_1 to R and may wish to grant R 'read' permission but not 'write' permission.
  - Or *O1* may want to prevent alteration of any kind and to localize the alteration made by an attacker A who gets I_1 from *O1* and then sends it to R.
  - Or *O1* may want everyone to be able to read I_1 while only herself and R can make modification on the document.
- $I_1 \in O1 \cap O2$, i.e., I_1 might be a contract between *O1* and *O2*. If the copy in *O1's* hand is different from that of in *O2*'s, *O1* wants to prove that *O2*'s copy is a tampered copy of the original contract by checking the authenticity of *O2*'s copy. In addition, *O1* may want to point out where exactly *O2* altered the original contract.

In this paper, we shall mainly focus on the discussion of graph authentication. Unless otherwise specified, we concentrate on the $I_1 = G_1$ scenario.

1.1 General Framework

Text and graph are often referred to as binary images as well. The binary nature makes it particularly hard to insert any watermark due to low capacity of perceptual invisible noise. Previous proposed methodologies for content-based text authentication mainly rely on altering the word/line spacing or the length of strokes. These methodologies might be useful for certain applications of plain text authentication, it, however, can hardly be

extended to authentication of graph even though they share the same binary nature. Graph often does not observe the same characteristics as text even on the pixel level. For instance, in a flow diagram, the shape of each

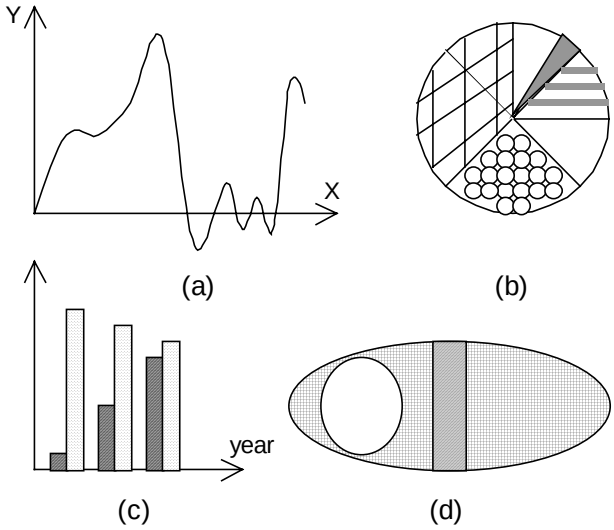


Fig 1 Other graphs

node may be very important whereas it often has fewer characters compare to a text paragraph. The number of objects that observe a vertical serif can be as low as several percent (of the total number of objects). Here, an object is referred to a line, a character or a curve. In the mean time, other kinds of graphs may not observe line spacing or vertical serif at all (see Figure 1).

On the other hand, the critical information of a graph (or text) is often contained in the object level rather than the pixel level. For instance, a useful application for document copy and copyright protection is to provide different level of access rights to different users while achieving the authentication capability that alteration of the original document can be detected as well as localized. In a graph, such as a procedure illustration shown in Figure2, the important information contained in the

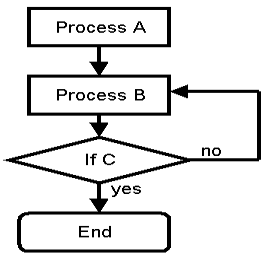


Fig2. A simple flow diagram

graph is the annotation of each node and the connections between nodes that illustrate the relationship of nodes. Whether the drawing of each box is a bit smaller or larger, the length of a line is longer or shorter, or the position of a node is a bit tilted to left or right is generally not as important. Consequently, the desired target of authentication is mainly on the flow level instead of pixel level. (It is worth to point out here that the above observation is generally true for plain text as well, namely, the sensitive information is contained in the character rather than the pixel. That is to say, the font or size of the text seldom can alter the content of it. Hence, in most cases, the desired target of authentication for text is on character level. In other words, character may be defined as the object of content (the physical entity) for marking and protection of text documents.)

Therefore, if we simply copy the previous proposed text authentication scheme, such as stroke-length-based scheme, or modify it to a corresponding scheme in graph, such as a line-length-based scheme, it merely can fulfill our authentication goal. New techniques for authentication of graph in text document are needed.

This paper presents new mechanisms, on both object lever and pixel level, that are suitable for authenticity check of binary graph presented in text documents. By building a bridge from graph to text on the character level, it allows authentication of graph using suitable text document authentication algorithm. When pixel level precision of a graph is required, a pixel level authentication can be added. This layer lets the owner detect as well as localize the change of graph on the pixel level. The general framework of a one-party owned system is illustrated in Figure 3. The hierarchical layout yields the application of several aforementioned scenarios. The first hierarchy is the pixel level authentication followed by an object level authentication hierarchy. These are done with owner *O1*'s private key. Notice here, either the pixel level or the object level protection is optional depending on the application. For ultimate protection, however, a dual-layer protection with a pixel protection layer plus an object protection layer is recommended since the two layers are orthogonal. Next, a meaningful watermark, such as a company logo, can be inserted, if desirable, back to the document. At last, the authenticated documents, text plus graph can be encrypted with public key encryption algorithm for secure transmission. Here the watermarking layer can be done either before or after the authentication layer. This again, depends on different applications. Access authorization can then be granted by distributing different keys to different users. For example, in the case of 'read' only, *R* will be given the public decryption key K_4 only. In the case of a multi-party owned document authentication, each party has a private key, the authentication is done by generating a key set with the private

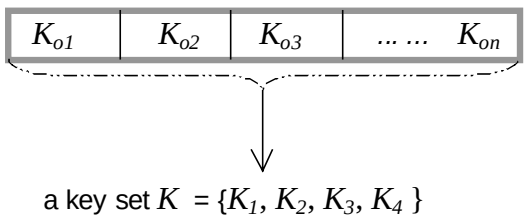


Fig 4 Illustration of a key set

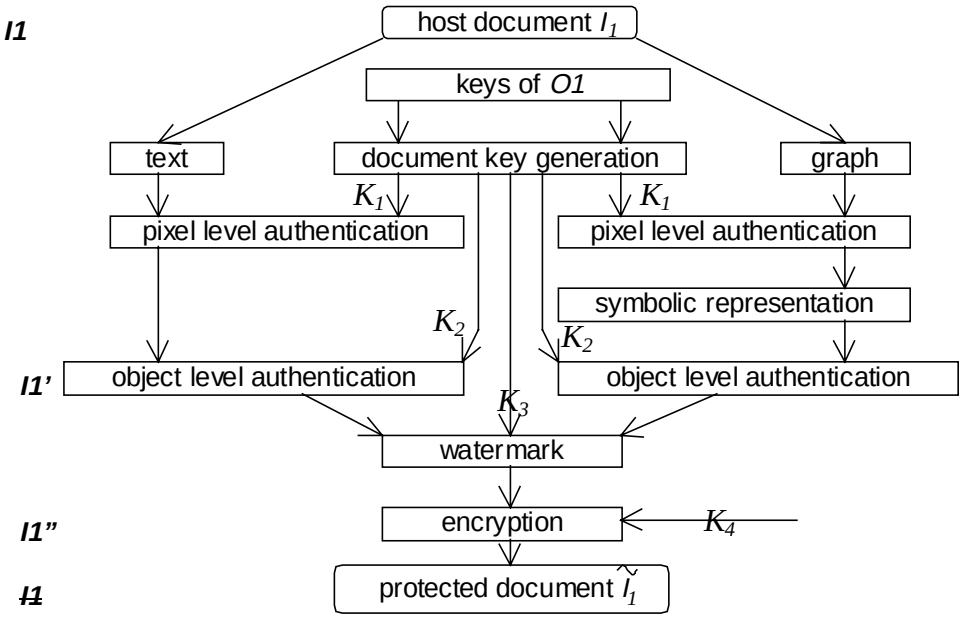


Fig 3 A general frame work for one-party owned document authentication

key from every party (see Figure 4). Modification of document with absence of any key, i.e., any party, will result in an unsuccessful tampering.[11]

Furthermore, we use context-dependent one way hash in the system, as suggested in [3]. This provides addition gain in robustness as opposed to earlier approaches.

1.2 Paper organization

The problems we need to solve include first how we can design a simple scheme for authentication of graph presented in text document, in another word, a graph in the context of plain text. And secondly, how can we authenticate a graph of 1bit/pixel without noticeable altering of the visual content and yet any meaningful (or say damaging) alteration can be easily localized?

In next section, we propose an intelligent symbolic representation language for authentication of graph which allows us to use suitable text document authentication algorithms to authenticate the graph together with the text in context. Visible and invisible pixel level authentication schemes are proposed in Section 3. At last, we discuss dual layer and coalescing authentication along with a brief summary. Flow diagram is used as an example throughout the following discussions.

2 Symbolic representation of graph

The idea comes from the following observation. Text and graph, although sharing the same binary nature, have different object level representations. Graph often contains lines and curves in addition to text characters, hence can not be directly authenticated with the same algorithm as was used for text authentication. If, however, we can create a bridge from graph to text, such as symbolic representation of graph, then graph authentication might be realized with proper text authentication algorithms and graph might therefore be authenticated along with the text in context. When the sensitivity of graph is on object level, any graph can be represented precisely with a series of relationship and specification symbols, which specify the exact characteristics and the

Relationship symbols	
< >	A tuple
∩	and
∪	or
≠	not
→	parent → child
↔	sibling relation
⇔	twin relation
←	child ← parent
>	contain relation
|	condition
.	.
.	.
.	.
:	unconnected
Specification symbols	
&	size
#	shape
@	position
◎	color

Table 1

exact relationship of nodes, along with the node annotation. Let's use Figure2, a typical system procedure diagram, as an example to illustrate the idea. It consists of nodes and lines. For illustration purpose, the annotation of each node is simplified. This diagram can be represented with a series of relationship symbols along with the node annotations as following[12]:

"<N_1{'Process A', #1, &reg, @mid} → N_2{'Process B', #1, &reg, @mid}→ N_3{'If C', #3, &reg, @mid}→< N_4{'End', #2, &reg, @mid}|yes; N_2|no>> **Fig2** A simple flow diagram."

In the above symbolic representation, N_1 N_2… are node names with the property of each node contained in {}, < > is a tuple, and → and | are relationship symbols as de-

fined in Table 1. The properties of nodes and lines, the shape, size, color, and position, can be described with the specification symbols. For those specification insensitive graphs, the symbols between each pair of {} can be simply ignored whereas in specification sensitive graphs, the specification symbols in each pair of {} give different level of details. This hierarchical representation provides additional flexibility.

After symbolization, a multimedia document (with text and graphs, see Figure 5(a)) is transferred to a symbolized (text-based, see Figure 5(b)) document which can be authenticated with suitable plain text authentication algorithm.

