Multimedia and Security
Workshop at ACM Multimedia

Recently security has certainly become one of the most significant and challenging 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 first ACM workshop on "Multimedia and Security" took place in conjunction with the ACM Multimedia'98 in Bristol, U.K., September the 12th. We focused on the analysis of specific security problems of multimedia systems and multimedia material in the digital environment.
The objective was 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. Beside technical approaches legal requirements for security solutions are further topics. The workshop provided space for intensive discussions among the addressed problems of security in and with multimedia. The proceedings show current solutions and still open problems which must be addressed in the near future.

A major field of the discussion was the identification of acceptance problems to use distributed multimedia production systems in digital marketplaces. Solutions for confidential transmission, authentication of original, copyright protection and Try&Buy transactions were addressed. As a main result the discussion in the workshop shows the need for a flexible and open watermarking environment for embedding robust watermarks. We would like to take the opportunity to thank the presentators for the excellent talks and the participants for the intensive discussions.

We understand that the interest and importance of security is reflected in the great number of participants in Bristol. We could claim a very international community and had a wide range of highly interesting topics. Due to the excellent feedback of the participants and the engagement in the preparation this was certainly one of the best international workshops in the multimedia-security area. Based on these excellent experiences we are planning the next workshop on Multimedia and Security at the ACM Multimedia '99 to continue the discussion and especially to see the advantages in digital watermarking, the robustness and the practical usage.
Additionally in the '99 event we want to address the topic, that existing multimedia security mechanisms are not realised by using multimedia tools applying security. Thus the discussion is extend to the use of multimedia to perform security. Though security is recognised as an important issue in multimedia it is, ironically, mostly not presented by the new media. Usually, security algorithms are seen as background processes, invisible to the user. 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.

Jana Dittmann
Workshop CoChair

Ralf Steinmetz
Conference CoChair

Contents

Requirements and Mechanisms of IT-Security Including Aspects of Multimedia Security

Petra Wohlmacher
University of Klagenfurt
Villacher Str. 161
A-9020 Klagenfurt
0043-463-2700854

petra@ifi.uni-klu.ac.at

ABSTRACT

In this paper we describe the most important security requirements, which must be fulfilled by today's IT-systems, and the security measures used to satisfy these requirements. These security measures are based on modern cryptographic mechanisms as well as on security infrastructures.

Regarding data security and communication security in particular in the field of multimedia, the requirements on security increase. If and in which way the discussed security mechanisms can be applied to multimedia security is difficult to analyse. This is mainly due to the complexity of multimedia data and their applications. This paper introduces the main issues of IT-security and represents the basis for solutions of security problems in the field of multimedia.

KEYWORDS

Security requirements, security measures, security mechanisms, multimedia, confidentiality, integrity, authenticity, non-repudiation, session key, one-way hash function, trapdoor one-way hash function, message authentication code, digital signature, authentication protocol, challenge-response protocol, security infrastructure, trust center, public key infrastructure, originality.

1 Introduction

IT-systems play an essential role in all areas of today's information community. By increasing the requirements for efficiency and the possibilities of IT-systems the needs for security and trustworthiness also increase. These needs are particularly important for security-relevant applications as well as for applications processing sensitive personal data.

In order to assess the trustworthiness of IT-systems, world-wide catalogues for security criteria have been published [2, 8, 16, 17, 19]. One of the most important ones is the Europe-wide valid ITSEC catalogue of criteria [8], which contains criteria for evaluating the security of IT-systems. This catalogue defines security criteria within different classifications regarding the following basic threats:

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

From these threats we may derive the basic requirements for the security of a given IT-system. Security requirements are met by security measures, which generally consist of several security mechanisms. Security services can be made available by security mechanisms.

Secure and trustworthy actions and interactions are important requirements for multimedia within the digitised world, too. Whether or not a multimedia application fulfils these requirements will have a substantial influence on the acceptance of this relatively new medium.

The remainder of this paper deals with the most important security requirements of today's IT-systems. Additionally, security measures and security mechanisms, which are fulfilling these requirements, are discussed. The presented requirements and measures may constitute the elementary basis for solutions of security problems of multimedia.

2 Requirements and Measures

The following security requirements are essential for IT-systems. They are met by the succeeding security measures:

- Confidentiality: Cipher systems are used to keep information secret from unauthorised 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.
- 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. The mechanisms are based on message authentication codes or digital signatures combined with notary services, timestamping services and evidence recording.

The security measures above mentioned use cryptographic mechanisms which we will explain in the next section.

3 Cryptographic Mechanisms

Cryptographic mechanisms can be implemented by the use of cryptosystems. These systems consist of a set of invertible functions, a set of keys, parameterising 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. In public-key cryptosystems each entity holds a key-pair (PK,SK). This pair consists of a secret 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 secret key SK. In other words, even with the most powerful computers it is not possible to deduce PK from SK during a period of time.

4 Confidentiality

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

A cipher system consists of a set of encryption functions, a corresponding set of decryption functions, and a set of keys. The data to be encrypted (plaintext) is transformed by the encryption function parameterised by a key. The result of this transformation is called ciphertext or cipher. The plaintext can be recovered by a decryption function also parameterised by a key.

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 we will give a more detailed discussion in the following section.

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 concatenation of x and y).

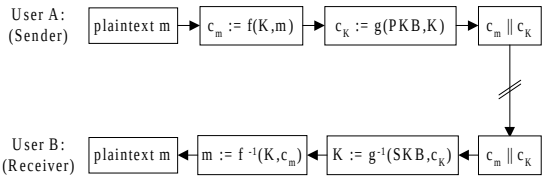


Figure 1: Session-key scheme

Plaintext m shall be encrypted with 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 f parameterised by the 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 the encryption function g, parameterised 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 g^{-1}, which is parameterised by the secret key SKB corresponding to PKB. Then he computes the plaintext m from the encrypted data c_m by use of the function f^{-1} of the private-key cryptosystem, parameterised by 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 the secret key SKB is able to recover the secret key K used for encryption. A possible solution for this problem will be given in section 8. Besides that it ensures that only the legal owner of the secret key SKB is able to recover the secret key K used for encryption.

Some examples of private-key cryptosystems which are used for cipher systems are DES [14], triple-DES [1] and IDEA [13]. Examples of public-key cryptosystems, used for encryption schemes, are RSA [21] and ElGamal [5]. Commonly used combinations for session-keys schemes are DES with RSA or IDEA with RSA.

5 Data Integrity

The integrity of data can be checked by means of so-called one-way hash functions. These functions are

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

often named manipulation detection code (MDC), message digest, digital finger print, cryptographic checksum or message integrity code (MIC). These mechanisms cannot prevent data manipulations, but they make these manipulations detectable. Therefore they are called detective mechanisms. The protected data remain in plaintext.

A one-way hash function H maps strings of arbitrary length to strings of a maximum or fixed length |n|: H: $\angle_0 \bullet [0,n] \subset \angle_0$ where $n \in \angle_0$. With respect to binary strings used as input, H can be defined as follows: H: $\{0,1\}^* \bullet \{0,1\}^n$, where n typically assigns one of the values 64, 128 or 160 bits. A hash function reduces the data m to its so-called hash value h := H(m).

Hash functions possess the characteristic that the image H(m) can be computed easily, but that it is computationally infeasible to find any preimage m such that m = H(m).

Since there exist infinitely many strings of arbitrary length, but only finitely many strings with a length $\bullet$ |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 preimages m_1 and m_2 which are mapped to the same hash value $H(m_1) = H(m_2)$.

Some examples of hash functions are MD5 [20], RIPEMD-128 [4], RIPEMD-160 [4] and SHA-1 [15].

Hash functions are public, i.e. no secret information is used for computing a hash value. Thus everyone who knows the function may compute the hash value and thereby check the integrity of the data.

Figure 2 illustrates how a one-way hash function is used.

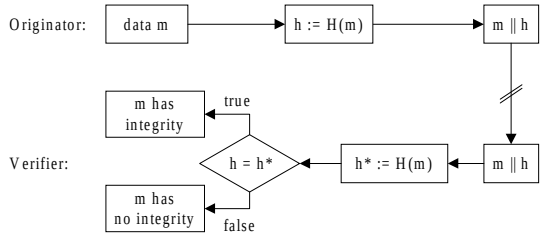


Figure 2: One-Way Hash Function H

To verify data integrity, the received hash value h is compared with the newly computed hash value h*=H(m). If h is equal to 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 the data 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).

6 Data Origin Authenticity

The following two mechanisms assure not only data integrity but also data origin authenticity:

- message authentication code (MAC), and
- digital signatures.

Just like the mechanisms for data integrity these mechanisms are detective, and the protected data again remains in plaintext.

6.1 Message Authentication Code

A message authentication code (MAC) is a one-way hash function h = H(k,m), which is parameterised 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):

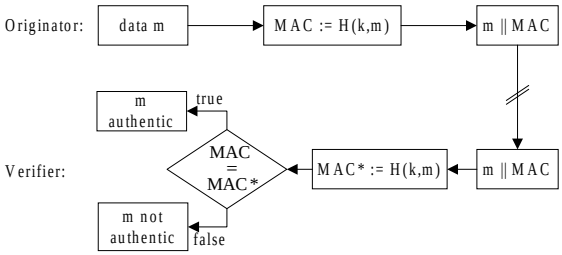


Figure 3: Message Authentication Code (MAC)

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.

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 parameterised 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).

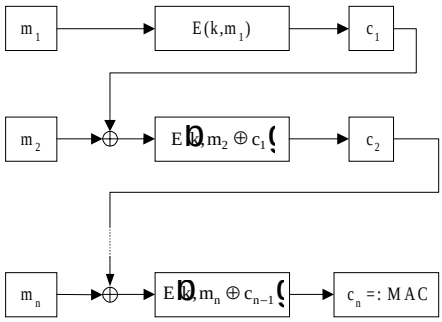


Figure 4: An example: CBC-based MAC

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

6.2 Digital Signatures

The idea and the term "digital signature" were introduced by Diffie and Hellman. In [3] 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 preimage x it is easy to calculate the image f(x), but it is computationally infeasible to find a preimage 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 secret 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 [21], DSS [18], ElGamal [5], GMR [7] and Fiat-Shamir [6].

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 9_n• 9_n where n = p·q or GF(p)• 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.

One-way hash functions used in digital signature schemes are for example MD5 [20], RIPE-MD 128 [4], RIPE-MD 160 [4] and SHA-1 [15].

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 [21]).

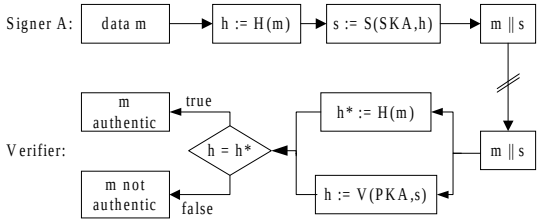


Figure 5: The Principle of a Digital Signature

Signer A wants to transmit data m and its signature to a verifier. For this A computes the hash value h 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 secret key SKA). Finally A transmits m and the corresponding digital signature s to the verifier.

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 secret key SKA, only A can compute the correct signature s to m. If h • 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 secret key used for the signature generation.

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

If the data are to be transmitted confidentially and authentically, the sender first signs the data with his

secret key and second encrypts m together with the signature using the recipient's public key.

7 Entity Authenticity

As described in the previous paragraph, data authenticity can be checked by digital signatures. Beyond that, it is often additionally 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 textfields. These fields may be used to exchange secret keys for a further confidential communication.

In the following we present the simplest version of an authentication protocol: the challenge-response protocol. This protocol can be implemented on the basis of a private-key or a public-key cryptosystem (see figures 6, 7 and 8) [11].

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 f and f^{-1} and need to share a key K. In the following we describe the unilateral authentication according to ISO 9798-2 (see figure 6).

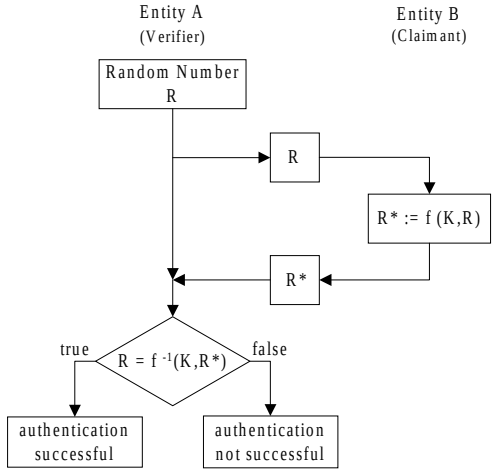


Figure 6: Unilateral Authentication Using a Private-Key Cryptosystem

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 f and the key K. Then he sends the resulting cipher R* (response) to A. Entity A decrypts the received cipher by use of f^{-1} and the 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 7):

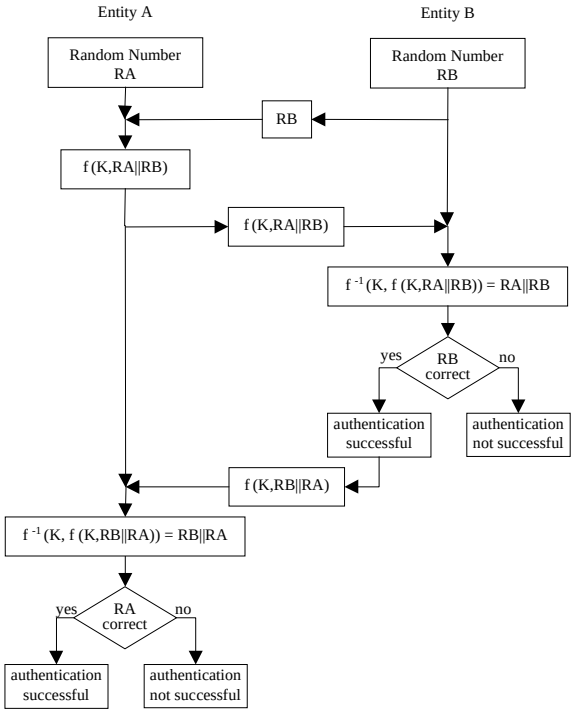


Figure 7: Mutual Authentication Using a Private-Key Cryptosystem

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 f(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 f(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 [14], triple-DES [1] and IDEA [13].
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 secret key of the claimant. The unilateral authentication is performed as follows (see figure 8):

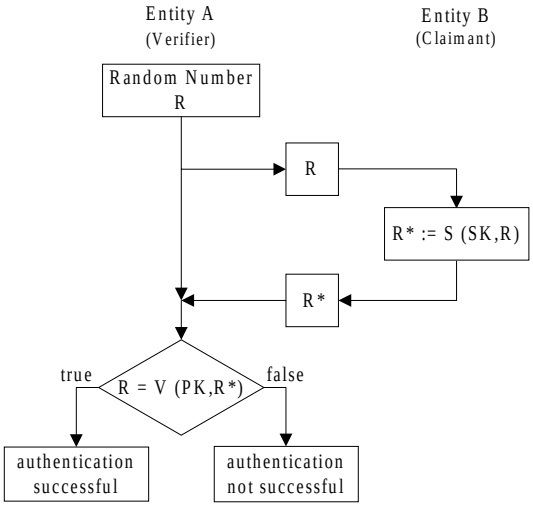


Figure 8: Unilateral Authentication Using a Public-Key Cryptosystem

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 secret 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 [11]. 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 [21], DSS [18], ElGamal [5], GMR [7] and Fiat-Shamir [6].
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 [11]. 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 synchronised 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 secret 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 secret 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 [12], 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 9).

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 secret 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.

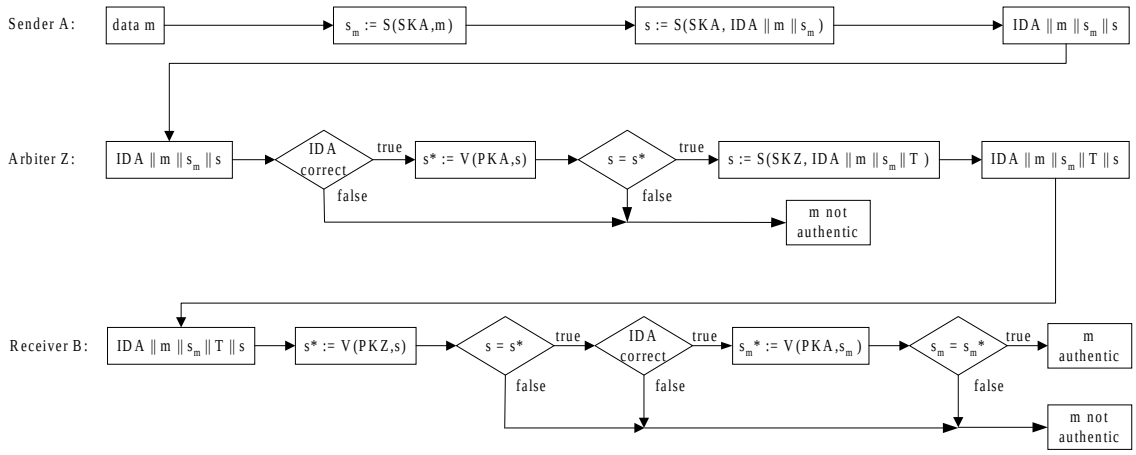


Figure 9: Arbitrated Digital Signature

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 secret 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 [9, 10]. For the issuing of certificates a trustworthy authority, a so-called trust center (TC), is needed. Trust centers authenticate the link of users to their public keys, and can provide further services like non-repudiation, revocation handling, timestamping, 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 Security for Multimedia

Whether or not the presented security functions can be used easily for multimedia data and multimedia applications, must be checked for each kind of application separately. The following problems may result due to the data formats and the amount of data:

- For reasons of performance, instead of encrypting the whole data only special parts of the entire data are encrypted (partial encryption). If the selection is well chosen, a sound confidentiality of the whole data can be achieved.
- All security functions described in this paper that can be used for checking the integrity and authenticity of some data have the property that if one bit of the input is changed, the checks will fail. Thus, if the authenticity of, say, graphic data is needed, it seems to be difficult to define a suitable input for digital signature schemes. Here a kind of data has to be used, which is not altered by the allowed operations such as scaling and conversion of picture formats. Appropriate methods include using characteristic vectors, which typify the graphic data as unique and are not influenced by allowed graphical operations.
- In order to provide non-repudiation services a proper security infrastructure has to be established and a security policy must be defined.

Furthermore the three basic threats, which we presented in paragraph 2, cannot cover the whole spectrum of the security requirements on multimedia. The essential, fourth basic threat to multimedia is:

- threat of originality (unauthorised duplicating of data).

The originality of data guarantees that they are presented in an unchanged form and not in a copy. For the protection of originality detectives mechanisms are used, e.g. copyright protection, digital watermarking and steganography. These mechanisms are still an issue of the present research. Additionally legal regulations, such as copyright protection, patent protection and computer criminal law, are trying to find countermeasures against this threat.

With respect to the security of multimedia these few examples show that there still exist a lot of open problems, which result in particular from the complexity of the multimedia data and their applications. It has to be analyzed if and in which way the IT-security mechanisms presented in this paper can be used to guarantee multimedia security.

11 References

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

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

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

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

[5] 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.

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

[7] 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.

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

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

[10] 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.

[11] 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).

[12] 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).

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

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

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

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

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

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

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

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

[21] 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.

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

The Functions of Digital Signatures from a Legal Point of View

Anja Miedbrodt
Research Area 403
Senckenberganlage 31
60054 Frankfurt/Main
049-69-79823361

a.hesse@jur.uni-frankfurt.de

ABSTRACT

This paper provides on overview of the necessity of the digital signature for electronic commerce and describes the legal requirements of the German Digital Signature Act and the Signature Ordinance compared with the Proposal of the European Commission on a common framework for electronic signatures.

KEYWORDS

Electronic signature, value of evidence of digital signatures, technical requirements of the German Digital Signature Act and the Signature Ordinance.

1 Introduction

The success of the Internet depends on the offered contents. It is doubtful if originators will publish their work in the Internet without a sufficient legal and technical protection. Equally the general public will only use this medium, if there is a lot of information available and if it's integrity is guaranteed.[16]

Because of the possibility of:

- digital storage and sending of data without any lost of quality,
- cheap creation and distribution of copies,
- exact access to every point of a stored and individual retrieval work, without necessity to buy the whole work,
- digital alteration, combination and disfigurement of work [18]

the right of the originator to exploit his work is endangered. The protection is only guaranteed in the interaction of legal and technical facilities. The protection has to be orientated on the attacks of third parties.

Beside the problem of proving the integrity of a message the impossibility to allocate a message to it's originator (the problem of authenticity) also exists in electronic networks. This threats the electronic commerce, because it is impossible to enforce obligations. Without an evidence of integrity and authenticity a court could not be convinced of obligations.

According to the state of art one possible resolution for that is the concept of digital signatures. But there are quite a number of technical and organizational requirements for digital signatures to fulfil their functions. One of these requirements is for example the cryptographic (mathematical) security of used procedures. A secure generation of the keys, distribution, allocation, administration and maintenance of revocation lists as well as the storage of the private keys are further requirements.

Specification of technical requirements for providers of digital signature products is a widespread object of state legislation[2] or the work of standards organisations[3] as well as intended international agreements[4] and transnational guidelines[5].

The German Digital Signature Act (enacted on 1st August 1997) and some Digital Signature Acts of several States of the United States are forerunners on the level of the state regulation.

According to a provisional stature of research these regulations could be divided at least into two approaches, which are partly founded on opposite goals:

- On the one hand there are detailed legal technological and organizational requirements for digital signatures to actually determine the integrity and the authenticity of a message. This approach is for example followed by the German Digital Signature Act. These Acts aim to promote the electronic commerce by legal conditions of actually secure digital signatures. In summary this approach aims to provide the guarantee to prove obligations by technical law.

---

[2] Beside the German Digital Signature Act is in Europe the Italian one enacted. Other European countries like Austria and Denmark plan to include in their drafts the expected Guideline of the European Union. In the United States, Utah, California, Florida, Illinois and Massachusetts for example did enact Signature Acts.

[3] X 509, FIPS 140- 1, ITSEC, „Department of Defense Trusted Computer Evaluation Criteria", Common Criteria (CC).

[4] OECD-Crypto-Guidelines 1997, UNCITRAL- Draft Uniform Rules On Electronic Signatures.

[5] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD).

- On the other hand the legislation aims at removing legal obstacles for digital signatures and electronic commerce. These provisions aim at promoting digital signatures by nonregulation. Mainly they face provisions, requiring hand-written signatures, because of historical reasons. Object of these Acts are for example the explicit admission of digital signatures for the communication with offices or as an evidence in proceedings. The guarantee of the technical security is left to the market. An example for this approach is the Proposal of the European Commission on a common framework for electronic signatures (COM (1998)297/2/98/0191 (COD), passed at 13th Mai 1998.
- Between them there are a lot of hybrid approaches.

2 How are the digital signatures embedded in the German legal system?

2.1 Writing Form

Generally according to the German civil law oral contracts are valid, unless a rule of law requires a hand-written signature[6] or the parties have arranged by contract the use of the writing form.[7] The reason for writing requirements are:

- to protect the parties against precipitation,
- to prove the agreement and to create certainty about the obligations and rights,
- sometimes to make possible supervision by State.[14]

The evidence function of the hand-written signature is based on cultural experience. The connection of the declaration with a durable medium guarantees the integrity. The authenticity is provided by the connection between the declaration and hand-written signature as a genuine biometric feature of the originator.[6]

At the moment the digital signature doesn't satisfy this signature requirement, because it is not hand-written. But the legislator is considering if it is necessary to introduce the electronic form which will be equal to the hand-written signature.

2.2 Evidence Law

In the course of a proceeding, parties declare facts to justify their claims. If any facts are contentious, they have to be proved. According to German code of civil procedure, parties can use every evidence, including digital signed declarations. This is not natural, because other legal systems require, that contacts have to be in written form to be enforceable.[8]

These evidences will be considered by court in the course of free consideration of evidence[9], which means that the judge has to be convinced of the facts. There is only a practical level of certainty required, because the complete certainty can not be achieved.[15]

How the judge assets several evidences is entirely up to him. Only in the course of the documentary evidence he is limited, because there are some provisions, which attach some facts to legal presumptions of genuineness and freedom of damage of a declaration included in a document.[3] That's why the document is a reliable evidence.

But a document is only a mental declaration in letters. [15] Because of the last requirement the digital signed document can not serve as a documentary evidence. [4,17]

But it can provide as real evidence and expert evidence.[17] The actually capability of the digital signature to provide the evidence of integrity and authenticity plays an important role for the consideration of evidences.

That's the reason why the German Digital Signature Act aims in accordance with § 1 paragraph 1 Signature Act at the establishment of general conditions under which digital signatures are deemed secure.

It is possible, that the increasing experience in the usage of digital signature products could be appreciated by the means of prima facie evidence.[6]

Prima facie evidence is a way to limit the free consideration of evidence by the judge. If facts are certain, normally based on particular reasons, in adjudicating a dispute, a court shall presume that these reasons are proved.

In opposite to the fundamental principal that the plaintiff has to prove all facts, which support his action, in the case of prima facie evidence he only needs to demonstrate and prove the facts, which indicate the typical reasons. The opponent, which declares the divergence of these typical reasons has to prove it.

3 The German Digital Signature Law

As above mentioned the Digital Signature Act provides an infrastructure for secure digital signatures.

Important corner pillars for the actual security of digital signatures according to the Germans Digital Signature Act and the Digital Signature Ordinance, enacted on 1st November 1997, are:

- governmental-licensed certification authorities,[10]
- by the certification authority informed holders of the keys,[11]

[6] § 126 BGB.

[7] § 125 sentence 2 BGB.

[8] for example § 2-201 Statute of Frauds (Uniform Commercial Code).

[9] § 286 ZPO.

[10] § 4 Digital Signature Act.

[11] §§ 6, 16 number 3 Digital Signature Act, § 4 Signature Ordinance.

- qualified suitable technical components. The requirements are described as aims to offer space for innovation.[12]
- supervision by the competent authority and regularly inspections. [8][13]

The legal requirements for the qualified suitable technical components are applied to the certification authorities as well as to final-consumers and to those, who offer their technical components commercially to final-consumers. A governmental control of the compliance with these legal requirements takes only place in the course of the licensing of the certification authorities. These legal requirements shall be deemed met when the competent authority has been notified by means of a security concept of measures ensuring compliance with the security requirements in this Act and the Ordinance and their implementation has been checked and confirmed by a body recognised by the competent authority.[14] The security concept shall include all security measures and especially an overview of the applied technical components and a description of the procedures used in certification. The concept shall be modified without delay in cases of security-relevant changes.[15]

The competent authority shall keep a catalogue of suitable security measures and shall publish this catalogue in the Federal Gazette[16]. These measures shall be taken into account in the preparation of the security concept.[17] The catalogue shall be prepared in keeping with provisions of the Federal Agency for Security in Information Technology. Experts from the areas of industry and science shall be consulted in this regard.[18]

These technical requirements of the Digital Signature Act and the Signature Ordinance shall be faced in the following.

It should be distinguished between the requirements for the keys, the procedure for the generation and examination of the signature and the requirements for the services that have to be provided by the certification authorities.

3.1 Requirements for the Keys

The technical components required for generation of signature keys must work in such a manner that it is nearly certain that any given key can occur only once and that a private key cannot be derived from the relevant public key.[19]

The competent authority shall publish in the Federal Gazette an overview of the algorithms and pertinent parameters considered suitable for generation of signature keys. Information published in this way shall include the date until the suitability is valid. This date should be at least six years after the time of assessment and publication. The suitability shall be redetermined on a yearly basis and as required. Suitability shall be considered present if, throughout a certain time period, any undetectable forging of digital signatures or manipulation of signed data can be ruled out with near certainty, by means in keeping with current scientific and technological standards. Suitability shall be determined in keeping with provisions of the Federal Agency for Security in Information Technology, taking relevant international standards into account. Experts from the areas of industry and science shall be consulted in this regard. [20]

The secrecy of private keys must be assured, and it must not be possible to duplicate keys.[21]

This requires for the storage of the key a technical component (for example a smartcard), which could not be compromised according to the state of the art.

Security-relevant changes in technical components must be apparent for the user.[22]

If such changes have been taken place, the security of the technical component doesn't work sufficiently any more. It could be apparent for example through failure.

This should protect users from security-relevant manipulation, especially from disclosure of their private keys.

Testing of technical components must conform to the E 4 Standard of the "Criteria for assessment of the security of information technology systems" (ITSEC) and must be rated as "high".[23]

The Ministry of Home Affairs with the agreement of the Ministry of Commerce has announced the Common Criteria for Information Technology (CC) version 1.0. There are now valid evaluation critera. Manufacturers and vendors of information technology products and governmental offices could apply for a certificate based on the Common Criteria at the

---

[12] § 14 Signature Act, §§ 16, 17 Signature Ordinance.

[13] §§ 13, 16 number 5 Signature Act, § 15 Signature Ordinance, 8.

14 § 4 paragraph 3 Sentence 3 Signature Act Such recognised authorities are Federal Agency for Security in Information Technology, Debis Systemhouse Security Services GmbH, TÜV Information Technology GmbH, TÜV product Service GmbH (DuD 1998, 236).

[15] § 12 paragraph 1 sentence 2 Signature Ordinance.

[16] § 12 paragraph 2 and § 16 Paragraph 6 Signature Ordinance. A first draft was presented on 18th November 1997 by the Federal Agency for Security in Information Technology. In the meantime drafts have been published by the competent authority. The final draft will follow.

[17] §§ 12 paragraph 2 sentence 2, 16 paragraph 6 sentence 2 Signature Ordinance.

[18] §§ 12 paragraph 2 sentences 3-4, 16 paragraph 4 sentences 3-4 Signature Ordinance.

[19] § 16 number 6 Signature Act , § 16 paragraph 1 sentence 1 Signature Ordinance.

[20] § 17 paragraph 2 Signature Ordinance.

[21] § 16 paragraph 1 sentence 2 Signature Ordinance.

[22] § 16 paragraph 1 sentence 3 Signature Ordinance.

[23] § 17 paragraph 1 Signature Ordinance.

Federal Agency for Security in Information Technology and at other institutions recognised by the competent authority. [13]

3.2 Requirements for the Procedure of Establishment and Testing the Signatures

3.2.1 No Derivation Of The Private Key

The technical components required for generation or verification of digital signatures must function in such a manner that the private signature key cannot be derived from the signature and the signature cannot be forged by any other means.[24]

This provision takes into account, that an attacker which has the public key and sufficient time as well as computering capacity, could try out all keys of the limited number of keys till he has found the right one. [10]

Digital signature products can only reach cryptographic security at most, that means it must be impossible with limited computering capacity in a sufficient short time to generate new and valid signatures. [10]

The used technical components have to be in accordance with the E 4 Standard of the ITSEC and must be rated as "high".[25]

Considered suitable algorithms and pertinent parameters for generation of the keys shall be published in the Federal Gazette.[26]

3.2.2 Signature Component

Use of the private signature key must be possible only following identification of the holder and must require proper possession and knowledge; the key must not be disclosed during use.[27] To realise these requirements the use of a smartcard is necessary. Protection from software is not sufficient.

Biometrical characteristics may also be used for the identification of the signature key holder.[28] The technical components required for collecting identification data must function in such a manner that they do not reveal identification data and that the identification data is stored only on the data storage medium with the private signature key.[29] Security-relevant changes in technical components must be apparent for the user.[30]

The used technical components have to be in accordance with the E 4 Standard of the ITSEC and must be rated as "high".[31]

3.2.3 Component For Display [32]

Because of a technical manipulation or other technical failure it could be happen, that data will be unintentional signed or that other data will be signed as displayed.

That's why the Act requires that the technical components for display of data for signing must work in such a manner that the signing person can reliably determine what data is to receive the signature; that a digital signature is provided only at the initiation of the signing person; and that such initiation is clearly indicated in advance.[33] The evaluation has to comply with E 2 of ITSEC and must be rated as "high.[34]

If technical components are commercially provided to third parties for use, clear and reliable interpretation of the relevant data must be assured, and the technical components must automatically be checked for genuineness when used.[35] Security-relevant changes in technical components must be apparent for the user.[36]

Technical components have to be in accordance with E 4 of ITSEC and must be rated as „high".[37]

3.2.4 Component For Verification

The technical components required for verifying signed data must function in such a manner that the verifying person can reliably establish what data has received the digital signature; that the verifying person can reliably establish the identity of the signature key holder; and that the correctness of the digital signature is reliably verified and appropriately displayed.[38]

The technical components for verifying certificates must permit clear and reliable determination of whether verified certificates were present, without having been invalidated, in the register.[39] The technical components must permit adequate determination, as necessary, of the contents of signed data.[40]

If this technical components are commercially provided to third parties for use, clear, reliable interpretation of the relevant data must be assured, and the technical components must automatically be checked for genuineness when used.[41] Security-relevant changes in technical components must be apparent for the user.[42]

---

[24] § 16 paragraph 2 sentence 1 Signature Ordinance.

[25] § 17 paragraph 1 sentence 2 Signature Ordinance.

[26] § 17 paragraph 2 Signature Ordinance.

[27] § 16 paragraph 2 sentence 2 Signature Ordinance.

[28] § 16 paragraph 2 sentence 3 Signature Ordinance.

[29] § 16 paragraph 2 sentence 4 Signature Ordinance.

[30] § 16 paragraph 2 sentence 5 Signature Ordinance.

[31] § 17 paragraph 1 Signature Ordinance.

[32] Regarding to the problem, that the Signature Ordinance doesn't determines the format of the data. [11]

[33] § 16 paragraph 3 sentence 1 Signature Ordinance, § 14 paragraph 2 sentence 1 Signature Act.

[34] § 17 paragraph 1 Signature Ordinance.

[35] § 16 paragraph 3 sentence 5 Signature Ordinance.

[36] § 16 paragraph 3 sentence 6 Signature Ordinance.

[37] § 17 paragraph 1 Signature Ordinance.

[38] § 14 paragraph 2 sentence 2 Signature Act.

[39] § 16 paragraph 3 sentence 3 Signature Ordinance.

[40] § 16 paragraph 3 sentence 4 Signature Ordinance.

[41] § 16 paragraph 3 sentence 5 Signature Ordinance.

42 § 16 paragraph 3 sentence 6 Signature Ordinance.

The technical components have to be in accordance with E 2 of ITSEC and must be rated as "high", unless these components are commercially provided to third parties for use. In this case they have to be conform to E 4 of ITSEC.[43]
On overview of suitable algorithms are published by the competent authority.[44]

3.3 Requirements For The Services Performed By The Certification Authorities

3.3.1 Generation Of Keys

The Signature Act does not contain any provision about the question, who has to generate keys, either the user or the certification authority.
If the certification authority provides signature keys, this authority shall take precautions to prevent any disclosure of private keys[45] and any storage of private keys by the certification authority.[46] Similar precautions shall also apply to personal identification numbers and other data used to identify the signature key holder in conjunction with the data storage medium with the private signature key[47] and to prevent from unauthorised access.[48] Storage of private signature keys by the certification authority shall not be permitted for the future, because this would endanger the possibility to prove obligations.
In this case somebody, who sends signed data could declare, that the certification authority has been copied and misused his private key. [9][49]
If the signature key holder generates signature keys, the certification authority shall reliably establish whether the signature key holder uses suitable technical components, pursuant to the Digital Signature Act and the Signature Ordinance, for storage and use of the private signature key.[50]

3.3.2 Issue Of Certificates

The certification authority shall take precautions to protect the components used to prepare the certificates against unauthorised access.[51]
In accordance with the state of art the validity period for a certificate, which has to be contained in the certificate[52], shall be no longer than five years and shall not exceed the period during which the applied algorithms and pertinent parameters are assessed to be suitable pursuant to § 17 (2) of the Signature Ordinance.[53]
The certification authority shall take measures to prevent undetected forgery or manipulation of the data intended for certificates.[54] These measures require especially repeated internal inspections and spot checks, which compare the content of certificates and application for certificates. [8]

3.3.3 Repository

To provide a control of the validity of the certificates the certification has to maintain a repository.[55]
The technical components used to store certificates in verifiable form[56] must function in such a manner that only authorised persons can make entries and changes[57], that the invalidation of a certificate cannot be undetectably rescinded, and that information can be checked for genuineness.[58] Protection against unauthorised retrieval are necessary. The information must include mention of whether the verified certificates were present at the given time, without having been invalidated, in the register of certificates.[59] Security-relevant changes in technical components must be apparent for the user.[60]

3.3.4 Time Stamping Service

Timestamps are necessary to provide an evidence that data has been presented at a certain moment. This is important for instance if the certificate has been revoked.
The certification authority shall take precautions to protect the technical components used to generate time stamps from unauthorised access.[61]
The technical components with which time stamps are generated must function in such a manner that the valid official time[62], without any distortion, is added to the time stamp when it is generated.[63] Security-relevant changes in technical components must be apparent for the user[64].