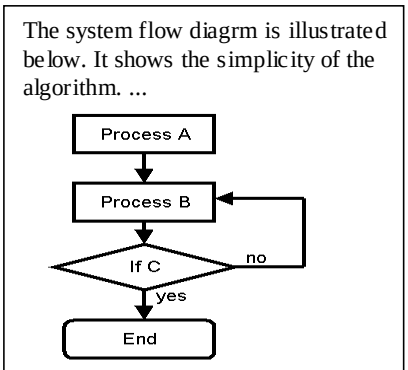


Fig5. A sample document paragraph (a) and its corresponding symbolized version (b)

For example, we can use a two- or multi-dimensional checksum to verify the authenticity. Let $T(p,q)$ represent the (p,q)th character. $S(p,q) = s^1(p,q)\ s^2(p,q)\ \cdots s^J(p,q) = f(T(p,q))$ is coded representation of $T(p,q)$ via map f. $s^1(p,q)\ s^2(p,q)\ \cdots s^J(p,q)$ represent the first, the second, … and the Jth bit of $S(p,q)$ that are in the order of the most significant bit to the least significant one. Let $Sum_p^j = \sum_{p=1}^{P} s^j(p,q)$ and $Sum_q^j = \sum_{q=1}^{Q} s^j(p,q)$, where P & Q are dimensional sizes. Thus, the position (p,q) of any alteration $Sum_p' \neq Sum_p$, $Sum_q' \neq Sum_q$ can be localized.

Of course, utilizing content-dependent one way hash function shall give a higher level of security. We can use the methodology suggested in [3]. Let B denotes the block size and K denotes a private key. In the case of a multi-party document, K is a function of K_{O1}, K_{O2}, …, i.e., $K = f1(K_{O1}, K_{O2}, \ldots)$. Assume it is a Jbits coding with the 1^{st} to $(J-1)^{th}$ bit the code bits and the lowest bit, J^{th} bit, the verification bit. Figure 6 shows a teximage of the document paragraph I shown in Figure 5. It uses 9bits coding. We choose the one way hash algorithm MD5. The encoding procedure is as following. Pad the source text I an exact multiple of 512 in length. For each 128-length set, I_o, choose its neighborhood set, $\underline{I_o}$=512 characters with $I_o \subset \underline{I_o}$. Assume

$S_o = \{S_o(i), i \in [1,128]\} = \{s^1{}_o(i)\ s^2{}_o(i) \cdots s^J{}_o(i)\} = f(I_o)$

and

$\underline{S_o} = \{\underline{S_o}(i), i \in [1,512]\} = \{\underline{s^1{}_o}(i)\ \underline{s^2{}_o}(i) \cdots \underline{s^J{}_o}(i)\} = f(\underline{I_o})$

are coded representation of I_o and $\underline{I_o}$ respectively.

1. Concatenate the code bits of the neighborhood set $\underline{I_o}$,
2. calculate the 128bits hash value of it, $h_o = H(\underline{S_o})$,
3. generate message $h_o' = Sgn(K, h_o)$ by signing h_o with public cryptography method, and
4. put h_o' into the J^{th} bit, the lowest bit, of $S_o(i)$, i.e., let $s^J{}_o(i) = h_o'(i)$, $i \in [1,128]$.

The decoding process is similar to the encoding process with the verification done through an XOR operation.

$$Auth_o(i) = \tilde{h}_o'(i) \oplus s^J{}_o(i)$$

If $Auth_o(i)==1$ for $\forall i \in [1,128]$, the I_o set has been altered.

Fig 6 The corresponding Teximage of that in Figure 5(b)

3 Pixel Level Authentication

For position sensitive or pixel level precision critical graph, object level algorithms can not fulfill the goal. In this section, we present two schemes for pixel level graph authentication.

Let $XxY=128$ be the defined block size. Cut G into XxY blocks. Assume the number of blocks is L,

① we concatenate the bits of the (x,y)th pixel of every block to the 1^{st} block and form a Lbits truncated image $TrunG$. Therefore, a Lbits/pixel image $TrunG$, with image size XxY, of graph G is generated. Let $TrunG(x,y)^l$ denotes the l^{th} bit of pixel (x,y) of $TrunG$. Notice here, it is desirable to form the truncated image $TrunG$ in a way that $TrunG(x,y) \neq 0$. Also note that to get a higher level of protection, a random number generator should be used to cut the graph.

Method I

② Collect all bits of all XxY pixels into a $XxYxL$ bits message $M1$. Pad $M1$ into an exact multiple of 512 in length with as many 0s as needed and get message $M1'$.

③ Compute the 128bits hash value of it using MD5, $M2=h(i)=H(M1')$.

④ Signing $M2$ with public key cryptograph method and generate $M3=h'(i)=Sgn(K, M2)$.

⑤ Generate a bounding box Gb of G with $Gb(i)=h'(i)$. Figure 7 illustrates the idea of it.

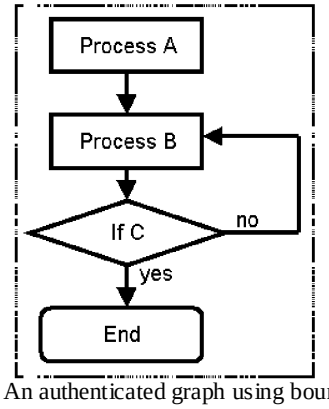


Fig 7 An authenticated graph using bounding box

This scheme is especially suitable for pixel level graph authentication because an added bounding box will not interfere with the graph appearance. (Note that in the case of text, an added bounding box can hardly be acceptable due to the change in appearance.) In addition, by enlarging the width of the box and repeating the 128bits a multiple times along the bounding box, it can easily survive distortion caused by common signal processing, even photocopy or fax processing.

An alternative way is to use a 1D or 2D bar code instead of a bounding box for authentication as shown in the figure below.

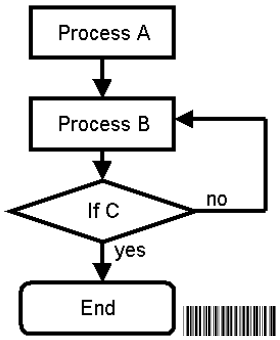


Fig 8 An authenticated graph using barcode

When invisible authentication is required or desirable, a less robust scheme that modifies the graph itself can be used.

Method II

② Pick 1bit $TrunG(x,y)^l$ =1 out of the *L*bits of each pixel (*x,y*) in *TrunG* to be the verification bit. For better imperceptibility and a higher lever of security, it should be picked in a way that it is as much spread out as possible.

③ Collect the rest (*L-1*)bits of all *XxY* pixels into a *XxYx(L-1)* bits message *M1*. Pad *M1* into an exact multiple of 512 in length with as many 0s as needed and get message *M1'*.

④ Compute the 128bits hash value of it using MD5, *M2=h(i)*=H(*M1'*).

⑤ Signing *M2* with public key cryptograph method and generate *M3=h'(i)*=Sgn(*K*, *M2*).

⑥ Embed *M3* into *TrunG* in the following fashion.

- If $h'(i)=h'((y-1)*X+x))=0$ and $|TrunG(x,y)|=$ odd, let $TrunG(x,y)^l=0$.
- If $h'(i)=h'((y-1)*X+x))=1$ and $|TrunG(x,y)|=$ even, let $TrunG(x,y)^l=1$.

Where $|TrunG(x,y)|$ denotes the cardinality of *TrunG(x,y)*, i.e., the number of bits that are '1's among the *L*bit of *TrunG(x,y)*.

Decoding can be similarly done.

It is easy to see that the advantage of the second scheme is the invisibility while its disadvantage lies in its low robustness.

For both methods, we, however, prefer a coalescing method as described in Section 4.2. Instead of using the pixel level value to generate the hash value, it uses the object level (character) value of *G* to generate the hash value and embeds the hash value into the pixels as discussed above.

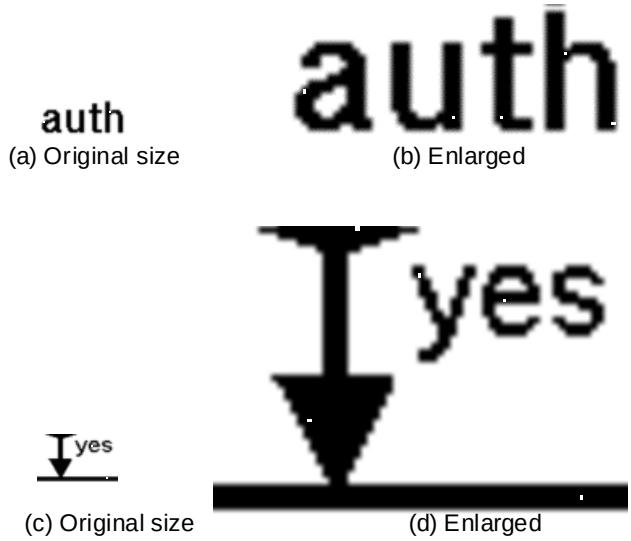


Fig 9 Sample results

Figure 9 illustrates two sample results. The lower one is cropped from our sample diagram in Figure 2. To give a better view of the results, each is enlarged to ≥400% of its original size.

4 Discussion and Summary

So far, we have used flow diagram as an example to demonstrate how content-based graph authentication can be done in both object level and pixel level. Other graphs, such as those shown in Figure 1, can be authenticated using the same methodology. Note that different scheme has different robustness level, should be chosen based on different application requirement.

In the following, we shall give a brief summary of the methodology proposed in this paper.

4.1 Dual-layer Protection

Object level authentication has a better survivability with regards to various kinds of processing noise. Even if the font, paragraph layout, or page layout is changed, the authenticity of the content may still be verifiable. That is, object level authentication is more robust com-

pare to pixel level authentication. However, object level algorithm is powerless when pixel level precision of a graph is required or when transparency is required, yet new coding is not applicable. Since the proposed object level signature is orthogonal to the pixel level signature, a more plausible scheme is to authenticate the graph with a pixel layer followed by an object layer.

4.2 Coalescing

Coalescing is different from dual-layer. To authenticate *I* with *N* symbols, for example, to authenticate the symbolized paragraph in Figue 5(b) which has *N*=248 characters, we compute the one way hash of *I* on the character level first. This is done by putting all the bits of the 248 characters together, pad it to an exact multiple of 512 in length, and calculate the hash value of it. Then, we embed the 128bits hash value on the pixel level as described in Section 3.

4.3 Embed Watermark

Fragile watermark can be inserted by XORing the watermark bits with the verification bits. For instance, suppose we want to embed "PTIPINTL" to the graph. First, encode "PTIPINTL" to a binary code *W*. Let *W(i)* be the *i*th bit and embed $h(i) \oplus W(i)$ instead of *h(i)*.

Of course, choosing an orthogonal feature to embed the watermark can also be done and is desirable in lots of the cases.

4.4 Color and Gray-scale Graph

The above proposed algorithms can be extended to authenticate multi-bits/pixel graph. For the object level algorithm, all it takes is an additional set of symbols. For the pixel level algorithm, however, a simple way is to extend one bit representation to multi-bit representation.

It is important to point out here that in the case of invisible marking, when the problem is extended from binary to multiple bits, the data hiding capacity of the host graph is subsequently increased. Consequently, the complexity of the problem is relatively reduced.

4.5 A Comparison Study

We compared our algorithms for authentication of graph with space-shifting method and serif-modification method. Table 2 summarizes the comparison results.

Notice that when we prepend the hash value to the document, special coding is not needed for object level authentication; otherwise, it is needed. Similarly, in the case of pixel level (or coalesced) authentication, special coding is not needed with Method I while it is needed for Method II. Here, special coding means a new code other than commonly accepted codes, such as ASCII code and Unicode.