Proposal for a European Parliament and Council Directive on a common framework for electronic signatures

[43] § 17 paragraph 1 Signature Ordinance.
[44] § 17 paragraph 2 Signature Ordinance.
[45] § 5 paragraph 2 sentence 1 Signature Ordinance.
[46] § 5 paragraph 4 sentence 3 Signature Act.
[47] § 5 paragraph 2 Sentence 2 Signature Act.
[48] § 11 Signature Ordinance.
[49] to the encryption-dispute [5], to the risk of a key recovery [1]
[50] §§ 14 paragraph 1 Signature Act, 16 number 3, § 5 paragraph 1 Signature Ordinance.
[51] § 11 Signature Ordinance.
[52] § 7 paragraph 1 number 5 Signature Act.
[53] § 16 number 4 Signature Act, § 7 Signature Ordinance.
[54] § 5 paragraph 4 sentence 1 Signature Act.
[55] § 4 paragraph 5 sentence 3 and § 5 paragraph 1 sentences 2 Signature Act.
[56] § 4 paragraph 5 sentence 3 and § 5 paragraph 1 Sentence 2 Signature Act.
[57] § 11 Signature Ordinance.
[58] § 16 paragraph 4 sentence 1 Signature Ordinance.
[59] § 16 paragraph 4 sentence 2 Signature Ordinance.
[60] § 16 paragraph 4 sentence 4 Signature Ordinance.
[61] § 11 Signature Ordinance.
[62] § 1 paragraph 4 Time Act
[63] § 16 paragraph 5 sentence 1 Signature Ordinance.
[64] § 16 paragraph 5 sentence 2 Signature Ordinance.

On 13th Mai 1998 the European Commission has passed a Proposal for a common framework for electronic signatures.
The European Parliament has to consent to this Proposal. A decision is still pending. If this guideline will be enacted the Member States have to transform it in their legal systems.
This proposal aims at facilitating the use of electronic signatures as well as providing for their legal recognition[65]. It aims at enabling the use of electronic signatures with an area without internal frontiers by focusing on the essential requirements for certification services and leaves detailed implementation provisions to the Member States.[66]

3.3.5 Field of application

The field of application of the Proposal is unlimited.

3.3.5.1 Technology- neutrality

While the digital signature technology is a recognised procedure to provide a proof of the integrity and authenticity of a message, according to the opinion of the European Commission, a Directive at the European level should be technology-neutral and should not focus only these kinds of signatures. That's why the Proposal describes the electronic signature functionally and not technically.[12] Since a variety of authentication mechanisms is expected to develop, the scope of the Directive should be broad enough to cover a spectrum of "electronic signatures", which would include digital signatures based on public-key cryptography as well as other means of authenticating data.[67] Only in the Annex there are special provisions concerning the content of a certificate[68] and the obligations of certification service providers in a public-key- infrastructure, based on digital signatures.

3.3.5.2 Contractual freedom

The freedom of the parties to agree among themselves the terms and conditions under which they accept electronically signed data should be respected to the extent allowed by national law.[69]
That's why electronic signatures used within closed groups, for example, where contractual relationships already exists, should not automatically fall within the scope of the Proposal. Contractual freedom should prevail in such a context.[70]

3.3.6 Concept of regulation

The European Commission recognises, that a legal framework is mainly needed for certificates to enable the authentication of electronic signature of an signing individual.[71]
Nevertheless in the opposite of the German Digital Signature Act the possibility to prove obligations shall be not provided by detailed technical requirements. Ensuring legal recognition of electronic signatures and of certification services is deemed to be the most important issue in the area.[72] The guarantee of technical security is left to the market. But this concept requires a functioning market, in which no monopoly exists and where security mechanism is not subject to a limiting regulation and security is an effective argument for purchase. [12]
It might be possible, that the lack of detailed technical requirements hinder the internal market of certification services, because the Proposal provides no common level and no comparability for the security. This would be the opposite of the objectives of the Proposal.[73]

3.3.6.1 Liability rules

The liability rules[74] shall support the trust-building process for both customers and business, that rely on the certificates and service providers, and thus shall promote the broad acceptance of electronic signatures.[75]
The certification service provider are liable to any person who reasonably relies on the certificate for:

- accuracy of all information in the certificate as of the date it was issued, unless the certification service provider has stated otherwise in the certificate. They are not liable for errors for information provided by the person to whom the certificate is issued, if the certification service provider can demonstrate that he has taken all rea-

[65] Art. 1 Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD).

[66] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), need for harmonisation, page 5.

[67] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), background, note 2. page 3, aim and scope of the Directive note 2, page 6.

[68] A certificate which complies with the requirements of Annex I is called qualified certificate.

[69] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), note 9.

[70] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), background, Note 4, page 3.

[71] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), aim and scope of the Directive, Note 5, page 6.

[72] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), background, Note 5, page 3.

[73] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), need for harmonisation, page 5.

[74] Art. 6 Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD).

75 Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), aim and scope, note 8, page 7; note 11, page 10.

sonably practicable measures to verify the information,[76]

- assurance that the holder identified in the certificate held, at the time of the issuance of the certificate, the signature creation device corresponding to the signature verification device given or identified in the certificate,[77]
- in cases where the certification service providers generates the signature creation device and the signature verification device, assurance that the two devices functioning together in a complementary manner.[78] The certification service providers can limit their liability by including limits of the use of the certificates[79] and by indicating a limit on the value of transactions.[80] This provision has to be incorporated in the national legal systems. Further liability provisions are based on the national laws.

3.3.6.2 Technical requirements

The legal recognition of electronic signatures is based on criteria, which are described in Annex II. The compliance with these requirements are not linked to any prior authorisation or accreditation.[81] Certification Service providers should in general be free to offer their services without prior authorisation. In accordance with the opinion of the European Commission there is no immediate need to ensure the free circulation of certification services by harmonising justified and proportionate national restrictions on the provision of these services.[82]

That's why the Proposal determines, that Member States shall not make the provision of certification services subject to prior authorisation.[83]

Regardless of that, Member States may introduce or maintain voluntary accreditation schemes aiming at enhanced levels of certification service provision as a means to gain the confidence of customers[84] and to give the certification service provider the possibility to benefit from the legal validity of the associated electronic signatures by means of voluntary accreditation schemes linked to common requirements.[85]

These requirements are not as detailed as the requirements, demanded by the German Digital Signature Act. The catalogue requires beside organizational and personal requirements, the usage of trustworthy systems and use of products that ensure protection against modification of the products so that they can not be used to perform functions other than those for which they have been designed. They must use electronic signature products that ensure the technical and cryptographic security of the certification services supported by the products. Further more provisions order to take measures against forgery of certificates and in cases where the certification service providers generate private cryptographic signature keys, they shall guarantee the confidentiality during the process of generating those keys.[86]

The renunciation of minimal technical requirements can complicate the enforcement of a suitable security level [12] and with it, the recognition of certificates, issued by provider from foreign countries.

3.3.6.3 Legal recognition of electronic signatures

The Proposal orders[87], that Member States shall ensure that an electronic signatures, which are based on a qualified certificate according to Annex I issued by a certification service provider, which fulfils the requirements set out in Annex II,

- satisfy the legal requirements of a hand-written signature;
- are admissible as evidence in legal proceedings in the same manner as a hand-written signatures.

This provision shows the contrary nature of the Proposal and the German Digital Signature Act. The Proposal is problematic, because it determines legal effects without safeguarding technical security.

The equal status of the hand-written signature and the electronic signatures requires clarity about the functions of the hand-written signature and that the electronic signature provides actually the proof of the integrity and authenticity.

This requires technical requirements, which can not only determined and regulated by the market. The liability rules are not a suitable instrument, because they can't achieve the same security like a governmental licensing scheme.[7]

---

[76] Article 6 (1) (a) (2) Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD).

[77] Article 6 (1) (d) Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD).

[78] Article 6 (1) (d) Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD).

[79] Article 6 (3) Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD).

[80] Article 6 (4) Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD).

[81] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), aim and scope of the Directive, Note 7, page 7.

[82] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), Note7, page 9.

[83] Article 3 § 1 Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD).

[84] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), background, Note 3, page 3.

[85] Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD), aim and scope of the Directive, Note 4, page 6.

[86] Annex II (f) Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD).

[87] Art. 5 Proposal on a common framework for electronic signatures COM(1998)297/2/98/0191(COD).

4 Acknowledgments

My thanks to Dr. Johann Bizer for giving useful advices and help.

5 References

[1] Abelson, Harold, Anderson, Ross, Bellovin, Steven M., Benaloh, Josh, Blaze, Matt, Gilmore, John, Neumann, Peter G., Rivest, Ronald L., Schiller, Jeffrey I., Schneider, Bruce, Risiken von key Recovery und Trusted Third Party-Verschlüsselung, DuD 1998, 14 .

[2] Begründung zur Signaturverordnung,.

[3] Bizer, Johann, Hammer, Volker, Elektronisch signierte Dokumente als Beweismittel, DuD 1993, 619.

[4] Bizer, Johann, Beweissicherheit im elektronischen Rechtsverkehr in A. Haratsch/D. Kugelmann/U. Repkewitz (Hrsg.), Herausforderungen an das Recht der Informationsgesellschaft, Stuttgart (Boorberg Verlag) 1996, 141.

[5] Bizer, Johann, Kryptokontroverse Der Schutz der Vertraulichkeit in der Telekommunikation, DuD 1996, 5.

[6] Bizer, Johann, Digitale Dokumente im elektronischen Rechtsverkehr, in: Detlef Kröger (Hrsg.), Internet für Rechtsanwälte und Notare, Neuwied (Luchterhand) 1997, 148.

[7] Bizer, Johann, Miedbrodt, Anja, Die digitale Signatur im elektronischen Rechtsverkehr, Deutsches Signaturgesetz und der Entwurf der europäischen Richtlinie, in: Marc Andre Gimmy/Detlef Kröger (Hrsg.), Rechtspraxis im Internet (i.E.)

[8] Deutscher Bundestag: Drucksache 13/7385 vom 09. April 1997.

[9] Federrath, Hannes, Schlüsselgenerierung in Trust Centern, DuD 1997, 98.

[10] Fox, Dirk, Fälschungssicherheit digitaler Signaturen, DuD 1997, 69.

[11] Fox, Dirk, Zu einem prinzipiellen Problem digitaler Signaturen, DuD 1998, 386.

[12] Fox, Dirk, Grimm, Rüdiger, Entwurf einer EU-Richtlinie zu Rahmenbedingungen „elektronischer Signaturen", DuD 1998, 407.

[13] Mackenbrock, Common Criteria (Version 2.0), Gemeinsame Kriterien für die Prüfung und Bewertung der Sicherheit in der Informationstechnik, Seite 1, http://www.bsi.bund.de/literat/doc/cc_20d.htm.

[14] Palandt, 56.Auflage, München 1997, § 125 Rdn. 1.

[15] Putzo, Thomas, ZPO, 19. Auflage, München 1995, § 286 Rdn.2.

[16] Röhm, Alexander W., Wilop, Karsten, Urheberrechtlicher Schutz im Internet, DuD 1998, 250, 251.

[17] Roßnagel, Alexander, Das Signaturgesetz, DuD 1997,

[18] Thomaschki, Kathrin, Europäisches Urheberrecht in der Informationsgesellschaft, DuD 1998, 265.

Secure Container Technology as a Basis for Cryptographically Secured Multimedia Communication

Ulrich Kohl

IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120, USA
+1 (408) 927 1867

kohl@almaden.ibm.com

ABSTRACT

Multimedia applications pose high requirements on their security. In this paper, two different categories of security technologies are described and discussed with regard to their ability to secure communication and commerce of multimedia documents. After the introduction, the security requirements of multimedia systems are depicted. Section 4 gives an overview of cryptographic operations and their use in existing Internet security solutions. Section 5 describes the concept of cryptographically secured containers using the IBM Cryptolope technology as an example. A summary concludes the paper.

KEYWORDS

Information Commerce, Multimedia Security, Internet Security, Secure Containers, Encryption

1 Introduction

Information commerce, enabled by the new communication technologies, is one of the most demanding applications for security. The reason is obvious, as (real) money is inherently involved. Moreover, commerce of multimedia documents can address a mass market, being able to handle electronic versions of goods like:

- News, papers, magazines, books (texts and graphics),
- Music, songs, albums (audio),
- Video clips, movies (video),
- Computer software (binary data).

The goal of multimedia information commerce is the trade of digital content. Of particular interest is not only the data, but also its useright and copyright. The range of the communication and business models is large. For example, the communications may be point-to-point, multicast, or broadcast; the business model may be pay-per-view or subscription; try&buy features may be required.

The requirements in the electronic marketplace are basically the same as these in the traditional marketplace: Somebody who paid for something should be allowed - and restricted - to use the media in the way it was negotiated. This imposes requirements in all fields of security.

The shortcomings of Internet protocols in the security area were well known and tolerated ever since the protocols were first deployed in the scientific world they originated from [2]. New concepts were developed and existing ideas were adapted in order to facilitate secure electronic commerce on the Internet. All of these new mechanisms - such as IPSP, SSL, SSH or S/MIME - attempt to secure the communication channel between two parties communicating over the Internet.

Secure Container technology takes a different approach. Instead of securing the connection, the content itself is protected. A secure container can be transmitted as user data over a non-secure network. None of the protocols of the network have to be modified in order to obtain security. Instead, dedicated secure container handlers on both the sender and the recipient are used together with a clearing house to run the secure container transactions.

2 Multimedia Security Requirements

Multimedia applications already pose strong requirements on the basic functionality of communication and storage systems and user interfaces. For example, high data rates have to be processed, communicated, stored or displayed in real-time. Systems were designed to satisfy the requirements; in order to facilitate some problems, data compression techniques have been developed.

Regarding security requirements, multimedia systems pose the same requirements as standard information commerce systems:

- At the beginning of a transaction, both content provider and client want to make sure that their respective partner is the one he claims to be, i.e. have to authenticate each other.
- Likewise, both parties will require that the content is authentic, i.e. that it has been really pub-

lished by the provider, and that the content is intact, i.e. that nobody has altered it. To be secure from eavesdroppers, the content never should be transmitted or stored in a readable format.

- Both authenticity and integrity requirements are not only applicable to the content, but also to the contract. The content provider needs to prove that a client accepted the terms, and a client needs to prove he has acquired a certain set of rights.
- Privacy of a client also may be important. No party, sometimes not even the content provider, should be able to track which client was purchasing or using which piece of content.

Additional requirements come from special properties of multimedia data. The immense data volume which has to be processed, and the production/consumption pattern of many multimedia applications favor more advanced communication protocols than just point-to-point. Broadcast or multicast communication is used to reduce the network load. The concept of broadcasting or publishing encrypted content which is purchased and decrypted by the customers on demand is called superdistribution [13]. Even on broadcast systems, a pay-per-view service should be able to be deployed. A subscription model is very reasonable for repetitive transactions, e.g. daily purchases of news, stock prices, journals, premium TV channels etc.

Several types of multimedia content also draw high attention to security issues. Movies or music albums are valuable types of content, address a mass market and thus promise high revenues for both the legal content owners and content pirates. With traditional media, professional and amateur piracy of music and movies is already well-established and will be even made easier using recordable digital media without security mechanisms.

3 Internet Security Mechanisms

Most of the current security systems, regardless whether they are incorporated in protocols or working stand-alone, are based on cryptographic algorithms. In this section, the main characteristics of some cryptographic algorithms are explained and their use in secure electronic commerce applications is outlined.

3.1 Building Blocks Of Security Solutions

3.1.1 Cryptographic algorithms

Cryptographic algorithms can generally be divided into two categories. In the case of s*ymmetric* cryptographic algorithms, the same key is used for encryption and decryption. Commonly found examples for this kind of algorithms are DES or RC4; the new Advanced Encryption Standard will fall under this category, too.

Asymmetric cryptographic algorithms (or public-key-algorithms) use a different key used for encryption and decryption. Public-Key-operations are performed with key pairs: every message which is encrypted with one of these keys can only be decrypted using the other one. If one of these keys is kept secret (the private key) and the other one is published (the public key), asymmetric cryptographic algorithms serve two purposes:

- Data which is *encrypted* with a recipient's public key can only be *decrypted* with the corresponding private key which is only known to the recipient.
- Data which is encrypted with a sender's private key can be decrypted by everybody who is in possession of a copy of the corresponding public key. This property serves as the foundation for digital signatures: The sender *signs* with the private key and the signature can be *verified* with the corresponding public key[88].

The most prominent example for an asymmetric cryptographic algorithm being used for both encryption and digital signatures is RSA; another one which is only suited for digital signatures is the Digital Signature Algorithm (DSA). Since asymmetric algorithms are relatively slow, they are never used for the encryption of large amounts of data. Today's systems typically make use of the following two concepts:

- Large amounts of data are symmetrically encrypted with a random key. This so called *session key* is asymmetrically encrypted using the recipient's public key and can thus be safely transferred. In connection oriented protocols, this phase is referred to as *key exchange.*
- In the case of digital signatures, one way *hash functions* are used to condense the data to be signed to a fixed length hash value, which is then encrypted with the sender's private key. Examples for one-way hash functions are MD5 or SHA.

Interested readers may consult a number of excellent books on cryptography, e.g. [11], and its use for secure electronic commerce solutions [5].

3.1.2 Rights Management Language

The use of cryptography helps a lot to achieve security: only the owner of the right key is able to decrypt and use encrypted information. Content providers may wish to express more complex terms and conditions of usage, though. For example, they may restrict the operations for a client (view only, copy

[88] Of course, the verifier has to trust the correctness of the public key. This can be accomplished by providing public key certificates.

once), offer rebates under special circumstances (club memberships, mass rebates). Such terms and conditions cannot be controlled by mere key ownership. Instead, flexible licensing mechanisms with so-called *rights management languages* (RML) have to be deployed. An RML is used to specify the credentials required by user to access a digital document and determines the resulting usage rights. An example for an advanced RML can be found in [4].

3.1.3 Watermarking

Partners in electronic commerce don't necessarily know each other, so a content provider may not want to believe that a client adheres to the terms and conditions and does not violate the copyright. Like in the physical world, watermarks, which ideally are not reproduceable or removable by an attacker (for attacks see [16]), can be applied to the information as a tracking mechanism.

Digital watermarks [3] are barely perceptible transformations of digital data (image, audio or video data) which can be extracted computationally. The use of digital watermarks enables different scenarios [12]:

- *Ownership watermarks* (fingerprints) can be used in order to convey ownership information. In this scenario they identify the recipient of digital documents and facilitate the detection of copyright violations. Due to its nature, ownership watermarking can only be applied when the customer is already known. In the case of superdistribution, ownership watermarking can only be done at the client station.
- *Originator watermarks* are applied by the content owners in order to mark their copyright or automatically trace slight alterations of their intellectual property. Additionally, they may be visible to prevent undetected illegal copying to media outside the system, e.g. with screen captures.
- *Captioning* is an application of digital watermarking which refers to the integration of metadata into the digital documents. This allows for example to embed usage restrictions such as "copy once" to be encoded in digital movies.

3.1.4 Tamper-resistant systems

All these security mechanisms only work properly if they are executed correctly at each participating entity. Especially the client station has to be considered a major point of attack because it is under full control of a potentially malicious owner. Servers have a higher degree of physical security, but are vulnerable for insider attacks as well. A solution to ensure security on all stations are tamper-resistant systems. A tamper-resistant system tries to detect whether attempts are made to use it improperly and stops executing then.

Tamper-resistance is important, because the information which is sold has a legal usage and thus is allowed to be decrypted for this use at the client station, so not only the cleartext, but also the parts of the system which perform the decryption and the intellectual property protection have to be protected from tampering by applying the mechanisms described above to the software itself. An architecture for tamper-resistant code and a classification of the possible attacks can for example be found in [1]; however, also tamper-resistant devices can be broken [8].

3.2 Securing Connections

Security services can be introduced at different levels of the layered structure of the Internet protocols.

Figure 1 shows the four-layered protocol stack of the Internet protocols: The Internet Protocol (IP) is a connectionless, packet-oriented transport mechanism. TCP and UDP are transport layer protocols which provide, respectively, connection oriented and connectionless transfer control services to the application level protocols. Figure 1 uses the application level protocols HTTP (WWW) and SMTP (e-mail) as examples.

An exhaustive comparison of the security mechanisms described in this section can also be found in

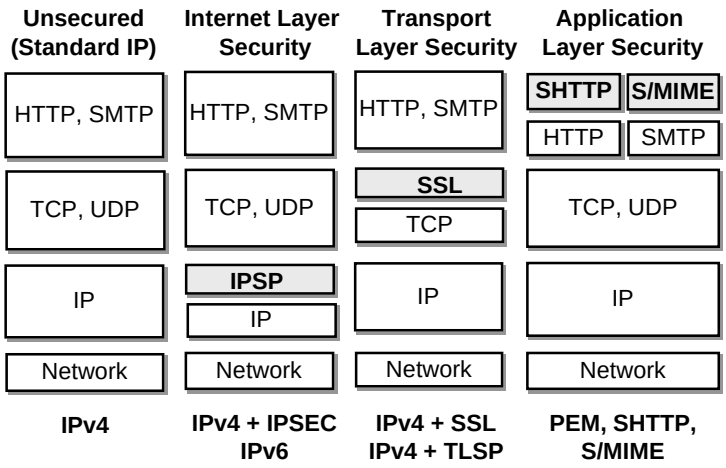


Figure 1: Integrating Security Services in the Internet Protocols

[15]; [14] focuses on the overall building blocks of a W3 Security architecture.

Internet Layer Security. As figure 1 shows, encryption and signing of all transmitted data can be integrated into the family of Internet protocols at the Internet layer. As a consequence, these security services are equally available for all application protocols, which do not even have to be recompiled in order to benefit from the security infrastructure. An IETF working group called IPSEC is in the process of standardizing the necessary protocol structures which will be available as an addition to the Internet Protocol in its current version (the so called IP Security Protocol IPSP) and as a part of the next version IPv6. The proposals are based on the use of DES for bulk data encryption and MD5 for hashing; a mechanism for performing key-exchange has not yet been standardized.

Since they involve changes to the basic Internet protocol, the main use of IP layer security mechanisms is currently in routers and firewall solutions in order to implement security gateways and virtual private networks.

Transport Layer Security. Transport layer security means protection of the transmitted data above the transport layer. The most prominent example of transport layer security is Netscape's Secure Socket Layer (SSL), which is layered on top of TCP [6]. SSL provides the services to authenticate a server, and optionally a client, to encrypt a session, and to authenticate messages.

An IETF working group is in the process of standardizing a so called Transport Layer Security Protocol (TLSP), which is in most aspects based on SSL.

SSL is intended to protect a single connection between two communicating applications at the socket layer. It protects any higher level protocol built on sockets, such as Telnet, FTP or HTTP. In order to achieve this, SSL places two layers on top of TCP. The lower layer is the SSL Record Protocol which is used for encapsulation of various higher level protocols. The upper layer is the SSL Handshake Protocol. It allows the end systems to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocols start to transmit over the encrypted channel.

SSL in version 3 supports the use of several different symmetrical algorithms for the encryption of bulk data (among them DES and RC4). Integrity checks are based on MD5 or SHA hash functions, several public key algorithms are supported for performing an initial authentication. The credentials used for performing the initial authentication and key exchange operations are X.509 certificates.

In detail, SSL provides the following security services to the higher protocols:

1. Authentication: The identities of the server and optionally also of the client are verified.

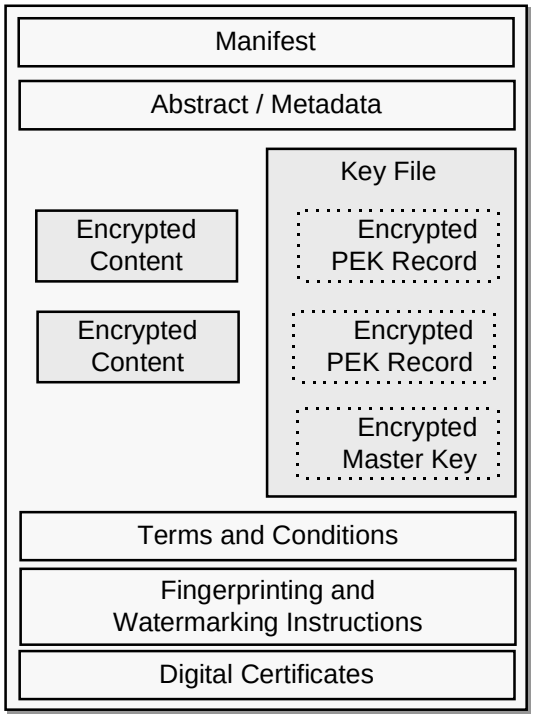


Figure 2: A Cryptolope Container

2. Confidentiality: Data to be transmitted is encrypted with the session key.
3. Integrity: Additionally to the user data, a message authentication code (MAC) is generated and transmitted.

The application-independence of SSL has the disadvantage that it can only offer point-to-point protection of the data during the communication process. In both the source and destination systems the data is in the clear. It is not within SSL's capabilities to protect the data when a host is compromised or to detect and fix the problem when a key is compromised.

Application Layer Security. Finally, as shown in figure 1, security services can be integrated into the Internet protocols at the application level. This refers to the design of new or the adaption of existing application protocols in order to integrate security features into the protocol elements.

One example of this approach is SHTTP, an extension of HTTP for security services [17]. SHTTP is a superset of HTTP and adds authentication, confidentiality, integrity and non-repudiation. The system is not tied to any particular cryptographic system, key infrastructure, or cryptographic format. Messages are encapsulated within SHTTP in various ways including encryption, signing, or MAC based authentication. Messages may be encapsulated multiple times to achieve multiple security features. Header definitions for key transfer, certificate transfer, and similar administrative functions are provided.

SHTTP does not rely on a particular key certification scheme. It includes support for RSA, in-band, out-of-

band and other forms of key exchange. Public key certificates can be provided in a message, or obtained elsewhere. As in SSL, client public keys are not required if client authentication is not needed.
Similar mechanisms are available for other application protocols as well: PEM and S/MIME, for example, are used to realize secure electronic mail and SSH is in widespread use for secure remote command execution and file transfer.

4 Protection on the Document Level

For multimedia applications, the solutions presented in the previous section have several shortcomings.
The protected multimedia documents are separated from the associated usage rights and conditions.

-There is no way for the user to prove that a document was received under certain usage conditions. Once the document is transmitted to the user, these usage conditions cannot be enforced.

-The content is decrypted at the end of the communication channel. This is acceptable if the content is only shown or played, but not if it would have to be stored in the clear at the client station.

-A connection oriented security mechanism does not allow superdistribution of larger amounts of data.

Document protection requires the document to be wrapped in a secure container at the content provider's site, and only to be unwrapped at the end user's computer. As a result, no further protection is needed, neither for the communication channel nor for the intermediate servers. Also, all of the intents of the content provider (protection, marking, etc.), and all the terms and conditions he is offering, can expressed in a tamper-evident digitally signed package. This enables superdistribution; the package can be moved freely from place to place without losing its intactness, its authenticity, and its associated terms and conditions.

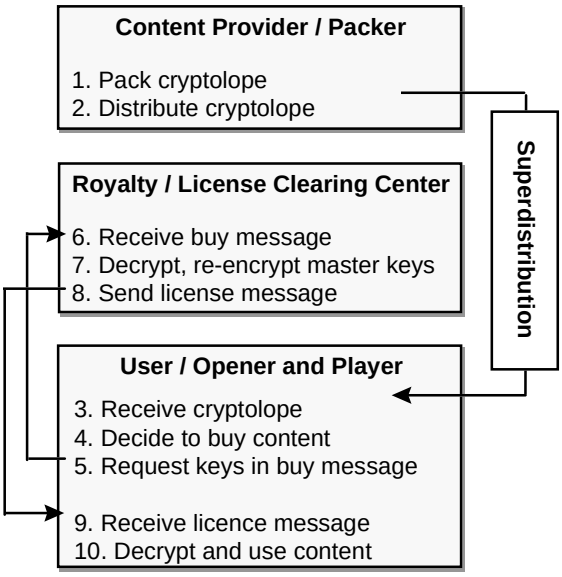


Figure 3: Cryptolope processing

IBM has coined the name Cryptolope™ (cryptographic envelope) for its document encapsulation technology [see 4]. There are others, for example DigiBox™ [18] or a system for the electronic distribution of audio data proposed by AT&T Labs [9].
As shown in figure 2, a cryptolope consists of multiple parts. In addition to the encrypted document, a cryptolope contains a clear text description of the encrypted content which serves to support a user's purchase decision. The metadata gives information about the contents as a whole, such as author, size or format and instructions on how the content may be purchased. The "real" information is stored in the encrypted content parts. For each part, a different part encryption key (PEK) is chosen. The PEKs are themselves encrypted using a master key and stored in the key records of the cryptolope.
A Cryptolope Container The cryptolope further contains the terms and conditions describing the rights associated with the content and fingerprinting and watermarking instructions which specify when and how identifying information is to be added to the documents. Digital signatures and certificates included in the cryptolope serve the purpose to authenticate the contents and optionally the users [7].
A cryptolope is created by the provider of content and can be distributed on arbitrary channels. Its security is inherently guaranteed because everybody can check the checksums and signatures, so nobody can tamper with a cryptolope and nobody can use the content without purchasing the PEKs.
The purchasing transaction requires a clearing center which acts on behalf of the content provider. A client who decides to buy some content is directed by the cryptolope instructions to an appropriate clearing center. The buy request message contains the encrypted PEK and public key certificate. The clearing house knows the master key (which could be its own private key or a shared secret symmetric key), decrypts the PEK and re-encrypts it using the client's public key. After the client received the license message containing the encrypted PEK, it can decrypt it using its private key and use it to decrypt the content itself. Figure 3 depicts the cryptolope process [10].
A cryptolope-based solution is well suited to meet multimedia security requirements:

1. Entity authentication is needed just between client and clearing house: the content provider needs not to have a special relationship with each user.
2. Every cryptolope and every message is digitally signed and includes the certificate of the signer, so it can be checked easily. The signature process is explicitly driven by the end user, so the signature can be considered as an expression of free will to sign a contract.
3. Checksums and signatures of the content parts allow to check the authenticity and integrity of the content.

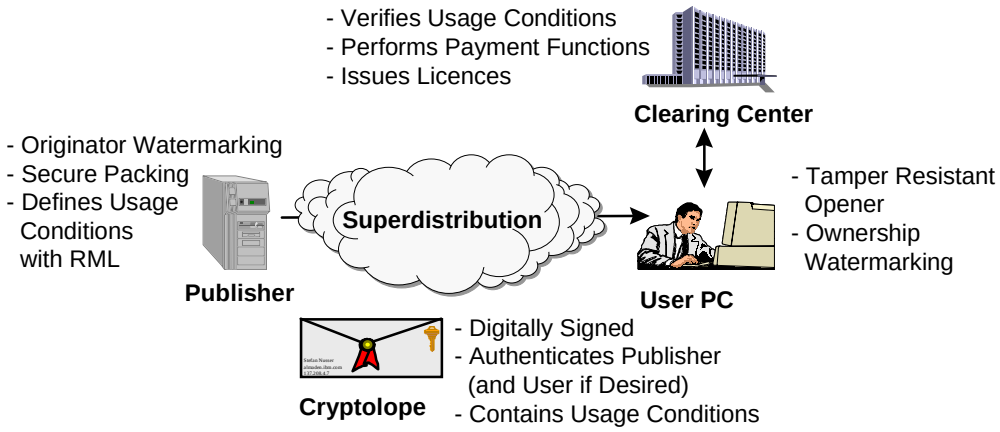


Figure 4: Use of security building blocks in the cryptolope architecture

4. Each encrypted part is confidential and can only be decrypted by an owner of the key, i.e. the content provider who created this key and the client who buys the key from a clearing center. A clearing center is able to decrypt and sell the key, but generally does not decrypt the content. The information is in clear text only at the content provider's and the client's side within the cryptolope processing environment.
5. As cryptolope processing requires dedicated opener and viewer software running on the client, code signing techniques can be applied to make the software on the client side hard to tamper with.
6. Try&buy applications are possible. Free (perhaps lower-quality) samples of the content can be added in the clear to the cryptolope. With the RML, arbitrary usage patterns can be allowed.

Clients do not need to get a cryptolope directly and online from a content provider, but can copy a cryptolope from the nearest cache and purchase the unlocking key from any authorized clearing house.

As a summary, figure 4 shows how the Cryptolope architecture makes use of the building blocks of security. Document layer security does not attempt to secure a specific communication channel; it is not even dependent on the Internet protocols as a transport mechanism.

5 Summary

With more and more computer and communication systems capable to process multimedia data, commerce of multimedia data opens a promising market. However, multimedia commerce poses even higher security requirements than established electronic commerce systems. This paper discussed different approaches to add security mechanisms to existing Internet protocols.

Internet security systems which protect communication channels often are not sufficient to meet the requirements of multimedia document commerce: the data is only protected during the communication process, but not before and afterwards. The control of the use of the information after the transmission becomes a major requirement, though.

Secure container technology promises to be a solution to this problem. Here, the main idea is to encrypt the information at its source and provide a means for a consumer to be able to decrypt it on demand. The information is encrypted and packed into a secure container which contains additional information, e.g. what information is contained, what its price is, or where and under which conditions a client can purchase the unlocking keys. Secure containers can be distributed via arbitrary, unsecure channels. Clearing centers are used to process the purchase transactions. IBM cryptolopes are an example of secure container technology. They can encompass all of the above described features and can be used to realize the idea of superdistribution.

6 Acknowledgments

The research work described in this paper was performed jointly with Jeffrey Lotspiech and Stefan Nusser.

7 References

[1] David Aucsmith, Gary Graunke: Tamper Resistant Software: An Implementation. In Proc. First International Workshop on Information Hiding, 1996.

[2] Steve M. Bellovin: Security Problems in the TCP/IP Protocol Suite. In Computer Communication Review, Vol. 19, No. 2, April 1989.

[3] Hal Berghel: Watermarking Cyberspace. In Communications of the ACM, Vol. 40, No 11, November 1997.

[4] M. Blaze, J. Feigenbaum, J. Lacy: Decentralized Trust Management. In Proc. 1996 IEEE Symposium on Security and Privacy.

[5] Warwick Ford, Michael S. Baum: Secure Electronic Commerce. Prentice Hall, 1997.

[6] Alan O. Freier, Philip Karlton, Paul C. Kocher: The SSL Protocol Version 3.0. IETF Internet-Draft, 1996.

[7] Marc A. Kaplan: IBM Cryptolopes, SuperDistribution and Digital Rights Management. Working Paper, V1.3.0,12/96. http://www.research.ibm.com/people/k/kaplan/.

[8] Markus Kuhn, Ross Anderson: Tamper-Resistance: A Cautionary Note. In Proc. Second USENIX Workshop on Electronic Commerce, 1996.

[9] Jack Lacy, James Snyder, David P. Maher: Music on the Internet and the Intellectual Property Protection Problem. White paper, http://www.a2bmusic.com/about/papers/.

[10] J.B. Lotspiech, U. Kohl, M.A. Kaplan: Cryptographic Containers and the Digital Library. In Proc. VIS '97, Vieweg Verlag, October 1997.

[11] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone: Handbook of Applied Cryptography. CRC Press, October 1996.

[12] F. Mintzer, J. Lotspiech, N. Morimoto: Safeguarding Digital Library Contents and Users – Digital Watermarking. In D-Lib Magazine, December 1997.

[13] Ryoichi Mori, Masaji Kawahara: Superdistribution: The Concept and the Architecture. In Transactions of the IEICE, Vol. E 73, No. 7, July 1990.

[14] Gustaf Neumann, Stefan Nusser: A Framework and Prototyping Environment for a W3 Security Architecture. In Proc. CMS'97, Chapman & Hall, September 1997.

[15] R. Oppliger: Internet Security – Firewalls and Beyond. In Communications of the ACM, Vol. 40, No. 5., May 1997.

[16] Fabien A.P. Petitcolas, Ross J. Anderson, Markus G. Kuhn : Attacks on copyright marking systems. In Proc. Second Workshop on Information Hiding, 1998.

[17] E. Rescorla, A. Schiffman: The Secure HyperText Transfer Protocol. Internet-Draft, March 1997.

[18] Olin Sibert, David Bernstein, David Van Wie: The DigiBox: A Self Protecting Container for Electronic Commerce. In Proc. USENIX 95 Electronic Commerce Workshop.

Video Protection by Partial Content Corruption

Carsten Griwodz
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany
Tel.: (+49) 6151 166159
griff@kom.tu-darmstadt.de

MOTIVATION

Many on-demand applications require that the same content is delivered to many different receivers in short sequence. In video on demand the goal of the content provider is the frequent sale of content in the most popular phase of their life cycle [1], which could be exploited by the introduction of caching and prefetching techniques [6]. It is not commercially feasible to restrict content access to a small group of receivers.

For such applications we want to provide a simple approach that is able to protect the content owner from data theft in the wide area network while protecting the infrastructure from an unnecessary number of transmissions. We want to provide a straightforward mechanism for these applications which can interoperate with caching systems as well as reasonably powerful servers. The mechanism should be computationally cheap, in order not to overload the server with the task of modifying the content (e.g. watermarking or encryption) for an arbitrary number of concurrent unicast transmissions. We believe that partial content corruption provides similar protection for video content as full content encryption but can still make efficient use of caching and pre-distribution for the bulk of the content, using protected unicast only for a minimal amount of data. We propose a novel scheme for inhibiting and investigating copyright violations.

1 Protecting the Cache

The initial approach towards video encryption was encryption of the whole stream. Various more efficient encryption algorithms were implemented. Maples and Spanos present in [4] an approach of encrypting only I-frames of MPEG videos. Qiao et al. [5]) propose a video encryption algorithm that works exclusively on the data bytes and does not parse the MPEG video. Still, each byte of the video data is manipulated once for each transmission. Kunkelmann et al. present [3] a variety of approaches to the partial encryption of the complete video stream for use with a security gateway. They consider a partial encryption of 10% of meaningful data appropriate for VoD applications, while full protection requires a major part of all data to be encrypted to prevent reconstruction. All of these approaches are compute intensive and put a heavy strain on a VoD server. Kunkelmann et al. report an increase of CPU utilization by 10.5% for the playback in comparison to unencrypted content.

In the typical design of throughput-oriented commercial video servers, the computing power is considered sufficient for the envisioned scenarios. This does not hold when the server re-encrypts the content for each customer of the service in real-time. The use of caching and pre-distribution with an acceptable compromise to protection reduces further the load on servers and networks. Figure1 shows a sketch of the distribution system we envision for our approach. From a video, we create two files by writing bytes from the original video to a small data slice and destroy those bytes in the original. When the larger part of the video is corrupted in this way, it can be distributed freely because it is useless by itself, while the small slice is protected and unicasted on demand. The unicast access informs the content owner of each use. The corruption is content-independent, the small slice is encrypted on the server side using a personal key of the receiver. The computational load of encrypting this portion of the video is relevantly below that of content-aware methods. At the receiver's side, the unicasted slice is decrypted and synchronized with the main part of the data coming from a cache server. For synchronization the methods described in [2] can be applied.

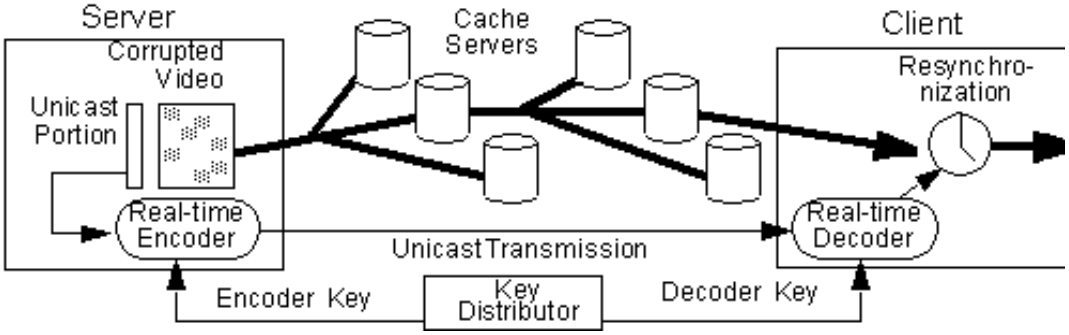


Figure 1: Distribution System

For videos encoded in MPEG-1, Huffmann encoding improves the effectiveness of the corruption of single bytes of data. Since the Huffmann algorithm is bit-oriented rather than byte-oriented, a Huffmann decoder is unable to recover from the error for the rest of a data segment. Furthermore, a complete Huffmann decoding of the data is necessary before the corruption is detected. As a result of this error propagation from a corrupted byte to the rest of a data segment in a frame, the number of bytes that need to be destroyed to corrupt compressed data is much lower than for an uncompressed frame. Thus, the destruction of larger blocks with the same overall ratio of corrupt to correct bytes is not feasible. The corruption of single bits may be as efficient as the corruption of bytes, but it increases the computational load.

We considered potential attacks to the scheme. In experiments we distinguished the selection of fixed or variable byte values used for the corruption of the original stream, and the applications of this corruption at periodic or variable offsets from each other. An attacker can identify both a periodic offset (by the use of auto-correlation) and a fixed replacement value (by gathering statistics on frequencies, see Figure2). Both information should be concealed as good as possible.

To prevent the identification of offsets it is essential to vary them. We use the Poisson distribution to compute offsets because of its memory-less property and select a random seed per video. The seed value is distributed to the receiver at the start of the encrypted unicasted transmission. The receiver's implementation of the distribution function must behave identically to the sender's to find exactly the same bytes. Figure2 shows that constant values are easily detected. Since the value is irrelevant for the reconstruction, we use values that are well hidden in the stream instead. At each insertion point, we insert the least frequent byte from the beginning of the file, counting the number of occurrences of bytes in the corrupted rather than the original file (see Figure3).

If an attacker would look for these infrequent bytes in the same way, he could identify them if the insertion of the values would still leave them the least frequent. Since the entropy of MPEG streams is extremely high (we found entropy values between 97.4% - 99% in our example videos), even a single insertion of a value might make a different one least frequent.

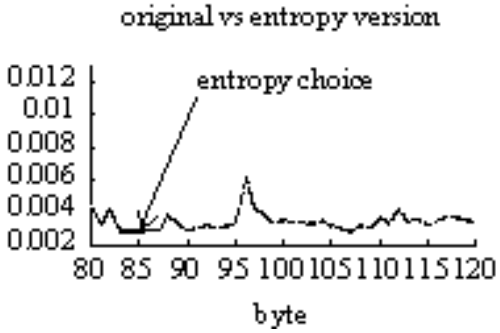


Figure 3: Rare values

It is known that header reconstruction is simple when the encoder is known, so we conducted our experiments with reconstructed headers. The remaining errors are disturbing enough to yield results that are unacceptable for commercial exploitation. We started experiments with a destruction ratio of 1%, which rendered videos (with correct headers) un-

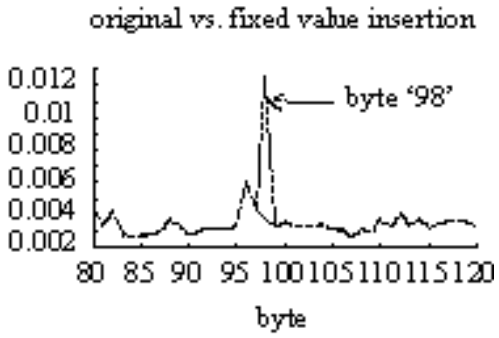


Figure 2: Fixed Value

playable to two MPEG players (ActiveMovie, VideoCharger Player) and showed nothing but artifacts in another (MpegTV). We found 0.5% to deliver bad quality and 0.1% to provide a quality sufficient to read text in large font which remains unmoving for several seconds. All of our numbers are adequately above current capabilities of restoration to good quality which handle bit error rate of about to 10-4 well.

2 Protecting the Delivered Video

The scheme presented so far protects from the theft of data that is located in caches. However, the authenticated receiver of the video, who has the full quality data available, may choose to record and resell it. We consider the insertion of random sequences of very scarce bit errors into the unicast portion of the stream means, not to prevent, but to prove copyright violation. Like watermarking, this can be exploited to prove copyright violation in a way that makes the danger of manipulation to the decoder software irrelevant.

The unauthorized reseller may decide to request the video multiple times in order to use a voting mechanism and eliminate the bit errors (since we assume that the technique is known). It is relevant to find a scheme that will yield a sufficiently large number of remaining bit errors to single out the unauthorized reseller and take further measures to prove the contract violation. Bit errors that remain after the execution of voting steps to eliminate bit errors can be identified by the content provider using a brute force approach of computing these values based on the seed values on file.

We have examined a couple of schemes that insert infrequent bytes into the video stream randomly and found that completely random errors are easily fixed by applying voting mechanisms. Our current idea is to choose for each video a random sequence of intervals of the unicast portion. For each delivery of the stream, a uniform distribution is applied to put one byte error into each interval. Similar to the distortions of a watermark, each copy can be identified by these randomly inserted errors when the provider keeps the random seed values in a database. If unauthorized copies of the video are uncovered, the bit error sequences can be compared with the series of bit errors which are generated by the seed value on file using a brute force approach.

If the attacker chooses a 3-copy voting to eliminate the bit errors, errors remain with some probability that can be used to identification the original customer. Let the length of the video be Sf, the unicast portion Su, with . If the average offset is O and the length of each interval is I, there is a probability of that a least one byte error remains. For a 1GB MPEG-1 video, 0.5% encrypted transmission, bytes (resulting in a byte error rate in the video) and , this computes to 0.537. Smaller intervals increase this probability considerably.

However, the necessary length of the video for the application of this idea is large, so further investigations are necessary to understand whether this is feasible.

3 Conclusion

We have presented a novel scheme for the protection of copyright in commercial video-on-demand systems that use caching and pre-distribution. The scheme exploits the error propagation of Huffmann encoding to corrupt large parts of a video, which can then be distributed freely, while the information that is necessary to reconstruct the content is delivered in a secure way. To help proving resale of videos by authenticated customers, we propose to add the insertion of infrequent byte errors to this scheme.

4 References

[1] C. Griwodz, M. Bär, L. C. Wolf: "Long-term Movie Popularity Models in Video-on-Demand Systems or The Life of an on-Demand Movie", ACM Multimedia 1997, November Seattle, WA, USA, November 1997

[2] J. Jarmasz, N. D. Georganas: "Designing a Distributed Multimedia Synchronization Scheduler", Proc. IEEE Multimedia Systems'97, Ottawa, June 1997

[3] T. Kunkelmann, R. Reinema, R. Steinmetz, T. Blecher: Evaluation of Different Video Encryption Methods for a Secure Multimedia Conferencing Gateway, Proc. of the 4th Int'l COST237 Workshop, Lisboa, Portugal, Dezember 1997, pp. 75-89

[4] T. B. Maples, G. A. Spanos: "Performance Study of a Selective Encryption Scheme for the Security of Networked, Real-Time Video", Proc. of the 4th Int'l Conf. on Computer Communications and Networks, Las Vegas, Nevada, September 1995

[5] L. Qiao, K. Nahrstedt, I. Tam: "Is MPEG Encryption Using Random Lists instead of Zig Zag Order Secure?", IEEE International Symposium on Consumer Electronics, December 1997, Singapore

[6] R. Tewari, M. Dahlin, H. M. Vin, J. S. Key: "BeyondHierarchies: Design Considerations for Distributed Caching on the Internet", UTCS Technical Report TR98-04, UTexas, 1998.

Applying Encryption to Video Communication

Thomas Kunkelmann
Darmstadt University of Technology
Information technology Transfer Office
Wilhelminenstr. 7
64283 Darmstadt, Germany

kunkel@ito.tu-darmstadt.de

ABSTRACT

In multimedia communications, the need for confidentiality and privacy gains more and more in importance, particularly in open networks like the Internet. This paper presents an overview of the security requirements of multi-media conferencing systems and of applicable security functions. For real-time video trans-missions there is a special need for se-lective encryption of the transmitted data. Existing methods are investigated and their strengths and weaknesses will be shown.

Besides video standards like MPEG and H.261, scalable video codecs become more and more popular. A scalable codec transmits a video signal in different layers, each encoded at its own bit rate. Applying encryption methods to them is a straightforward task and can be integrated easily.

KEYWORDS

Multimedia communication, Security, Encryption, Partial encryption

1 Introduction

Communication and cooperation in heterogeneous distributed environments are playing a rapidly increasing role in the business processes of today's enterprises. Nowadays several enterprises with distributed locations shift their personal communication and meetings more and more to so-called virtual meetings via computer links. In these cases confidential information has often to be passed securely over open networks like the Internet.

Another kind of distributed multimedia applications with a high demand for security mechanisms are video databases and *video-on-demand* (VoD) services. The security policy for these applications is not focused on optimal protection of highly confidential data, rather on protecting data against illegal access. Therefore the encryption methods needed here tend to be fast, with respect of the high data bandwidth of video streams, and to be cheap to implement in order to supply an emerging market of private users (Pay-TV, VoD). The expense to break into an encryption scheme needs not to be high, but it should be more expensive than the legal access to the video service.

In all these distributed multimedia applications, the cryptographic functionalities must cover different aspects of security, like confidentiality, integrity and authenticity. Therefore different modules of encryption mechanisms must be available to the application. Scalability for encryption methods can be achieved by partial encryption of multimedia data.

The main focus of this paper considers *encryption methods* to provide confidentiality, since their application to multimedia data streams will cause time-critical problems when encrypting the whole data stream. Besides integrity checks, the other security functionalities do not cause any problems concerning the real-time constraints.

The rest of this paper is organized as follows: Section 2 deals with general aspects of combining multimedia data with encryption. Section 3 presents some methods for the partial encryption of video data streams. Those methods will be evaluated for MPEG video compression in Section 4. In Section 5 the application of partial encryption to scalable video is presented. Section 6 concludes this paper.

2 Multimedia Data and Encryption

Several multimedia data formats require a special treatment in terms of encryption. In particular, these are data formats with real-time properties, like audio and video communication. Here encryption methods cannot be applied straightforward due to the severe time constraints for data processing and the complexity of secure encryption standards. Either encryption must be realized with special hardware, which is not available on many platforms, or the data streams have to be subdivided in order to separate data portions relevant to the human perception for encryption. The latter case is known as *partial encryption* schemes.

2.1 Data Formats For Video Transmission

For the partial encryption of multimedia data it is important to see how video data is organized in the data stream, in order to develop applicable methods for extracting the relevant data portions. Therefore, a

short survey over the common data formats used in toady's video conferencing systems is presented. A more general survey can be found in [1] and [2].

2.1.1 Motion-JPEG

The *Motion-JPEG* (M-JPEG) video format is not standardized, it consists of a sequence of single video images (frames) encoded with the *JPEG* format [3]. The JPEG image encoding technique leads to a high compression ratio for continuous-toned images. It is based on a combination of applying the *Discrete Cosine Transformation* (DCT) to blocks of 8×8 image pixels, followed by an entropy coding [1]. The M-JPEG video format is used mainly for video conferencing tools due to a symmetrical expense for encoding and decoding.

2.1.2 MPEG-1 and MPEG-2

The *MPEG* format for coding and transmitting video signals along with the corresponding audio information has been standardized by the ISO [4]. For MPEG there are three different standards specified, MPEG-1, MPEG-2 and MPEG-4 (standard scheduled for November 1998). MPEG-1 is today's commonly used video compression standard due to its availability for many platforms and appropriate hardware support. It covers data rates of about 1.2 to 1.85 Mbit/s. An MPEG data stream is formed of different layers, responsible for the synchronization of audio and video, and providing pre-defined starting points for re-synchronization. MPEG utilizes the compression techniques of JPEG, along with interframe relationships (*prediction* and *motion compensation*).

2.1.3 H.261 and H.263

H.261 and H.263 are widespread standards adopted by the ITU [5] for transmitting video data streams. The intention of H.261 is to provide video information at a data rate of p×64 Kbit/s (with $p \in \{1, ... 30\}$), matching the ISDN specification. Therefore H.261 is toady's mostly used video compression standard for ISDN video conferencing systems. The codec (en*cod*ing and *dec*oding functionality) is designed for a symmetrical encoding and decoding process with a maximum end-to-end delay of 150 ms.

The H.261 standard also specifies many format parameters. The resolutions supported by H.261 are CIF (*Common Interface Format*, 352×288 pixels) and QCIF (1/4 CIF). The frame rate is defined as 29.97 fps. The encoding schemes for H.261 are similar to those used in MPEG.

2.2 Performance Aspects For Encrypted Video

As pointed out in [6], modern high-performance workstations and servers are capable of playing MPEG-1 or M-JPEG video, leaving about 20 to 60 percent CPU time for other jobs when using hardware JPEG support. On most desktop workstations such a computing power is not available. Here the frame rate or the pixel resolution has to be reduced to meet the limited CPU capacity. Performance measurements on a PC (100 MHz Pentium, Linux) showed that such a system can playback about three H.261 QCIF video streams with frame rates sufficient for video conferencing (between 11 and 12 fps).

Table 1 shows the performance evaluations of several hardware platforms decrypting video streams in software, with standard library implementations of the DES algorithm. The reason for investigating DES is the fact that cryptanalysists consider it to be a safe algorithm for ciphertext-only and known-plaintext attacks, except for its small key space.

For most scenarios, the need for reducing the encryption effort is obvious, the slower workstations are already overloaded with the DES decryption. For the H.261 scenario, an encryption CPU usage of 20 percent implies a frame reduction from e.g. 11 to 8.8, violating the lower bounds for human image perception. Therefore, partial encryption is a suitable solution also for this case.

2.3 Integration Of Security Functionalities In The System

Security functionalities can be built up on two different layers of a system dealing with the transmission of digital video information:

DES CPU usage	*1.5Mbit MPEG*	*2Mbit M-JPEG*	*3×128 Kbit H.261*
Intel Pentium-100, Linux	86.70 %	★115.62 %	21.67 %
DEC Alpha 1000/ 266	65.63 %	87.50 %	16.41 %
Sparc 20 (Solaris)	76.01 %	★101.34 %	19.00 %
Sparc 4c (SunOS)	★312.77 %	★417.03 %	78.19 %

Table 1: CPU utilization of different hardware systems for DES software encryption (★ = projected values). The MPEG and M-JPEG cases represent e.g. Pay-TV scenarios (16 - 25 fps), while the H.261 scenario describes an ISDN video conference with three video channels open (12 - 15 fps).

Figure 3: Encrypted parts of a video stream with the partial encryption method of [7]. (H: Header data; DC: low frequency (DC) coefficient; AC: higher-frequency DCT coefficient)

- Security in the transmission or networking layer, i.e., security is already provided by the networking protocol used (e.g., SSL, RTP [8], ATM [9]). An additional data manipulation by security applications is not necessary.
- Security in the data layer, i.e., before data is transmitted from a sender to a receiver it will be manipulated by the appropriate security functions in the application. The security functionality can either be applied to the application, or the application itself is designed to gain security for other programs, e.g. the Secure Shell (SSH) [10].

One of the drawbacks of network layer security mechanisms is the need for secure underlying transport protocols, which are not available at the moment. IPnG and ATM will provide these functionalities in the near future. The advantage of data layer security is that the transmitted data can be subdivided into parts with sensitive and insensitive data with respect to the human perception, necessary for partial encryption methods.

3 Partial Video Cryption Methods

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, which are summarized in this section.

3.1 SEC-MPEG

SEC-MPEG [11] is a toolkit for partial encryption of MPEG-1 data. The aim of this toolkit is to achieve confidentiality and integrity checks. Confidentiality is achieved by using the DES algorithm, integrity checks are carried out by a *cyclic-redundancy check* (CRC). The toolkit supports four levels of confidentiality, beginning with encrypting the header information, up to an encoding of the whole MPEG stream. In level 2 a subset of DCT blocks is selected, which will be partially encrypted, while level 3 encrypts all intracoded image information.

3.2 Partial Encryption Of Intracoded Frames

Some work has been done in partially encrypting only the intracoded frames (I-Frames) of an MPEG stream [12] or the intracoded blocks in intercoded frames. In [13] an example of this kind of encryption is given, the authors also show the limits of this technique. Video sequences with a high degree of motion still show a lot of details of the original scene. As a remedy the increase of intracoded- frames is suggested, but this will also vastly increase the size of video data.

3.3 Permutation Of DCT Block Information

A method for an encoding/ decoding process with no significant delay resulting from additional encryption is applicable to video compression techniques based on the JPEG algorithm. In [14] this method is described for the MPEG standard. It is based on the zigzag ordering of the DCT coefficients before entropy coding is applied, which is randomly permuted. The drawback of this method is the worse performance of the run length encoding, which results in an expansion of the encoded video data of about 20% to 40% for the tested video sequences.

3.4 Reducing The Amount For Strong Encryption

Statistical analysis of MPEG streams show that it is still sufficient to reduce the effort for encryption to one half of the video stream, and use these data as a

Figure 4: Maximal possible reconstruction for intracoded block encryption (center) and with the method of [7] (right), both frames with about 46% encrypted data (video *flowers*, 1/2 original size, the original image is shown left).

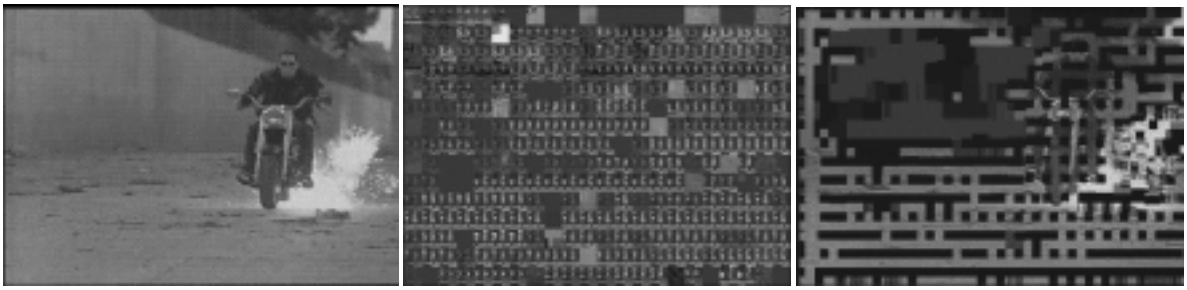


Figure 5: Video sequence *biker* (left) with 25% encrypted data, playback (center) and maximal possible reconstruction (right)

one-time pad for the other half of the stream, in order to obtain a strong cryptographic protection for the whole MPEG data [15]. The method needs about 53% of the effort for encrypting the whole data stream, its drawbacks are the usage of multiple encryption keys and the overwriting of some MPEG header fields, which makes the solution infeasible for most existing applications.

3.5 Scalable Method For JPEG-Based Video

In [7] we present a scalable partial encryption method, which allows a security level of nearly every granularity. It can be applied to all video compression methods based on the JPEG standard, in particular the formats mentioned above. This method is not prone to the motion prediction problems mentioned in 4.2. Our method takes advantage of decreasing importance for the image composition of the DCT coefficients, so it is sufficient to encrypt only the first few of them. The algorithm starts with encrypting a data block at the beginning of a DCT block and guarantees the protection of at least the first *n* DCT coefficients of a block, encrypting consecutive data portions in the video stream of the encryption method's block size. The parameter *n* of encrypted coefficients provides scalability for the security level. Table 1 gives an example (with n=3), which parts of an MPEG stream will be encrypted.

4 Evaluation of Results

First, some aspects on the safety of partial encryption methods for video data are presented. Based on these considerations, a comparison of the different methods with respect to safety, time consumption and communication overhead is given.

4.1 Possible Reconstruction Of Protected Data

With methods used in cryptanalysis, e.g., statistical and entropy evaluations, it may always be possible to detect those portions of a data stream which have been encrypted. However, this will be a difficult job for partially encrypted (MPEG or similar encoded) video streams due to the nearly redundancy-free Huffman encoding. An eavesdropper who succeeded in analyzing a partially encrypted video stream might probably reconstruct a video frame as in the examples of figure 2. Here the non-reconstructible protected information is set to zero, otherwise the random encrypted information would still obscure the reasonable information.

These examples motivate to protect truly confidential video information with an adequate method, e.g. the scalable approach presented in [7]. In other scenarios, where encryption is merely used to aggravate the access for the public, e.g., video-on-demand systems, the expense for reconstructing parts of a video is out of all proportion to the fee for joining the movie broadcast legally. In these scenarios, a simple encryption method might be considered as sufficient.

4.2 Experimental Results

All experiments are based on a series of different video sequences, which reflect several scenarios where digital video can be used. Movies are represented by the videos *flowers*, *biker* and *coastguard*, while *akiyo* is an example for a video conference scene.

In VoD scenarios the encryption effort need not to be high, even with a few percent of encrypted data the quality of the video material becomes intolerably poor [16]. In Figure 3 an example for an encrypted video image with about 25% of the data encrypted is presented. About 10 percent encryption can be considered as a satisfactory level for VoD applications, which complies with the fact that here the software and hardware effort must be minimized to keep the costs per set-top unit cheap.

For truly confidential video sequences it is not sufficient at all to pick some few video blocks or DCT coefficients for encryption, as it is done in most partial encryption schemes. Here the approach in [7] is a good choice for partial encryption. When using our scalable approach it is necessary to protect at least the first 10 to 12 DCT coefficients in order to keep a high level of confidence. This results in an encryption rate of 40% or more of the video data.

4.3 Comparison Of The Encryption Methods

In Table 2 the different partial encryption methods are compared with respect to security, scalability, time effort, and protocol signaling overhead.

An important factor is the signaling or control data overhead an encryption scheme generates. These data can be embedded in the video stream as it is done in SEC-MPEG with a special encryption header flag, or it can be transmitted via a separate control channel. A complete comparison of different partial encryption schemes can be found in [17].

5 Encryption of Scalable Video Streams

Besides video standards like Motion-JPEG, MPEG and H.261/H.263, *scalable video* codecs are becoming more and more popular [18]. A scalable codec transmits a video signal in different layers, each encoded at its own bit rate. Therefore it is possible to decode an already encoded video at different bit rates without any additional content parsing.

5.1 Scalable Video Coding With A Spatial Resolution Pyramid

The scalable video codec that is investigated here is based on a spatial resolution pyramid [19]. Figure 4 shows this two-layer pyramid. The original video signal is decomposed into two spatial layers. The codec expects an input signal corresponding to a CIF resolution at 30 frames per second. Layer 1 contains a spatially downsampled version of the original signal at QCIF resolution, layer 0 contains the signal at its full resolution.

By using a predictive pyramid coder, the spatial resolution pyramid is transmitted within a base and an enhancement layer, which carries refinement information. These refinement data are needed to reconstruct the frames within the CIF layer from those transmitted in the base layer at QCIF resolution. The scalable codec combines low complexity downsampling and interpolation filters with highly efficient E_8-lattice vector quantization. Decoder complexity is sufficiently low to allow software-only implementations on today's PCs and workstations [6]. Encoder complexity is mainly determined by motion estimation as it is also the case for all standardized motion-compensated hybrid codecs. Similar to MPEG and other video compression schemes, I- and P-frames are used. Motion-compensated prediction is based on 16×16 blocks. The scalable codec is described in more detail in [18].

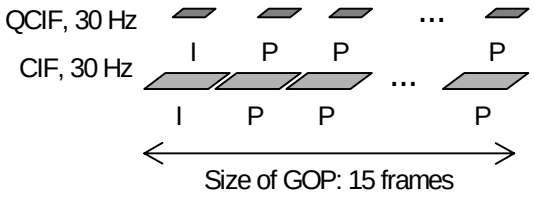


Figure 7: Spatial resolution pyramid used for the scalable codec

Figure 6: *Coastguard*: Original (left), transparent encryption with 75% protected data (right).

Method	Security	Scalability	Time overhead	Protocol overhead
SEC-MPEG	high	3 levels	DES encryption	about 17 to 32%(own data format)
Frame-type encryption	high	I: 25-40% IP: 70-85% IPB: 99%	DES encryption	none
Intra-block encryption	high	no	DES encryption	none
DCT permutation	breakable	no	None	none, data volume + 20% to 40%
Scalable method	high	full, from 8% to 100%	DES encryption	3-5%

Table 2: Comparison of different partial encryption methods

5.2 Partial Video Encryption

In contrast to the methods mentioned in Section 4, partial encryption can be easily included into a scalable codec. The scalable codec generates a natural partitioning of the video data into more important

and less important data without requiring additional content parsing. The type of scalability used here is *spatial* scalability. Other possible types of scalability are *temporal* and *SNR* [4] scalability. Spatial scalability allows two different encryption schemes, namely *base* and *enhancement* layer encryption [20].

Enhancement layer encryption is also known as *transparent encryption*. Its purpose is to restrict access to the full video quality only to receivers owning the correct decryption key. Other recipients can only decode the base layer(s). This scenario makes perfectly sense for Pay-TV applications, where the content provider allows free previews at a low quality. The decodable quality mainly depends on the number of encrypted layers. For transparent encryption it is important not to encrypt the headers and starting sequences of the upper layers, since a decoder should be able to discard the information of these layers if it does not possess the correct decryption key. Transparent encryption does not require any modifications at the decoder. An example of this kind of encryption is shown in figure 5.

Protecting only the base layer of a scalable video stream can already achieve a good content protection, since in terms of image perception most of the relevant information is concentrated in the base layer. The enhancement layer(s) only cater for minor details in the video scene and can be left unprotected in many cases [20]. An example for a reconstructed frame with an 'undecodable' base layer is shown in figure 6.

5.3 Partial Encryption Results For Mpeg-1 And The Scalable Codec

In table 3 simulation results for an MPEG-1 and a scalable video stream are compared. As test sequences *coastguard* and *akiyo* are used. Partial MPEG encryption is done with the method described in [7]. As can be seen, MPEG-1 needs 1071 kbps to encode *coastguard* at a PSNR of 28.7 dB and 123 kbps to encode *akiyo* at a PSNR of 33.7 dB. The corresponding rates needed by the scalable codec depend on the rate spent within the base layer. The values show that at low and medium base layer rates the scalable codec outperforms MPEG-1 in terms of coding efficiency.

By comparing the *energy* values E of both partial encryption methods it can be seen that the protection obtained from simple base layer encryption is comparable to the best known partial MPEG encryption method. For *akiyo*, an even higher protection can be obtained with the same encryption effort. Since base layer encryption needs no content parsing, its computational complexity is much lower than partial MPEG encryption.

Figure 8: *Akiyo*: original video image (left), reconstruction of 50 % partially encrypted video data from an MPEG-1 data stream (center) and from a scalable video stream (right)

6 Conclusions

This paper pointed out the security requirements needed for multimedia communication. A special

sequence (CIF, 30 Hz)	*encryption rate percentage*	MPEG-1			Scalable Codec		
		bit rate [kbps]	*PSNR [dB]*	*E*	*bit rate [kbps]*	*PSNR [dB]*	*E*
coastguard	~25 %	1071	28.7	344	948	29.4	212
	~50 %			162	984	28.9	130
	~75 %			49	1044	28.4	91
akiyo	~50 %	128	33.9	103	122	33.7	32
	~66 %			61	132	33.6	22
	~75 %			43	136	34.9	13

Table 3: Simulation results for partial encryption obtained with MPEG-1 and the scalable codec. Encryption rate percentage is the percentage of the encrypted bit rate with respect to the overall bit rate. For the scalable codec this percentage is identical to the percentage of the base layer bit rate with respect to the overall bit rate. The overall bit rate is the bit rate needed for transmitting a test sequence at the given PSNR. E denotes the energy contained in the decodable frames after the given rate percentage has been encrypted. All values are computed as averaged values over the first 100 frames of each test sequence.

treatment has to be applied for real-time video data due to the large amount of data to be protected. Partial encryption is a solution to solve this problem.
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 following blocks (intracoded blocks), from this it follows that only the referenced blocks have to be protected.
There are several sophisticated approaches for applying partial encryption methods to non-scalable standard-based hybrid video coding schemes. Nevertheless, the protection obtained from simple base layer encryption of a scalable encoded video based on a spatial resolution pyramid is comparable to the best known partial MPEG encryption method. 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.

7 Literature

[1] R. Steinmetz: Data Compression in Multimedia computing - standards and systems. Multimedia Systems, 1(4), pp. 187-204, Springer Verlag, Berlin 1994

[2] R. Steinmetz, K. Nahrstedt: Multimedia: Computing, Communications and Applications. Prentice Hall, München 1995

[3] ISO/ IEC International Standard 10918: Digital Compression and Coding of Continuous-Tone Still Images. 1993

[4] ISO/ IEC IS 13818-2: Generic coding of moving pictures and associated audio information: Video. 1996

[5] ITU-T Recommendation H.263: Video coding for low bit rate communication. 1996

[6] P. Bahl, P.S. Gauthier, R.A. Ulichney: Software-only Compression, Rendering, and Playback of Digital Video. Digital Technical Journal Vol. 7(4), 1995

[7] T. Kunkelmann, R. Reinema: A Scalable Security Architecture for Multimedia Communication Standards. Proc. 4th IEEE Int'l Conference on Multimedia Computing and Systems, Ottawa, Canada, 1997

[8] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson: RTP: A Transport Protocol for Real-Time Applications. RFC 1889, 1996

[9] The ATM Forum: ATM Security Framework 1.0. AF-SEC-0096.000, 1998

[10] T. Ylönen: The SSH (Secure Shell) Remote Login Protocol. http://www.cs.hut.fi/ssh/RFC, 1995

[11] J. Meyer, F. Gadegast: Security Mechanisms for Multimedia Data with the Example MPEG-1 Video. http://www.cs.tu-berlin.de/~phade/secmpeg.html, 1995

[12] T.B. Maples, G.A. Spanos: Performance Study of a Selective Encryption Scheme for the Security of Networked Real-time Video. Proc. 4th Int'l Conference on Computer and Communications, Las Vegas, NV, 1995

[13] I. Agi, L. Gong: An Empirical Study of Secure MPEG Video Transmissions. ISOC Symposium on Network and Distributed System Security, San Diego, CA, 1996

[14] L. Tang: Methods for Encrypting and Decrypting MPEG Video Data Efficiently. Proc. 4th ACM Int'l Multimedia Conference, Boston, MA, 1996

[15] L. Qiao, K. Nahrstedt: A New Algorithm for MPEG Video Encryption. Proc. 1st Int'l Conf. on Imaging Science, Systems and Technology, Las Vegas, NV, 1997

[16] T. Kunkelmann, H. Vogler, M.-L. Moschgath, L. Wolf: Scalable Security Mechanisms in Transport Systems for Enhanced Multimedia Services. Proc. 3rd European Conf. on Multi media Applications, Services and Techniques (ECMAST'98), Berlin, Germany, 1998

[17] T. Kunkelmann, R. Reinema, R. Steinmetz, T. Blecher: Evaluation of Different Video Encryption Methods for a Secure Multimedia Conferencing Gateway. Proc. 4th COST 237 Workshop, Lisboa, Portugal, 1997

[18] U. Horn and B. Girod: Scalable video coding for the Internet. Computer Networks and ISDN Systems, Vol. 29, No. 15, pp. 1833-1842, Nov. 1997

[19] M. Vetterli, K.M. Uz: Multiresolution coding techniques for digital television: A review. Multidimensional Systems and Signal Processing, Vol. 3: pp. 161-187, 1992

[20] T. Kunkelmann, U. Horn: Video Encryption Based on Data Partitioning and Scalable Coding - A Comparison. 5th Int'l Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'98), Oslo, Norway, 1998

Generating Robust Digital Signature for Image/Video Authentication

Ching-Yung Lin and Shih-Fu Chang

Department of Electrical Engineering
and New Media Technology Center
Columbia University
New York, NY 10027, USA

{cylin, sfchang}@ctr.columbia.edu

ABSTRACT

Image/video authentication techniques protect the recipients against malicious forgery. In this paper, we describe an image authentication technique that verifies the originality of the received images. The authentication signature can distinguish content-changing manipulations (such as pixel replacing) from content-preserving manipulations (such as JPEG compression). We also propose a video authentication method that generates robust signatures for compressed video. The signatures can survive some of the transcoding process of MPEG.