4.6 Summary

In this paper, we proposed methodologies for content-based authentication of graph, both at object level and pixel level. We also went one step further by combining the object layer with the pixel layer authentication. This is motivated by the application objective of achieving the authentication capability that alteration of the original document can be detected as well as localized. Details of the algorithms are given. Preliminary experiment results show that the proposed methodologies are effective.

5 References

[1] A Secure, Robust Watermark for Multimedia,IJ Cox, J Kilian, T Leighton, T Shamoon, *Info Hiding 96* pp 185-206.

[2] A review of watermarking and the importance of perceptual modeling, Ingemar J. Cox and Matt L. Miller, *Human Vision and Electrtonic Imaging II, SPIE 3016*, San Jose, CA, USA, February 1997 pp 92-99.

[3] Fragile imperceptible digital watermark with privacy control,, C. W. Wu, D. Coppersmith, F. C. Mintzer, C. P. Tresser, IBM Thomas J. Watson Research Ctr.; M. M. Yeung, *Electronic Imaging '99. Security and watermarking of multimedia content, SPIE 3657*, Jan, 1999.

[4] Electronic Marking and Identification Techniques to Discourage Document Copying, J Brassil, S Low, N Maxemchuk, L O'Garman, *IEEE Infocom 94* pp 1278-1287.

[5] Hiding Information in Documents Images, J. Brassil and S. Low and N.F. Maxemchuk and L. O'Gorman, *Conference on Information Sciences and Systems (CISS-95)*, March 1995.

[6] Marking and detection of text documents using transform-domain techniques, Y. Liu, J. Mant, E. Wong, S. Low, *Electronic Imaging '99. Security and watermarking of multimedia content, SPIE 3657*, Jan, 1999.

[7] Techniques for Data Hiding, W Bender, D Gruhl, N Morimoto, A Lu, *IBM Systems Journal* v 35 no 3--4 (96) pp 313-336.

[8] Digital Watermarks for Audio Signals, L Boney, AH Tewfik, KN Hamdy, *IEEE International Conference on Multimedia Computing and Systems*, Hiroshima, Japan 17--23 June 1996 pp 473-480.

[9] Robust MPEG Video WatermarkingTechnologies, J Dittmann, M Stabenau, R Steinmetz, *Multimedia 98* pp 71-80.

[10] Issues and solutions for authenticating MPEG video, C. Y. Lin, S. F. Chang, , *Electronic Imaging '99. Security and watermarking of multimedia content, SPIE 3657*, Jan, 1999.

[12] Document authentication, H. Yu, to submit.

[13] Intelligent system for symbolic representation of graph, H. Yu, to submit.

Table 2: Comparison result for graph

	W/o content-dependent one way hash		Our algorithms, w/ content-dependent one way hash		
(Text)	Traditional line spacing	Traditional serif length	Coalescing	Object level	Duel level with coalesc-ing
Special coding	Needed	Needed	May or may not needed	May or may not needed	May or may not needed
Imperceptibility	Good	Good	OK	Good	Good
Detectability	Bad	Bad	OK	Good	Good
Pixel-level detectability	Bad	Bad	Good if Method I OK if Method II	Can't detect	OK
Localization-ability	Bad	Some bad. Some OK	OK	Good	Good
Copy and print	Bad	Bad	Good if Method I, bad if Method II	Good	Good
Noise resistance-ability	Bad	OK	Good if Method I, bad if Method II	Good	Good
Robustness to scaling	Good	OK	OK if Method I, bad if Method II	Good	Good

** Detectability measures the correct detection rate when certain number of objects are altered in the document and the processing noise otherwise is zero. Localization-ability measures the ratio of localized alterations. The measures are mostly done on object level and alterations are mostly done on character level for text and sub-node/line level for graph unless otherwise specified.*

Active Data Hiding for Secure Electronic Media Distribution

Hong Heather Yu, Alex Gelman, Robert Fish

Panasonic Information and Networking Technology Lab

heathery@research.panasonic.com

ABSTRACT

Electronic media distribution imposes high demand on content protection mechanisms for secure distribution of media. Imperceptible data hiding for copy control and copyright protection is thus becoming a more and more attractive research area in recent years. Many questions, though, still need to be answered for efficient and effective application of such technique to more real-world scenarios. This paper proposes active data hiding for electronic media distribution. Compare to the traditional passive data hiding, active data hiding can improve renewability and interoperability, provide additional application values, and higher level of security to the multimedia content. In this paper, we will address the application value of such approach and propose a novel scheme for imperceptible active data hiding into audio signal. The system decoding is done without the presence of original host signal to ensure the suitability of the proposed scheme for digital packaged and/or networked media distribution.

KEYWORDS

Content-based authentication, media security, digital watermarking, data hiding, music distribution.

1 Introduction

1.1 The Age of Electronic Media Distribution

Advances in multimedia, communication, and networking are making electronic media distribution (will be referred to as EMD here after) possible. Average users are starting to access and will soon be looking forward to purchasing multimedia content through the Internet. This urges the development of secure content distribution technologies with which content owners will agree to electronic distribution of digital media such as video and audio. The problem is amplified by the fact that the digital copy technology such as DVD-R, DVD-RW, CD-R, and CD-RW are widely available. Thus, starting from mid 90's, more and more attention is attracted to the research area of data hiding for media security. Though, most of the research works have been concentrated on passive data hiding (such as digital watermarking, please see definition in Section 3) [1,2,3,4,5,6,7,8,9] for copyright protection or copy control. Passive data, as the name itself shows, can only act upon request. They can not perform any task actively by itself, neither renewal nor feedback, while renewal, such as key renewal, is an important way to fight again piracy and various attacks. With passive data hiding, those types of functionality can only be achieved through additional techniques/functions built into the players. This greatly limits the application domain and the renewability of the system when additional functions are not available to the players. In this paper, we propose a fresh new approach, active data hiding (see definition in Section 3), for secure e-distribution of multimedia content. A fragile active data hiding system that can put a JAVA executable imperceptibly into the base domain with lossless extractability is presented. Extraction is conducted without the presence of the host signal. For ease of presentation, we will concentrate on introducing such a scheme for audio data hiding. But the methodology can be easily extended to other media such as video, image, and text.

1.2 Paper organization

We introduce the concept, pose the problem, and address the application value of active data hiding in the next section. In the third section, a fragile active data hiding system is presented. Experiment result will be given along with a brief summary and future work direction in the last section.

2 Active data hiding

2.1 Data hiding

Here, we define *data hiding* as imposing a meaningful and extractable but perceptually invisible/inaudible signal onto a host signal such as audio, image, or video for various applications. The hidden information can be extracted and used for information retrieval, copyright protection, and other purposes. In this paper, we focus on data hiding techniques for secure EMD related applications, such as copyright protection, copy control, and fraud tracking (see Figure 1).

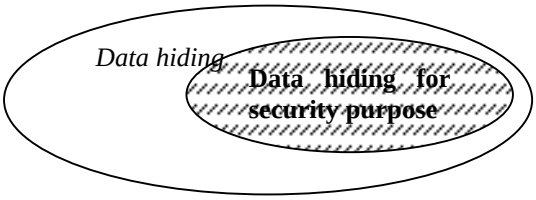


Figure 1: Data hiding

2.2 Active data hiding

Active data hiding is a subset of data hiding (see Figure 2). It is to hide an active data stream such as an applet or an executable file into the host signal. The active data stream can act as agent/agents to push (push model vs. pull model) the distribution of active information, to upgrade certain functionality/capability, or to force the renewal of access control, such as playback or record control, information.

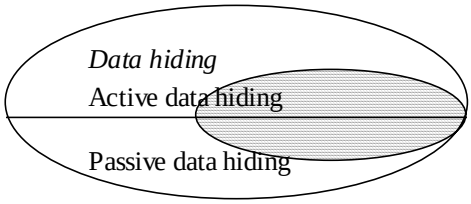


Figure 2: Active data hiding vs. passive data hiding

Passive data hiding is a complement of active data hiding set. It is to hide a passive data stream, such as a text stream or a random bit stream 1001110…, which can not perform any task actively by itself but acted upon, into the host signal.

In contrast to traditional passive data hiding, active data hiding introduces new applications for EMD. For example,

Scenario I. An agent sends feedback information to the server when streaming or online preview is performed.

Scenario II. An agent updates the copy control information hidden in the host signal/content.

Scenario III. An agent carries key renewal information and performs key update or refreshment.

Scenario IV. An agent shows relevant information to the user when requested.

Scenario V. An agent carries updated commercial information.

Scenario VI. An agent scrambles the host content when authenticity check is failed or when it is found to be an illegitimate copy of the content.

and so on so forth.

2.3 Active data hiding system design

2.3.1 Criteria

Active data hiding draws attraction to itself with the application values. Besides the basic imperceptibility and extractability requirement, however, it bears another set of technical requirement that yields higher level of difficulty of the problem. First of all, the size of an applet or an executable usually is at least several hundred bytes. This requires techniques for imperceptible high bit rate embedding instead of low bit rate embedding as is in the case of traditional passive data hiding. The difficulty of the problem thus rises. This is easy to imagine. For a fixed size host signal, the more data to hide into it, the more difficult it will be to satisfy the imperceptibility (see Figure 3(a)). Second of all, due to the sensitivity of error in executable files, the extracted hidden data set has to be virtually errorless, i.e., decoding has to be lossless. This puts an additional difficulty onto the problem.

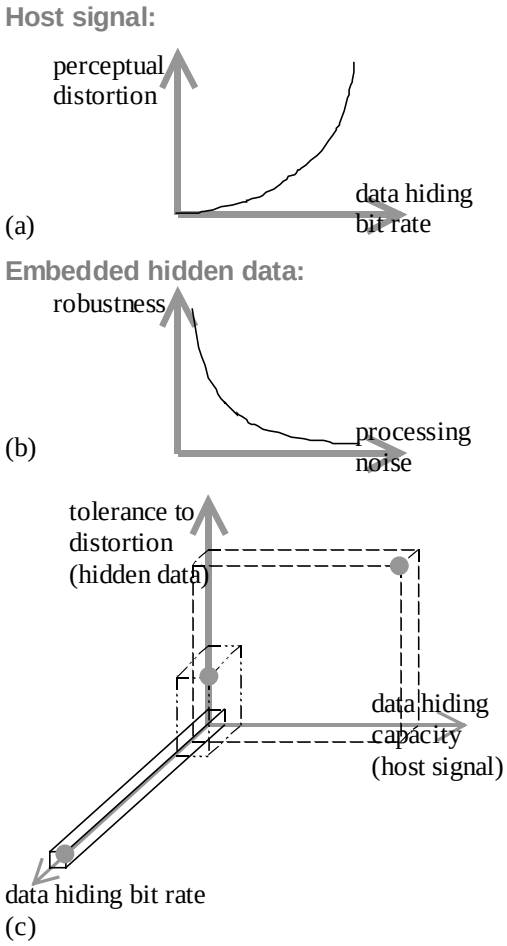


Figure 3: Illustration of the relationships between data hiding bit rate and distortion in host signal domain (a) and in embedded hidden data domain (b)&(c)

Figure 3 illustrates the general relationships between data hiding bit rate and perceptual distortion in the host signal domain, and between processing noise and robustness plus between data hiding capacity/data hiding bit rate and the hidden data's capability on tolerating distortion in the embedded hidden data domain. Taking Figure 3 © as an example, the hidden data's capability to tolerate distortion is generally proportional to the host signal's data hiding capacity and inverse proportional to the data hiding bit rate. For a fixed host signal, its data hiding capacity is fixed. Hence, by simply rely on the

one pass methodology for passive data hiding, i.e., embed only the primary hidden data (one pass embedding, see definition of primary hidden data in Section 3) into the host signal, we can hardly reach our goal: lossless and high bit rate embedding for active data hiding. This is especially true for audio signal due to its low data hiding capacity compare to visual media. Therefore, for active data hiding, not only do we need to explore as much data hiding capacity of the host signal as possible, but also is it desirable for us to add additional techniques and protection layers in order to satisfy the criteria of active data hiding. That is to ensure lossless extractability, one solution is to add additional protection layers, for instance, error correction bits to ensure low probability of decoding error. Notice here, adding an error correction layer requires additional hidden bits. Therefore, the design of a scheme that can well utilize the data hiding capacity of host signal is desirable.