KEYWORDS

Authentication, watermark, digital signature, manipulation, transcoding

1 Introduction

The concept of content-based image/video authentication builds upon the increasing need for trustworthy digital multimedia data in commerce, industry, defense, *etc*. Digital media become popular in the past few years partly because of their efficiency of manipulation. Editing or modifying the content of a digital image or video can be done efficiently and seamlessly. However, these advantages decrease the credibility of digital data. To ensure trustworthiness, content-based image/video authentication techniques are needed for verifying the originality of video content and preventing forgery [1]. Observers require them to verify either the "*reality*" of images/videos of natural events or the "*intactness*" of artificial images/videos such as motion pictures, film, *etc*.

The proof of the "*reality*" of a video clip or an image can be provided only by the digital camera that took the shot. Similarly, the proof of the "*intactness*" of a received image/video should be provided by the producer. A signature, which conveys the identification of the camera or the producer and is relative to the contents, can be the proof. Image/video authentication techniques are based on two methods: embedded watermark and external digital signature. Embedding a watermark in the image/video is equivalent to signing a specific digital producer identification (signature) on the content of images/videos [2,3]. Once the image/video is manipulated, this watermark will be destroyed such that the authenticator can examine it to verify the originality of contents. Another approach generates a content-based digital signature which includes the important information of contents and the exclusive producer identification [4-10]. The signature is generated by a producer-specific private key such that it can not be forged. Therefore, the authenticator can verify a received image/video by examining whether its contents match the information conveyed in the signature [4].

Today, most digital multimedia data are stored or distributed in compressed form. Moreover, to satisfy the various needs of broadcasting, storage and transmission, some transcoding of compressed digital images/videos may be required [11,12]. For instance, digital video clips are usually shot and stored in the compressed format with a pre-determined bit-rate. But the final distributed bit rate of them may be different. Another example is that digital images shot and stored in one format may need to be distributed in different formats. These transcoding processes change the pixel values of the digital image/video but not its content. Therefore, these processes should not alter the authenticity of the data. Robustness is an important concern in developing multimedia authentication techniques. Without robustness, an authentication method can only *verify* the images/videos at the final stage of transcoding processes, but not *authenticate* them. In other words, unless we trust all the transcoders in the processes, the "*reality*" or the "*intactness*" of the multimedia data cannot be proven without robust signatures.

Robustness consideration for authentication is different from that for general watermarking techniques [13-15]. Watermarks used for copyright protection are expected to be robust to most manipulations. But authentication signatures are expected to survive only acceptable transcoding or compression and reject other manipulations.

Of the two authentication methods, the embedded watermarking method is more convenient but usually does not work well with lossy compression. The watermarks are either too fragile for compression or too flexible for manipulations. In other words, a watermarking method that can reliably distinguish compression from other manipulations still has not been found. The external signature method is not as efficient because anyone who needs to authenticate the received image/video has to request the source to provide the signature. But since the signatures remain untouched when the pixel values of the images/videos are changed, they provide a better prospect for achieving robustness.

In this paper, we describe an effective technique for content-based image/video authentication that is based on the robust authentication signature we proposed in [8-10]. This signature can survive JPEG compression, because the content-based information included in the signatures is invariant before and after JPEG compression. The proposed video authentication signature is also robust to some of the transcoding processes of MPEG.

Section 2 describes the proposed robust image authentication system and its characteristics. Section 3 shows the process of generating robust signatures. Section 4 describes the authenticator. In Section 5, we describe two methods to enhance the performance of the authentication system. Section 6 shows the robustness of this robust digital signature. In Section 7, we show some experimental results of the image authentication system. Section 8 describes the common transcoding processes of MPEG compressed videos and a robust video authentication system. We present a brief conclusion in Section 9.

2 Image Authentication System

The proposed method is shown in Figure 1. Our method uses a concept similar to that of the digital signature method proposed by Friedman [4], but their technique doesn't survive lossy compression. A signature and an image are generated at the same time. The signature is an encrypted form of the feature codes or hashes of this image, and it is stored separately. Once a user needs to authenticate the image he receives, he should decrypt this signature and compare the feature codes (or hash values) of this image to their corresponding values in the original signature. If they match, this image can be claimed to be "authentic". The most important difference between our method and Friedman's "trustworthy camera" is that we use invariance properties in JPEG lossy compression as robust feature codes instead of using hashes of the raw images.

3 Signature Generation

The generation of a signature can be divided into two parts: feature extraction and feature encryption. Feature extraction is the core problem of this paper. From the compression process of JPEG, we have found that some quantitative invariants or predictable properties can be extracted.

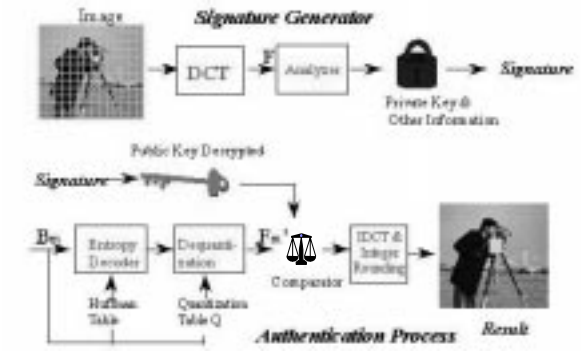


Figure 1: Signature Generator and Image Authentication

Because all DCT coefficient matrices are divided by the same quantization table in the JPEG compression process, the relationship between two DCT coefficients of the same coordinate position should remain the same after the quantization process. Furthermore, due to the rounding effect after quantization, the relationship of the two may be the same or become equal. For instance, if one coefficient $Fp(n)$ in the position n of block p is larger than the other coefficient $Fq(n)$ in the position n of block q, then after compression, their relationship, $Fp'(n) \geq Fq'(n)$, where $Fp'(n) =$ *Integer Round* $(Fp(n)/Q)\cdot Q$ and $Fq'(n) =$ *Integer Round* $(Fq(n)/Q)\cdot Q$, is guaranteed. It can be summarized as Theorem 1:

This property holds for any number of decoding and re-encoding processes.

The signature generation process is as follows: Each 8x8 block of an image captured directly by a digital camera, a digital camcorder, or computer graphic software is transformed to the DCT coefficients, and sent to the image analyzer. The feature codes are generated according to two controllable parameters in the analyzer: mapping function, W, and selected positions, b, in the DCT domain. Given a block p in an image, the mapping function is used for selecting the other block to form a block pair, *i.e.*, $q = W(p)$. A coefficient position set, b, is used to indicate which positions in a 8x8 block are selected. The feature codes of the image records the relationship of the difference value, $Fp(n)$-$Fq(n)$, and *zero*, at the b selected positions. If the difference is larger than or equal to *zero*, a bit *1* is represented; otherwise, a bit *0* is recorded. This process is applied to all blocks to

Theorem 1:

- **if** $Fp(n) > Fq(n)$ **then** $Fp'(n) \geq Fq'(n)$,
- **if** $Fp(n) < Fq(n)$ **then** $Fp'(n) \leq Fq'(n)$,
- if $Fp(n) = Fq(n)$ then $Fp'(n) = Fq'(n)$.

ensure the whole image is protected. (*i.e.*, each block has to be, at least, in a block pair.) In the last step, the feature codes are encrypted with a private key by using the Public Key Encryption method [4]. More detailed descriptions of signature generation process are in [10].

4 Authentication Process

The procedure of authentication process is also shown in Fig. 1. Given a signature derived from the original image and a JPEG compressed image bit-stream, *Bm*, for authentication, at the first step, we have to decrypt the signature and reconstruct DCT coefficients from *Bm*. Because the feature codes decrypted from the signature record the relationship of the difference values and *zero*, they indicate the sign of the difference of DCT coefficients, despite the changes of the coefficients incurred by lossy JPEG compression. If these constraints are not satisfied, we can claim that this image has been manipulated by another method.

5 Performance Enhancement

5.1 Tolerance Bound For Recompressing Noise

Rounding noises may be added during the JPEG compression process and they may cause false alarm. In practice, computer software and hardware calculate the DCT with finite precision. Because the error may accumulate throughout the multiple recompression processes, we have to introduce some tolerance bounds to prevent the authenticator from reporting some *false alarm* in the accepted recompression process. If we assign a tolerance bound, τ, to the authentication system, then the following property, should be considered as acceptable value changes in the authenticator.

5.2 Multi-Layer Feature Codes

Theorem 2:

- **if** $Fp(n)-Fq(n) \geq k$ **then** $Fp'(n)-Fq'(n) \geq k - 1/2 \cdot (Qp(n)+Qq(n))$,
- if $Fp(n)-Fq(n) < k$ then $Fp'(n)-Fq'(n) \leq k + 1/2 \cdot (Qp(n)+Qq(n))$

Given two DCT coefficients at the same positions of two blocks, not only their relationship after compression is constrained, but also the range of their difference after compression is limited. Defining *Qp* and *Qq* as the quantization matrix of the block *p* and *q*, respectively, the following theorem must be satisfied: Applying Theorem 2, we can use multi-layer feature codes to protect the DCT difference values within more precise ranges. For instance, the r-th layer feature codes record the relationship of the difference value, $Fp(n)-Fq(n)$, and a threshold, k_r. Therefore, they indicate the possible ranges of the difference of DCT coefficients, which will be tested in the authenticator.

6 Robustness

The feature codes generated in the Section 3 are based on the characteristics of JPEG compression. With the robust digital signature generated from these feature codes, images may be compressed and decompressed several times and still considered as authentic.

In some practical applications, some other manipulations are also considered acceptable, such as intensity enhancement, scaling, cropping, file format transformation, *etc*. These acceptable manipulations can be either pre-determined by the signature generator with special consideration on the controllable parameters, or decided by the authenticator with case-dependent tolerance bound. The methods for achieving robustness to these manipulations are discussed as follows:

- Intensity enhancement:

If a constant intensity change is applied to the whole image, it only changes the DC values of all the 8x8 DCT blocks. Because the authenticator compares the difference of DCT coefficients, this manipulation will be considered as acceptable. On the other hand, if the authenticator wants to reject it or limit the range of change, we can include the mean value of all DC coefficients in the signature such that the authenticator can reject large intensity changes.

- Cropping:

In most situations, cropping only selects a part of the image, such that it may introduce a different visual meaning to the cropped image. However, if this manipulation is allowed in some situations, we can design a robust signature with carefully selected mapping function. For instance, we can select block pairs from adjacent blocks. Then, the feature codes of those cropped blocks can be found in the original signature. In practical situations, the cropped image has to provide its related location on the original image to the authenticator. Because the origin point of the cropped image may not be at the grid points of the original image, (*i.e.*, each 8x8 block in the cropped image may cover parts of four 8x8 original blocks), the authenticator can only verify the cropped image excluding its boundary pixels. In this case, the recompress process may introduce different variation to pixels from recompressing the original image. Therefore, some tolerance may be needed in this situation.

- Scaling:

Scaling is a common operation on the images, which is accepted in many situations. For instance, a scanner may scan an image with a high resolution. This image may be down-sampled to an appropriate size later. In the scaling cases, the signature generator has to record the original size of the image. An authenticator can re-scale this scaled image to its original size before general authentication processes. Because the DCT transformations are linear and the difference in the pixel values of the original and the re-scaled image should not be too great, there will be no large changes on the DCT coefficients. Similar to the general recompression noise, these changes can be also considered as some kinds of noise that can be solved by allowing larger tolerance values in the authenticator.

Format transformation with other lossy compressions:

Other lossy compressions such as wavelet-based methods or color space decimation methods can be considered as introducing noises to the original image. Similarly, we can use larger tolerances in the authenticator to allow these lossy compressions.

Filtering and other operations:

Filtering, such as low-pass filtering and edge enhancement, may probably change more visual meaning of images. The authenticator would be hard to deal with these operations. However, if the changes on pixel values are not too great, we can still consider them as some kind of noise and use larger tolerance values. This method can also be applied to other operations.

7 Experimental Results

The 'Lenna' image is compressed with a compression ratio of 9:1. The authentication signature is generated based on the original image. The compressed bitstream is sent to the system for authentication. As predicted, the authenticator will verify the compressed image as authentic and decompress this image perfectly. The authentication result is shown in Fig. 2(b).

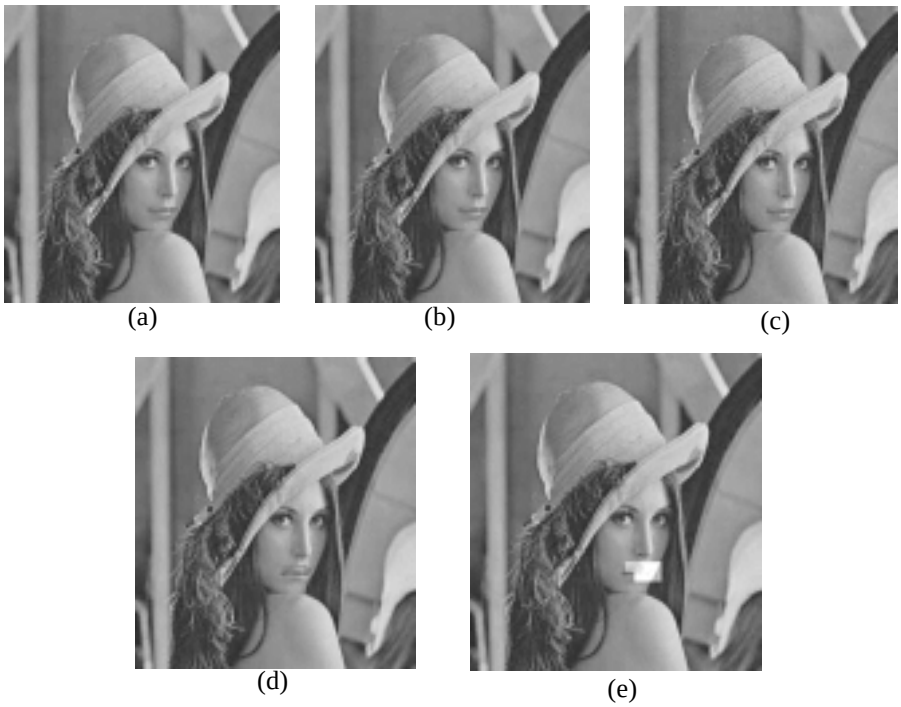


Figure 2: Experimental Results: (a) original image, (b) 9:1 JPEG compressed, (c) 9:1 JPEG recompressed from a 6:1 compressed image, (d) manipulated image, (e) authentication result of the manipulated image.

The original image is compressed with a compression ratio 6:1. Then, this image is decompressed by Photoshop 3.0, rounded to integral values, and recompressed into an image with compression ratio 9:1. In this case, the recompression process (9:1) does not trigger the manipulation detector and the final compressed image is still verified as authentic. The final decoded image is similar to Fig. 2(c).

In the third experiment, we flipped the mouth area of the image. It is shown in Fig. 2(d), with its authentication result shown in Fig. 2(e). It can be clearly shown that the manipulated part has been detected as fake and highlighted by the authenticator.

8 Video Authentication System

Similar to the image authentication system, a video authentication signature has to be robust to the transcoding processes. Regardless of the format transformation between different compression standards (such as MPEG-1, MPEG-2, H.261 and H.263), five transcoding processes may be applied to the compressed video [16,17]:

1. Dynamic Rate Shaping [18,19]: A real-time rate-control scheme in the compressed domain. This technique sets dynamic control points to drop the high-frequency DCT coefficients on each 8x8 block in a macroblock. Motion vectors are not changed.
2. Rate Control without Drift Error Correction [20,21]: This technique is also applied in the compressed domain. DCT coefficients are re-quantized to satisfy different bit-rate constraint. Motion vectors are not changed.
3. Rate Control with Drift Error Correction [16]: This technique improves the video quality, but it needs more computations. DCT coefficients of the residue of intercoded blocks are changed to satisfy the change of the re-quantized intra-coded blocks. Motion vectors are not changed in this case.
4. Transcoding with Mostly Consistent Frame Types [16,17,23]: The frame types (I, P and B), are kept unchanged in each generation. It may be used in creating a new sequence by cutting and pasting several video segments with consistent GOP units within each segment except the frames at the boundary.
5. Transcoding with Inconsistent Frame Types [16]: In some editing process, the compressed videos are transformed to the uncompressed bit-streams which are then re-encoded. The GOP structures of frames and the motion vectors may change in this case.

Video authentication signatures can be generated for different situations. For instance, to generate a signature that is robust to situations 1, 2 and 4, we can use the DCT coefficients of the luminance and chromatic matrices in each macroblock to generate the comparison pairs. Since the *quantization_scale* is specified for each macro-block [25], the relative relationships of the coefficients are invariant during transcoding. Therefore, similar to the signature generation process of images, we can use them to generate the feature codes. If a more flexible choice of comparison pair is necessary, the authentication system can generate signatures based on the criteria we have proposed in [9,10]. It should be noted that, in situation 4, the frames in the boundary of video segmentations cannot be verified by this method.

Because the drift error correction process changes the DCT coefficient values, statistical models of the changes can be used to provide tolerance bounds for the coefficient relationships, similar to that described in [10].

Situation 5 poses the most challenging case for authentication. The GOP structure in the video is changed and so is the relationship of DCT coefficients among blocks. The design scheme for generating a robust signature in this situation is still under study.

A more detailed description of the content-based video authentication techniques will be shown in [26].

9 Conclusion

In this paper, we have described a method for robust image/video authentication. Robust signatures can distinguish the JPEG lossy baseline compression from other malicious manipulations for images, and the Rate-Control Coding from other manipulations for compressed videos. Our analytic and empirical performance analyses have shown the effectiveness of the image authentication system and presented a possible direction for further video authentication research.

10 References

[1] Bearman, D., and Trant, J. Authenticity of Digital Resources: Towards a Statement of Requirements in the Research Process. D-Lib Magazine, June 1998.

[2] Yeung, M. and Mintzer, F. An Invisible Watermarking Technique for Image Verification. Proc. Of ICIP, Santa Barbara, CA, USA, Oct. 1997.

[3] Lin, C.-Y. and Chang, S.-F. A Watermark-Based Robust Image Authentication Method Using Wavelets. ADVENT Report, Columbia University, Apr. 1998. http://www.ctr.columbia.edu/~cylin/pub/a98wav.ps

[4] Friedman, G.L. The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image. IEEE Trans. on Consumer Electronics, Vol.39, No.4, pp.905-910, Nov. 1993.

[5] Quisquater, J.-J., Macq, B., Joye, M., Degand, N. and 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.

[6] Gennaro, R., and Rohatgi, P. How to Sign Digital Streams. CRYPTO '97, Santa Barbara, CA, USA, August 1997, pp.180-197.

[7] Gennaro, R., Krawczyk, H. and Rabin, T. RSA-based Undeniable Signatures. CRYPTO '97, Santa Barbara, CA, USA, August 1997

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

[9] Lin, C.-Y. and Chang, S.-F. An Image Authenticator Surviving DCT-based Variable Quantization Table Compression. CU/CTR Technical Report 490-98-24, Nov. 1997.

[10] Lin, C.-Y. and Chang, S.-F. A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation. CU/CTR Technical Report 486-97-19, Dec. 1997. http://www.ctr.columbia.edu/~cylin/pub/authpaper.ps

[11] Wells, N.D. The Atlantic Project: Models for programme production and distribution. Proceedings of the European Conference on Multimedia Applications Services and Techniques (ECMAST 96), Louvaine-la-Neuve, Belgium, May 1996, pp. 243-253.

[12] Brightwell, P.J., Dancer, S.J. and Knee, M.J. Flexible Switching and Editing of MPEG-2 Video Bitstreams. International Broadcasting Convention (IBC 97), Amsterdam, Sep. 1996, pp. 547-552.

[13] Cox, I., Kilian, J., Leighton, T., and Shamoon, T. Secure Spread Spectrum Watermarking for Multimedia. NEC Research Institute Technical Report, 95-10, 1995.

[14] Braudaway, G.W., Magerlein, K.A. and Mintzer, F. Protecting Publicly-Available Images with a Visible Image Watermark. IBM Research Division, T.J. Watson Research Center, Technical Report 96A000248, 1996.

[15] Meng, J. and Chang, S.-F. Embedding Visible Watermarks in the Compressed Domain. IEEE International Conference on Image Processing (ICIP 98), Chicago, IL, USA, Oct. 1998.

[16] Tudor, P.N. and Werner, O.H. Real-Time Transcoding of MPEG-2 Video Bit Streams. International Broadcasting Convention (IBC 97), Amsterdam, Netherlands, Sep. 1997, pp. 286-301.

[17] Werner, O.H. Generic Quantiser for Transcoding of Hybrid Video. Proceedings of the 1997 Picture Coding Symposium, Berlin, Germany, Sep 1997.

[18] Eleftheriadis, A. and Anastassiou, D. Constrained and General Dynamic Rate Shaping of Compre-ssed Digital Video. Proceedings of the 2nd IEEE International Conference on Image Processing (ICIP95), Arlington, VA, USA, Oct. 1995.

[19] Jacobs, S. and Eleftheriadis, A. Straming Video using Dynamic Rate Shaping and TCP Flow Control. Visual Communication and Image Representation Journal, Jan. 1998.

[20] Viscito, E. and Gonzales, C. A Video Compression Algorithm with Adaptive Bit Allocation and Quantization. SPIE Vol. 1605 Visual Communications and Image Processing '91.

[21] Ding, W. and Liu, B. Rate Control of MPEG Video Coding and Recording by Rate-Quantization Modeling. IEEE Trans. on Circuits and Systems for Video Technology, Vol. 6, No. 1, pp.12-19, Feb. 1996.

[22] Meng, J. and Chang, S.-F. Tools for Compressed-Domain Video Indexing and Editing. SPIE Conference on Storage and Retrieval for Image and Video Database, Vol. 2670, San Jose, CA, USA, Feb. 1996.

[23] Meng, J. and Chang, S.-F. CVEPS – A Compressed Video Editing and Parsing System. Proceedings of ACM Multimedia 96, Boston, MA, USA, Nov. 1996.

[24] Chang, S.-F. and Messerschmitt, D. G. Manipulation and Compositing of MC-DCT Compressed Video. IEEE Journal of Selected Areas in Communications, Vol. 13, No. 1, pp.1-11, Jan. 1995.

[25] Haskell, B.G., Puri, A. and Netravali, A.N. Digital Video: An Introduction to MPEG-2. Chapman and Hall, 1997.

[26] Lin, C.-Y. and Chang, S.-F. Issues and Solutions for Authenticating MPEG Video. SPIE Storage and Retrieval for Image and Video Databases, San Jose, CA, USA, Jan. 1999.

Weaknesses of Copyright Marking Systems

Fabien A.P. Petitcolas
University of Cambridge, Computer Laboratory
Pembroke Street, Cambridge CB2 3QG, UK
fapp2@cl.cam.ac.uk

Ross J. Anderson
University of Cambridge, Computer Laboratory
Pembroke Street, Cambridge CB2 3QG, UK
rja14@cl.cam.ac.uk

ABSTRACT

Hidden copyright marks have been proposed as a solution for solving the illegal copying and proof of ownership problems in the context of multimedia objects. We show that the first generation of systems does not fulfil the expectation of users through a number of attacks that enable the information hidden by them to be removed or otherwise rendered unusable.

KEYWORDS

Digital watermarking, fingerprinting, attacks.

1 Introduction

The ease with which digital media could be copied led people to propose techniques for embedding hidden copyright marks and serial numbers in still images, video and audio. We formed the view that useful progress might come from trying to attack all these first generation schemes. In the related field of cryptology, progress was iterative: cryptographic algorithms were proposed, attacks on them were found, better algorithms were proposed, and so on. Eventually, theory emerged: fast correlation attacks on stream ciphers and differential and linear attacks on block ciphers, now help us understand the strength of cryptographic algorithms in much more detail than before.

Electronic copyright management schemes have been proposed as a solution to the copying problem. These schemes might be imposed in applications such as Digital Versatile Disk (DVD) and video-on-demand where the idea is that DVD players would refuse to copy files containing suitable copyright marks. But such schemes suffer from a number of drawbacks. They rely on the tamper-resistance of consumer electronics – a notoriously unsolved problem [1]. The tamper-resistance mechanisms being built into DVD players are fairly rudimentary and the history of satellite TV piracy leads us to expect the appearance of 'rogue' players which will copy everything[89]. Electronic copyright management schemes also conflict with applications such as digital libraries, where 'fair use' provisions are strongly entrenched. Another problem, according to Samuelson, is that '*Tolerating some leakage may be in the long run of interest to publishers*' [2]. A European legal expert put it even more strongly: that copyright laws are only tolerated because they are not enforced against the large numbers of petty offenders [3].

Similar issues are debated within the software industry; some people argue, for example, that a modest level of amateur software piracy actually enhances revenue because people may 'try out' software they have 'borrowed' from a friend and then go on to buy it. Bill Gates' view is significant: '*Although about three million computers get sold every year in China, people don't pay for the software. Someday they will, though. And as long as they're going to steal it, we want them to steal ours.* […] *Then we'll somehow figure out how to collect sometime in the next decade.*' [4]

For all these reasons, we may expect leaks in the primary copyright protection mechanisms and wish to provide independent secondary mechanisms that can be used to trace and prove ownership of digital objects. Here too marking techniques are expected to be important.

2 Copyright marks

There are two basic kinds of mark: *fingerprints* and *watermarks*. One may think of a fingerprint as an embedded serial number while a watermark is an embedded copyright message. The first enables us to trace offenders, while the second can provide some of the evidence needed to prosecute them. It may ever, as in the DVD proposal, form part of the primary copy management system; but it will more often provide an independent back-up to a copy management system that uses overt mechanisms such as digital signatures.

In [5], we discussed various applications of fingerprinting and watermarking, their interaction, and some related technologies. Here, we are concerned with the robustness of the underlying mechanisms. What sort of attacks are possible on marking schemes? What sort of resources are required to remove marks completely, or to alter them so that they are read incorrectly? What sort of effect do various possible removal techniques have on the perceptual quality of the resulting audio or video?

[89] As a matter of fact techniques to bypass the territorial lock of certain DVD implementations are already available on the Internet.

The basic problem is to embed a mark in the digital representation of an analogue object (such as a film or sound recording) in such a way that it will not reduce the perceived value of the object while being difficult for an unauthorised person to remove. A first pass at defining robustness in this context may be found in a recent request for proposals for audio marking technology from the International Federation for the Phonographic Industry, (IFPI) [6]. The goal of this exercise was to find a marking scheme that would generate evidence for anti-piracy operations, track the use of recordings by broadcasters and others and control copying. The IFPI robustness requirements are as follows:

- the marking mechanism should not affect the sonic quality of the sound recording;
- the marking information should be recoverable after a wide range of filtering and processing operations, including two successive D/A and A/D conversions, steady-state compression or expansion of 10%, compression techniques such as MPEG and multi-band nonlinear amplitude compression, adding additive or multiplicative noise, adding a second embedded signal using the same system, frequency response distortion of up to 15 dB as applied by bass, mid and treble controls, group delay distortions and notch filters;
- there should be no other way to remove or alter the embedded information without sufficient degradation of the sound quality as to render it unusable;
- given a signal-to-noise level of 20 dB or more, the embedded data channel should have a bandwidth of 20 bits per second, independent of the signal level and type (classical, pop, speech).

Similar requirements could be drawn up for marking still pictures, videos and multimedia objects in general. However, before rushing to do this, we will consider some systems recently proposed and show attacks on them that will significantly extend the range of distortions against which designers will have to provide defences, or greatly reduce the available bandwidth, or both.

3 Attacks

This leads us to the topic of attacks and here we present some quite general kinds of attack that destroy, or at least reveal significant limitations of, several marking schemes: PictureMarc 1.51 [7], SysCoP [8], SureSign [9], JK_PGS (É.P.F.L. algorithm, part of the European TALISMAN project), EIKONAmark [10], [11], Echo Hiding [19], Giovanni [17] and the N.E.C. method [13]. We suspect that systems that use similar techniques are also vulnerable to our attacks.

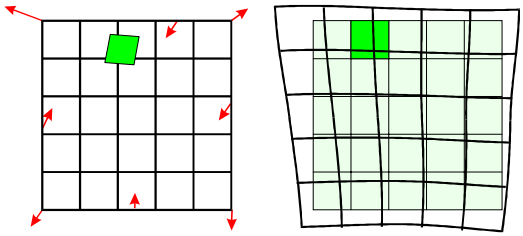


Fig. 1 — We exaggerate here the distortion applied by StirMark to still pictures.

3.1 The Jitter Attack

Our starting point in developing a systematic attack on marking technology was to consider audio marking schemes. A simple and devastating attack on these schemes is to add jitter to the signal by removing samples or duplicating other. In fact most simple spread-spectrum based techniques are subject to this kind of attacks. Indeed, although spread-spectrum signals are very robust to distortion of their amplitude and to noise addition, they do not survive timing errors: synchronisation of the chip signal is very important and simple systems fail to recover this synchronisation properly. So, in general time scaling based attacks are very efficient against audio marking systems.

3.2 Stirmark

Following this attack and after evaluating some watermarking software, it became clear that although many schemes could survive basic manipulations – that is, manipulations that can be done easily with standard tools, such as rotation, shearing, resampling, resizing and lossy compression – they would not cope with combinations of them. This motivated the design of StirMark, initially implemented by Markus G. Kuhn and enhanced and maintained by the first author [14].

StirMark is a generic tool developed for simple robustness testing of image marking algorithms and other steganographic techniques. StirMark simulates a resampling process, i.e. it introduces the same kind of errors into an image as printing it on a high quality printer and then scanning it again with a high quality scanner. It applies a minor geometric distortion: the image is slightly stretched, sheared, shifted and/or rotated by an unnoticeable random amount and then resampled using Nyquist interpolation.

With those simple geometrical distortions we could confuse most marking systems available on the market. More distortions – still unnoticeable – can be applied to a picture. We applied a global 'bending' and 'random displacement' to the image: in addition to the general bi-linear property explained previously, a slight deviation is applied to each pixel, which is greatest at the centre of the picture and almost null at the corners and to which is added a higher frequency displacement of the form

$\ddot{e}\sin(\omega_x x)\sin(\omega_y y) + n(x,y)$ – where n is a random number – is added (Fig. 1).
Finally a transfer function that introduces a small and smoothly distributed error into all sample values is applied. This emulates the small non-linear analogue/digital converter imperfection typically found in scanners and display devices.
In order for these distortions – which are practically unnoticeable as one can see from Fig. 2 – to be most effective, a medium JPEG compression should be applied after StirMark.[90]

Fig. 2 — 'Lenna' before and after StirMark used with default parameters.

We suggest that image-watermarking tools, which do not survive StirMark – with default parameters – should be considered unacceptably easy to break. This immediately rules out the majority of commercial marking schemes.
One might try to increase the robustness of a watermarking system by trying to foresee the possible transforms used by pirates; one might then use techniques such as embedding multiple versions of the mark under suitable inverse transforms; for instance Ó Ruanaidh and Pereira suggest using the Fourier-Mellin transform. However, the general theme of the attacks described above is that given a target marking scheme, we invent a distortion (or a combination of distortions) that will remove it or at least make it unreadable, while leaving the perceptual value of the previously marked object undiminished. We are not limited in this process to the distortions produced by common analogue equipment, or considered in the IFPI request for proposals cited above.
It is an open question whether there is any marking scheme for which a chosen distortion attack cannot be found.

3.3 The Mosaic Attack

This point is emphasised by a 'presentation' attack, which is of quite general applicability and which possesses the initially remarkable property that a marked image can be unmarked and yet still rendered pixel for pixel in exactly the same way as the marked image by a standard browser.

[90] We preferred not to include JPEG or similar compression techniques in StirMark in order to keep the program simple.

The attack was motivated by a fielded automatic system for copyright piracy detection, consisting of a watermarking scheme plus a web crawler that downloads pictures from the net and checks whether they contain a watermark.
It consists of chopping an image up into a number of smaller subimages, which are embedded in a suitable sequence in a web page. Common web browsers render juxtaposed subimages stuck together, so they appear identical to the original image (Fig. 3). This attack appears to be quite general; all marking schemes require the marked image to have some minimal size (one cannot hide a meaningful mark in just one pixel). Thus by splitting an image into sufficiently small pieces, the mark detector will be confused. The best that one can hope for is that the minimal size could be quite small and the method might therefore not be very practical.
There are other problems with such 'crawlers'. Java applets, ActiveX objects, etc. can be embedded to display a picture inside the browser; the applet could even de-scramble the picture in real time. Defeating such techniques would entail rendering the web page, detecting pictures and checking whether they contain a mark. An even more serious problem is that much current piracy is of pictures sold via many small services, from which the crawler would have to purchase them using a credit card before it could examine them. A crawler that provided such 'guaranteed sales' would obviously become a target.

3.4 A General Attack On Audio Marking

Audio restoration techniques have been studied for several years and have proved to be very useful to remove localised degradations (clicks, crackles, scratches, etc.) from old recordings [15], [16]. After finding the local degradations, these methods basically ignore the bad samples and interpolate the signal using the neighbouring ones.
Our attack is based on this idea: the signal is reconstructed block by block using the original data. The method we used assumes that the recorded data x is the realisation of a stationary autoregressive (AR) process of order p, i.e.

$$x_n = \sum_{k=1}^{p} a_k x_{n-k} + e_n \qquad n = p+1,\ldots,N \qquad (1)$$

where $\mathbf{e} = [e_{p+1},\ldots,e_N]^T$ is the 'excitation' noise vector. We suppose that we want to reconstruct a block of l consecutive samples starting at sample $m+1$ and assume to be unknown. Estimators for both a and x are chosen such that they minimise the quadratic error $E = \mathbf{e}^T\mathbf{e}$ which is a function of the unknown samples $\mathbf{x}_u = [x_{m+1},\ldots,x_{m+l}]^T$ and the unknown AR parameters $\mathbf{a} = [a_1,\ldots,a_p]^T$.

Fig. 3 —Screen-shot of a web browser while downloading an image after the *mosaic attack*. This attack chops a watermarked image into smaller images, which are stuck back together when the browser renders the page. We implemented software 2Mosaic that reads a JPEG picture and produces a corresponding mosaic of small JPEG images as well as the necessary HTML code automatically [1]. In some cases downloading the mosaic is even faster than downloading the full image! In this example we used a 350×280-pixel image watermarked using PictureMarc 1.51. Photography: Kings' College Chapel, courtesy of John Thompson, JetPhotographic, Cambridge.

Minimisation of E is non-trivial since it involves non-linear fourth order unknown terms but a suboptimal solution to the above problem can be used.

First E is minimised with respect to $\mathbf{a}$ by taking an arbitrary initial estimate for $\mathbf{x}_u$ (typically zero) in order to obtain an estimate $\hat{\mathbf{a}}$ of $\mathbf{a}$. If we note $\mathbf{x}_1 = [x_{p+1}, \ldots, x_N]^T$, then equation (1) can be written $\mathbf{e} = \mathbf{x}_1 - \mathbf{B}(\mathbf{x})\mathbf{a}$ and $\hat{\mathbf{a}}$ is given by:

$$\mathbf{B}^T\mathbf{B}\hat{\mathbf{a}} = \mathbf{B}^T\mathbf{x}_1 \qquad (2)$$

Then E is minimised with respect to $\mathbf{x}_u$ and using $\hat{\mathbf{a}}$. Equation (1) is written as $\mathbf{e} = \mathbf{D}_k(\mathbf{x})\mathbf{x}_k + \mathbf{D}_u(\mathbf{x})\mathbf{x}_u$ where $\mathbf{x}_k$ is the vector of known samples. After minimisation, the reconstructed block $\hat{\mathbf{x}}_u$ is given by:

$$\mathbf{D}_u{}^T\mathbf{D}_u\hat{\mathbf{x}}_u + \mathbf{D}_u\mathbf{D}_k\mathbf{x}_k = 0 \qquad (3)$$

These two steps can be iterated to get better results but it seems that one iteration is usually enough. For the attacks we just increase m in steps of the block length l and compute for each step an estimated block which is appended to the others. We end up with a fully reconstructed signal.

Other and better interpolation algorithms are available, but the least square AR interpolation technique, we briefly summarised, gives satisfactory results if the blocks are relatively small, up to 80 samples [15], [16].

Although we used it only against BlueSpike's method [17], this attack is quite general and could also be used against image marking too. Similar algorithms for image reconstruction are given in [18].

3.5 Attack On Echo Hiding

Echo hiding hides information in sound by introducing echoes with very short delays [19]. It relies on the fact that we cannot perceive short echoes (say 1 ms) and embeds data into a cover audio signal by introducing an echo characterised by its delay τ and its relative amplitude α. By using two types of echo it is possible to encode ones and zeros. For this purpose the original signal is divided into chunks separated by spaces of pseudo-random length; each of these chunks will contain one bit of information.

The echo delays are chosen between 0.5 and 2 milliseconds and the best relative amplitude of the echo is around 0.8. According to its creators, decoding involves detecting the initial delay and the autocorrelation of the cepstrum of the encoded signal is used for this purpose.

The 'obvious' attack on this scheme is to detect the echo and then remove it by simply inverting the convolution formula; the problem is to detect the echo without knowledge of either the original object or the echo parameters. This is known as 'blind echo cancellation' in the signal processing literature and is known to be a hard problem in general.

We tried several methods to remove the echo. Frequency invariant filtering was not very successful. Instead we used a combination of cepstrum analysis and 'brute force' search.

The underlying idea of cepstrum analysis is presented in [20]. Suppose that we are given a signal $y(t)$, which contains a simple single echo, i.e. $y(t) = x(t) + \alpha x(t-\tau)$. If we note Φ_{xx} the power spectrum of x then

$$\Phi_{yy}(f) = \Phi_{xx}(f)\left(1 + 2\alpha\cos(2\pi f\tau) + \alpha^2\right)$$

whose logarithm is approximately $\ln\Phi_{yy}(f) \approx \ln\Phi_{xx}(f) + 2\alpha\cos(2\pi f\tau)$. This is a function of the frequency f and taking its power spectrum raises its 'quefrency' τ, that is the frequency of $\cos(2\pi f\tau)$ as a function of f. The autocovariance[91] of this later function emphasises the peak that appears at 'quefrency' τ.

Experiments on random signals as well as on music show that this method returns quite accurate estimators of the delay when an artificial echo has been added to the signal. In the detection function we only consider echo delays between 0.5 and 3 milliseconds. Below 0.5 ms the function does not work properly and above 3 ms the echo becomes too audible.

Our first attack was to remove an echo with random relative amplitude, expecting that this would introduce enough modification in the signal to prevent

[91] $C(x) = E\left((x-\bar{x})(x-\bar{x})^*\right)$

watermark recovery. Since echo hiding gives best results for α greater than 0.7 we could use $\tilde{\alpha}$ – an estimation of α – drawn from, say a normal distribution centred on 0.8. It was not really successful, so our next attack was to iterate: we re-apply the detection function and vary $\tilde{\alpha}$ to minimise the residual echo. We could obtain successively better estimators of the echo parameters and then remove this echo. When the detection function cannot detect any more echo, we have got the correct value of $\tilde{\alpha}$ (as this gives the lowest output value of the detection function).

3.6 Protocol Considerations

The main threat addressed in the literature is an attack by a pirate who tries to remove the watermark directly. As a consequence, the definition commonly used for robustness includes only resistance to signal manipulation (cropping, scaling, resampling, etc.). Craver *et al.* show that this is not enough by exhibiting a 'protocol' level attack [21].

The basic idea is that many schemes provide no intrinsic way of detecting which of two watermarks was added first: the process of marking is often additive, or at least commutative. So if the owner of the document d encodes a watermark w and publishes the marked version $d + w$ and has no other proof of ownership, a pirate who has registered his watermark as w' can claim that the document is his and that the original unmarked version of it was $d + w - w'$.

Craver *et al.* argue for the use of information-losing marking schemes whose inverses cannot be approximated closely enough. However, our alternative interpretation of their attack is that watermarking and fingerprinting methods must be used in the context of a larger system that may use mechanisms such as timestamping and notarisation to prevent attacks of this kind.

Registration mechanisms have not received very much attention in the copyright marking literature to date. The existing references such as [22], [23], [25] and [26] mainly focus on protecting the copyright holder and do not fully address the rights of the consumers who might be fooled by a crooked reseller. Moreover a good registration and trading mechanism cannot be based on a weak marking technique.

3.7 Implementation Considerations

The robustness of embedding and retrieving techniques is not the only issue. Most attacks on fielded cryptographic systems have come from the opportunistic exploitation of loopholes that were found by accident; cryptanalysis was rarely used, even against systems that were vulnerable to it [27].

We cannot expect copyright marking systems to be any different and the pattern was followed in the first attack to be made available on the Internet against the most widely used picture marking scheme, PictureMarc, which is bundled with Adobe Photoshop and Corel Draw. This attack [28] exploited weaknesses in the implementation rather than the underlying marking algorithms, even although these are weak (the marks can be removed using StirMark).

Each user has an ID and a two-digit password, which are issued when she registers with Digimarc and pays for a subscription. The correspondence between IDs and passwords is checked using obscure software in the implementation and although the passwords are short enough to be found by trial and error, the attack first uses a debugger to break into the software and disable the password checking mechanism. We note in passing that IDs are public, so either password search or disassembly can enable any user to be impersonated.

A deeper examination of the program also allows a villain to change the ID and thus the copyright of an already marked image as well as the type of use (such as adult versus general public content). Before embedding a mark, the program checks whether there is already a mark in the picture, but this check can be bypassed fairly easily using the debugger with the result that it is possible to overwrite any existing mark and replace it with another one.

Exhaustive search for the personal code can be prevented by making it longer, but there is no obvious solution to the disassembly attack. If tamper resistant software [29] cannot give enough protection, then one can always have an online system in which each user shares a secret embedding key with a trusted party and uses this key to embed some kind of digital signature. Observe that there are two separate keyed operations here; the authentication (which can be done with a signature) and the embedding or hiding operation.

3.8 Robustness Against Insiders

Although we can do public-key steganography – hiding information so that only someone with a certain private key can detect its existence [30] – we still do not know how to do the hiding equivalent of a digital signature; that is, to enable someone with a private key to embed marks in such a way that anyone with the corresponding public key can read them but not remove them. But if the stego key is widely released (e.g. as part of a global law enforcement or in equipment) it is very likely to leak over time.

Another problem is that a public decoder can be used by the attacker; he can remove a mark by applying small changes to the image until the decoder cannot find it anymore. This was first suggested by Perrig in [26]. In [31] a more theoretical analysis of this attack is presented as well as a possible countermeasure: randomising the detection process. One could also make the decoding process computationally expensive. However neither approach is really satisfactory in the absence of tamper-resistant hardware.

Unless a breakthrough is made, applications that require the public verifiability of a mark (such as DVD) appear doomed to operate within the con-

straints of the available tamper resistance technology (one could use a number of marks with keys revealed in succession[92]), or to use a central 'mark reading' service. This is evocative of cryptographic key management prior to the invention of public key techniques.

4 Conclusion

We have demonstrated that the majority of copyright marking schemes in the literature are vulnerable to attacks involving the introduction of sub-perceptual levels of distortion. In particular, many of the marking schemes in the marketplace provide only a limited measure of protection against attacks. Most of the image marking systems are defeated by StirMark, a simple piece of software that we have placed in the public domain [14]. We have also shown specific attacks some audio marking systems.

This experience confirms our hypothesis that steganography would go through the same process of evolutionary development as cryptography, with an iterative process in which attacks lead to more robust systems.

Our experience in attacking the existing marking schemes has convinced us that any system which attempted to meet all the accepted requirements for marking (such as those set out by IFPI) would fail: if it met the robustness requirements then its bandwidth would be quite insufficient. This is hardly surprising when one considers that the information content of many music recording is only a few bits per second, so to expect to embed 20 bits per second against an opponent who can introduce arbitrary distortions is very ambitious.

Our more general conclusion from this work is that the 'marking problem' has been over-abstracted; there is not one 'marking problem' but a whole constellation of them. We do not believe that any general solution will be found. The trade-offs and in particular the critical one between bandwidth and robustness, will be critical to designing a specific system.

We already remarked in [5] on the importance of whether the warden was active or passive – that is, whether the mark needed to be robust against distortion. In general, we observe that most real applications do not require all of the properties in the IFPI list. For example, when auditing radio transmissions, we only require enough resistance to distortion to deal with naturally occurring effects such as multipath. Many applications will also require supporting protocol features, such as the timestamping service that we mentioned in the context of reversible marks.

So we do not believe that the intractability of the 'marking problem' is a reason to abandon this field of research. On the contrary; practical schemes for most realistic application requirements are probably feasible and the continuing process of inventing schemes and breaking them will enable us to advance the state of the art rapidly.

Finally, we suggest that the real problem is not so much inserting the marks as recognising them afterwards. Thus progress may come not just from devising new marking schemes, but in developing ways to recognise marks that have been embedded using the obvious combinations of statistical and transform techniques and thereafter subjected to distortion. The considerable literature on signal recognition may provide useful starting points.

[92] This is what happens for bank note printing in some countries: notes have a number of 'anti-copy' features, which are publicised in succession. Forgers are less likely to reproduce them since they do not know their existence.

5 Acknowledgments

The first author is grateful to Intel Corporation for financial support under the grant 'Robustness of Information Hiding Systems'.

6 References

[1] Ross J. Anderson and Markus G. Kuhn. Tamper Resistance – A Cautionary Note. In *Second USENIX Workshop on Electronic Commerce*, pages 1–11, Oakland, CA, USA, November 1996. ISBN 1-880446-83-9.

[2] Pamela Samuelson. Copyright and Digital Libraries. *Communications of the ACM*, pages 15–21, 110, 38(4), April 1995.

[3] Alastair Kelman. Electronic Copyright Management – The Way Ahead. Security Seminars, University of Cambridge, 11 February 1997.

[4] *The Bill & Warren Show*. Fortune, page 44, 20th July 1998. Public dialogue between Bill Gates, founder and CEO of Microsoft Corporation, and Warren Buffett, chairman of Berkshire Hathaway Inc.

[5] Ross J. Anderson and Fabien A.P. Petitcolas. On The Limits of Steganography. *IEEE Journal of Selected Areas in Communications (J-SAC) – Special Issue on Copyright & Privacy Protection*, pages 474–481, 16(4), May 1998. ISSN 0733-8716.

[6] International Federation of the Phonographic Industry. Request for Proposals – Embedded Signalling Systems Issue 1.0. 54 Regent Street, London W1R 5PJ, June 1997.

[7] Geoffrey B. Rhoads. Steganography methods employing embedded calibration data. Digimarc Corporation. US Patent 5,636,292, 3 June 1997.

[8] E. Koch and J. Zhao. Towards Robust and Hidden Image Copyright Labeling. In *Workshop on Nonlinear Signal and Image Processing*, pages 452–455, Neos Marmaras, Greece, 20–22 June 1995. IEEE.

[9] Signum Technologies – SureSign digital fingerprinting. http://www.signumtech.com/, October 1997.

[10] Alpha Tec Ltd. EIKONAmark. http://www.generation.net/~pitas/sign.html, October 1997.

[11] I. Pitas. A method for signature casting on digital images. In *International Conference on Image Processing*, volume 3, pages 215–218, September 1996.

[12] Ross J. Anderson, editor. *Information hiding: first international workshop*, volume 1174 of *Lecture notes in Computer Science*. Springer Verlag, Berlin, Germany, May 1996. ISBN 3-540-61996-8.

[13] Ingemar J. Cox, Joe Kilian, Tom Leighton and Talal Shamoon. A Secure, Robust Watermark for Multimedia. In Anderson [12], pages 183—206.

[14] Markus G Kuhn and Fabien A.P. Petitcolas. StirMark. http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/, November 1997.

[15] Saeed Vahed Vaseghi. *Algorithms for restoration of archived gramophone recordings*. PhD thesis, Emmanuel College, University of Cambridge, UK, February 1988.

[16] Simon J. Godsill, Peter J.W. Rayner and Olivier Cappé. Digital audio restoration. In Mark Kahrs and Karlheinz Brandenburg, editors, *Applications of Digital Signal Processing to Audio and Electroacoustics*. Kluwer Academic Publishers, 1998.

[17] Giovanni audio marking software. Blue Spike company. http://www.bluespike.com/, May 1998.

[18] Raymond Veldhuis. *Restoration of lost samples in digital signals*. International Series in Acoustics, Speech and Signal Processing. Prentice Hall, Hertfordshire, UK, 1990.

[19] Daniel Gruhl, Walter Bender and Anthony Lu. Echo hiding. In Anderson [12], pages 295–315.

[20] Bruce P. Bogert, M.J.R. Healy and John W. Tukey. The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Ceptstrum and Saphe Cracking. In M. Rosenblatt, editor, *Symposium on Time Series Analysis*, pages 209–243, New-York, USA, 1963. John Wiley & Sons, Inc.

[21] Scott Craver, Nasir Memon, Boon-Lock Yeo and Minerva M. Yeung. Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications. *IEEE Journal of Selected Areas in Communications (J-SAC) – Special Issue on Copyright & Privacy Protection*, pages 573–586, 16(4), May 1998. ISSN 0733-8716.

[22] Marc Cooperman and Scott A. Moskowitz. Steganographic method and device. The DICE Company. US Patent 5,613,004, 18 March 1995.

[23] Alexander Herrigel, Adrian Perrig and Joseph J.K. Ó Ruanaidh. A Copyright Protection Environment for Digital Images. In *Verläßliche IT-Systeme* '97, Albert-Ludwigs Universität, Freiburg, Germany, October 1997.

[24] David Aucsmith, editor. *Information hiding: second international workshop*, Lecture Notes in Computer Science, Portland, Oregon, USA, 1998. Springer Verlag, Berlin, Germany. (to appear)

[25] Alexander Herrigel, Joseph J.K. Ó Ruanaidh, Holger Petersen, Shelby Pereira, and Thierry Pun. Secure copyright protection techniques for digital images. In Aucsmith [24], pages—.

[26] Adrian Perrig. A copyright protection environment for digital images. Diploma dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, February 1997.

[27] Ross J. Anderson. Why cryptosystems fail. *Communications of the ACM*, 37(11):32-40, November 1994.

[28] Anonymous (zguan.bbs@bbs.ntu.edu.tw). Learn cracking IV – another weakness of PictureMarc. news:tw.bbs.comp.hacker mirrored on http://www.cl.cam.ac.uk/~fapp2/watermarking/image_watermarking/digimarc_crack.html, August 1997. Includes instructions to override any Digimarc watermark using PictureMarc.

[29] David Aucsmith. Tamper resistant software: An implementation. In Anderson [12], pages 317-333.

[30] Ross J. Anderson. Stretching the limits of steganography. In Anderson [12], pages 39-48.

[31] Jean-Paul M.G. Linnartz and Marten van Dijk. Analysis of the sensitivity attack against electronic watermarks in images. In Aucsmith [24], pages –.

Audio Watermarking and Data Embedding - Current State of the Art, Challenges and Future Directions -

Mitchell D. Swanson[1], Bin Zhu[1], and Ahmed H. Tewfik[1,2]

[1]Cognicity, Inc.
7171 Ohms Lane, Edina
MN 55439 USA
(612) 841-7100

info@cognicity.com

[2]Dept. of Electrical and Computer Engineering
University of Minnesota
4-174 EE/CSCI Building
200 Union St. SE, Minneapolis
MN 55455 USA
(612) 625-6024

tewfik@ece.umn.edu

ABSTRACT

Data embedding algorithms embed binary streams in host multimedia signals. The embedded data can add features to the host multimedia signal or provide copyright protection. We review developments and requirements in transparent data embedding techniques for audio signals. We describe our latest audio embedding algorithm and include experimental results indicating remarkable robustness to low bit rate MPEG-Layer 3 and Dolby AC-3 coding. We conclude with a discussion of future research directions.

KEYWORDS

Data embedding, data hiding, watermarking, copyright protection, steganography

1 Introduction

The past few years have seen an explosion in the use of digital media. Digital media offers several distinct advantages over analog media including easy access, manipulation, and transmission. These advantages have opened many new possibilities. In particular, it is possible to hide data (information) within signals. The information is hidden in the sense that it is perceptually and statistically undetectable. With many schemes, the hidden information can still be recovered if the host signal is modified.

Digital data embedding has many applications. Foremost is passive and active copyright protection. Many of the inherent advantages of digital signals increase problems associated with copyright enforcement. For this reason, creators and distributors of digital data are hesitant to provide access to their intellectual property. Digital watermarking has been proposed as a means to identify the owner or distributor of digital data.

Data embedding also provides a mechanism for embedding important control, descriptive or reference information in a given signal. This information can be used for tracking the use of a particular clip, including billing for commercials and audio broadcast. It can be used to track audio creation, manipulation and modification history within a given signal without the overhead associated with creating a separate header or history file. It can also be used to track access to a given signal. This information is important in rights management applications.

Data embedding is also ideally suited for covert communications. Data embedding can securely hide large amounts of potentially encrypted information into an audio signal.

A most interesting application of data embedding is providing different access levels to the embedded data. For example, the quality of an audio signal can be controlled. A person with a high access level can hear details that another person with a lower access level would not hear. Similarly, data embedding allows users to tailor an audio signal to their needs, e.g., by listening to a song broadcast over a single channel in a particular rating. In this case, data embedding is used to embed alternative lyrics in a given version of the song that is broadcast.

The goal of this paper is to present an overview of the challenges and issues that need to be addressed by successful watermarking and data embedding techniques and the current state of the art. In the next section, we review requirements for data embedding algorithms. Insertion of the data into audio signals is then described, followed by previous work in the field. Our latest research results for a robust perception-based audio data hiding algorithm are then presented. We conclude with a dis-

cussion on future directions for audio data embedding.

2 Data Embedding Requirements

As mentioned in the Introduction, data embedding can be used in many different applications. Obviously, different applications will have different requirements. Therefore, there is no unique set of requirements that all data embedding techniques must satisfy. Nevertheless, certain requirements must be satisfied in several application areas. In this section, we shall review some of these requirements and indicate when they are important [1].

2.1 Perceptual Transparency

In most applications, such as copyright and usage tracking, the algorithms must embed data without affecting the perceptual quality of the underlying host signal. Furthermore, data embedding should not produce artifacts that are perceptually dissimilar from those that may be detected in an original host signal.

2.2 Recovery Of Data With Or Without Access To Original Signal

In some applications, such as copy tracking and copyright protection, the data extraction algorithms may use the original signal to decode the embedded data. However, in most applications, data embedding algorithms do not have access to the original audio signal while extracting of the embedded signal. This inability to access the original signal limits the amount of data that can embedded in a given host signal. It also renders data extraction more difficult.

Specifically, the embedded data may be considered as information transmitted on a communication channel and corrupted by a strong interference and channel effects. The strong interference consists of the host signal. Channel effects correspond to post-processing operations. Most data extraction procedures are inherently projection techniques on a given direction. Ideally, a larger projection value will indicate the presence of one type of data, e.g., a binary symbol or a watermark that represents an author. A segment of the original host signal that is highly correlated with the projection direction will provide a *false detection*. Furthermore, it may be impossible to modify that segment to reduce its correlation with the projection direction without affecting the perceptual quality of the host signal. Hence, the algorithm may be unable to embed useful data into that segment.

Note that the projection direction cannot be easily changed since the decoder does not have access to the original host signal. Any change in that direction must be accomplished through an algorithm that uses the received modified host signal. Note also that the probability of getting a high correlation between an arbitrary segment of the host signal and the projection direction decreases as the size of the segment increases, i.e., an increase in process gain. However, as that size increases, the amount of data that can be embedded in the host signal decreases.

Post-processing effects can complicate the detection process. For example, synchronization problems may arise as a consequence of temporal rescaling, cropping, resampling, etc. Many modifications lead to new signals which have a different number of samples than the original signal with embedded data. To extract the embedded information, the extraction algorithm must adapt to the new signal with fewer samples automatically or access the original to register the signal. Note however that loss of synchronization does not imply that the embedded data has been erased.

2.3 Bit Rate Of Data Embedding Algorithm

Some applications of data embedding, e.g., insertion of a serial number or author identification, require that relatively small amounts of information be incorporated repeatedly in the signal. However, in some envisioned applications of data embedding, e.g., covert communications, the algorithms must be able to embed an amount of data that is a significant fraction of the amount of data in the host signal.

2.4 Robustness

Lossy signal processing operations are frequently applied to the host audio. Operations that damage the host signal also damage the embedded data. Furthermore, third parties may attempt to modify the host signal to thwart detection of the embedded data. The data embedding algorithm must often survive modifications including:

- additive and multiplicative noise;
- linear and nonlinear filtering, e.g., lowpass filtering;
- compression, e.g., MPEG audio layer 3, Dolby AC-3;
- local exchange of samples, e.g., permutations;
- quantization of sample values;
- temporal scaling, e.g., stretch by 10%;
- removal or insertion of samples;
- averaging multiple watermarked copies of a signal;
- D/A and A/D conversions;
- a second embedded signal;
- frequency response distortion;
- group-delay distortions;
- frequency notches and hopping.

Several of these requirements were proposed by the Recording Industry Association of America (RIAA) and the International Federation of the Phonographic Industry (IFPI). In an effort to protect owners of digital audio, the entities issued a Request for Proposals in mid 1997 seeking a technology to inaudibly embed data in audio signals. The evaluation process was carried out by the MUSE Project that is jointly funded by the recording industry and the European Union. Several commercial systems were submitted. However, perceptual tests revealed that some of the systems tested were audible. Furthermore, the robustness tests indicated very mixed results [2]. The systems did not meet the requirements, despite initial claims. As a result, a second round of proposals was requested. The second round is currently under investigation.

2.5 Security

In many applications the embedding procedure must be secure in that an unauthorized user must not be able to detect the presence of embedded data, let alone remove the embedded data. Security requirements vary with application. The most stringent requirements arise in covert communication scenarios. Security of data embedding procedures is interpreted in the same way as security of encryption techniques. A secure data embedding procedure cannot be broken unless the unauthorized user has access to a secret key that controls the insertion of the data in the host signal. Hence, a data embedding scheme is truly secure if knowing the exact algorithm for embedding the data does not help an unauthorized party detect the presence of embedded data. An unauthorized user should not be unable to extract the data in a reasonable amount of time even if he knows that the host signal contains data and is familiar with the exact algorithm for embedding the data. Note that in some applications, e.g., covert communications, the data may also be encrypted prior to insertion in a host signal.

2.6 Copyright Protection And Ownership Deadlock

Data embedding algorithms may be used to establish ownership and distribution of data. In fact, this is the application of data embedding or watermarking that has received most attention in the literature. Unfortunately, most current watermarking schemes are unable to resolve rightful ownership of digital data when multiple ownership claims are made, i.e., when a deadlock problem arises. The inability of many data embedding algorithms to deal with deadlock, first described by Craver et al. [3], is independent of how the watermark is inserted in the multimedia data or how robust it is to various types of modifications. Solutions to the issues described in [3] were derived independently in [4] and [5].

3 Signal Insertion: The Role Of Masking

The first problem that all data embedding and watermarking schemes need to address is that of inserting data in the digital audio without deteriorating its perceptual quality. Of course, we must be able to retrieve the data from the edited host signal, i.e., the insertion method must also be invertible. Since the data insertion and data recovery procedures are intimately related, the insertion scheme must take into account the requirement of the data embedding application. In many applications, we will need to be able to retrieve the data even when the host signal has undergone modifications, such as compression, editing or translation between formats, including A/D and D/A conversions.

Data insertion is possible because the digital media is ultimately consumed by a human. The human hearing system is an imperfect detector. Audio signals must have a minimum intensity level before they can be detected by a human. These minimum levels depend on the temporal and frequency characteristics of the human auditory system. Further, the human hearing system is characterized by an important phenomenon called masking. Masking refers to the fact that a component in a given audio signal may become imperceptible in the presence of another signal called the masker. Most signal coding techniques (e.g., [6]) exploit the characteristics of the human auditory system directly or indirectly. Likewise, all data embedding techniques exploit the characteristics of the human auditory system implicitly or explicitly. In fact, embedding data would not be possible without the limitations of the human auditory system. For example, it is not possible to modify a binary stream that represents programs or numbers that will be interpreted by a computer. The modification would directly and adversely affect the output of the computer.

4 The Human Auditory System

Audio masking is the effect by which a faint but audible sound becomes inaudible in the presence of another louder audible sound, i.e., the masker [7]. The masking effect depends on the spectral and temporal characteristics of both the masked signal and the masker.

Frequency masking refers to masking between frequency components in the audio signal. If two signals which occur simultaneously are close together in frequency, the stronger masking signal may make the weaker signal inaudible. The masking threshold of a masker depends on the frequency, sound pressure level (SPL), and tone-like or noise-like characteristics of both the masker and the masked signal. It is easier for a broadband noise to mask a tonal, than for a tonal signal to mask out a broadband noise. Moreover, higher frequency signals are more easily masked.

The human ear acts as a frequency analyzer and can detect sounds with frequencies which vary from 10 Hz to 20000 Hz. The HAS can be modeled by a set of bandpass filters with bandwidths that increase with increasing frequency. The bands are known as the critical bands. The critical bands are defined around a center frequency in which the noise bandwidth is increased until there is a just noticeable difference in the tone at the center frequency. Thus if a faint tone lies in the critical band of a louder tone, the faint tone will not be perceptible.

Frequency masking models are readily obtained from the current generation of high quality audio codecs, e.g., the masking model defined in ISO-MPEG Audio Psychoacoustic Model 1, for Layer I [8].

Temporal masking refers to both pre- and post-masking. Pre-masking effects render weaker signals inaudible before the stronger masker is turned on, and post-masking effects render weaker signals inaudible after the stronger masker is turned off. Pre-masking occurs from 5-20 msec. before the masker is turned on while post-masking occurs from 50-200 msec. after the masker is turned off [7]. Note that temporal and frequency masking effects have dual localization properties. Specifically, frequency masking effects are localized in the frequency domain, while temporal masking effects are localized in the time domain.

5 Previous Audio Work

Several techniques have been proposed in the literature. Most are based on spread spectrum methods and are inherently projection techniques on a given key-defined direction.

Several approaches are described in [9]. The techniques include embedding data by modifying the phase values of Fourier Transform (FT) coefficients, spread spectrum, and echo coding. Another audio data embedding technique is proposed in [10], where FT coefficients over the middle frequency bands, 2.4 to 6.4 kHz, are replaced with spectral components from a signature. Pruess et. al. [11] embed data into audio by shaping a pseudo-noise sequence according to the shape of the original signal.

Some commercial products are also available. The *Identification Code Embedded* (ICE) system from Central Research Laboratories inserts a pair of very short tone sequences into an audio track. Solana Corporation's *Electronic DNA* (E-DNA) embeds data into subbands of the audio signal using a spread spectrum technique. The Dice Company also has a technique for encoding information into digital multimedia data. Cognicity, Inc., offers *Audio-Key*, an audio data embedding algorithm based on the algorithm discussed below.

Advanced audio embedding algorithms take into account perceptual masking. Moses [12] proposes a technique to embed data by encoding it as one or more whitened direct sequence spread spectrum signals/FSK signals and transmitted such that the signal is masked by the audio signal. In [13], the authors present an audio watermarking algorithm that exploits temporal and frequency masking by adding a perceptually shaped pseudo-random sequence.

6 Current Research

Our current work on perceptual audio data embedding techniques aggressively pursues the requirements mentioned in Section 2. The approach is designed to be flexible, e.g., embedding data rates that range from low to high, depending on the application. The algorithm employs a projection of an audio's frequency subbands onto a pseudo-random direction dictated by a secret key. The projection is followed by a *non-linear quantization step* to avoid the need for the original audio signal during extraction. Furthermore, the detection process includes a sophisticated searching mechanism to properly synchronize with the embedded data without access to the original audio signal. Note that the process does *not* require long random sequences to obtain the significant process gain factor required by the popular spread-spectrum systems. The technique uses a non-linearity to avoid the conventional use of matched filters.

The data embedding algorithms supports many features, including the ability to embed data into multiple (potentially overlapping) frequency bands. The bands are modified in such a way as to produce minimal interband distortion. Furthermore, the data in multiple bands may be embedded all at once, or in multiple passes. Such a feature is beneficial in copyright ownership and tracking environments where an audio signal may be repeatedly stamped with sales and tracking information. It is also useful for maintaining a modification history of an audio clip.

The data embedding algorithm is designed to be robust to many distortions. To illustrate the robustness to distortions, 21 mono and 14 stereo audio signals representing a large assortment of audio characteristics, e.g., impulses, tonals, etc., were tested. Eight of the stereo signals, muse_1, muse_2, etc., are components of the MUSE Embedded Signalling audio test material described in Section 2.

Text data was embedded into the audio at a rate of 42 bits/second (i.e., 6 printable characters/second). Two bands of the audio are used in the experiment. The embedded text data was random in nature. No knowledge of the data structure or length, e.g., an *N* Byte repeating ID code, was used to improve detection performance. A total of 119042 bits were embedded in the 35 audio signals.

The audio signals were encoded using Sonic Foundry's commercial Dolby AC-3 software codec. The

mono signals were encoded at a rate of 56 kbps, while the stereo signals were encoded at 96 kbps. The detection results for the mono and stereo signals are shown in Tables 1 and 2, respectively. The left column in each table consists of the audio clip's name. The next two columns, B1 and B2, list the number of bit errors made by the detection algorithm in each band. The last column lists the total number of bits embedded in the clip. Of the 119042 bits embedded into the mono and stereo audio signals, only 255 bits were incorrectly decoded. The resulting mean bit error rate (BER) is 0.21%. Note that the MUSE tracks contain 160 errors out of 91392 bits for a mean BER of 0.17%.

ac3, 21 bps	Bit Errors		Bits
Audio	B1	B2	Embedded
bach	0	0	798
castanet	0	0	252
clarinet	0	0	546
cooder	0	1	1974
drum	7	2	882
lovettl	0	0	1428
lovettr	1	2	1428
moon	12	9	672
piano	10	11	420
prokofiev	0	1	756
ritenourl	3	6	1330
ritenourr	3	9	1330
svega	1	1	966
tchaikov	0	0	672
titanic_a10m	0	1	420
titanic_a30m	2	3	1344
titanic_b10m	0	0	420
titanic_b30m	0	1	1344
vivc	1	0	462
yoyomal	0	0	1302
yoyomar	1	3	1302
TOTALS	41	50	20048
		BER	0.45%

Table 1. Bit errors in mono signals after AC-3 coding at 56 kbps.

ac3, 21 bps	Bit Errors		Bits
Audio	B1	B2	Embedded
lovett	0	0	1428
ritenour	1	3	1344
titanic_a10	0	0	420
titanic_a30	0	0	1344
titanic_b10	0	0	420
titanic_b30	0	0	1344
muse_1	5	9	11718
muse_2	2	2	17332
muse_3	30	40	7658
muse_4	3	1	11046
muse_5	0	8	10668
muse_6	13	14	8820
muse_7	5	2	11760
muse_8	12	14	12390
yoyoma	0	0	1302
TOTALS	71	93	98994
		BER	0.17%

Table 2. Bit error in stereo signals after AC-3 coding at 96 kbps.

A similar test was conducted for the MPEG Layer-3 (mp3) audio codec. The software used to encode the signals was Opticom's commercial .mp3 Producer Pro v 2.1 based on the MPEG Layer-3 audio compression technology and software implementation licensed from the Fraunhofer IIS. The detection results for the mono and stereo clips after coding the audio are shown in Tables 3 and 4. Again, the mono signals were encoded at a rate of 56 kbps, while the stereo signals were encoded at 96 kbps. The mean BER for the stereo and mono signals is 1.73%. The eight MUSE tracks have a mean BER of 1.92% at *96 kbps* stereo mp3 coding. Recall that the first round of competitors in the MUSE proposal *failed* to satisfy robustness requirements at *128 kbps* stereo mp3 coding.

mp3, 21 bps	Bit Errors		Bits
Audio	B1	B2	Embedded
bach	1	1	798
castanet	2	2	252
clarinet	0	1	546
cooder	4	5	1974
drum	7	5	882
lovettl	4	4	1428
lovettr	5	2	1428
moon	13	13	672
piano	8	12	420
prokofiev	5	5	756
ritenourl	6	5	1330
ritenourr	8	10	1330
svega	4	4	966
tchaikov	1	1	672
titanic_a10m	0	0	420
titanic_a30m	2	1	1344
titanic_b10m	1	1	420
titanic_b30m	1	1	1344
vivc	0	0	462
yoyomal	3	0	1302
yoyomar	8	3	1302
TOTALS	83	76	20048
		BER	0.79%

Table 3. Bit errors in mono signals after mp3 coding at 56 kbps.

mp3, 21 bps	Bit Errors		Bits
Audio	B1	B2	Embedded
lovett	13	14	1428
ritenour	12	17	1344
titanic_a10	2	5	420
titanic_a30	11	10	1344
titanic_b10	6	3	420
titanic_b30	13	14	1344
muse_1	23	33	11718
muse_2	157	149	17332
muse_3	49	67	7658
muse_4	28	31	11046
muse_5	136	127	10668
muse_6	79	64	8820
muse_7	227	244	11760
muse_8	170	168	12390
yoyoma	16	11	1302
TOTALS	942	957	98994
		BER	1.92%

Table 4. Bit error rates in stereo signals after mp3 coding at 96 kbps.

Of course, the mean BER drops further as the embedded data rate is reduced to 28 bits/s and 14 bits/s. For example, the mean BER for the AC-3 coded audio with an embedded data rate of 14 bits/s drops to 0.01%.

Extensive experimental results indicate that the algorithm is capable of surviving multiple sampling rates, time scaling, D/A and A/D conversions, and RealNetwork's streaming audio format.

The new algorithm includes further enhancements to the perceptual quality of the embedded audio signal using additional characteristics of the temporal and frequency masking phenomena. A formal investigation into the perceptual quality is currently underway with third-party "golden ear" professionals. Preliminary tests performed by audio engineers indicate that it outperforms our previous audio data embedding algorithm that proved transparent in a series of blind tests on a mixed background audience.

7 Future Directions

As described in Section 2.2, many of the current audio data embedding techniques are based on spread spectrum techniques and are inherently projection techniques on a given direction. Ideally, a larger projection value will indicate the presence of a binary symbol that represents an author. To reduce the probability of a false detection, the length of the audio segment and pseudo-random direction are increased to reduce the chances of a high correlation between the original host signal and the pseudo-random sequence. This is inevitable in the audio environment, where typical audio signals have a strong broadband nature that interferes with the spectrum of the pseudo-random sequence which represents the projection direction. As the size increases, the amount of data that can be embedded in the host signal decreases. Furthermore, long blocks increase problems associated with distortions. For example, the computational requirements of synchronization algorithms are frequently higher than order N, where N is the length of the audio block. As a result, the detection speed performance after the numerous distortions listed in Section 2.4, most of which require synchronization, will be very poor. Future audio data embedding algorithms should avoid the overused spread spectrum/matched filter approach. As our knowledge of masking improves, the capacity and robustness of these algorithms will improve. Further, future data embedding algorithms are likely to implement active control over the audio clips and use more sophisticated signal dependent keys.

8 References

[1] M. Swanson, M. Kobayashi, A. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies," Proc. of IEEE, Vol. 86, No. 6, June 1998, pp. 1064-1087.

[2] IFPI MUSE Project: Embedded Signalling http://www.ifpi.org/technology/muse_embed.html

[3] S. Craver, N. Memon, B-L. Yeo, M. Yeung, "Can invisible watermarks resolve rightful ownership?," IBM Research Report RC20509, July, 1996. Also SPIE Storage and Retrieval for Image and Video Databases V, vol. 3022, pp. 310-321, Feb. 1997.

[4] S. Craver, N. Memon, B-L. Yeo, M. Yeung, "Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications," IBM Research Report RC20755, March, 1997.