2.3.2 For EMD applications

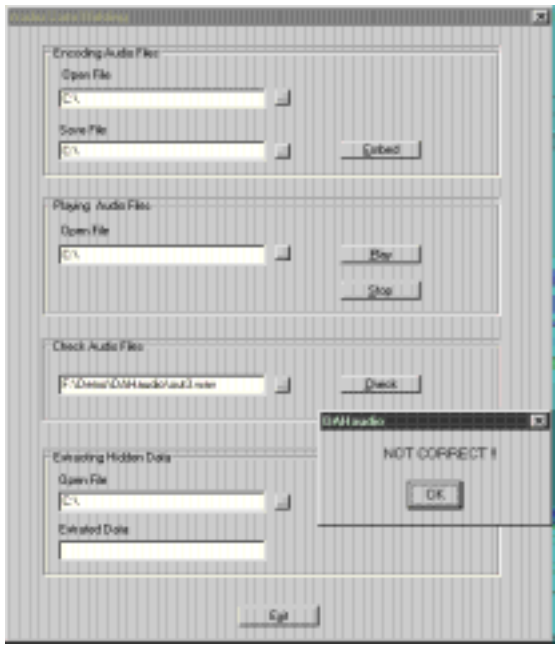
Let's consider a simple application scenario: A customer downloads a song from an Internet music database, writes the digital song in flesh memory, and then plays the song in a portable music player. Since only the protected medium, the song, is available to the portable player, the extraction of any hidden data, i.e., the decoding process, has to be performed without the original host medium. Hence, the design of a scheme that can perform blind detection of the hidden data is required.

In Section 3, we present a multi-layer, primary (or called ruling) hidden data layer plus secondary (or named governing) hidden data layer, data hiding scheme. The methodology can be used to satisfy the lossless and high bit rate criteria of active data hiding. The designed algorithm can also satisfy the blind detection criteria for EMD use. Such a system is implemented in DAHaudio package in our lab. In the next subsection, Section 2.4, we briefly introduce our DAHaudio system design and capabilities.

2.4 DAHaudio system

DAHaudio is an audio DAta Hiding package developed at PINTL (Panasonic Information and Networking Technologies Laboratory). It has active data hiding as well as passive data hiding capability and corresponding functionalities. Figure 4 shows the DAHaudio interfaces.

(a) DAHaudio



(b) Fragile data hiding input-output interface

Figure 4: DAHaudio interfaces

The above shown system can perform embedding, playback (preview/pay-n-play), authenticity check, and decoding (agent extraction/execution) interactively. Figure 5 illustrates the running result of a sample agent, a Java program which has approximately 500bytes (≈4000bits). When runs, it would show a window on the player screen with a commercially meaningful text. For ease of presentation, we will use this Java program $\mathcal{A}$ as the embedded agent throughout the paper.

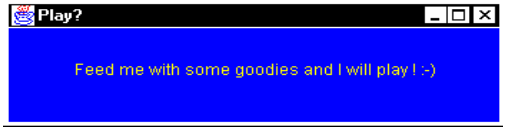


Figure 5: A Java agent

DAHaudio is targeted at secure networked and packaged audio distribution. In this paper, however, we only focus on the introduction of its fragile active data hiding functionality and system methodology. For methodologies on robust active data hiding with DAHaudio, please refer to our other papers.

3 Multi-layer data hiding

3.1 Terminology

Host data (*host signal*) is defined as the original data to be protected.

Embedded data designates a modified version of the host signal that has secondary meaningful data embedded into it.

Primary hidden data refers to the hidden data for user use. For instance, an embedded active agent data stream or a watermark data stream. The rules contained in the data can provide various important information about the host signal and on how to use the host signal. They are also called *ruling data* in this paper.

Secondary hidden data refers to the hidden data for control use. For instance, the error correction bits. This kind of data can be used to govern or control the use of the *primary hidden data*. Hence, are also called *governing data*.
Primary data (or primary data plus secondary data) is called *embedded hidden data*.

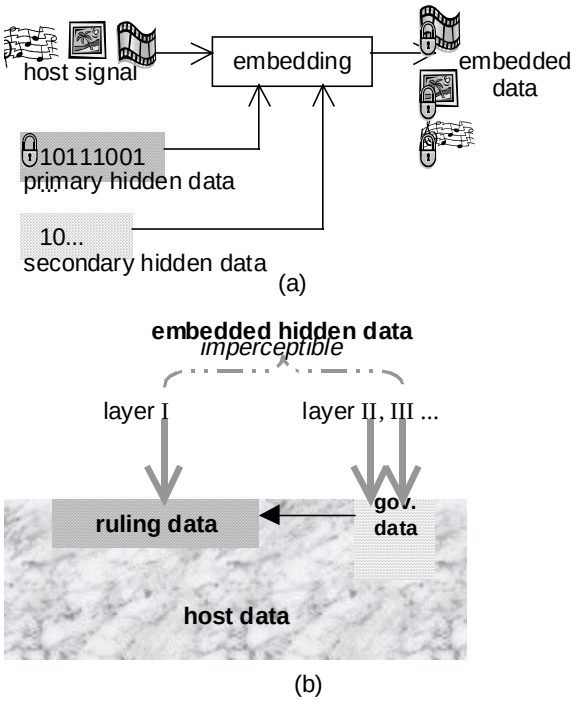


Figure 6: Data layers

3.2 General framework

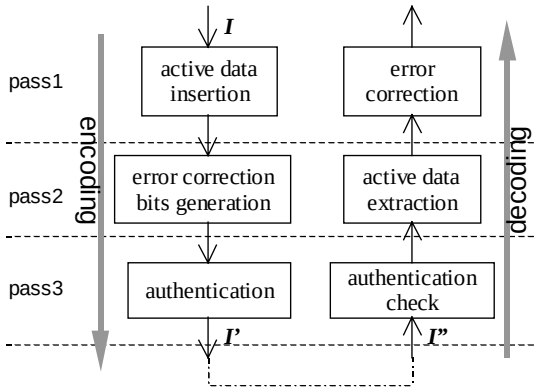


Figure 7: System three-pass architecture

The system has a three-pass architecture as shown in the figure above. First, the meaningful active agent data stream ***S*** is mapped into a sequence of binary data ***Sb*** = $Sb_1,Sb_2,\ldots Sb_M$ of length ***M*** which are inserted imperceptibly into the host signal ***I***. Then, the error correction bits, ***E*** = $E_1,E_2,\ldots E_Q$, are generated and embedded into the host signal in the second pass. In the third pass, cryptography techniques are used to authenticate the host signal as well as the embedded hidden agent.

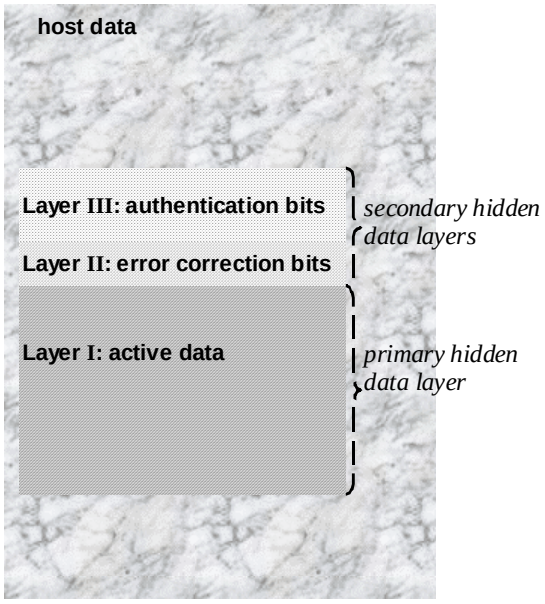


Figure 8: Illustration of the system data layers

Taking playback and record control as an example, they can be done by proper extraction of the active agent along with authenticity verification. The authenticity of an input audio ***I"*** is first checked with the key ***K'*** (see Figure X appended at the end of the paper). If ***I"*** is verified to be authentic, the active agent data stream can further be extracted with key ***K'*** (notice here the two keys can be the same or different depending on the application); otherwise access can be restricted based on the content management rule. For instance, prohibits 'record' and high quality 'playback' but not low quality 'playback' when authenticity check fails. After the active data stream ***Sb"*** is extracted, an error correction process is performed on ***Sb"*** and ***S'*** is further generated. Agent data stream ***S'*** = ***S*** can then be invoked.

3.3 Some characteristics of the proposed scheme

- Use perceptual model to guarantee inaudibility.
- Ensure lossless extraction of the active agent with secondary hidden data: authentication and error correction bits.
- Ability to localize alteration (w/ authentication layer) and to correct minor alteration (w/ authentication + error correction layers).
- Added security with public key cryptography.
- Base domain embedding assures fast extraction performance.

With regards to the features of the embedded agent, it

- may send feedback information to server when streaming or online preview is performed.
- may renew keys or management rules.

- may perform scrambling on the audio signal to prevent further unauthorized use of the content.
- may allow play-once-preview and also play unlimited times with just one time downloading when proper key is purchased.
- may carry additional info to display to the users.

3.4 Algorithms

3.4.1 Imperceptible active data embedding

The hidden data imperceptibility is ensured with proper usage of perceptual model. It takes advantage of human auditory system's inability to distinguish noise under conditions of auditory masking. That is, the presence of a strong audio signal makes a temporal or spectral neighborhood of weaker audio signals imperceptible. Our empirical study also shows that human ear can not distinguish the differences when a minor change is made on a singular point or maskee point (under the condition it is still a maskee point before and after the modification). In the base domain, the masking ability of a given sample depends on its loudness; while in the spectrum domain, the masking ability of a given signal component depends on its frequency position and its loudness. Empirical results also show that the noise masking threshold at any given frequency is solely dependent on the signal energy within a limited bandwidth neighborhood of that frequency and at any given time is solely dependent on the signal energy within a limited temporal neighborhood. In this paper we shall focus on base domain embedding only. Compare to the spectrum domain embedding, the advantage of it lies in the decoding performance in terms of speed. The disadvantage, however, is its low survivability over compression. The definition of singular point, masker point, and maskee point are given below.