[5] M. Swanson, B. Zhu, A. Tewfik, "Multiresolution Video Watermarking using Perceptual Models and Scene Segmentation," IEEE J. on Selected Areas in Communications, vol. 16, no. 4, May 1998, pp. 540-550. Also vol. II, pp. 558-561, Proc. ICIP '97.

[6] N. Jayant, J. Johnston, R. Safranek, "Signal compression based on models of human perception," Proc. of the IEEE, Vol. 81, Oct. 1993, pp. 1385-1422.

[7] P. Noll, "Wideband speech and audio coding," IEEE Communications Magazine, Nov. 1993, pp. 34-44.

[8] ISO/IEC IS 11172, Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage up to about 1.5 Mbits/s.

[9] D. Gruhl, A. Lu, W. Bender, "Techniques for data hiding", IBM Systems Journal, Vol. 35, Nos. 3 & 4, pp. 313-336.

[10] J. F. Tilki, A. A. Beex, "Encoding a Hidden Digital Signature onto an Audio Signal Using Psychoacoustic Masking", in Proc. 1996 7th Int. Conf. on Sig. Proc. Apps. And Tech., pp. 476-480.

[11] R. Preuss, S. Roukos, A. Huggins, H. Gish, M. Bergamo, P. Peterson, "Embedded Signalling", U. S. Patent 5,319,735, 1994.

[12] D. Moses, "Simultaneous Transmission of Data and Audio Signals by Means of Perceptual Coding," U. S. Patent 5,473,631, 1995.

[13] M. Swanson, B. Zhu, A. Tewfik, L. Boney, "Robust Audio Watermarking Using Perceptual Masking", Signal Processing, vol. 66, no. 3, May 1998, pp. 337-355.

Watermarking in the Real World: An Application to DVD

Matt L. Miller
Signafy, Inc.
4 Independence Way
Princeton, NJ 08540
(609) 734-7620
mlm@signafy.com

Ingemar J. Cox
NEC Research Institute
4 Independence Way
Princeton, NJ 08540
(609) 951-2722
ingemar@research.nj.nec.com

Jeffrey A Bloom
Signafy, Inc.
4 Independence Way
Princeton, NJ 08540
(609) 734-7620
bloom@signafy.com

ABSTRACT

The prospect of consumer DVD recorders highlights the challenge of protecting copyrighted video content from piracy. Digital watermarking can be used as part of a copy protection. We describe the copy protection system currently under consideration for DVD. We will also highlight some implementation issues that are being addressed.

KEYWORDS

Watermarking, DVD, copy protection

1 Introduction

Digital multimedia watermarking is a field that has received an increasing degree of interest from researchers in both academic and practical settings. The fundamental challenge is to hide a piece of information into a digital image file or a video or audio stream (also referred to as the cover material) such that the information is not perceived and cannot be removed without causing significant perceptual degradation to the cover [1]. Since the watermark is embedded into the media, it has the property that it will undergo the same transformations as the media and can thus be used as an indicator of what those transformations may have been.

Some potential applications include the use of a watermark as a signature identifying the copyright owner, as a fingerprint identifying the customer of the cover media, as an authentication key describing some feature of the media which would likely change if the cover were manipulated, or as a copy control mechanism indicating copy permission. Most of these applications rely on the property that watermarks are not easily separated from the content or cover media and, consequently, research into watermarking has focused on the problem of making watermarks difficult to remove without making them perceptible[2].

Since the middle of 1996, we have been working on a copy control application in which watermarks will be one part of a system for protecting video on digital versatile disks (DVD). While the difficulty of removing watermarks is an important problem in this application, we have been confronted with a wide variety of other problems that have been given much less attention in the literature. In this paper we will briefly describe the DVD copy protection framework in which watermarking technology is to be applied and present some of the technical challenges which have not yet been adequately addressed.

2 Application Framework – DVD Copy Protection System

In 1996, the Motion Picture Association of America (MPAA), the Consumer Electronics Manufacturers Association (CEMA), and members of the computer industry put together an ad hoc group to discuss the technical problem of protecting digital video from piracy, particularly in the domain of DVD [3]. This group, the Copy Protection Technical Working Group (CPTWG), is open to anyone who wishes to participate, and has no official decision-making power. However, over the past year and a half, it has succeeded in designing the major part of a copy protection system that is likely to become the defacto standard for DVD.

Two major principals have guided the CPTWG's work. The first principal is that the copy protection system should not be mandatory. This immediately divides devices into two categories: "compliant" devices, which implement the protection system, and "non-compliant" devices, which do not. The media to be protected must be scrambled in such a way that it cannot play on non-compliant devices, or else there will be no protection at all.

The second principal is that the system must be cost-effective. This means it is unlikely to be secure against determined hackers, since that level of security would require more computing power than is reasonable in low-cost consumer devices. Rather, the aim is to come up with a system that is cheap, and good enough to prevent casual copying by the average user. The design mantra is "keeping honest people honest."

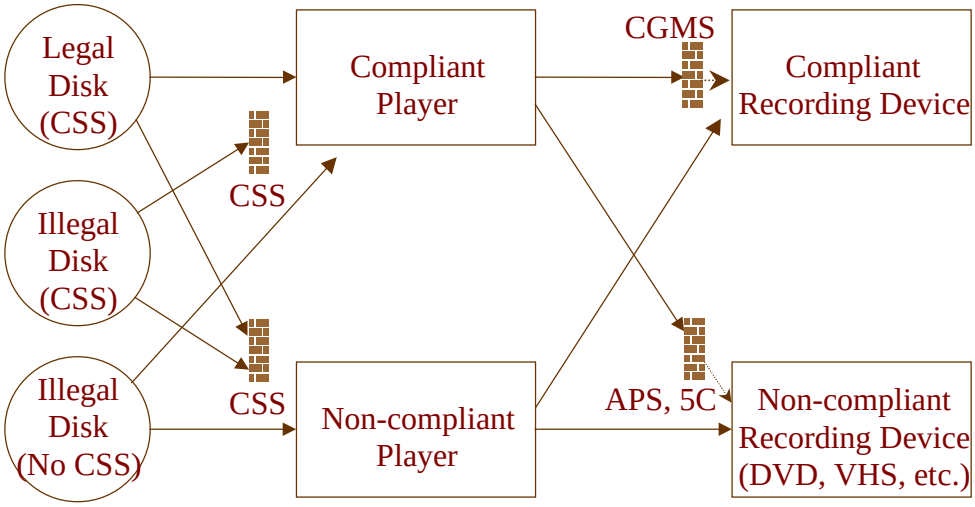


Figure 1. DVD copy protection system *without* watermarking

The system designed by the CPTWG is still a work in progress. At present, there are three components that are already being built into consumer devices. These are the Content Scrambling System (CSS), the Analog Protection System (APS), and the Copy Generation Management System (CGMS). Two additional components are being seriously considered: a system for secure communications across a PC bus (designed by a coalition of 5 companies, and hence referred to as 5C), and watermarking. The watermarking component, of course, is the topic of this paper. The other four components are briefly described below.

- CSS is a low-cost method of scrambling MPEG-2 video, developed by Matsushita. To descramble the video, a device requires a pair of keys. One of the keys is unique to the disk, while the other is unique to the MPEG file being descrambled. The keys are stored on the lead-in area of the disk, which is generally only read by compliant drives. Keys can be passed from a DVD drive to a descrambler over a PC bus using a secure handshake protocol (different from 5C).

The purpose of CSS is twofold. First and foremost, it prevents byte-for-byte copies of an MPEG stream from being playable, since such copies won't include the keys. Second, it provides a reason for manufacturers to make compliant devices, since CSS scrambled disks won't play on non-compliant devices. Anyone wishing to build compliant devices must obtain a license, which contains the requirement that the rest of the copy protection system be implemented.

- The APS system, developed by Macrovision, is a method of modifying NTSC signals so that they can be displayed on televisions, but cannot be recorded on VCR's. It works by confusing the automatic gain control in VCR's, and this usually leads to unwatchable recordings. Before being adopted for DVD, it has been widely used on videocassettes.

Of course, the data on a disk is not NTSC encoded, so APS has to be applied by the NTSC encoder in a DVD player. The information of whether a given video stream should have APS applied, and details about how it should be applied, is stored in the MPEG stream header.

- CGMS is simply a pair of bits in the header of an MPEG stream that encode one of three possible rules for copying: "copy-always" (the video may be freely copied), "copy-never" (the video may never be copied), or "copy-once" (a first generation copy may be made, but no copies may be made of that copy). The copy-once case is included to support such uses as time shifting, where a copy of broadcast media is made for later viewing. Copy-once is unlikely to appear on pre-recorded disks, but it is important for DVD recorders to support it.

- The proposed secure transmission system, 5C, provides a mechanism for pairs of compliant devices on a computer bus to exchange keys, so they can send encrypted data to one another that no other devices can decrypt. The system is more secure than the handshake used for CSS.

Development of 5C was prompted by the advent of high-speed computer busses such as 1394, which can potentially carry uncompressed digital video from a player or set-top-box to a monitor. The fear is that a pirate could tap into the bus and record any unencrypted video being transmitted.

The role of these copy protection devices is illustrated in Figures 1 and 2. Figure 1 shows the system without watermarking and demonstrates the need for watermarking. In this illustration we assume that

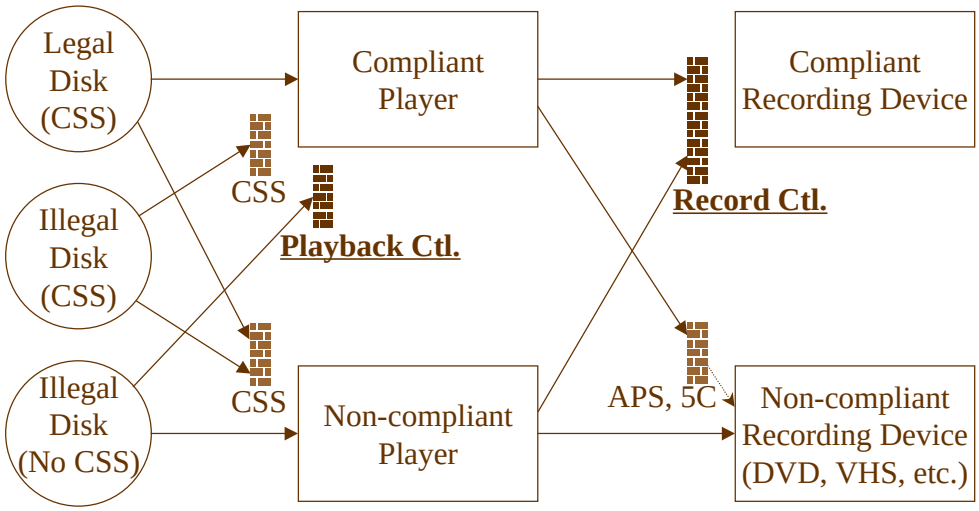


Figure 2. DVD copy protection system *with* watermarking

available in the marketplace will be both compliant and non-compliant players and recording devices. Three possible types of disks are considered: factory-pressed, legal disks containing copy protected video, bit-for-bit illegal copies of these disks, and illegal copies made of the video after descrambling.

Legal disks will be scrambled with CSS and can be played only on compliant devices. Bit-for-bit copies of these disks won't be playable on any devices, because they won't contain the descrambling keys. This in ensured by storing the keys on the lead-in area of the legal disk, which is only read by compliant drives. The compliant drives take precautions to prevent the keys from being copied.

CGMS is intended to prevent illegal copies, however a non-compliant player may strip out these copy control bits from the header, leaving the video in the clear, or unprotected. At this point there is nothing left to indicate copy restrictions to the compliant recording device and DVD RAM disks without CSS or CGMS can be generated.

Another potential weak point in the system is in the protection against copies being made on non-compliant recorders. APS works only on VCR's, and 5C works only when the display device is a compliant, digital monitor. If the output of the player is, for example, analog RGB, a pirate can simply route it into an appropriate non-compliant recorder and make an unencrypted copy. Of course, such a copy would not contain the CGMS bits.

Because of these two weaknesses, it can be expected that many unprotected, illegal copies will be made. These can be widely distributed, since they will play in either compliant or non-compliant devices. The purpose of introducing watermarking into this system is twofold: first, to improve the protection provided by CGMS by making the copy-control information harder to remove, and, second, to reduce the value of illegal, unencrypted copies when they are made, by making them unplayable on compliant devices.

Figure 2 shows the same scenario except that now watermarking is included. The two functions of the watermark mentioned above are referred to as "record control" and "playback control", respectively. Record control takes over the job of CGMS. It works regardless of how the video reaches the compliant recorder, since the watermark that contains the CGMS data is never removed by normal video processing.

Copy-once control can also be implemented in the compliant recording device. Recording of source data containing this copy-once watermark is allowed, however some modification is made to indicate a third state called copy-no-more which can be treated the same as copy-never.

Playback control introduces a new point of protection in the system. Should a pirate be successful in generating a DVD RAM copy of a protected video without CSS, this copy will still contain the watermark. Watermarking allows compliant players to recognize as illegal a video marked with copy-never that is being read from an unscrambled DVD RAM and refuse playback. This playback control limits the potential market for pirated DVD to those consumers who own non-compliant players, which will not play legal disks.

In the summer of 1997, after receiving presentations on watermarking technologies from several companies, the CPTWG set up the Data Hiding SubGroup (DHSG) to evaluate these systems and determine whether the technology is mature enough for inclusion in the copy protection system. The CPTWG issued a call for proposals [4] in July 1997. Eleven companies responded with proposals. After the initial round of testing, seven proposals remain under consideration.

The remainder of this paper describes some of the challenges that are faced by the companies that submitted proposals to the DHSG.

3 Challenges

As the copy protection system described above and illustrated in Figure 2 is implemented an array of challenges related to the watermarking technology have arisen. The issue of watermark removal is often addressed in watermarking literature and remains an important concern [5]. There are a number of other issues, some technical and some non-technical, which have also come to play an important role. In the remainder of this section we briefly introduce and discuss the following issues: enforcement, system tampering, detector placement within the system, computational cost of the detector, effects of geometric distortion, interaction between the watermarking and compression systems, false positive rates and analysis, and copy generation control.

Enforcement - One interpretation of Figure 2 is that the DVD world may be split in two, one compliant and one non-compliant. The copy protection system, specifically the watermarking technology and the CSS, will prevent legal copies from being played on non-compliant players and illegal copies from being played on compliant players. This does not stop consumers from owning two players, one compliant and one non-compliant, and does not prevent the sale of a "dual" player containing both compliant and non-compliant drives. The approach taken to discourage the manufacture of "dual" players is to note that both the CSS and watermarking technologies are protected by patents and may only be used in a DVD player with the proper licenses. These licenses will specify that the player must not possess the capability of playing non-compliant DVD sources. We will then rely on the expense of owning two DVD players and the fact that non-compliant DVD copy protected source is illegal as a violation of the content provider's legal copyright, to help "keep honest people honest."

System Tampering - The illegal copy without CSS of the Figure 1 scenario was rendered unplayable by the watermarking technology in Figure 2. This suggests that the pirate has an interest in being able to remove the watermark [5]. Watermarks that are image independent can easily be reconstructed by frame averaging and, once found, can be subtracted from the watermarked video source. Another documented "attack" on watermarks is called sensitivity analysis in which a detector is used to reconstruct the watermark in a frame by a systematic degradation of the image. Again, once found, the watermark can be subtracted from the video source. The field of watermark removal is very active and the robustness of watermarking techniques is constantly being challenged. While possession, sales, and distribution of illegal copies are prohibited by law, there are no such constraints on the sales of watermark removal hardware or software.

There are two common approaches to this problem. The most obvious approach is to invent a watermark that is truly tamper resistant. The other, perhaps more realistic approach may seem at first to be counter intuitive. A company that relies on the tamper resistance of a watermarking technology may wish to actively seek out, invent, and patent any reasonable technique for removing that watermark. Any watermark removal software or hardware using these techniques would then represent a patent infringement. A third approach is to introduce and pass legislation to outlaw the sale of watermark removal hardware or software. We understand that this is being considered, particularly as many countries must update their copyright law to support recent changes by the WIPO.

Beyond watermark removal there are other ways to circumvent the copy protection system. These include hardware modification to disable watermark detection and source scrambling such that the watermark detector does not recognize the source as watermarked video. In this latter case the video must be descrambled after it passes by the watermark detector. Neither of these two approaches can be used to generate an illegal copy that will play on compliant players.

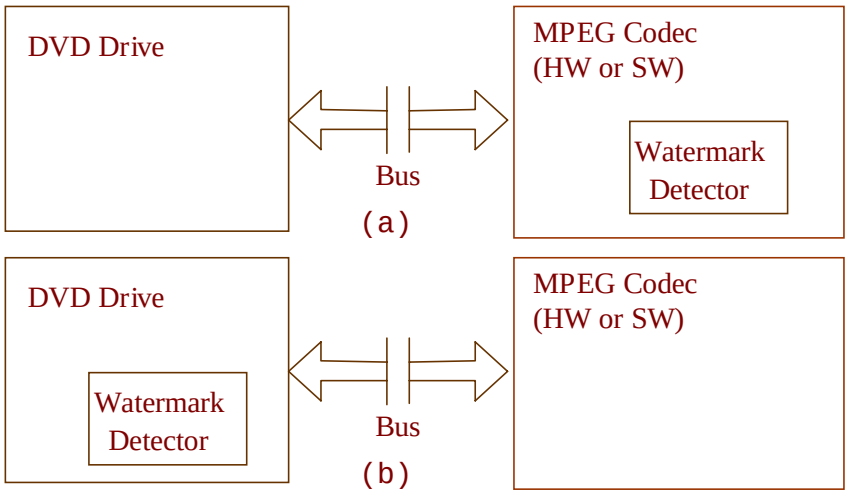


Figure 3. Watermark Detector Placement

Detector Placement – An issue of significant debate within the DHSG involves the physical placement of the watermark detector in the system. This is of particular interest for DVD drives installed in personal computers. Two reasonable approaches are shown in Figure 3. In the scenario of Figure 3a the watermark detector is located inside the MEPG codec and in Figure 3b it is in the DVD drive. Each of these solutions has its advantages and its disadvantages. Having the detector in the MPEG codec is an efficient solution since both the codec and the detector can share many of the same elements (tables, buffers, etc.). However, this solution also allows easy creation of a "dual" system in a computer, since most MPEG decoding applications will use non-compliant MPEG decoders.

The second scenario, which is currently leading in the debate, places the watermark detector in the DVD drive. This has the advantage that it is more tamper resistant. Record control will prevent watermarked, non-compliant MPEG bitstreams from being recorded. The DVD player also has knowledge of the disk type (ROM or RAM) from which the video is being read and can check for an allowed combination of disk type and watermark (e.g. copy-never and copy-once should not be found on a RAM disk).

Detector Computational Cost – Adding a watermark detector to a DVD RAM drive will require some degree of redesign. In order to minimize that cost, drive manufacturers have indicated that the detector must fit onto unused silicon that already exists in the drives. This restriction on the cost of the watermark detector in the DVD application means that the detector must be implemented in about 30k gates. A significant implication is that the detector may not use a frame buffer and must process the video in real time without reference to previous frames. This shows the asymmetry between the watermark embedder and decoder since the motion picture industry is likely to accept an embedder with very high computational cost and physical cost on the order of $100,000.

Geometric Distortion – DVD players have the facility to geometrically alter the video in two important ways. Letterbox is a technique which changes the aspect ratio from 4:3 to 16:9. Panscan represents a cropping of the larger image. The watermark must survive these geometric distortions as well as more arbitrary scaling and cropping which a pirate may use to avoid watermark detection. While these issues are generally addressed in watermarking literature, this special case where a frame buffer may not be available is particularly difficult.

Watermark/Compression Interaction – It can be argued that a goal of video compression, to remove all visually imperceptible information, makes the challenge of imbedding a visually imperceptible watermark much more difficult. If the watermark is placed in perceptually significant component, the source may be more difficult to compress.

In the DVD application, MPEG-2 compression is used and it is required that the watermark be detectable in both the compressed data stream and the reconstructed video. The former case requires detection in the block-based DCT domain (without frame buffers as previously mentioned) and both cases require that the watermark survive MPEG quantization. Another requirement is that the watermarks be modifiable in the compressed data stream without complete decompression and that the modifications not affect the bit-rate or position of I-frames.[6] The scalability features of MPEG-2 further complicate watermark detection and modification in the bitstream.

False Positive Rate – Watermark detection can generally be expressed as a binary decision and there are penalties associated with incorrect decisions. In the DVD application, when the detector decides that a watermark is present in video that does not contain a watermark, the result will be that a user cannot do some action that should be allowed. A couple might never be able to watch their wedding video. A football fan might not be able to record the Super Bowl for time shifting. The latter example is particularly catastrophic; if a piece of the Super Bowl triggers a false positive, *no one* will be able to record it on DVD. Our estimates of the required false positive rate are about one in 10^{11} or 10^{12} distinct frames. A recent model for predicting the false positive rate can be found in [7].

Copy Generation Control – There are a number of proposed methods for using watermarks in a copy generation control system. The goal is to detect a copy-once state and change it to a copy-no-more state as the video is being recorded. One approach is to use a watermark that can actually be changed. Recall that this will need to be done in the MPEG stream without changing the bitrate. This approach is likely to be more susceptible to tampering since the ability to change a watermark implies the ability to remove it.

Another approach involves the addition of a separate watermark. Thus the copy-once state will be indicated by the presence of one watermark and copy-no-more by the presence of both. To do this, the DVD recorder, with it's limited computational complexity and cost, must be able to insert the copy-no-more watermark. As with the other watermarks, this copy control watermark must be unobtrusive, indelible, and robust.

The opposite approach can also be taken where the presence of two watermarks, one of which is fragile, represents the copy-once state. The recorder then has the task of removing the fragile watermark. An interesting example of this can be found in the Macrovision/Digimarc proposal [4] in which the fragile watermark is a visible pattern (placed in the overscan area of the frame so that it will be hidden by the edge of a television screen). This pattern is designed in such a way that it cannot be recorded on a VCR. Thus, a copy on a VCR removes the fragile mark, and automatically converts copy-once into copy-no-more. Of course, a digital recorder will still have to remove the fragile watermark explicitly.

A completely different approach to generation control is to use information that is not embedded in the watermark, but must be available to the recorder if copy-once video is to be recorded. Such information, often referred to as a "tag" or "ticket", might be stored in MPEG headers or in the vertical blanking interval of analog video. In such a system, the copy-once state is represented by the presence of a copy-once watermark *and* an appropriate tag. The watermark without the tag would indicate copy-no-more. Since no mechanism would be provided for copying the tag, any copy would necessarily be labeled with copy-no-more. A weakness of this method is that all devices that do not copy the video, such as set top boxes, must preserve the tag. Current set top boxes would have to be modified for this purpose.

4 Conclusion

We have described here several of the difficult problems encountered in designing a real-world application of watermarking for copy control in DVD. While the problems of fidelity and robustness have received significant attention in the literature, several of the problems encountered here are less studied. The most notable of them are

- Interaction with compression algorithms
- Overall system design to avoid circumvention
- False positive rates
- Issues of computational costs, such as designing detectors without using frame buffers

The details and relative importance of these problems change with different applications. But they all pose fundamental challenges that must be met before watermarking can fulfill its promise as a tool for copyright protection.

5 References

[1] Cox, I.J.; Kilian, J.; Leighton, F.T.; Shamoon, T., Secure spread spectrum watermarking for multimedia, *IEEE Transactions on Image Processing*, vol.6, no.12, p. 1673-87, 1997.

[2] Cox, I.J. and Miller, M.L., Review of watermarking and the importance of perceptual modeling, Proc. SPIE, vol.3016, p. 92-9, 1997.

[3] Bell, A., Personal communication, 15 May, 1998.

[4] DHSG Call for Proposals, http://www.dvcc.com/dhsg.

[5] Cox, I.J. and Linnartz J-P., Some General Methods for Tampering with Watermarks, *IEEE Journal on Selected Areas in Communications*, Vol. 16, pp. 587-93, 1998.

[6] Hartung, F. and Girod, B., Digital watermarking of MPEG-2 coded video in the bitstream domain, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2621-4, 1997.

[7] Hernández, J. R., et.al., Performance analysis of a 2-D-multipulse amplitude modulation scheme for data hiding and watermarking of still images, *IEEE Journal on Selected Areas in Communications*, Vol. 16, pp. 510-24, 1998.

Digital Watermarking for Compressed Video

Frank Hartung
Telecommunications Laboratory
University of Erlangen
Cauerstr 7
D-91058 Erlangen
Germany
Phone +49 9131 8527116
hartung@nt.e-technik.uni-erlangen.de

Jonathan K. Su
Telecommunications Laboratory
University of Erlangen
Cauerstr 7
D-91058 Erlangen
Germany
Phone +49 9131 8527103
su@nt.e-technik.uni-erlangen.de

Bernd Girod
Telecommunications Laboratory
University of Erlangen
Cauerstr 7
D-91058 Erlangen
Germany
Phone +49 9131 8527100
girod@nt.e-technik.uni-erlangen.de

ABSTRACT

The ease of reproduction, distribution, and manipulation of digital documents creates problems for authorized parties that wish to prevent illegal use of such documents. To this end, digital watermarking has been proposed as a last line of defense. A digital watermark is an imperceptible, robust, secure message embedded directly into a document. The watermark is imperceptible both perceptually and statistically. Robustness means that the watermark cannot be removed or modified unless the document is altered to the point of no value. The watermark is secure if unauthorized parties cannot erase or modify it. Current watermarking schemes employ principles adopted from spread-spectrum communications systems, which transmit a message redundantly using a low-amplitude, pseudo-noise carrier signal. For compressed video, the embedding is done in the transform domain of the DCT encoded signal. With appropriate rate control and drift compensation mechanisms included, the bit-rate of the compressed video is not increased due to watermarking, and the watermark is not visible. The complexity is similar to the complexity of a video decoder, and the rate of the embedded watermark information is typically a few bytes per second. The principle applies to all video compression schemes employing motion compensation and DCT residual encoding, like MPEG-1, MPEG-2, MPEG-4, ITU-T H.261, and ITU-T H.263.

KEYWORDS

digital watermarking, multimedia security, video, compressed video, MPEG-2, MPEG-4.

1 Introduction

Digital media are replacing traditional analog media and will continue to do so. By digital media, we mean digital representations of audio, text documents, images, video, three-dimensional scenes, etc. These media offer many benefits over their analog predecessors. Analog media - such as audio cassettes and video tapes - degrade each time they are copied. Distribution of analog media is regulated and often requires special equipment (e.g., broadcasting equipment). In contrast, digital data can be stored, duplicated, and distributed with no loss of fidelity. The data can also be manipulated and modified easily, and editing software is readily available.

Perhaps the most important of these properties is the ease of distribution. With only a personal computer, some free or inexpensive software, and an Internet connection, virtually anyone can begin distributing digital media, which is accessible to millions of people. Clearly, digital media offer many benefits, but they also create problems for parties who wish to prevent illegal reproduction and distribution of valuable digital media (e.g., copyrighted, commercial, privileged, sensitive, and/or secret documents). Two classic methods for protecting valuable documents are encryption and copy protection. While encryption can protect documents against unauthorized access, the decrypted document can be copied and distributed easily. Likewise, many copy-protection mechanisms employ a header, which indicates whether or not the document may be copied. Bypassing the copy-protection mechanism is a relatively simple task. As a safeguard against failures of encryption and/or copy protection, digital watermarking [1] has been proposed as a "last line of defense" against unauthorized distribution of valuable digital media. A digital watermarking system embeds information directly into a document. For ex-

ample, information about copyrights, ownership, timestamps, and the legitimate receiver could be embedded. Thus, the document itself contains the information. Digital watermarking cannot by itself prevent copying, modification, and re-distribution of documents. However, if encryption and copy protection fail, watermarking allows the document to be traced back to its rightful owner and to the point of unauthorized use.

2 Digital watermarking

2.1 Requirements

In general, a watermark should comply to the following requirements:

- **Robustness**:The watermark should be reliably detectable after alterations to the marked document. Robustness means that it must be difficult (ideally impossible) to defeat a watermark without degrading the marked document severely.
- **Imperceptibility or a low degree of obtrusiveness**: To preserve the quality of the marked document, the watermark should not noticeably distort the original document.
- **Security**: Unauthorized parties should not be able to read or alter the watermark.
- **Fast embedding and/or retrieval**: The speed of a watermark embedding algorithm is important for applications where documents are marked "on-the-fly" (i.e., when they are distributed). The large bandwidth necessary for video also requires fast embedding methods.
- **No reference to original document**: For some applications, it is necessary to recover the watermark without requiring the original document.
- **Unambiguity**: A watermark must convey unambiguous information about the rightful owner of a copyright, point of distribution, etc. This requirement is a cryptographic and protocol issue.

Of these properties, robustness, imperceptibility, and security are usually the most important. When speaking of robustness, we often talk about attacks on a watermark. An attack is an operation on the marked document that, intentionally or not, may degrade the watermark and make the watermark harder to detect. For images and video, compression (e.g., JPEG or MPEG), filtering, cropping, resizing, and other signal processing manipulations (even printing and rescanning) must not destroy the watermark.

3 Digital watermarking of compressed video

3.1 Principle

Virtually all proposed watermarking methods build on the same basic principles, namely the application of small, unobtrusive, random-looking changes to the data that are however deterministic and can lateron be re-discovered by correlation [2]. For redundancy and robustness, each bit of watermark information is embedded into many pixels. For security, it is further modulated using a pseudo-random signal, as supplied by a random number generator, and added to the data, obeying amplitude limitations for imperceptibility.

We apply the principle to compressed video, as it is stored and distributed in real-world video distribution systems, like the WWW or video-on-demand servers. Direct manipulation of the video pixels is not possible, because decompression, watermarking and compression are far too complex.

In order to avoid decompression and re-compression, we apply a block-wise transform of the watermark signal using the DCT, that is, the same transform that is used in hybrid video compression schemes like MPEG-1 and MPEG-2. The compressed video sequence is then partly decoded in order to have access to the encoded DCT coefficients. The corresponding DCT coefficients of the watermark are then added to the coefficients of the video and re-encoded. All other parts of the video bitstream are simply copied into the new, watermarked, video bitstream. If the bit-rate of the watermarked video must not exceed the bit-rate of the unwatermarked video, a rate control can easily be applied which prevents such excess, at the cost of less robust watermark embedding. Details of the proposed watermarking method can be found in [3,4].

The watermark recovery is easily done after decompression of the watermarked sequence by employing a correlation receiver that knows the pseudo-noise signal used for embedding. The original (unwatermaked) video sequence is not required. Details can again be found in [3,4].

3.2 Properties Of The Proposed Method

The proposed method allows to robustly embed an invisible watermark into compressed video sequences. The watermark can carry arbitrary information, like information about souce and destination of the data, or copyright statements.

The embedding is done in the transform domain of the DCT encoded signal. When using the rate control mechanism, the bit-rate of the compressed video is not increased due to watermarking. A drift compensation mechanism avoids visible distortion in the sequence which could otherwise occur due to the iterative structure of video compression employing motion-compensated prediction. The embeddding complexity is similar to the complexity of a video decoder, and the rate of the embedded watermark information is typically a few bytes per second. The principle applies to all video compression schemes employing motion compensation and DCT residual encoding, like MPEG-1, MPEG-2, MPEG-4, ITU-T H.261, and ITU-T H.263.

With appropriate extensions of the basic scheme as described in [5], the watermarks resist all known at-

tacks and modifications, including collusion attacks and geometrical manipulations of the video sequence.

4 References

[1] H. Berghel and L. O'Gorman, "Protecting Ownership Rights Through Digital Watermarking", *IEEE Computer*, May 1996, pp. 101-103.

[2] I. Cox, J. Kilian, T. Leighton, T. Shamoon, "Secure Spread-Spectrum Watermarking for Multimedia", technical report, NEC, 1995.

[3] F. Hartung and B. Girod, " Digital watermarking of MPEG-2 coded video in the bitstream domain", Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, April 1997, vol. 4, pp.2621-2624.

[4] F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video", *Signal Processing*, vol. 66(3), pp. 283-301, May 1998.

[5] F. Hartung, J. Su, and B. Girod, "Spread Spectrum Watermarking: Malicious Attacks and Counter-Attacks", submitted to SPIE Conference on Security and Watermarking of Multimedia Contents 99.

Image Distribution with Scrambling and Watermarking

Takehito Abe
NTT Information and Communication Systems Labs.
1-1 Hikarinooka
Yokosukashi
Kanagawa, Japan
+81-468-59-2837
take@isl.ntt.co.jp

Hiroshi Fujii
NTT Information and Communication Systems Labs.
1-1 Hikarinooka
Yokosukashi
Kanagawa, Japan
+81-468-59-2639
fujii@dq.isl.ntt.co.jp

Youichi Takashima
NTT Human Interface Labs.
1-1 Hikarinooka
Yokosukashi
Kanagawa, Japan
+81-468-59-3990
yoh@mistral.hil.ntt.co.jp

KEYWORDS
image distribution, scrambling, watermarking, copyright.

1 Introduction

The advent of computer networks and mass storage media such as CD-ROMs has made it possible for anyone to distribute digital information easily and economically. In this new environment, image distribution methods suitable for electronic commerce are being widely studied. The most serious problem is piracy, which is an obstacle to spread digital image distribution over an open network. Hence, secure image distribution methods are strongly required.

2 Image Distribution

Digital images must be protected against piracy in the following two phases. The first is the dealing phase between providers and users. The second is the post distribution phase at the users' end. In first phase, users need to see a sample of the image prior to its purchase. However, illegal copying must be prevented. For this phase, we have proposed an image scrambling technique called the 'image partial scrambling method' [1], in which the image data are scrambled and only legal users can descramble the image. In the second phase, the distributed images must be protected from illegal copying. It is hard to prevent images from being copied with ordinary copy protection methods (e.g. computer program copy protection) because the image data must be exposed for display. For this phase, a watermarking technique is usually used [2]. By embedding copyright information with this technique, providers can prove if there has been a copyright violation.

If an individual user's information (e.g. user ID) is embedded in the image using this watermarking technique, providers will be able to administer their users. However, in order to embed individual user information in each image, providers must embed the information before distribution. Hence, these images cannot be distributed by mass-distribution method like CD-ROM or broadcasting. We propose a new image distribution method that uses image scrambling and watermarking. Our method makes it possible to embed individual user information into images distributed through mass-distribution media. In following section, we propose our image distribution method and describe the scrambling and watermarking technique used in our distribution method. We then describe its implementation to the practical system.

3 Protocol

The image distribution protocol is shown as Figure 1.

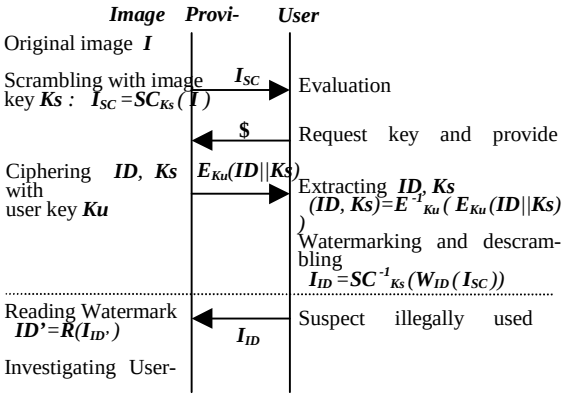


Figure 1. Protocol of digital image distribution and image investigation.

Description:
$SC_{Ks}( )$: scrambling with image key ***ks***
$SC^{-1}_{Ks}( )$: descrambling with image key ***ks***
|| : combining process
$E_{Ku}( )$: enciphering with user key ***ku***
$E^{-1}_{Ku}( )$: deciphering with user key ***ku***
$W_{ID}( )$: watermarking ***ID***
$R( )$: reading watermark

A provider makes a scrambling image I_{SC} using the image key ***Ks*** and then transmits I_{SC} to a user. The user evaluates I_{SC} and requests the key to descramble the image. The provider makes confidential data by mixing and enciphering ***Ks*** and the user information ***ID*** using user key ***Ku***, and then the provider sends the data to the user. In the descrambling module at

the user's end, ***Ks*** and ***ID*** are deciphered using ***Ku***, and the image is watermarked and descrambled at the same time. As a result, the user obtains an unscrambled image by purchasing only a little data, including cipher key, and moreover, the images are watermarked with individual user information.