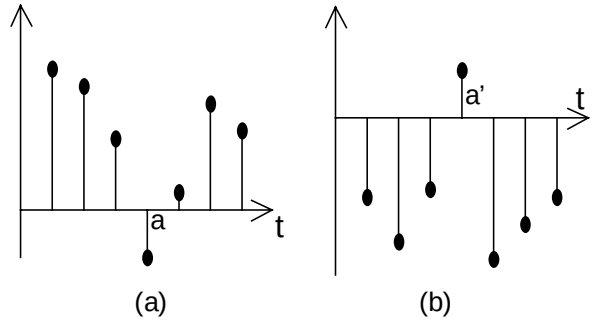


Figure 9: Singular points

Define $\boldsymbol{I}(j)$ to be a singular point iff sign($\boldsymbol{I}(j)$) = - sign($\boldsymbol{I}(j\text{-}1)$) & iff sign($\boldsymbol{I}(j)$) = - sign($\boldsymbol{I}(j+1)$). Figure 9 illustrates two singular points ***a*** in (a) and ***a'*** in (b).

A masker point $\boldsymbol{I}(j)$ is defined as a point with a intensity value larger than a threshold δ, i.e., amp($\boldsymbol{I}(j)$)≥ δ, whereas a maskee point $\boldsymbol{I}(j^k)$ is defined as a point that is under the mask of a masker point $\boldsymbol{I}(j)$, i.e., amp($\boldsymbol{I}(j^k)$)≤ mask(amp($\boldsymbol{I}(j)$)) (see Figure 10 where sample ***a*** is a masker point and sample ***b***, ***c***, & ***d*** are maskee points).

Several methods can be used to embed bits into the singular and maskee points. Here, as an example, we list one simple method in the following to embed a sequence of bits $Sb_1,Sb_2,\ldots Sb_M$ into the singular bits $Isng_1$, $Isng_2$ … $Isng_M$, of a host signal $I_1,I_2,\ldots I_n, \ldots I_N$.

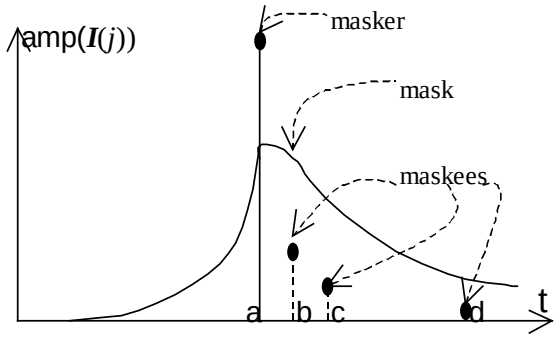


Figure 10: Perceptual mask

Encoding:

- If I(j)==0, set I(j)=I(j)+1.
- If the embedding bit Sb_m is 0 and the mth singular point is $Isng_m$, then set $Isng_l$ to 0.

  If the embedding bit Sb_m is 1, then leave $Isng_m$ unchanged or set $\varepsilon_1 \leq Isng_m \leq \varepsilon_2$, where ε_1 and ε_2 are lower and upper bound with ε_2 controlled by perceptual mark.

Decoding:

- Let m=1.
- If In =0 (or In is a singular point and In <ε1), set Sbm to 0 and m++.
- If In is a singular point (and In ≥ε1), set Sbm to 1 and m++.

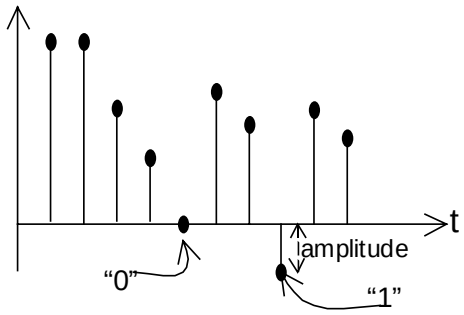


Figure 11: Adjustable parameters and decoding

Note that proper scrambling can increase the security level of the embedded bits.

3.4.2 Orthogonality

The bases of different hidden data layers should be orthogonal to each other to ensure maximum detectability. That is, the primary hidden data layer (the active data layer) and the secondary hidden data layers (the error correction data layer and the authentication layer) should be orthogonal to each other. In our system, multiple orthogonal features, singular points and maskee points, are

used with which different feature is employed to hide different data layer.

3.4.3 Error correction

An error correction layer shall require additional data hiding capacity upon the capacity required for hiding the primary hidden data. Therefore, the smaller the error correction sequence is, the more desirable it is. One simple way is to use 2D or multi-D checksum error correction. Assume the error correction bit number is $\boldsymbol{Q}$ and the active data stream bit number is $\boldsymbol{M}$. Then in the case of 2D checksum, the error correction stream length (number of bits) satisfies $\boldsymbol{M}=(\boldsymbol{Q}/2)^2$. For instance, our Java agent $\mathcal{A}$ has a data stream length of 4000bits. The error correction bits needed for $\mathcal{A}$ is thus only 64x2≈128bits in the case of 2D checksum.

Below, the 2D checksum algorithm is given.

Encoding

- Let Q=ceiling[$2M^{1/2}$], i.e., let Q be the smallest integer which is no less than ($2M^{1/2}$).
- Arrange $\boldsymbol{Sb} = Sb_1,Sb_2,\ldots Sb_M$ into $\boldsymbol{Q}$/2 chucks $\boldsymbol{SB}(1)= SB(1)_1,SB(1)_2,\ldots SB(1)_{Q/2} =Sb_1,Sb_2,\ldots Sb_{Q/2}$, $\boldsymbol{SB}(2) = SB(2)_1,SB(2)_2,\ldots SB(2)_{Q/2} = Sb_{Q/2+1},\ldots Sb_Q$ … and $\boldsymbol{SB}(Q/2) = SB(Q/2)_1,SB(Q/2)_2,\ldots SB(Q/2)_{Q/2} = Sb_{(QQ-2Q)/4+1}$,…Sb_M
- Let E_q=least($SB(q)_1+SB(q)_2+\ldots+SB(q)_{Q/2}$) for $q\in(1, Q/2)$ and E_q=least($SB(1)_q+SB(2)_q+\ldots+SB(Q/2)_q$) for $q\in (Q/2, Q)$, where least(S) denotes the least significant bit of S.

Decoding can be similarly done as in encoding process.

3.4.4 Authentication

We can use an authentication scheme similar to that of [9] in which they suggested to place the authentication value into the least significant bit of each sample. In this case, to ensure orthogonality, ε_1 shall be set to 2 or larger for both singular points embedding and maskee points embedding.

A brief outline of the algorithm is described below.

- Choose verification block size $\boldsymbol{B}$ and dependent block size $\boldsymbol{D}$(for example, $\boldsymbol{B}$=128 & $\boldsymbol{D}$=512bits). Assume the host signal is a 16bits audio, concatenating all the high bits (all the bits except the least significant bit) of the 512 samples yields a message $\boldsymbol{Mb}$ of 15x512=7680bits. Now by further concatenating a key of 512bits (or a key of shorter length which is padded to 512bits), a message $\boldsymbol{MB}$ of 8192bits is produced.
- Compute the one way hash with the MD5 algorithm, $\boldsymbol{MB'}=\boldsymbol{h}$=H($\boldsymbol{MB}$), to generate a 128bits message $\boldsymbol{MB'}$. (Append time or other secondary hidden data, such as the error correction bits, host signal length, and/or owner information, if $\boldsymbol{B}$>128bits.)
- Use public key (or secret key, depends on different applications) cryptography method and signing $\boldsymbol{MB'}$ with secret key $\boldsymbol{K}$, $\boldsymbol{MB''}$=Sgn($\boldsymbol{K}$, $\boldsymbol{MB'}$).
- Insert the $\boldsymbol{B}$bits message $\boldsymbol{MB''}$ into the least significant bit of each sample, from 1→0 if embedding 0 or 0→1 if embedding 1, into the verification block.

Another alternative is to embed the authentication bits to the feature points instead of all the samples. This method shall give it a little bit more space to tolerate scaling compare to the first method. Here, scaling refers to intensity scaling only.

- Choose block size $\boldsymbol{B}$ (for example, 2048+128 samples). Reserve 128 feature points (singular points or maskee points). Assume the host signal is a 16bits audio, concatenating all the bits of the rest 2048 samples which yields a message $\boldsymbol{MB}$ of 16x2048=32768bits.
- Compute the one way hash, $\boldsymbol{MB'}=\boldsymbol{h}$=H($\boldsymbol{MB}$), with the MD5 algorithm.
- Use public key (or secret key, depends on different applications) cryptography method and signing $\boldsymbol{MB'}$ with secret key $\boldsymbol{K}$, $\boldsymbol{MB''}$=Sgn($\boldsymbol{K}$, $\boldsymbol{MB'}$).
- Modify the 128 feature points to embed $\boldsymbol{MB''}$.

Decoding can be similarly done with the public key plus an XOR operation to check the authenticity of the signal.

4 Experiment result and summary

4.1 Experiment result

The system is implemented on PC with VC++. We used a small test set of 10 pieces of music which include both rock-n-roll and classical music. The testing music is about 2-4 minutes long each. The decoding time is about 7Mbytes/sec on average with Pentium400 CPU. The test is conducted in our lab & office environment with low noise. The average SNR is about 26.6dB on our test set.

Table 1 lists our perception test results. Notice that due to our limited capability on studio access, the system has only been tested in lab, office, and home environment. Whether the system will survive studio environment shall be tested in the future. Table 1 also gives a sample result on embedding active agent $\mathcal{A}$ into a short piece of sound clip of 5 seconds in length. Although the embedded data streams can still be extracted without error, the low data hiding capacity determines the high perceptibility of distortion.

Table 1. Perception test

person	song	Distortion perceptibility
Office environment		
15 persons	Song 1	0%
Person A	Song 1~10	0%
Person B	Song 1~10	0%
Person A	Sound file x*	Perceptible
Lab environment		
Person A	Song 1~10	0%
Home environment		
Person A	Song 1-10	0%
Person C	Song 1~10	0%

*Sound file x is a 5 seconds, 231KB long (in .wav format) sound clip of nature sound. It is embedded with the same Java agent $\mathcal{A}$ in the above test.

Table 2 gives the results of our extractability test.
Figure Y shows two plots of sample results. The changes on singular points before and after active data embedding (point A) are marked in Figure Y (b). This set of tests also shows that the minor changes on singular points and maskee points indeed do not interfere with the perceptual quality of the audio clips.

Table 2. Extractability test results

Songs and processing	extractability
Song 1~10, no post processing	Yes, no error
Sound clip x, no post proc.	Yes, no error
Song 1, intensity scaling 93.75%	Yes, no error
Song 1, sample dropping (drop the1st 100 samples)	Yes, no error

4.2 Summary

A fragile active data hiding system is presented in this paper. We have also developed robust active data hiding system for audio/visual signal. For details of the methodologies and algorithms of our robust audio active data hiding system, please refer to our publication [10], in which robustness against MP3 compression and common signal processing attacks are assured on top of imperceptibility and lossless extractibility.
Future works include large database test and benchmarking on our fragile/robust active data hiding scheme. Also frequency domain fragile active data embedding is another future task with which the active data can be expected to survive compression while alteration can still be detected as well as localized.