4 Constituent technique

4.1 Scrambling

We have developed methods for scrambling and descrambling digital image data coded by JPEG or MPEG. In our method, an image is scrambled by altering the value of DCT coefficients directly in accordance with random numbers created from the ciphering key. The rough outlines of images remain after scrambling. The original image can be obtained by descrambling with the (de)cipher key. The scrambling process is very efficient. The quality of the images can be controlled by choosing which coefficients to alter and the degree of alteration. In addition, one image can be scrambled repeatedly.

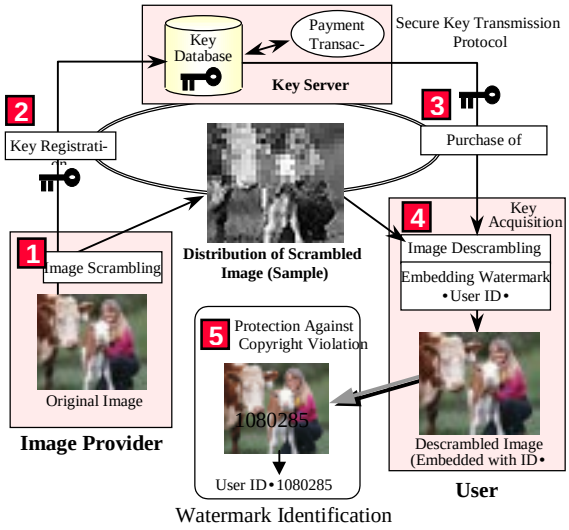


Figure 2. Image distribution system, InfoProtect.

4.2 Watermarking

Watermarking is a technique which imperceptibly embeds
sub-information into the main digital contents. There are two main requirements for watermarking scheme. One is that quality degradation due to watermarking should be minimum, and second is that rewriting the watermark without quality degradation should be difficult. Our method [2] is based on well-established process of spreading watermark data energy across the picture area by use of a block transform. Security is provided by a "key", which is a seed to an appropriate pseudo random number generator. A number of other features are included to counter any attempt discover the watermarking details through statistical attacks.

5 Implementation

We have developed a practical image distribution system, 'InfoProtect' [3] using the method discussed above (Figure 2). In this system, users, who are registered with providers and get individual user IDs, purchase the descrambling key for their favorite images and pay by credit card or electronic coupon. The purchased images are embedded with the user IDs using watermarking technique when the images are descrambled at users' PC.

6 Summary

We described a new digital image distribution method that watermarks individual user information when descrambling an image at the user's end. In this method, images can be distributed through mass-distribution media and copyright is administrated after distribution. We implemented our method in the practical system.

7 References

[1] H. Fujii, N. Taniguchi, and Y. Yamanaka. "Scrambling Digital Images for Distribution through Network", Proc. of the PTC '96, Honolulu (1996), p. 447

[2] T. Nakamura, H. Ogawa and Y. Takashima. "A Watermarking Technique for Still Images", NTT R&D Vol.47, No.6 (1998), pp.711-714, (*In Japanese*)

[3] http://www.mmlab.ntt.ocn.ne.jp/

Watermarking Multiple Object Types in Three-Dimensional Models

Ryutarou Ohbuchi
IBM Tokyo Research Laboratory
1623-14 Shimotsuruma
Yamato-shi
Kanagawa, 242-8502, Japan
ohbuchi@acm.org

Hiroshi Masuda
The University of Tokyo
7-3-1 Hongo, Bunkyoku
Tokyo, 113-8656, Japan
masuda@nakl.t.u-tokyo.ac.jp

Masaki Aono
IBM Tokyo Research Laboratory
1623-14 Shimotsuruma
Yamato-shi
Kanagawa, 242-8502, Japan
aono@acm.org

ABSTRACT

Three-dimensional (3D) graphical model is about to become a full-fledged multimedia data type, prompted by increasing popularity of Virtual Reality Modeling Language (VRML) [7] and imminent standardization of MPEG4 [8].

Following an introduction on data embedding, this paper presents a discussion on potential targets of data embedding that exist in both VRML and MPEG4 formats. We then present several algorithms that embed data in shape (i.e., geometry and topology of the shapes) and shape attributes associated with shape (e.g., per-vertex texture coordinates).

KEYWORDS

Three-dimensional computer graphics, geometrical modeling, information security, digital watermark.

1 Introduction

The advantages of digital media, such as the Internet and CD-ROMs lies in the fact that the duplication, distribution, and modification of contents are much easier than the older media, such as printed media. For example, duplication of a digital content can be performed without any loss of its quality. These advantages, however, are double-edged swords. Digital media made unauthorized duplication, distribution, and modification of their valuable contents easier.

Data embedding, or *(digital) watermarking* put structures called *watermarks* into digital contents (e.g., images) in such a way that the structures do not interfere with intended use (e.g., viewing) of the contents. The watermarks carry information that can be used to manage the contents, for example, to add annotations, to detect tampering, or to authenticate rightful purchasers. While data can be embedded in an analog media, digital media provided an opportunity for a robust data embedding with significant data capacity.

In the past, a multimedia content typically meant a content that includes text, image, video, and audio data types. As a result, data embedding techniques for these "traditional" digital content data types has been studied by many [18, 19, 23, 1, 2, 11, 17, 3, 5, 10, 21, 22]. As 3D model gains status as an important member of multimedia data types, prompted by increasing popularity *of Virtual Reality Modeling Language* [6] and imminent standardization of *MPEG-4* [7], we added 3D polygonal model of geometry to the list of data embedding targets [12, 13, 14, 15].

In this paper, we will first introduce data embedding in general, followed by a discussion on embedding targets that exists in 3D models that follows VRML and MPEG4. We will then present three embedding algorithms, each of which is based on vertex coordinate modification, vertex topology modification, and texture coordinate modification, respectively.

1.1 Data Embedding Classifications

In this paper, following recommendation in [16], the act of adding watermark is called (data) *embedding or watermarking*, and retrieving the information encoded in the watermark for perusal is called *extraction*. The object in which the information is embedded is called *cover-<datatype>*, the object with watermark is called *stego-<datatype>*, and the information embedded is called *embedded-<datatype>*. The suffix "<datatype>" varies with data types, such as image, text, or 3D model. For example, an embedded-text is embedded in a cover-polygonal mesh to produce a stego-polygonal mesh with embedded-text.

A watermarks can be classified by its (1) *visibility* (or, more generally, *perceptibility*) and (2) *robustness*, as suggested by Mintzer, et al. [10]. A *visible watermark* is made intentionally visible to serve their purposes, for example, to deter a third party from unauthorized sales of contents. On the other hand, an *invisible watermark* is imperceptible without processing by mechanical means. A *robust watermark* should resists both intentional and unintentional modifications of the watermarked content. A *fragile watermark*, on the other hand, must be altered by intentional (and some unintentional) modifications so that it could detect tampering of or damage to the content. Here, *unintentional modifications* are the

kind a content should expect during a course of its intended use, while *intentional modifications* are the kind that are applied with an intention of destroying or altering the watermark**.**

A watermark can be classified further by its use of cover data for extraction. If an extraction algorithm requires original cover data as well as the (possibly corrupted) stego-data, the scheme is called *private watermarking*. Otherwise, the scheme is called *public watermarking*. An embedding scheme by Cox et al [3] is an example of private watermarking.

A watermarking scheme may employ a random sequence generator to make an embedded message secure from being read by a third party. For example, in an image watermarking, positions of pixels to be modified for watermarks can be scrambled by a pseudo-random sequence generated from a stego-key (or stego-keys) by using a public-key cryptographic method [9]. The scrambling can also be used to erase (reduce) statistical signature in order to make watermarking less detectable. Both public-key cryptography and shared-(private-)key cryptographic method can be used for this purpose.

Data embedding has many potential applications. Obviously, requirements for data embedding scheme vary depending on its intended application(s). Some of the potential applications are listed below.

- **Theft deterrence:** A robust, visible yet unobtrusive watermark in an image could deter unauthorized sales of the image by lowering commercial value of the image.
- **Copyright notification:** A copyright could be embedded as a robust invisible watermark into an image. Such notification could direct users of the model to the web site of the model's copyright owner.
- **Tamper detection:** Images taken by a digital still camera can be marked in the camera with a fragile invisible watermark so that modification made to the image afterward can be detected.
- **Content integrity check:** Since MPEG4 contents are editable, content creators might fear that a part of her/his creation is extracted and played without context, or a part of the content might is substituted. Watermarks in polygonal models and other 3D model contents could be used to detect such tampering.
- **Fingerprinting:** If an image is "fingerprinted" with the identities (e.g., digital signatures) of its purchaser and seller by using a robust watermarking technique, circulation of unauthorized copies of the image could be traced to the purchaser.
- **Play or duplication control:** Robust invisible watermark could control hardware devices to stop delivery of pornographic or violent digital-video contents (a la v-chip for broadcast TV in USA), or to prevent unauthorized duplication.

2 Embedding Target Objects In 3d Models

3D models in VRML [6] and MEPG4 [7] formats contain many types of objects. Among them, we consider objects in the following list to be important targets for data embedding. These objects are important since they have relatively large quantity of redundancy that can be exploited for data embedding.

1. Shape

- Polygonal Mesh Topology and Geometry
- Regular Mesh Geometry
- Elevation Grid Height Field Values

2. Shape attributes

- Vertex color (opacity), vertex texture coordinate, vertex normal vector, etc.
- Line color, etc.
- Face color (opacity), face normal vector, index of refraction, etc.
- Volume color (opacity), etc.

3. Animation parameters

- Interpolators
- Point/vertex coordinate and orientation..
- Colors and normal vector.
- Camera position and orientation.
- Face and Body Animation Parameters
- Parameterized position of eyes, tongue, etc.
- Angle of joints, etc.
- Animated Mesh
- Vertex coordinates displacements.

4. Others

- 2D still texture image, movie texture.
- Sampled sound.
- Text string, text position and orientation, text color, etc.

Text-to-speech phoneme strings and synthetic sound symbol sequences (e.g., a MIDI command sequence) contain little redundancy to be used as good embedding targets.

2.1.1 Shapes

Shapes, or *geometrical components*, of 3D objects are arguably the most important class of target, for without shape, 3D model means little. While point set and poly-lines are viable candidate for data embedding targets, *polygonal mesh* is probably the most important target for data embedding in 3D models.

Two components, *vertex coordinates* and *vertex topology*, define shape of a 3D polygonal mesh. Vertex coordinate combined with vertex topology defines more complex *geometrical primitives*, that are, lines, polygons, and polyhedrons. These geometrical primitives have their own quantities such as length of a line segment and volume of a polyhedron that are called *geometrical quantities* in this paper. The geometrical primitives have topology of their own, which are, for example, connectivity of triangles and tetrahedrons.

Data can be embedded in a 3D shape by modifying either geometry or topology of its geometrical primitives. A unit of such modification is called *embedding primitive*. It is also important to arrange these embedding primitives in an order so that the arranged set of embedding primitive as a whole carry a significant amount of information. Arrangement can be created either by topology or by quantity of geometrical primitives.
Details of fundamental methods to embed data into shapes can be found in papers [14] and [15]. Section 4 of this paper presents two data embedding algorithms, one that targets geometry and the other targets topology. (These algorithms have previously appeared in [14] and [15], but included in here for completeness.)

2.1.2 Shape Attributes

Shape-*attributes*, such as vertex color, per-vertex texture coordinates, per-face color and per-volume refractive index, are essentially sets of numerical values that can be modified to embed data. While less important than a shape itself, a shape attribute still is an important class of target for embedding.
The approach to embedding using shape attributes is similar to those used for algorithms that employ coordinate embedding primitives; Values of the attributes are modified and the modifications are ordered to embed a significant amount of information.
Details of a data embedding algorithm that targets texture coordinate will be presented in Section 4. (This algorithm has previously appeared in [15].)

2.1.3 Animation Parameters

Animation parameters have potentials to become very important targets for data embedding. For example, if polygon-based counterparts of music videos are made, moves of a popular musician captured to animate his/her figure could carry a very high value.
VRML provides various *interpolators* for animation. An interpolator is a sequence of multiple sets of values that are linearly interpolated to produce continuously varying values. These varying values can be used to translate, rotate, or deform objects.
Figure 1 shows an example of VRML interpolator data generated by a 3D modeling and animation software. It is a plot of trajectory of the torso of a skateborder model as it performs a manuever called "540". In the torso alone, this animation sequence contained 53 coodinate points, each of which is a 3D coordinate. Combined with the other parts, such as head, upper-arm, lower-arm, camera, etc., there are siginificant amount of data that can be exploited for embedding in the animated 3D model.
The MPEG4 proposal [7] contains other types of data objects for animation. It contains animated (deforming) regular mesh whose vertices move over time given a continuously transmitted list of incremental displacements. The displacements may be coded in two ways, either by using simple difference or by using discrete cosine transformation of a short sequence of displacement values.
The MPEG4 proposal also contains human face and body models that can be animated by transmitted animation parameters. For example, a facial model is controlled by a set of about sixty integer values, each of which specify location of eyebrows, eyes, a tounge, ears, etc. Most of these parameters use small

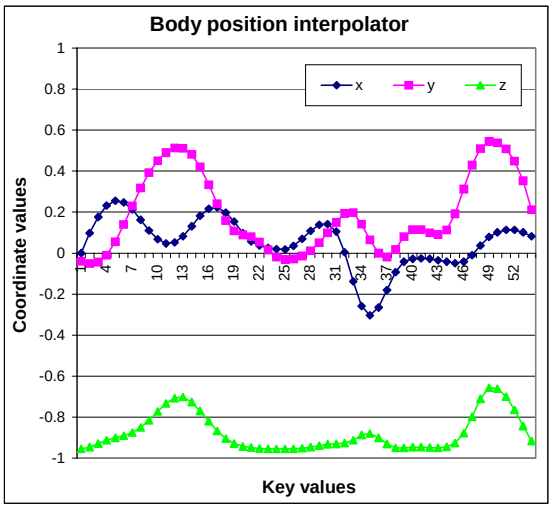


Figure 1. Plot of key values of a VRML coordinate interpolator in an animation sequence.

quantization levels (from 2 to 5), except for a few. The body model is controlled by over 170 parameters, each of which has quantization levels of 256. While face animation parameters with its small quantization levels may lack redunduncy for data embedding, body animation parameters will have enough redunduncy to be exploited for data embedding.
It must be noted, however, that the MPEG4 proposal contains extensive list of both lossless and lossy compression algorithms for shapes, shape attributes, and for animation parameters. Data compression algorithms in general try to find and remove redundancies that are necessary for data embedding algorithms. Embedding algorithms must take these data compression algorithms into account.

2.1.4 Others

Sampled sound, 2D still texture and 2D movie texture are obvious targets of embedding by using data embedding algorithms developed previously for respective data types. However, care must be taken in using these objects for embedding since these objects in a 3D model can be removed effortlessly.

3 Embedding Algorithms For 3D Polygonal Meshes

In this section, we will present algorithms that target shape and a shape attribute of polygonal meshes for data embedding. All the algorithms in this section are implemented by using a kernel for a non-manifold modeler [9]. The system employs *radial edge* struc-

ture [20] to represent the topological relationship among vertices, edges, faces, and regions.

3.1 An Algorithm Based On Geometrical Quantity Modification

A pair of dimensionless quantities, for example, $\{e_{14}/e_{24}, h_4/e_{12}\}$ in Figure 2, defines a set of similar triangles. The algorithm described in this section, Triangle Similarity Quadruple (TSQ) algorithm, uses such dimensionless quantity pair as the geometrical embedding primitive to watermark triangular meshes.

The TSQ algorithm can be classified as a public watermarking scheme. Watermarks produced by the TSQ algorithm withstand translation, rotation, and uniform-scaling transformations of the stego-polygonal-meshes. An embedded message is resistant to resection and local deformation if it is repeatedly embedded over a mesh. The watermarks are destroyed, among other disturbances, by a randomization of coordinates, by a more general class of geometrical transformation, or by a topological modification such as re-meshing.

In order to realize subscript ordering, the algorithm uses a quadruple of adjacent triangles in the configuration depicted in Figure 2 as a Macro-Embedding-Primitive (MEP). Each MEP stores a quadruple of symbols *{Marker, Subscript, Data1, Data2}*. In Figure 2, the triangle marked *M* stores a marker, *S* stores a subscript, and *D1* and *D2* stores data values. A marker is a pair of values that identifies MEPs. As mentioned above, this public watermarking scheme does not require cover-polygonal-mesh for extraction. However, the marker value pair is necessary for extraction. A watermarked mesh would contain multiple MEPs to embed a significant amount of data as shown in the example of Figure 3. While each MEP is formed by topology, a set of multiple MEPs is arranged by quantity of the subscript.

The TSQ algorithm embeds a message according to the following steps. (For the detailed explanation and execution examples, please refer [14].)

(1) Traverse the input triangular mesh to find a set of four triangles to be used as a MEP. MEPs must not share edges or vertices to avoid interference.
(2) Embed the marker value by changing a dimensionless quantity pair in the center triangle of the MEP. In Figure 2, it is $\{e_{14}/e_{24}, h_4/e_{12}\}$. This modifies positions of vertices v_1, v_2, and v_4.
(3) Embed a subscript and two data symbols in a similar manner by displacing vertices v_0, v_3, and v_5. Subscript is embedded in the pair $\{e_{02}/e_{01}, h_0/e_{12}\}$, and two data symbols are embedded in the pairs $\{e_{13}/e_{34}, h_3/e_{14}\}$ and $\{e_{45}/e_{25}, h_5/e_{24}\}$.
(4) Repeat (1) to (3) above until all the data symbols of the message are embedded.

For each triangle, the algorithm first modifies the ratio h_i/e_{ij} by changing h_i only. Then the algorithm modifies the ratio e_{ij}/e_{kl} while keeping the height h_i constant. In order to embed the message repetitively, steps (1) to (4) are repeated many times.

Figure 4 shows triangles that formed MEPs in darker gray. Due to the mutual exclusion rule described in the step (1) above, MEPs do not share vertices.

Given a watermarked mesh and two numbers that identify marker triangles, extraction proceeds according to the following steps.

(1) Traverse a given triangular mesh and find a triangle with the marker, thereby locating a MEP.
(2) Extract a subscript and two data symbols from the triangles in the MEP.
(3) Repeat (1) to (2) above for all the marker triangles on a given triangular mesh.
(4) Sort the extracted symbols according to their subscripts.

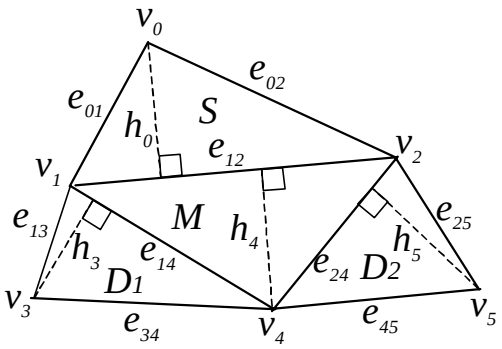


Figure 2. A macro-embedding-primitive. In the figure, v_i are vertices, e_{ij} are lengths of the edges, and h_i are heights of the triangles.

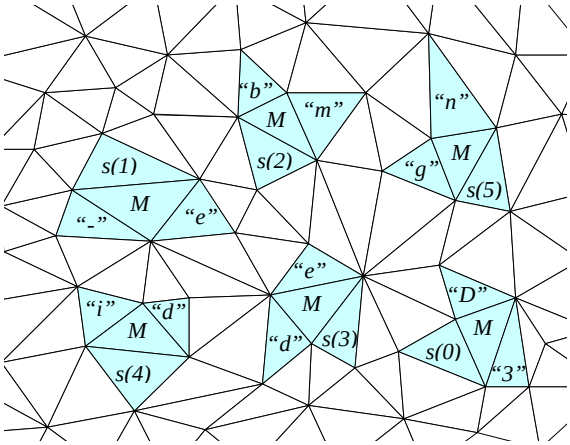


Figure 3. In this example of TSQ watermarking, six macro embedding primitives on a mesh embed a string "3D-embedding". Subscripts (denoted *s(i)* for a subscript *i*) arranges the MEPs.

The TSQ algorithm embedded 210 bytes of data, that is, 0.15 byte/triangle, in the model of Figure 4, which consisted of 1406 triangles. Experiments using seven

polygonal mesh models showed that the TSQ algorithm was able to embed 0.15-0.18 byte/triangle.

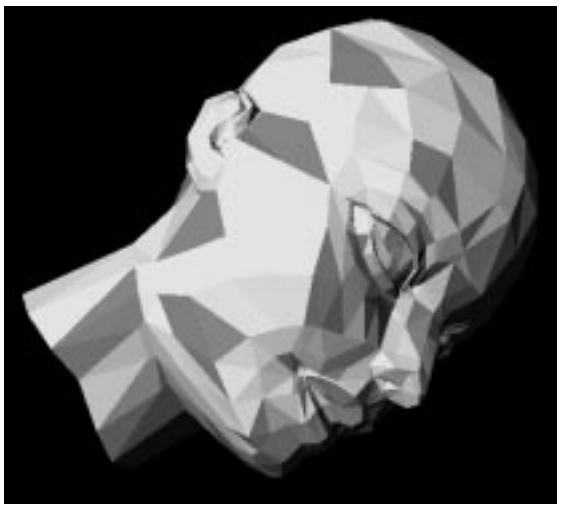


Figure 4. Macro embedding primitives, each of which consists of four adjacent triangles, are shown in dark

3.2 An Algorithm Based On Topological Modification

The *Triangle Strip Peeling Symbol sequence (TSPS)* embedding algorithm that will be presented in this section is a public watermarking scheme based on a topological embedding primitive. It employs, as its embedding primitive, an adjacency of a pair of triangles in a triangle strip, each of which encodes a binary bit of information. One-dimensional arrangement of embedding primitives is induced by the adjacency of triangles on the triangle strip. To recognize the triangle strip with watermark, the strip is peeled off from the original mesh.

Since both embedding primitive and arrangement are topological, watermarks produced by the algorithm are immune to geometrical transformation. Repetitive embedding makes the watermarks resistant to resection. The watermarks can be destroyed by topological manipulations, for example, by polygon simplification algorithms. A disadvantage of this algorithm is its low space efficiency compared to many algorithms based on geometrical primitives.

Inputs to this embedding algorithm are an orientable triangular mesh and a message bit string. The TSPS embedding algorithm embeds data according to the following steps. (See Figure 5.)

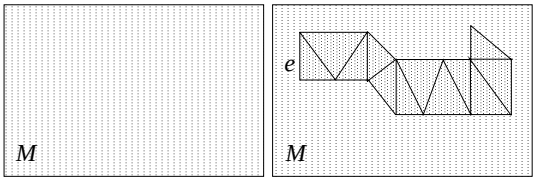
(1) Starting from an edge *e* selected from the input mesh *M*, grow a triangle strip *S* on *M* by using the message bit-string to determine the direction of growth of the strip. Observe that a triangle at the end of (current) strip has two "free" edges, i.e., edges that are not adjacent to triangles of the current triangle strip. Since *M* is orientable, these two edges can be ordered on the triangle by traversing the edges in a fixed order (either counterclockwise or clockwise). Depending on the data bit, choose one of the two free edges as the edge to be shared with the next triangle of the strip. (See Figure 6.)

(2) "Peel off" the triangle strip *S* from *M* by splitting all the edges and vertices on the boundary of *S* except the initial edge *e*. The strip *S* is connected to the rest of the mesh only by the edge *e*.

The edge *e* serves as the initial condition for finding the triangle strip. Arrangement of embedding primitives is induced naturally by the connectivity of triangles on the triangle strip. Since the peeled strip caps the hole completely, proper colors and vertex normal vectors make the watermark invisible.

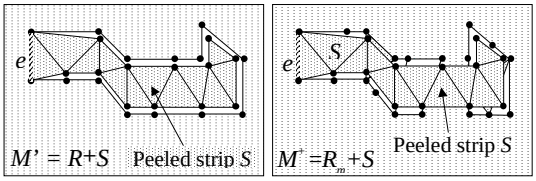
Figure 6 shows an example of a triangle strip. The strip drawn with solid lines, which start at edge *e*, embeds a bit string "10101101011" in a sequence of 12 triangles. Each bit of the bit string steers the direction of growth of the triangle strip. If the last bit of the string is "0" instead of "1", the last triangle will become the one that is drawn with broken lines.

Steering by message bit strings produces strips whose shape may not fit in a given mesh, depending on a given bit string. In the example of Figure 6, a message bit string with all "1" would keep steering the strip to the left. If the message string is sufficiently long, the strip will either hit the boundary of the mesh or circle back to itself. To avoid this problem, shapes, locations and orientations of the strips must be controlled carefully. We manipulate the shape of the triangle by using *steering symbols*. A steering symbol is a bit that does not carry information but simply steer direction of growth of a triangle strip. Steering symbols are interleaved with data symbols, that are, symbols that encode embedded data, in order to control shape of triangle strips. Obviously, steering symbol halves the embedding data capacity. Our current implementation determines initial locations, directions of growth, and shapes of triangle strip manually.



(a) Original mesh.

(b) Generate triangle strip *S* based on the message bit string.



(c) Cut out the strip *S* from the mesh *M*, except at the edge *e*.

(d) Scramble mesh topology of the stenciled mesh (*M*-*S*) by adding vertices, edges, and polygons, if necessary.

Figure 5. Triangle strip *S* encoding a message bit string is peeled off from the cover-polygonal mesh *M*. (The cracks around *S* in the figure is for illustration purpose only.)

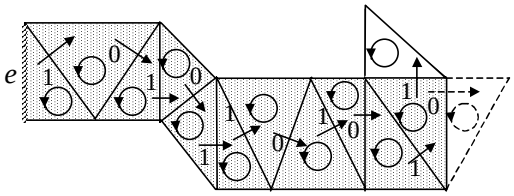


Figure 6. Connectivity of 12 triangles (drawn with solid lines) in a triangle strip encodes the bit string "10101101011" (11 bits). If the bit string is "10101101010" (change in the last bit), the last triangle will be the one drawn with broken lines.

Extraction of a message is carried out according to the following steps.

Traverse the watermarked mesh and find an edge with topological features that starts a triangle strip of known length that is attached to the stencil mesh by an edge.

Starting from the initial edge, traverse the triangle strip to the open end as embedded bits are extracted.

Figure 7 shows a simple example of TSPS embedding, in which a triangle strip of length 27 is peeled off from a mesh that consisted of 214 triangles. The triangle strip encodes 13 data bits and 13 steering bits. Selection of steering bits in this case was done manually. As another example, a model of triceratops (499 triangles) in Figure 8 is marked with a triangle strip of length 19 triangles, which encodes 9 bits. (The colors of the strips in these examples are intentionally changed to show their location.)

Watermarks produced by the TSPS embedding algorithm can be erased if a geometrical "mending" program, for example the one similar to [4], which would stitch the triangle strip back to the stencil mesh. Such mending can be prevented to some extent by modifying topology of stego-polygonal-mesh in order to confuse mending algorithms. For example, vertices, edges, and polygons can be added into the stencil mesh *R* so that finding correspondence of edges and vertices to be stitched together is difficult (Figure 5(d)).

3.3 An Algorithm Based On Shape Attribute Modification

An algorithm explained below embeds data in texture coordinates of polygonal mesh. A similar algorithm can be used to embed data in other per-vertex attributes, such as vertex colors. Data embedding into per-face attributes of a polygonal mesh surface is also possible; Modify per-face attributes and then arrange these modified attributes.

A set of texture coordinates associated with vertices of a polygonal model is a good target for data embedding. This is because a set of roper texture coordinate is crucial to properly render texture mapped objects, and a set of texture coordinates is difficult to regenerate once it is lost.

The algorithm we experimented modulates amplitude of texture coordinates based on message bit string.

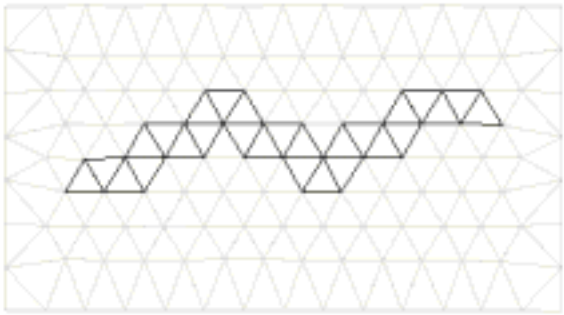


Figure 7. A triangle strip consisting of 27 triangles was cut out from a flat triangular mesh (214 triangles). The triangle strip, displayed in darker gray, encodes 13 data bits interleaved with 13 steering bits.

Let s_i be *i*th bit of a bit string *S*. The embedding algorithm modifies a coordinate value x_i (e.g., either *u* or *v*) of a texture coordinates by the following steps, given a modulation amplitude *A*.

r=x_i-x_i/A;

if s_i='0' **then** *b*=*A*/4 **else if** s_i='1' **then** *b*=*A**3/4;

x_i= r+b;

This is just an example of modulation method. Many other alternatives, including multi-valued modulations, are possible. Whatever the modulation method, we can make two such modulations per 2D-texture coordinates. Thus, if we embed one bit per floating point number, we can embed 2*N* bits into *N* 2D-texture coordinates.

In modifying the texture coordinate, the modulation amplitude A must be chosen so that the watermark is robust enough without degrading quality of texture mapped objects. We conducted experiments to see how amplitude affect appearances of texture-mapped 3D polygonal mesh objects. Some of the results are shown in Figure 9 and Figure 10.

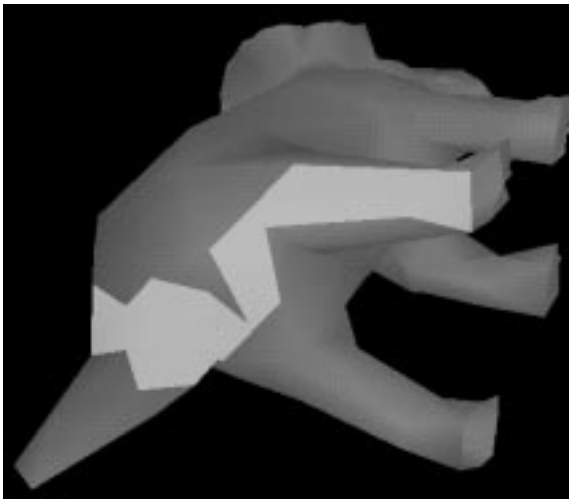


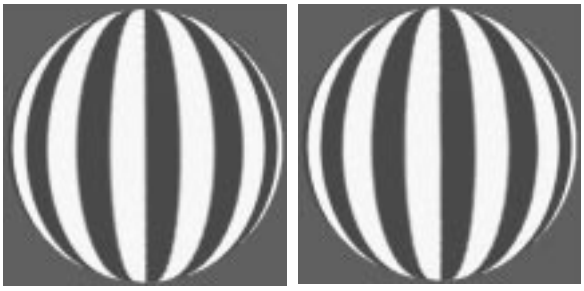
Figure 8. A triangle strip, 19 triangles long and shown in a light gray, is generated and peeled off from a model of a triceratops (499 triangles). (A part of the strip is not visible from this viewpoint.)

Texture images are a synthetic red-and-white stripe image (256 x 256 pixels) and a photograph of a human face with a tree leaves in background (300 x 300 pixels image area, 1024 x 1024 pixels overall). These images were texture-mapped onto a model of a sphere tessellated into 1800 triangles, which contained 961 vertices (and thus 961 texture coordinate). We can embed a maximum of 961 bytes in the sphere if we modify four bits per single-precision floating-point number. In this experiment, we embedded a 358 byte long text.

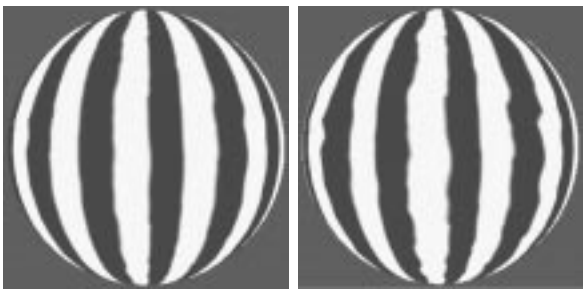
In the figures, *Ar* is the modulation amplitude relative to the range of texture coordinate variation on the model. In these examples, the texture coordinates varied in the range [0,1] in both *u* and *v* coordinates so that the maximum variation range of texture was 1.0. In another word *Ar*=0.1 % means amplitude of 0.001.

In Figure 9, in which the red-and-white stripe texture is used, distortion in the rendered image is perceptible in rendered images when *Ar*=0.5 % (Figure 9c) and *Ar*=1 % (Figure 9d). Complex, less geometrical, texture images reduced perceptibility of texture distortions. Distortions of the human face texture shown in Figure 10 were difficult to perceive. Even for the image with *Ar*=1 % (Figure 10c), a careful comparison with the original image (Figure 10a) was necessary to reveal distortions.

In our prototype implementation, we used the order of appearance of texture coordinates in the input file as the arrangement for embedding. This arrangement



(a) *Ar=0* (No embedding) (b) *Ar*=0.1 %



(c) *Ar*=0.5 % (d) *Ar*=1 %

Figure 9. A red-and-white stripe image is mapped onto a sphere model (1800 triangles). Texture coordinates are modulated with relative amplitudes *Ar*=0 % to Ar=1 %.

(a) *Ar=0*. (No embedding) (b) *Ar*=0.1 %

(c) *Ar*=0.5 % (d) *Ar*=1 %

Figure 10. A photograph of a human face is mapped onto a sphere model (1800 triangles). Texture coordinates are modulated with several relative amplitudes *Ar*=0.0% to 1%.

is destroyed easily by shuffling the positions of the texture coordinates in the file. If this is a problem, there are alternative methods to introduce ordering into a set of texture coordinates. Since each texture coordinates is associated with a vertex, ordering vertices implies ordering of texture coordinates. Several examples of methods to order vertices are described in [14]. It is also possible to arrange texture coordinate by using a non-geometrical quantity itself. In the example of texture coordinate, texture coordinate or quantity derived from it can be used to order vertices. Note that watermark that modifies geometry and/or topology of a polygonal mesh do not interfere directly with non-geometrical attributes. It is possible to combine an attribute-modifying algorithm (e.g., the one described in this section that modifies texture coordinate) with an algorithm that modifies geometry or topology (e.g., the triangle strip peeling algorithm).

This experiment showed that, if modification amplitude is chosen appropriately, data embedding into texture coordinates is possible without noticeable change in the models rendered appearance.

4 Summary And Future Work

In this paper, we first presented introduction to data embedding technology. It is followed by a discussion on possible data embedding targets that exist in 3D models, that are, shape (both topological and geometrical components of shape), shape-attributes (e.g., texture coordinates and vertex color), and others, such as mesh animation parameters and face/body animation parameters. As examples, we presented three algorithms. Two of the algorithms embed data in shape, using both geometry and topology of 3D polygonal meshes. The other algorithm embeds data in a shape-attribute, that is, texture coordinates, of 3D polygonal mesh models.

In the future, we would like to experiment with algorithms that embed data in animation parameters that exists in MPEG4 and VRML formats. We need to evaluate effects of data compression algorithms used in these formats to compress shape, shape attributes, and animation parameters. We also would like to develop and test realistic scenarios employing data embedding algorithms for 3D models.

5 REFERENCES

[1] W. Bender, D. Gruhl, and N. Morimoto, Techniques for Data Embedding, *IBM Systems Journal*, Vol. 35, Nos. 3 & 4, 1996.

[2] G. Braudway, K. Magerlein, and F. Mintzer, Protecting Publicly-Available Images with a Visible Image Watermark, *IBM Research Report*, TC-20336 (89918), January 15, 1996.

[3] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, Secure Spread Spectrum Watermarking for Multimedia, *IEEE Trans. on Image Processing*, Vol. 6, No. 12, pp1673-1678, 1997.

[4] A. Gueziec, G. Taubin, F. Lazarus, and W. Horn, Cutting and Stitching: Efficient Conversion of a Non-Manifold Polygonal Surface to a Manifold, *IBM Research Report* RC-20935 (92693), July, 1997.