5 References

[1] E. Koch, J. Zhao, Towards robustness and hidden image copyright labelling, in *Proc. Workshop on Nonlinear Signal and Image Processing*, 1995, P452-455.

[2] S. H. Low, N. F. Maxemchuk, J.T. Brassile, L. O'Gorman, Document marking and identification using both line and word shifting, *in Proc. Infocom'95.*

[3] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, Secure spread spectrum watermarking for images, audio, and video, in *Proc. ICIP'96*, P243-246.

[4] L. Boney, A. H. Tewfik, K. N. Hamdy, Digital watermark for audio signals, in *Proc. ICMCS'96*, P473-480.

[5] M. Yeung, F. C. Mintzer, An invisible watermarking technique for image verification, in *Proc. ICIP'97*, P680-683.

[6] M. D. Swanson, B. Zhu, B. Chau, A. H. Tewfik, Multiresolution video watermarking using perceptual models and scene segmentation, in *Proc. ICIP'97.*

[7] C. Podilchuk, W. Zeng, Image-adaptive watermarking using visual models, *IEEE Journal on Selected Areas in Comm.*, 16(4):525-539, 1998.

[8] O. Benedens, Geometry-based watermarking of 3D models, in IEEE Computer Graphics and Applications, V19, 1999, P46-55.

[9] C. W. Wu, D.Coppersmith, F. C. Mintzer, C. P. Tresser, M. M. Yeung, Fragile imperceptible digital watermark with privacy control, in *Proc. SPIE'99* V3657.

[10] X. Li, H. Yu, Robust active data hiding for digital audio, to submit.

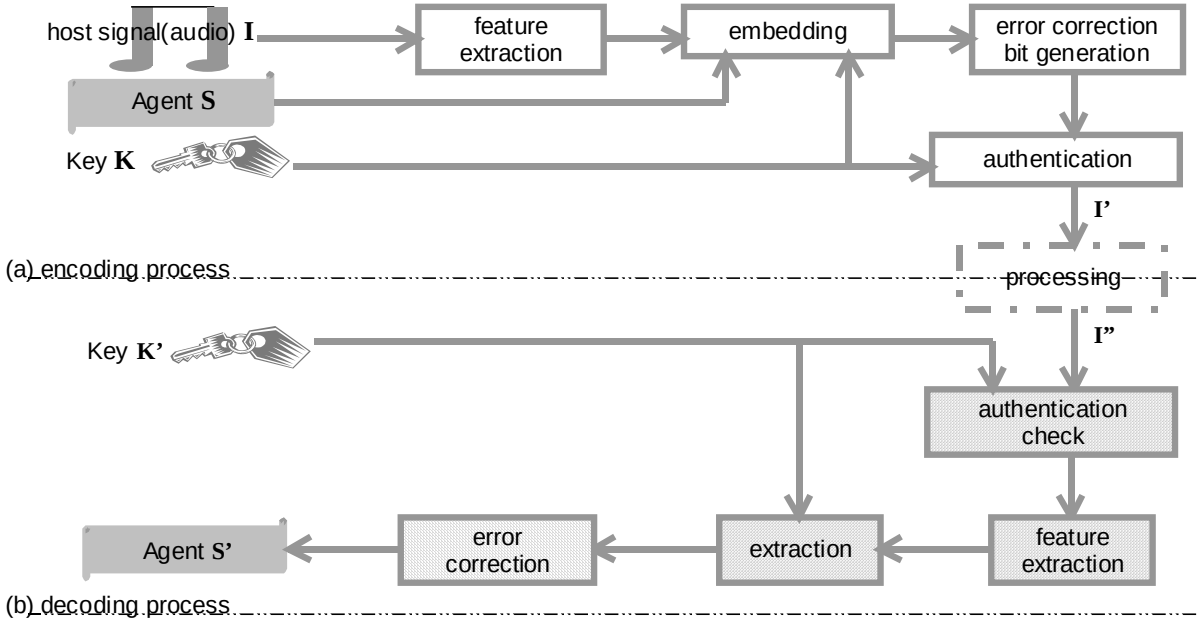
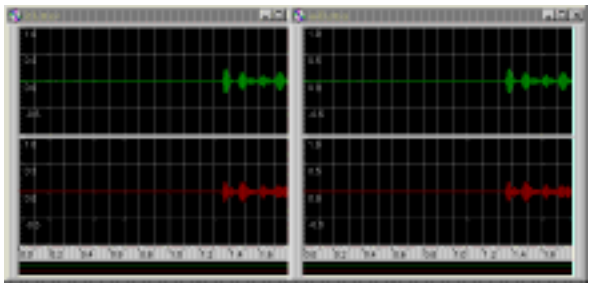
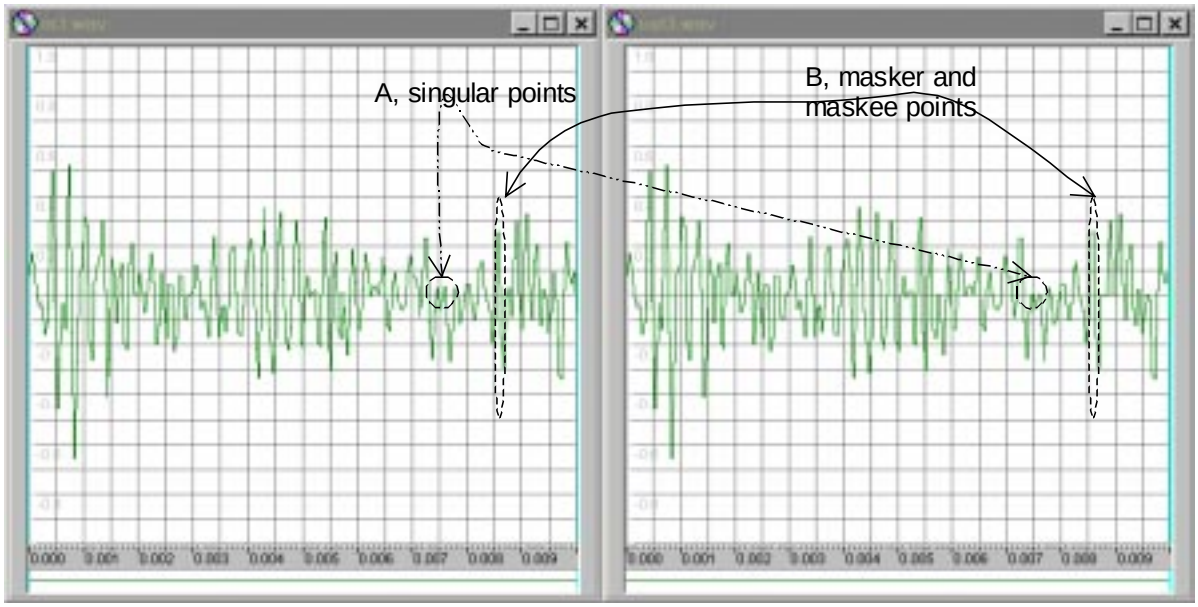


Figure X. Encoding and decoding process outline



(a) (Left: original audio signal; Right: embedded with active agent $\mathcal{A}$; Upper: channel one; Lower: channel two;)



(b) (Left: original audio signal; Right: embedded with active agent $\mathcal{A}$)
(notice the change after embedding in the marked areas A(singular points))

Figure Y. Sample results plot

Digital Watermarking for MPEG Audio Layer 2

Jana Dittmann, Martin Steinebach, Ralf Steinmetz

GMD – Forschungszentrum Informationstechnik GmbH
Institut IPSI
Darmstadt, Germany

{dittmann,steinebach,ralf.steinmetz}@darmstadt.gmd.de

ABSTRACT

MPEG-Audio has become a standard in the area of audio compression. It is used for a wide range of applications like online music distribution or in the audio parts of MPEG-Videos. In this paper we show how to secure the audio stream by watermarking without conversion to PCM-Wave-Data and without encoding the MPEG-Data in a special way. The original MPEG-file is not necessary to read the embedded information.

The key to our watermarking algorithm is to change the scale factor information of MPEG-frames. Small patterns are created producing a stream of information bits hidden in the data stream. Multiple streams can be included by using different patterns without critical reduction of perceived quality or robustness of the single watermarks.

KEYWORDS

Digital watermarking for MPEG audio, copyright protection, scale factor manipulation.

1 Motivation

Today audio watermarking is used mainly for copyright protection. We want to provide watermarking technologies for authentication and the prove of integrity and research robust and fragile audio watermarking schemes. This new approaches can be used to improve security of multi media streams.

A possible attack against the integrity of an audio recording would be the removal of words, so that a sentence like “I am not guilty” could be changed into “I am guilty”. A watermark could be used to verify if the original has been changed by embedding a time stream.

2 Digital Watermarking

Digital watermarking is a way to embed data within another data stream or signal using aspects of the carrier signal like quantisation noise in A/D-conversion. The technique of watermarking digital audio data has been the object of previous researches, e.g. [BTH1996], [SZAT1998], [SM1998] and [CKLL1997]. They provide a number of attributes which are important for watermarking:

- **perceptual transparent**: The watermark should not produce audible artifacts or reduce the quality of the audio data
- **robust**: The watermark should not be removable without seriously damaging the carrier audio data
- **statistical invisible**: Even when the algorithm for insertion of the watermark is known to the public, it should not be possible to destroy, fake or remove it without knowing a special key
- **self-clocking**
- **embedded directly in the data** (as headers can be removed or replaced easily)
- **multiple watermarks** should be possible
- the **expense** necessary to embed the watermark should stand in relation to its estimated effect
- **compression characteristics** of watermark and original data should be the same or at least similar
- **unambiguous**: the watermark should reliably identify the owner

Most of the previous works are based on marking PCM-Data. Many claim to be robust against MPEG-encoding, but tests have shown that the coding and decoding deletes most watermarks.

Watermarking algorithms are not robust against MPEG compression if they are based on the same principles: Parts of the audio data are masked or are not perceptible because of psycho-acoustic laws ([LAKa1996], [LAKb1996]) .

So either we have to embed the watermark in a perceptible area of the data, which would mean loss of quality, or we have to work directly on the MPEG Data.

With MP3Stego ([Pet1998]) there is an algorithm that inserts a watermark in layer 3 files. But to do so the audio data has to be encoded to MPEG by the algorithm. The resulting watermark is not robust against de- and re-compression and may not be robust against MPEG-attacks.

3 MPEG Audio Layer 2

The audio signal we used to test our algorithm is a MPEG-Audio Layer 2 stereo signal, 44.1 kHz and

160 bps. The basic idea of the algorithm makes it compatible with every Layer 1 or Layer 2 MPEG audio stream.
Only scale factor information of the MPEG stream is used in our algorithm. The following graph shows how the needed bits are extracted:

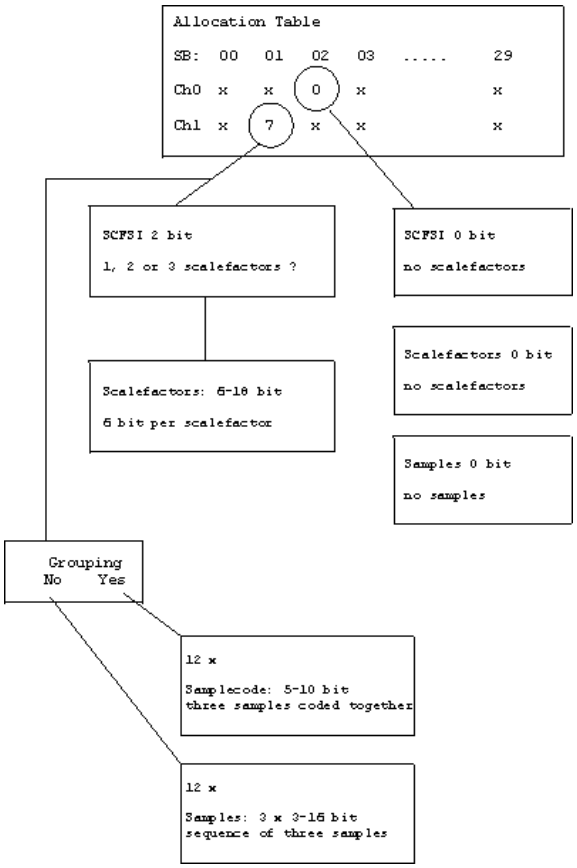


Figure 1: Scale factor location in MPEG-Stream

In Layer 3 scale factors are encoded in a different way. So to embed data in a Layer 3 audio stream we will have to change the extraction algorithm.
The data is divided in channels, subbands and one to three scale factors per frame. The allocation table tells which channels and subbands are encoded in the frame. Then we have to look at the scale factor selection information (SCFSI) to see how many scale factors are used for the samples. There are four different possibilities that use up to three scale factors. When we know how many scale factors are used we can extract them for further use. Knowledge of the used size of the samples is only necessary to stay synchronous with the data stream.

4 Watermarking principle

Figure 2 shows an overview of our algorithm. Given a MPEG-file, a text to embed and a group of three patterns we encode the text into a sequence of patterns and extract the scale factors from the frames of the MPEG-file.
Difference patterns based on this scale factors are calculated and the central algorithm changes these patterns until a sufficient number matches our desired sequence of patterns.
The whole watermark is inserted in this way, if there are more frames than needed the watermark is inserted multiple times.
Then the new scale factors are inserted in the source file, overwriting the old ones and so creating a watermarked MPEG-file.

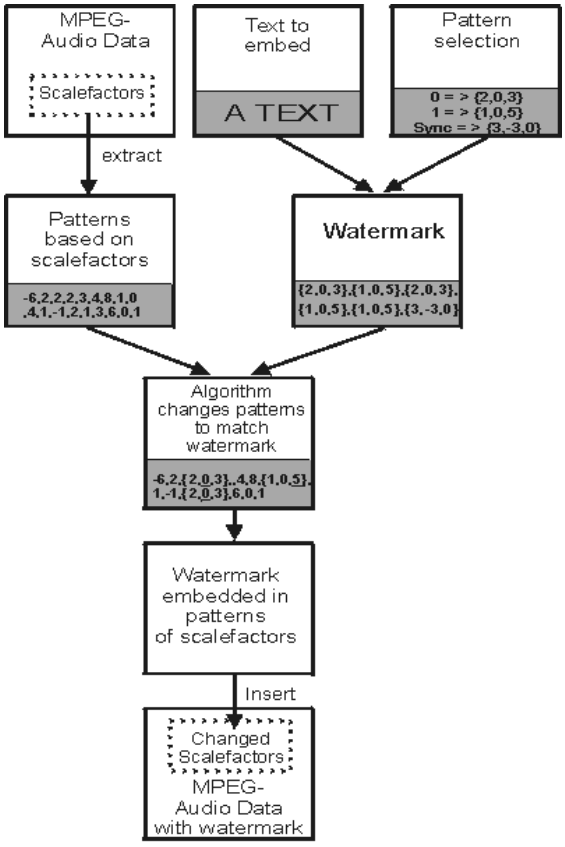


Figure 2: Watermarking principle

Embedding watermarks in the scale factor information has already been proposed in [NQ1998]. While our algorithm has first been developed uninfluenced by this work, we used some knowledge and experiences gained in [NQ1998] to improve audio quality. We also choose another way to embed and receive data and offer more security against attacks that would destroy the watermarks proposed in [NQ1998].
A main difference between the two algorithms is that the one in [NQ1998] needs the original signal do read the watermark. This also makes it resistant against inversion attacks where a third party could prove ownership by subtracting their watermark from the original marked by the true owner. As we build our algorithm for applications where ownership can be proven by other ways, it did not have to be resistant against this attack and therefor does not need the original to read the watermark. Only a very small amount of data has to be transferred if it is used online - a huge benefice when a fast solution is necessary.

The key to our watermark is always a group of three patterns, one to code „0", one for „1" and another for „sync". The last one is used for self clocking and robustness against cropping. These patterns consist of a few numbers that must match the differences between a starting point and the following scale factors in the data stream. An example: Given the list of scale factors {10,8,12,14} the first one would be used as a starting point and the pattern would be {-2,2,4}.

An information bit (0,1,sync) is present in a part of the stream when its number of occurrences is higher then the ones of the other two patterns. Therefor the flow of scale factors is searched for matching or nearly matching patterns while the other two patterns are destroyed. Then the nearly matching patterns are changed by adding or subtracting small numbers so that they finally match the requested pattern. Imagine we were trying to insert the pattern {-2,2,2} in the example above. We would have to change the last number (14) by -2 to match the pattern. So the work can be done easily by subtracting the existing pattern from the wanted pattern (a) and applying the result to the scale factors (b):

(a) {-2,2,2} - {-2,2,4}= {0,0,-2}

(b) {8,12,14}+ {0,0,-2} = {8,12,12}

This would be the final list of scale factors. By doing so we create an area in the MPEG-stream where one of the patterns is found quite often while the other two are found rarely or not at all.

5 The central algorithm

Based on the idea explained above, we created an algorithm that changes patterns in the sequence of scale factors so that two of three patterns are subdued and one is inserted a certain number of times.

The parameters of the algorithm are:

- the three patterns for „0", „1" and „sync"
- a maximum tolerance that states how strong the changes in the scale factors may be to create matching patterns
- a number of frames in which the changes take place
- the minimal number of patterns that must be equal to the pattern of the information bit we want to encode in the area of frames

As one can see in figure 3 the central algorithm consists of one loop. Given the wanted and unwanted patterns, minimum of matching patterns, tolerance and the sequence of scale factors, the algorithm first distorts any appearing pattern of the two unwanted and then starts to count the patterns corresponding to the information bit to be embedded. Usually there won't be enough matching patterns to satisfy the given minimum. Now the algorithm will start to look for patterns similar to the wanted one.

Based on the sum of the squares of the difference between found and wanted patterns it will decide which patterns could be changed to the desired pattern without serious loss of quality. In the first round of the loop only patterns which differ by one will be changed, in the next the used tolerance is increased until either the given maximal tolerance or the minimal number of patterns is reached.

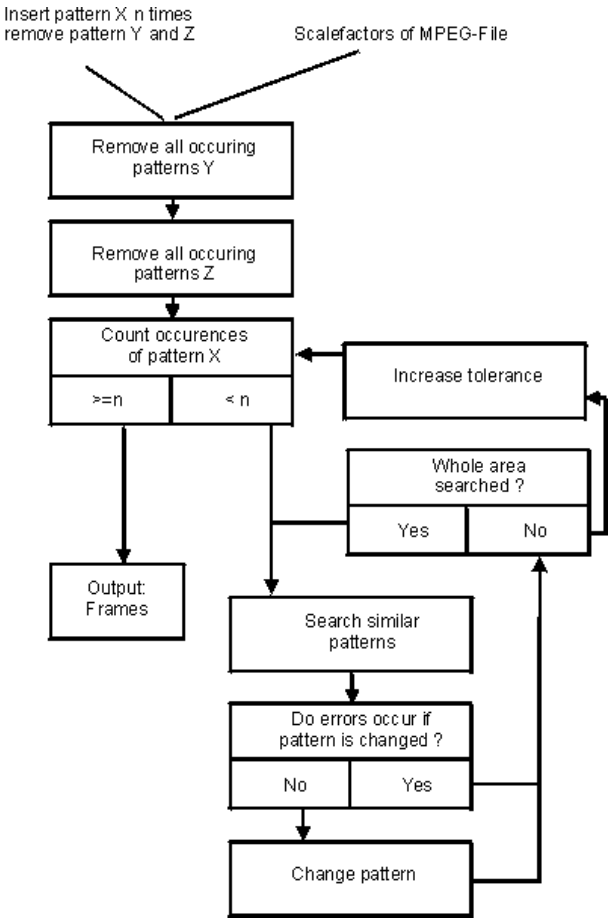


Figure 3: Pattern insertion

This means that at low levels of maximal tolerance the algorithm will not insert as many patterns as desired. This will protect audio quality by the loss of security.

The reason why the sum of the squares and not only the sum is used is that large changes are more audible than multiple changes, so it is better to change three scale factors by 1 (which means a difference of 3 ($1^2+1^2+1^2$)) than one by 3 (which would mean a difference of 9 ($3^2+0^2+0^2$)).

Figure 4 shows a small excerpt of a file marked with two watermarks at a bit rate of 1 bps. It was computed by comparing the scale factors of the original MPEG file and the marked one. Wherever gray bars are visible, scale factors exist. The white bars are the ones that were changed. The height of the bars is depended on the size of the scale factor.

The distribution of the scale factors is subband/channel in the Y-axis and SCFSI-number and frame in the X-axis. The same as used while searching for patterns.

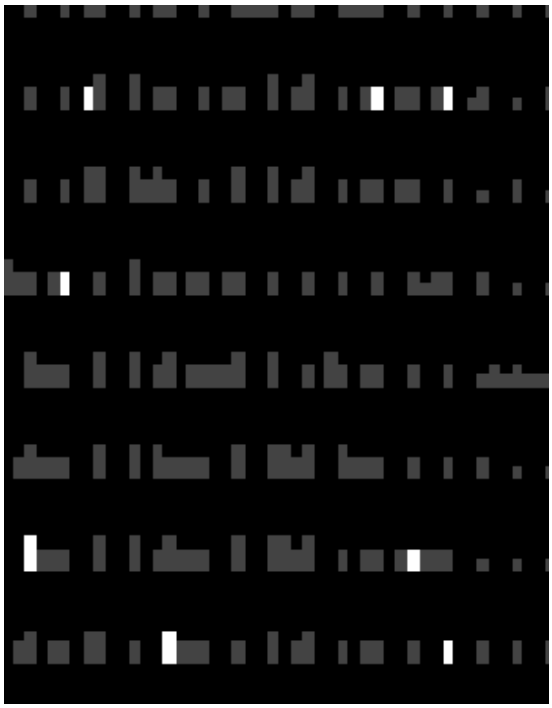


Figure 4: Differences of scale factors

5.1 Pattern selection

It is vital for the resulting audio quality and the security of the watermark to carefully choose the right patterns. Some patterns occur quite often and should not be used for data embedding as they would have to be changed each time they are found where another pattern should be inserted.
First steps in research have lead to the following „Top-20" of pattern occurrences:

Rank	Pattern			Abs	Hits
1	0	-1	-1	2	393
2	0	0	0	0	375
3	-1	-1	-1	3	367
4	1	0	0	1	351
5	0	0	-1	1	331
6	0	-1	0	1	322
7	-1	-1	-2	4	316
8	0	0	1	1	316
9	0	1	1	2	310
10	1	1	1	3	310
11	0	1	0	1	309
12	1	1	0	2	306
13	0	-1	-2	3	303
14	0	-2	-1	3	302
15	1	0	1	2	301
16	-1	0	0	1	299
17	-1	-2	-2	5	297
18	-1	-1	0	2	294
19	0	-2	-2	4	294
20	1	0	-1	2	292

Abs: Sum of pattern absolutes
Hits: How many times the pattern has been found in so far four MPEG-Files with 63831 patterns.

Table 1: Occurrences of patterns

In our tests patterns with at least one number as large as three (e.g.{0,0,3}) or with steps larger then two (e.g. {0,-2,2}) worked fine. Of course the used patterns should differ as much as possible when using multiple watermarks, as two similar patterns would be changed to match each other and then the watermark embedded first would be destroyed or at least heavily distorted.

5.2 Security

The knowledge about the used patterns is the key to the security of our algorithm. To further improve security the patterns can be modulated. Thereby detection of key patterns becomes harder.
Example(pattern shifting):

starting pattern	{2,0,2}
modulation step 1	{0,2,2}
modulation step 2	{2,2,0}

The modulation process can be synchronized with the sync-bit, for example at the beginning of each encoded letter.
Embedding with a very high redundancy r_1 enforces a lot of matching patterns which can be located more easily. A sensible threshold has to be found here.
The text information can be encrypted before embedding it, so even finding the right pattern combination will not necessarily lead to identifying it as no readable output is produced.
If an attacker knows the combination of the three patterns, he can easily destroy, change or replace the watermark.

5.3 Robustness

An example how to attack our watermarks inside of MPEG-files would be to change the scale factors randomly. But as our patterns are distributed over the whole range of subbands and are embedded in two directions (time and subband/channel) a large amount of scale factors would have to be changed. This would mean an audible loss of quality.
Another kind of attack would be the separation of one channel, thereby creating a mono channel out of one of the two a stereo channels. Almost all patterns distributed over the subband axis would be destroyed, but the ones in the time axis survive this attack.
In the current implementation the watermarks are not robust against decoding to PCM-Wave and back to MPEG. In our tests we found almost no similarity between the patterns of the original and the re-coded MPEG-file. Massive changes occurred which made the survival of patterns impossible. But we have to remember that decoding and re-coding to MPEG always means a serious loss of quality, which will stop pirates form choosing this way.

6 Detection of a watermark

Thanks to the open design of the algorithm we tested different ways to embed and extract data.
The first method was to use a fixed number of frames for every information bit. This makes it sim-

ple to extract the data if no trimming occurred: In the given region the three patterns are searched for and counted. The one with the most hits is the embedded bit.
For robustness against trimming the sync-bits can be used. With a fixed number of frames and a fixed number of bits to encode a letter the sync bit can be used as a header to resynchronize the algorithm at the beginning of each watermark.
The second method uses no fixed number of frames. Only the minimum of embedded bits and a tolerance are given. The algorithm steps through the frames and changes patterns if the tolerance allows it. When the minimum number of patterns to embed is reached, the algorithm continues with the next information bit. Thereby a given level of quality is always ensured, but the watermark is also embedded. The used number of patterns can vary as needed.
The detection process is more complicated as in method 1: We search for the dominant of the three patterns in the frames, and every time the dominance changes, the previously dominating bit is treated as a found information bit. To make this process robust against noise and false detection, weighting of the frames and filtering of very short dominance phases is used. The sync-bit is used to divide the "0" and "1" bits.
This method can be used to embed bit patterns in the MPEG stream. The detection rate of these patterns is high.

7 Test results

Most of the attributes of a watermark we mentioned in part 2 were realized in our algorithm. While points like security and robustness have already been mentioned, the following test results will try to complete the picture.

7.1 Transparency

We are still testing in this area. Right now you can say that a certain loss of quality is not deniable, but it is not strong enough to be found annoying. The following results are based on a test with ten students. The audio material has been burned on CD and was played on a usual stereo set in a natural listening environment.
The examples were rated on a scale from one to five where one was „no audible difference" and five was „bad FM-receiver or scratches on a record".
The test contained the following audio data:

- original (CD-Quality)
- only MPEG-encoding (44,1 kHz, 160 bps)
- MPEG with one watermark
- MPEG with two watermarks.

All examples were about 30 seconds long.
The watermarks were embedded with a bitrate of 1bps but with the requirement of twenty matching patterns.
The examples were:

- **Form**: Electronic dance music
- **Sheila**: Female ethnic singing with synthesizer background
- **Waaberi**: Male ethnic singing with native instruments
- **Crowd**: Talking group of people
- **Serenade**: Spoken poem

The averaged results in table 2 show us that in no case the perceived quality with the watermarks was a full step worse than the one with only the quality loss produced by MPEG-compression. Most results are in the range of two, which meant a difference like between two stereo-sets.
The last example, „Serenade" was the one where the changes were heard most often. The algorithm produces some slightly audible noise in the reverb between the words .

Example	MPEG	1 wm	2 wms
Form	1,60	1,90	2,10
Sheila	1,60	1,90	2,10
Waaberi	1,60	1,90	2,10
Crowd	1,90	2,40	2,40
Serenade	1,90	2,40	2,70

Table 2: Averaged results of test

A second test was done to show if a listener could hear differences between the unmarked and the marked MPEG file. While the listeners heard the original at the beginning of each sequence, now only the unmarked MPEG was given to compare with.
We created sequences of ten 4 second long audio pieces of the examples „Form", „Sheila" and „Serenade". Five types of material existed:

- unmarked
- 20 frames / 2 bps
- 10 frames / 4 bps
- 5 frames / 8 bps
- 3 frames / 14 bps

The given number of frames tells how many frames where used to encode at least 20 patterns according to the desired information bit. As a frame in this case is 23 ms long, there are about 43 frames per second. The bits per seconds (bps) are calculated by dividing these 42 frames by the used number of frames.
Ten sequences were created. The six listeners had to determine whether the actual piece of the sequence was a marked one or not.
The results of the test are shown in table 3. The last column tells how may percent of the times an audio piece of the according type was found to be changed. We can see that the unmarked pieces were chosen more times (10,4 %) than the ones with 2 and 14 bps (7,2% and 8,9 %). The ones with 4 and 8 bps were most often selected as marked, but not even in 20% of the times they occurred.

Type	n.o. app.	marked	%
none	40	25	10,4
2 bps	23	10	7,2
4 bps	13	12	15,4
8 bps	9	9	16,7
14 bps	15	8	8,9

n.o.app.: number of appearance in a total of 100 pieces
marked: how many times did the six listeners hear a loss of quality one of the pieces of the according type

Table 3: Results of watermark perception test

This shows that it is very hard to detect the changes made by our algorithm, even at higher bit rates like 14 bps. Only in 39 of 360 cases the watermark was heard, which is a percentile chance of 10,8 %. This is almost the same chance as the one of false detection (10,4 %).

7.2 Bitrate / Capacity

In our tests we ensured 20 matching patterns in a area of 3 frames, which was audible, but not annoying and good enough for most internet movies or previews. With about forty frames per second the given bit rate would be fourteen.
For our first audio test we used a bit rate of one bit per second, but with two parallel watermarks. For the second test we used bitrates up to 14 bps. In both cases detection success of the embedded data was 100 %.
Further research will be necessary to find out if there is correspondence between MPEG-bitrate and possible embedding-bitrate and which bitrates can be used with MPEG Audio Layer 1 and 3.

7.3 Complexity

Our algorithm has shown to be quite fast as the used calculations are mainly additions and subtractions on integers or bytes.
It also uses only a few kilobyte of memory. This is because we only have to look at a small part of the scale factor information at one time and can leave the rest of the file unchanged.
It can be assumed that a real-time application for detecting watermarks can be implemented without serious changes in the algorithm.

7.4 Self-clocking

Self-clocking was achieved with the „sync" information bits: Depended on the way we insert our data, we can determine the number of frames used for one information bit.
In our current tests, we use one „sync" bit to separate letters and two „sync" bits to mark the end of the message.
But we could also reduce the amount of inserted data by only embedding a sequence of „1"s and „0"s and using a „sync" at the end of the message. Then we would have to look for the number of frames used for this „sync" and use this information to decode the other bits if we did not know the number of used frames.

8 Conclusion

We introduced a way to mark MPEG-Audio Layer 2 files with one or multiple watermarks. Tests have shown that it doesn't reduce audio quality significantly even with multiple watermarks.
The strength of our algorithm is that it works on existing MPEG-Files without the need of decoding to PCM-data and that it doesn't need the original MPEG-file to read the watermark. Only small transfer rates would be necessary if used online, and due to low complexity of the used calculations the algorithm works very fast.
It is build to be secure against attacks imaginable against MPEG-files by the way the data is distributed in the scale factor information.
Removal is possible by decoding the audio file to a PCM-file and back again, but this would result in a high quality loss. A loss not acceptable in most situations a watermark is necessary.

9 References

[BTH1996] L. Boney, A.H. Tewfik, K. N. Hamdy: Digital watermarks for audio signals.

[SZAT1998] Mitchell D. Swanson, Bin Zhu, and Ahmed H. Tewfik: Audio watermarking and data embedding - Current state of the art, challenges and future directions

[NQ1998] Lintian Qiao and Klara Nahrstedt: Non-Invertible Watermarking Methods For MPEG Encoded Audio, June, 1998.

[SM1998] Scott Moskowitz: So this is Convergence? Technical, Economic, Legal, Cryptographic, and Philosophical Considerations for Secure Implementations of Digital Watermarking, Blue Spike inc., 1998.

[CKLL1997] Ingemar J. Coxy, Joe Kiliany, Tom Leightonz and Talal Shamoony Lambda: Secure Spread Spectrum Watermarking for Multimedia, Published in IEEE Trans. on Image Processing, 6, 12, 1673-1687, 1997.

[LAKa1996] Laurence Boney, Ahmed H . Tewfik , and Khaled N. Hamdy: "Digital Watermarks for Audio Signals", 1996 IEEE Int. Conf. on Multimedia Computing and Systems June 17-23, Hiroshima, Japan, p. 473-480.

[LAKb1996] Laurence Boney, Ahmed H. Tewfik, and Khaled N. Hamdy: "Digital Watermarks for Audio Signals", EUSIPCO-96, VIII European Signal Proc. Conf., Trieste, Italy, September, 1996.

[Pet1998] Fabien A.P. Petitcolas: MP3Stego, Computer Laboratory, Cambridge, August 1998.