[5] F. Hartung and B. Girod, Copyright Protection in Video Delivery Networks by Watermarking of Pre-Compressed Video, *Lecture Notes in Computer Science*, Vol. 1242, pp.423-436, Springer, 1997.

[6] ISO/IEC 14772-1 Virtual Reality Model Language (VRML).

[7] ISO/IEC JTC1/SC29/WG11 *MPEG-4 Visual* and *MPEG 4 SNHC*.

[8] H. Masuda, Topological Operations for Non-Manifold Geometric Modeling and Their Applications, *Ph. D dissertation*, Department of Precision Machinery Engineering, University of Tokyo, 1996 (in Japanese).

[9] A. J. Menezes, P. C. van Oorshot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[10] F. Mintzer, G. W. Braudway, and M. M. Yeung, Effective and Ineffective Digital Watermarks, Proceedings of the *IEEE International Conference on Image Processing (ICIP) '97*, Vol. 3, pp. 9-12, 1997.

[11] J. J. K. O'Ruanaidh, W. J. Dowling and F. M. Boland, Watermarking Digital Images for Copyright Protection, *IEE Proc.-Vis. Image Signal Process.*, Vol. 143, No. 4, pp. 250-256, August 1996.

[12] R. Ohbuchi, H. Masuda, and M. Aono, Embedding Data in 3D Models, in Steinmetz, et al. eds, *Lecture Notes in Computer Science* No. 1309, pp.1-11 (Proceedings of the *IDMS '97*, Darmstadt, Germany, September) 1997.

[13] R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models, *Proceedings of the ACM Multimedia '97*, Seattle, Washington, USA, November 1997, pp. 261-272.

[14] R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications, pp. 551-560, *IEEE Journal on Selected Areas in Communications*, May 1998.

[15] R. Ohbuchi, H. Masuda, and M. Aono, Geometrical and Non-Geometrical Targets for Data Embedding in Three-Dimensional Polygonal Models, to appear in August 1998 issue of the *Computer Communications*, Elsevier.

[16] B. Pfitzmann, Information Hiding Terminology, in R. Anderson, Ed., *Lecture Notes in Computer*

Science No.1174, pp. 347-350, Springer-Verlag, 1996.

[17] J. R. Smith and B. O. Comiskey, Modulation and Information Hiding in Images, in R. Anderson, Ed., *Lecture Notes in Computer Science* No.1174, pp. 207-296, Springer, 1996.

[18] K. Tanaka, Y. Nakamura, and K. Matsui, Embedding Secret Information into a Dithered Multilevel Image, *Proc. 1990 IEEE Military Communications Conference*, pp. 216-220, 1990.

[19] S. Walton, Image Authentication for a Slippery New Age, *Dr. Dobb's Journal*, pp. 18-26, April 1995.

[20] K. Weiler, The Radial Edge Structure: A Topological Representation for Non-Manifold Geometric Boundary Modeling, *Geometric Modeling for CAD Applications*, North Holland, pp. 3-36, May 1986.

[21] M. M. Yeung, F. C. Mintzer, G. Braudway, and A. R. Rao, Digital Watermarking For High-Quality Imaging, Proceedings of the *First IEEE Workshop on Multimedia Signal Processing*, Princeton, NJ, USA, June, 1997, pp. 357-362.

[22] M. M. Yeung and F. Mintzer, An Invisible Watermarking Techniques for Image Verification, Proceedings of the *IEEE ICIP '97*, Vol. 2, pp. 680-683, 1997.

[23] J. Zhao and E. Koch, Embedding Robust Labels into Images for Copyright Protection, Proc. of *the Int'l. Congress on Intellectual Property Rights for Specialized Information, Knowledge, and New Technologies*, Vienna, August 1995.

Non-Invertible Watermarking Methods for MPEG Video and Audio*

Klara Nahrstedt
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
217-244-6624

klara@cs.uiuc.edu

Lintian Qiao
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
217-244-6624

l-qiao@cs.uiuc.edu

ABSTRACT

Various digital watermarking techniques have been pro-posed in recent years as the methods to protect the copyright of multimedia data. However, the rightful ownership problem has not been properly solved as it is a non-trivial task to construct a non-invertible watermarking process. In this paper we give a brief overview of our watermarking solutions which were proved to be non-invertible in [4] and are successful in resolving rightful ownership of watermarked MPEG video and audio.

We will discuss various issues of the watermark construction process, and watermark embedding schemes applied to MPEG video and audio.

KEYWORDS

Watermarking, Non-invertibility, Ownership, Copyright, MPEG Video and Audio

1 Introduction

With the growth of multimedia systems in distributed environments, the research of multimedia security as well as multimedia copyright protection becomes an important issue. Digital watermarking techniques have been proposed in recent years as the methods to protect the copyright of multimedia data. There are various watermarking schemes applied to images and several methods applied to audio and video streams. Among them, a large class of watermarking schemes addresses invisible watermarks. Craver et al [2] also pointed out the problems of non-invertibility[1] for watermarking schemes and how to resolve the rightful ownership of the invisible watermarking schemes. Craver et al attacked existing watermarking techniques by providing counterfeit watermarking schemes that can be performed on a watermarked image to allow multiple claims of ownerships. We refer to their attack as the CMYY (Craver-Memon- Yeo-Yeung) attack. The rightful ownership problem is either not addressed at all or not addressed properly within current existing watermarking techniques.

In this paper we will briefly outline our solutions towards watermarking schemes applied to MPEG Video and Audio which have the properties of being invisible and non-invertible. We introduce requirements on the watermark construction so that the whole watermark has to be created by using a standard encryption function, e.g. DES. Because of these requirements, the non-invertibility of the proposed scheme is easily proved as shown in [4]. Furthermore, we brie y discuss the watermarking methods for MPEG video and audio with the main goal to make the watermark invisible.

The paper is outlined as follows: Section 2 presents the rightful ownership and non-invertibility problem, Section 3 and 4 discuss the non-invertible schemes for MPEG video and audio and Section 5 concludes the paper.

2 Rightful Ownership and Non-invertibility Problem

The purpose of a watermark is to protect the owner's copyright. But without a careful scheme design and proper requirements on the watermark, an attacker can easily confuse everyone by manipulating the watermarked video (image, audio) and claim that he/she also is the original owner. This is called the "rightful ownership" problem.

Craver et al provided the following scenario: given a watermarked video (image), it is possible for an attacker to watermark the watermarked video (im-

[1] Informally, non-invertibility means that it is computationally impossible for an attacker to find a pair of a faked image and a watermark such that the pair can result in the same watermarked image created by the real owner.

* This research was supported by National Science Foundation Career Grant CCR-96-23967 and Research Board of University of Illinois at Urbana-Champaign.

age) again using any watermarking scheme. This twice watermarked video (image) has both original and attacker's watermarks on it. Both the original owner and the attacker can claim the ownership, therefore, defeat the purpose of using watermark.

Using the original video clip (image) in the verification process can prevent the multiple ownership problems in some cases. However, even with the presence of the original video clip (image), the rightful ownership problem still exists. Craver et al showed that the following scenario is possible:

Assume that the original video clip (image) is V. The owner of the video clip uses watermark W to create a watermarked video clip (image) V_W, and publishes V_W. We denote this process as

$V \oplus W => V_W$ (1)

The attacker creates a watermark WF , without knowing V, extracts WF from VW and creates a counterfeit video clip (image) VF . Notice that,

$V_F \oplus W_F => V_W$ (2)

In this way, the attacker can use V_F as his \original" and claim the ownership of V_W.

If (2) can be achieved, then that scheme is called invertible watermarking. Otherwise, it is called noninvertible watermarking.

There is a class of invisible watermarking schemes which do not use original image (video clips) in the verification process. We will refer to these schemes as Self-Proof Class. Because the use of the Self-Proof Class is simpler than that of other watermarking schemes, it is interesting to study its non-invertibility. We want to know „Are there any non-invertible schemes which belong to the Self-Proof Class?". Unfortunately, the answer is NO as shown in [4].

Of course, one can add the requirement of watermark construction to the Self-Proof Class schemes. But in order to verify the watermark construction, something other than watermarked image itself has to be presented. This contradicts to the concept and the definition of Self-Proof Class scheme.

Another attempt to resolve the ownership problem is provided by Wolfgang and Delp [5]. Their method uses timestamp to generate the watermark. Then the owner with the earliest timestamp is the true owner. However, this scheme can be easily defeated because timestamps can be manipulated. For example, for a certain event such as "the Berlin Wall came down", everybody knows exactly when it happened. If an attacker uses that time as his/her timestamp in watermarking the video/image taken during that event, then who is the original owner? The following example may be more realistic: in some videos, the real time clocks may be displayed somewhere in the picture, such as news broadcasting, basketball games, etc. The timestamps can be accurately decided in these cases. Again, if an attacker uses these timestamps, then who is the original owner? Based on the above discussion, we can conclude that the use of timestamps is not a good way to solve the watermark ownership problem.

3 Non-invertible Scheme for MPEG Video

One possible approach against CMYY attack is to make a strict requirement on the construction of the watermark and bind the watermark with the original video (image) itself. This will greatly limit the choices of the watermark W_F and the falsified „original" V_F for an attacker. Clearly, if it is computationally impossible for an attacker to find both W_F and V_F which satisfy formula (2), then the non-invertibility is achieved. In this section, we will derive such a scheme which applies to the MPEG-encoded video streams. Of course, the derived schemes must not belong to the Self-Proof Class.

We will refer to V as a single I frame within the I, P, B sequence of a MPEG stream.

3.1 Watermark Construction

Here we only describe the watermark construction on a single image V which is encoded in JPEG format. It can be easily extended to the I frames of MPEG video clips. The method which we will use is based on the ideas from direct sequence spread spectrum communications.

First, we choose a standard encryption function, such as DES, and a key KEY . In Zig-Zag order, we scan each block of the DCT-transformed image.

Let $AC_{b,l}$ denote the value of the l-th AC coeficient in the block b, where b = 1 ... nb, and nb is the number of blocks in V , l = 1...63. Let

$NZ_{b;l} = 1$ if $AC_{b;l}$ $6 \neq 0$; 0 otherwise. (3)

Let nAC_l; l = 1:::63 denote the total number of nonzero AC coeficients at the l-th position of each block, i.e.,

$nAC_l = \sum_{1 \leq b \leq nb} NZ_{b,l}$ (4)

For example, if $AC_{1;5} = 1$, $AC_{2,5} = -2$, $AC_{1001,5} = 4$, $AC_{b,5} = 0$ in all other blocks, then $nAC_5 = 3$.

Second, we transform first m ($1 \leq m \leq\_ nb$) nAC_l into binary numbers and concatenate those binary numbers to form a PAD, i.e., $PAD = nAC_1...nAC_m$. The choice of m will depend on the number of bits which can be embedded into the frame V . For example, for a standard MPEG-1 video with picture size 352x288 (pixels), there are totally 1584 blocks. Therefore, 11 bits are required to represent each nAC_l because $2^{10} = 1024 < 1584 < 2048 = 2^{11}$. If we are allowed to embed totally 101 bits, m will be [101/11] = 10.

Third, we apply DES with KEY to PAD and get $EPAD = DES_{KEY}$ (PAD).

Forth, let us consider definitions of the following bit sequences:

- Let Aj be a bit sequence, where j=1 ... total number of bits allowed to embedded in V, such that

  $A_j = -1$ if $EPAD_j = 0$; 1 otherwise (5)

- Let B_i be a bit sequence, such that
  $B_i = A_j, \quad (j \times C_r) \le i < ((j + 1) \times C_r)$ (6)
  where C_r is the chip-rate[2].
- Let p_i be a bit sequence, such that
  p_i= -1 if i-th bit of $DES_{KEY}(V)$ is 0; 1 otherwise (7)

Last, we construct the watermark bit sequence by combining the sequences B_i and p_i:

$w_i = \alpha \times B_i \times p_i$; i = 1...(Width × Height) (8)

We will discuss how we select the scaling number α in the next subsection.

3.2 Watermark Embedding Procedure

There are several assumptions which need to be mentioned before we describe our scheme.

1. We will derive a watermarking scheme for MPEGcompressed video without fully decoding the video to raw image sequence and then encoding again. We decode the video up to the DCT coeficients and then embed a watermark into the stream at the DC and AC coeficient level.
2. We set α = 1. Tests on various MPEG clips show that, with chip-rate 1000, if we watermark the DC coefficients, then the watermarked video has lower quality and the watermark is visible. One way to solve this problem is to increase the chip-rate, for example, to 5000, but this will decrease the amount of information that can be embedded by a factor of 5. Therefore, this is not a good solution. We choose not to mark any DC coefficients. (Setting α to a small number can also avoid the quality degradation, but this means that there are many choices for α in one scheme and different α can be chosen in different schemes. This uncertainty gives an attacker freedom to manipulate the verification process. Therefore, we fix α = 1.)
3. We only watermark I frames because I frames are the most significant frames in a video stream. The watermark of I frames will be carried over to P and B frames due to the dependency between I, P, and B frames. It is true that an attacker could drop all watermarked I frames to get an unwatermarked video stream. However, although P and B frames may contain some I-blocks which can be de-coded by themselves, there is enough dependency between the I frame and the P and B frames, therefore, the quality of the P or B frames will be bad enough if the I frames are dropped. Furthermore, we only watermark the luminance blocks because they are more significant than the chrominance blocks.
4. We want to achieve that the total length of a watermarked stream is less than or equal to the total length of the original stream, so that the watermarking process does not defeat the compressing process of MPEG.
5. In our scheme, the original video stream and a key are needed for both the watermark creation and the verification process.

With the considerations of the above assumptions and requirements, we design an algorithm as follows:

1. 1. A new watermark is created based on the encrypting information from the original I; We denote the process as $w_i = DES_{KEY}(v_i)$. We then put the sequence w_i into a two dimension matrix of (Height × Width). The result is a watermark in the form of a raw image.
2. DCT is applied to the watermark;
3. An AC coefficient in the original image V is marked if the length of the resulting VLC (Variable Length Code) does not increase.
4. An AC coefficient is marked and the last non-zero AC coefficient (in the zig-zag order) of a block in the original image V is dropped if the total length of the resulting VLC does not increase. No more than two AC's can be dropped within a block.[3]

We now discuss the results and the implications of the proposed scheme.

1. Formula (nAC_l in V_W) < (nAC_l in V) always holds. This is because, for each block, there are three possible outcomes: the number of non-zero AC's decreases by 0 (unchanged), 1, or 2 in comparison with the original video frame V . The outcome depends only on the original V and the key KEY . It is easy to see, the total number of nonzero AC coefficients in V_W decreased. In addition, the values of some non-zero AC coeficients in V_W could also be changed and the changes also depend only on V and KEY.
2. The randomness of the embedded information is guaranteed by the encryption function DES. p_i in the spread spectrum scheme is the Pseudo Random Noise Code. By applying DES to V, we can create a Pseudo Random sequence of values 1/-1 which can be used as p_i. By choosing different KEY, a different p_i sequence is created. The KEY is safe due to the DES encryption algorithm which means that it is computationally impossible for an attacker to find out the key even with the knowledge of V.
3. The information contained in PAD is essentially the number of first m non-zero AC coefficients in the frame V. This information is un-

[2] Chiprate: In spread spectrum systems, this is the rate at which the discrete signal is applied.

[3] Two AC's can be dropped only when one becomes 0 and another, which must be the last non-zero AC coefficient, is dropped.

known without the knowledge of the original V.

3.3 Verification Process

The verification process justifies the ownership claim and consists of three steps:

1. The claimant is required to provide his/her original video clip V, the key KEY, and his/her watermark W. Then the trusted third party verifies the creation of the watermark. If the watermark is confirmed, then Step 2 is applied. Otherwise, the ownership is denied to the claimant.
2. The trusted third party applies the watermark embedding algorithm. Let the resulting watermarked video/image be V_υ. Goto Step 3.
3. The trusted third party compares the resulting V_υ with the published watermarked video clip/image V_W. If V_υ and V_W are similar, i.e. $C(V_\upsilon; V_W; \delta) = 1$, then the ownership is granted to the claimant, otherwise the ownership is denied.

The watermarking process including the watermark construction and the watermark embedding procedure were proven to be non-invertible in [4].

3.4 Discussion

Note that the construction of the watermark plays an essential role in our scheme to solve the rightful ownership problem. By using the encryption function as a „black-box", the watermark can not be arbitrarily chosen by an attacker. Because of this, checking the watermark construction is needed during the watermark verification process.

In the watermark construction and in the verification process we indicated, that without the original, we can not verify if a watermark is a legitimate one. However, the „original" does not have to be a true image or part of the video clip. For example, we can Xor the original frame with the first 100000 digits of π and use the result as the „original".

We also notice that EPAD is actually redundant in the watermark creation process. We can simplify the process and use the random sequence p_l as the watermark as long as p_l is created by applying DES (or other encryption functions) to V.

Finally, there is a concern that watermark embedding algorithm may be subject to the multiple-document attack: the AC coefficients which were dropped in one watermarked version may appear in other versions which were created by applying different keys; the AC coefficients which were modified in one watermarked version might be untouched in other versions. Therefore it is possible for the attacker to guess the original value of the AC coefficients if there are enough versions available. One way to solve this problem is to use a master key to preprocess the original video clip (image) by applying the proposed watermarking algorithm and then always use this preprocessed version as the „original". In doing so, the original information which were changed by the preprocessing will never be exposed.

4 Non-invertible Scheme for MPEG Audio

Audio data as well as other types of multimedia data are often stored and transmitted in compressed format, such as MPEG audio. The watermarking schemes should target the compressed data domain. Notice that, although the watermarking schemes such as Boney's scheme [1] work well in uncompressed data domain, extra decoding and re-encoding steps have to be taken if the audio data is already available in compressed format. Because the Boney's scheme uses the MPEG psychoacoustic model to mask the watermark, there would be no watermark presented in MPEG audio stream. Therefore, the study of watermarking in MPEG audio bit stream is interesting.

In this section, we cover the watermarking procedures for MPEG Layer II audio streams. However, the presented methods are easily extended to other layers of MPEG audio.

4.1 Watermark Construction

First of all, in order to be immune against CMYY attack, the watermarking scheme has to be non-invertible. There are several different approaches to implement the non-invertibility. One of them is to create the watermark by applying standard encryption algorithm such as DES to the original data as we discussed it in Section 3 for MPEG Video.

The watermark construction for MPEG audio has the following steps:

First, a key KEY is selected, and for each MPEG audio frame a_j , j=1, ..., N (number of audio frames), we apply DES with KEY to it to get a random byte sequence RBS:

$$RBS = DES_{KEY} \text{ (one audio frame } a_j) \quad (9)$$

Second, let RBS_i be the i-th byte of the random byte sequence and w_i be the i-th bit of watermark bit stream, then the watermark can be created by:

$$w_i = -1 \text{ if } RBS_i = \text{even number; 1 otherwise} \quad (10)$$

The watermark bit sequence is applied repeatedly to the audio data of the same audio frame if the length of the watermark bit sequence (the number i) is less than the number of samples in a frame. Normally, i is in a range from 100 to more than 1000, therefore, it is secure enough to use it repeatedly. In the case that i is larger than the number of samples in a frame, the over-produced bits are ignored.

4.2 Watermark Embedding Procedures

Notice that there are two major parts in data fields of an MPEG audio frame. One is the scale factors and the other is the encoded samples. Both of these

two parts can be used to embed the watermark. We describe the two procedures below.

4.2.1 Watermarking Scale Factors

Scale factor is the multiplier that makes the samples to fully use the quantizer range. The decoder multiplies the scale factor with decoded quantizer output to reconstruct the quantized subband samples. Each scale factor takes 6 bits, therefore, we have as many as 63 levels of scale factors (indexed from 0 to 62, 63 is not used by the standard). The level changing of scale factor has an auditory effect that the sound becomes stronger when the scalefactor level increase (index decrease) and becomes weaker when the scalefactor level decreases (index increase). However, tests show that a small change of scalefactor level (for example, increases or decreases by 1) normally can not be detected by the people. Our first MPEG audio watermarking procedure is based on this observation.

The watermarking procedure is very simple and just adds the watermark bit w_i to the index of the corresponding scale factor with two exceptions: (1) if index is 0 and $w_i = -1$ then do nothing; (2) if index is 62 and $w_i = 1$ than do nothing.

Let $SF_i(index)$ be the i-th scale factor with the level indicated by index and SFW_i be the i-th watermarked one. The watermarking procedure can be described as:

$$SFW_i = \begin{cases} SF_i(index) & \text{if index}+w_i = -1 \text{ or } 63; \\ SF_i(index+w_i) & \text{otherwise} \end{cases}$$

This scheme has drawbacks. The first one is that, for some audio streams, there are only a few scale factors for a frame, therefore, our frame-based watermarking scheme does not have much data to watermark. One solution is to group frames together and to apply the watermarking procedure at a group of frames.

When the scale factor is increased by 1 level (index decrease by 1) or decreased (by 1 or multiple levels), there is no significant audio distortion. However, when the scale factor is increased by 2 levels or more, there is often perceivable audio distortion and the noise can be heard. This introduces a problem that multiple watermarks can not be applied. The reason is that when multiple watermarks are applied, certain scale factors would be increased by multiple levels and perceivable noise would be introduced.

This also creates another problem because an attacker can lower the scale factors dynamically by 2 levels or 3 levels (increase the index by 2 or 3) and destroy the watermarks.

4.2.2 Watermarking Encoded Samples

The other choice is to embed the watermark into the sample data. The basic idea is to add the watermark (-1/1 bit sequence) to encoded sample sequence. However, changing of these encoded samples is very sensitive. Tests show that, if we change every encoded sample by either adding 1 or -1, the distortion of the resulting audio is easily detected by human ear and the watermarked audio is not acceptable.

In order to solve this problem, we introduce a spacing parameter sp which has the following meaning: in approximately every sp samples, we randomly select 1 or 2 samples to watermark. By choosing a good spacing parameter, the distortion can be minimized. This conclusion is also supported by our experimental tests (see next section). The use of spacing parameter gives us a method to adaptively watermark MPEG audio depending on different audio streams.

The watermark creation procedure is slightly modified to incorporate the spacing parameter:

$$W_i = \begin{cases} -1 \text{ if } RBS_i = 0 \pmod{sp} \\ 1 \text{ if } RBS_i = 1 \pmod{sp} \\ 0 \text{ otherwise} \end{cases}$$

The watermarking procedure is similar to the previous one except that we have to make sure that the watermarked sample does not have the format of „111...1" because this kind of sample coding is illegal in MPEG audio standard.

Let S_i be the i-th sample in an audio frame and SW_i be the i-th watermarked sample. Let $nbal_i$ be the number of bit allocation for i-th sample (the information of bit allocation for each sample comes from the Bit Allocation field in an audio frame). The watermarking procedure can be described as:

$$SW_i = \begin{cases} S_i & \text{if every bit of } (S_i + w_i) \text{ is 1;} \\ S_i + w_i & \text{otherwise} \end{cases}$$

4.2.3 Discussion

Both MPEG audio watermark schemes are simple extensions of the concepts from spread spectrum communications. They are designed to apply watermark in the compressed data domain so that the expensive decoding/re-encoding can be avoided.

Because the creation of watermark is based on applying standard encryption function to the original audio data, the non-invertibility of these schemes can be easily proved.

Another requirement is that the original MPEG audio stream must be presented in the verification process because, in general, the watermark schemes which do not use the originals in their verification are invertible. (See also [4, 3]).

Finally, we want to point out that, because the human ear is more sensitive to audio distortion than human eye to image distortion, the amount of data which can be embedded as watermark is very lim-

ited and depends on the content of the audio streams which we showed experimentally [3]. For this reason, in the 2nd scheme, we introduced the spacing parameter which can be thought as an indicator to measure how much data can be embedded. By choosing a smaller spacing parameter, we could reach the upper limit which means that if we embed more data, the distortion will be noticeable. This also means that multiple watermarks can not be applied.

5 Conclusion

In this paper, we presented watermarking methods which embed the watermark directly into the MPEG video and audio bit streams. This is very much desired if the video and audio streams are already in MPEG encoded format. Otherwise, if we use watermarking schemes designed for uncompressed video and audio, we have to go through the expensive decoding-watermarking-encoding process.

Our experimental tests show that the watermarking methods minimize the visual and acoustic distortion and provide good video and audio quality for various types of MPEG encoded streams. Although we use only watermarking on MPEG I frames and MPEG Audio Layer II streams in our experimental tests, the proposed schemes can be applied to other compressed/uncompressed images/videos and MPEG Audio Layer I streams and Layer III streams. We also take the rightful ownership problem into our consideration. By creating the watermark using a standard encryption function such as DES, the non-invertibility property can be achieved and the unique ownership can be determined.

6 References

[1] L. Boney, A. H. Tewfik, and K. N. Hamdy. Digital Watermarks for Audio Signals. In Proceedings of 1996 IEEE International Conference on Multimedia Computing and Systems, pages 473{480, Hiroshima, Japan, June 1996.

[2] S. Craver, N. Memon, B. Yeo, and M. Yeung. Can Invisible Watermarks Resolve Rightful Ownerships? Technical Report RC 20509, IBM Research Division, July 1996.

[3] L. Qiao and K. Nahrstedt. Non-invertible water- marking methods for mpeg encoded audio. Tech- nical report, University of Illinois at Urbana- Champaign, Urbana, IL, June 1998.

[4] L. Qiao and K. Nahrstedt. Watermarking Method for MPEG Encoded Video: Towards Resolving Rightful Ownership. In IEEE International Con- ference on Multimedia Computing and Systems, Austin, TX, June 1998.

[5] R. B. Wolfgang and E. J. Delp. A Watermarking Technique for Digital Imagery: Further Stud- ies. In Proceedings of the International Conference on Imaging Science, Systems, and Technology, Las Vegas, Nevada, JULY 1997.

Copyright and Content Protection for Digital Images based on Asymmetric Cryptographic Techniques

Alexander Herrigel
Digital Copyright Technologies
Stauffacherstr. 149
8004 Zurich, Switzerland
Fax: +41-1-923.81.31
Email: herrigel@usa.net

S. Voloshynovskiy
State University "Lvivska polytechnika"
Faculty of Radio Engineering Devices
290646 Lviv
S. Bandery Str. 12, Ukraine
Email: svolos@polynet.lviv.ua

ABSTRACT

This paper[93] presents a new approach[94] for the copyright protection of digital multimedia data. The system applies cryptographic protocols and a public key technique for different purposes, namely encoding/decoding a digital watermark generated by any spread spectrum technique and the secure transfer of watermarked data from the sender to the receiver in a commercial business process. The public key technique is applied for the construction of a one-way watermark embedding and verification function to identify and prove the uniqueness of the watermark. Our approach provides secure owner authentication data who has initiated the watermark process for a specific data set. Legal dispute resolution is supported for multiple watermarking of digital data without revealing the confidential keying information. Content protection for images is provided by ciphering/deciphering the data in the transform domain.

KEYWORDS

Copyright protection, asymmetric cryptographic techniques, spread spectrum techniques, copyright holder, copyright certificate center, digital copyright certificate, digital multimedia data.

1 Introduction

Confronted with the need for the rapid development and deployment of flexible information services for the information highway, the Internet community has recently developed new technologies, protocols, and interfaces which enable the fast provision of interactive multimedia and hypermedia services. Based on this development, there is a growing need in the commercial market for the fast provision of IT-based multimedia services. Current market estimates show that for example in Germany, the multimedia market will rapidly grow from 6 Billions DM to at least 25 Billions DM within the next five years. Some providers such as commercial image archives have already started new services since these new technologies enable a worldwide access for their specific customer groups. In addition, the implementation and deployment of these new technologies for a specific commercial business process results in much less manual operations and in a faster business process execution time. Different service providers are, however, confronted with an important problem. The adoption of these new technologies comes along with the loss of control concerning the commercial distribution of the multimedia data prepared for the different customers. If a digital multimedia data set may be copied and distributed in milliseconds without any expensive equipment and quality burden, who can then differentiate between the original and its copy? If unauthorized buyers distribute the data, how can a provider issue a successful court case? The provider's loss of control over the distribution may result in severe financial damages and reputation problems.

The IT-technologies developed so far have been mainly based on the TCP/IP protocol suite with its network services. Originally, this protocol was developed with the emphasis on availability only. In the last years additional security requirements have been identified to execute commercial business processes on the Internet. Specific security architectures such as the Secure Socket Layer Protocol [1] have been designed and implemented to address the identified threats. These security architectures enable the generation of an authenticated and trusted virtual communication channel between a client and a provider system. They offer, however,

[93]This work has been funded by the Swiss National Science Foundation under the SPP program (Grant. 5003-45334) and by the EC (ESPRIT Project No. 25530: JediFire).

[94]All methods, procedures, and schemes presented in this paper are based on patent applications.

no means for the copyright protection of digital multimedia data distributed in a commercial environment. The lack of the new IT-technologies for the different aspects of the copyright protection problem has resulted in the development of so-called digital watermarking techniques. The main emphasis of these techniques is the embedding of copyright owner authentication data in a multimedia data set such that only the copyright owner can identify and verify this authentication data.

A number of different approaches [2-17] have recently proposed. This development was based on technical exploitation issues only and has often not addressed legal aspects. Different groups in academic research and industry are very much concerned with the provisioning of robust watermark techniques, which are resistant against transformations an attacker may apply to destroy the watermark. Some approaches are quite promising. But even if these promising approaches are applied for the provisioning of authentication data within the multimedia data set, some important legal aspects remain unsolved. Suppose Alice is a copyright holder of a digital image and Bob a customer very much interested to buy this image. Since Bob knows that this image is in his country very popular, he decides to redistribute it without any permission from Alice. Having received the image from Alice protected by her digital watermark he embeds also his own watermark before his redistribution process is invoked. If Alice has detected the copyright infringement she issues a court case. The court, however, is in a serious dilemma. Confronted with the illegal redistribution Bob tells the court that he is the original copyright holder, since he can detect his watermarked data. He asked the court to punish Alice for copyright infringement. In addition, he may claim that the detection of specific authentication data does not provide substantial proof that it was really generated by Alice and not by other parties[95]. A solution to this problem is an important prerequisite for the commercial acceptance of the Internet technology by multimedia service providers. These providers will refuse to adopt the new IT-technologies, if the multimedia distribution process is not under their control.

Content protection can be provided applying cryptographic algorithms developed in the last period [18]. These techniques, however, are not robust against specific denial of service attacks, such as loosely compression of ciphered images.

[95] Referring to different legal frameworks [20-23] the results of the watermarking verification procedure (extraction) must be based on input data which can be verified by third parties such as a court of law to be used in a copyright infringement case for legal evidence.

2 Definitions

The copyright protection of a multimedia data set is considered as the process of proving the intellectual property rights to a court of law against unauthorized reproduction, processing, transformation, or broadcasting on the basis of digital evidence data. This process is based on a watermarking process WP and a registration process RP. RP is executed after WP has been initiated and finished. RP is executed by a third party, which represents a different legal entity as the Copyright Holder (CH), and provides digital evidence data for the CH required for verifying copyright ownership. The notation applied for the WP follows the stenographic terminology reported in [16]. We use the expression cover-data and stego- data. The specific cover- or stego-data is digital image, audio, or video data. The WP embeds or extracts owner authentication data in or from multimedia data sets. This owner authentication data is embedded such that the commercial usability of the multimedia data set is not affected. For this purpose, a key is applied to embed encoded owner authentication data, called the watermark, into the cover-data set I, resulting in a stego-data set I*. The watermark data can then be extracted from the stego-data if the correct key is used. The embedding and verification process is illustrated in Fig. 1.

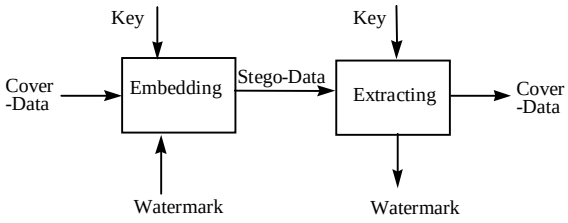


Figure 1: The watermark embedding and verification process.

We assume that the WP applied for the embedding is a one-way function [18], collision resistant [18] and robust, i.e. it is for an unauthorized third party not possible to overwrite or delete this watermark without the cryptographic keying information. In the following, WP is based on any perceptually adaptive spread spectrum technique, a specific type of a symmetric cryptographic system [18]. In order to embed or extract a watermark, it is necessary to know the exact values of the seed used for the generation of pseudo random sequences used to encode the watermark. Because spread spectrum signals are statistically independent (and therefore virtually orthogonal), more than one watermark may be encoded into the multimedia data set. Depending on the seed applied for the embedding and verification, we distinguish between a private and a detection watermark. A private watermark is defined as encoded owner authentication data embedded with a cryptographic signature as the seed. A detection watermark is defined as encoded owner authentica-

tion data embedded with a cryptographic secret key as the seed. We differentiate between copyright protection and content protection.
Content protection is considered as an additional process applied during the trading transaction between a service provider and a customer. Our content protection is based on the image data and not on cryptographic ciphering algorithms applied during the communication between the service provider and the customer, since these cryptographic algorithms are not robust against loosely compression and other image transformations.

3 Security Requirements

The security requirements for the applied business processes are partitioned into three classes, namely the requirements for the secure off- or on-line exchange of business data, the requirements for the content protection, and the requirements for the copyright protection. We envision a comprehensive security solution, which covers all security requirements for the business process. Since every buyer in a commercial environment can embed additional watermark data, the detection and verification of a specific watermark does not resolve a possible legal conflict. We have, therefore, introduced in our architecture an entity, called the Copyright Certificate Center (CCC).
This entity offers an on-line and fault-tolerant service accepting copyright requests. The CCC then generates on the basis of the provided data from the stego-image, its classification and the sender's address a digital copyright certificate which is first stored locally and then sent back to the sender. This digital copyright certificate is based on a sequence number continuously increased from request to request. It is, therefore, possible to identify the time a copyright ownership was claimed. If a buyer embeds an additional watermark and registers his claim also at the CCC, the associated sequence number of the copyright certificate will be larger as the one of the original copyright holder. The CCC generates, therefore, digital evidence data to be used later in a court case to resolve the copyright ownership.

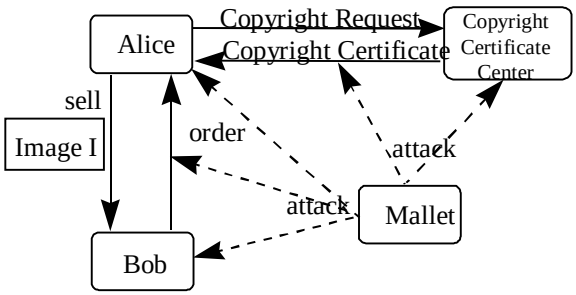


Figure 2: Thread analysis scenario of a business process.

With respect to a typical business process of a provider such as a commercial image archive (Fig. 2), the following security requirements have been derived on the basis of an extensive thread analysis presented in [17].
Security requirements for the on- or off-line communication:

- Mutual authentication
- Integrity
- Non-repudiation

Security requirements for the copyright protection:

- Copyright registration with digital copyright certificate96
- One-way watermark function
- Cryptographic key based robust watermarking
- Oblivious watermarking

Security requirements for the content protection:

- Confidentiality

4 Security Architecture

We envision a copyright protection system operating an open environment like the Internet with different interconnected computers. Users may be located anywhere and may sell or buy multimedia data. If legal dispute resolution for multiple watermarks is needed the Copyright Holder (CH) sends copyright information and from the stego-data derived authentic information to the Copyright Certificate Center (CCC).

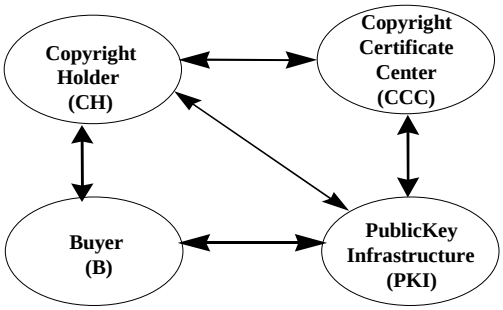


Figure 3: The entities of the derived security architecture.

After having received a copyright certificate from the CCC, the CH can sell his digital multimedia data, for example, via a shopping mall, to a buyer (B). The Public Key Infrastructure (PKI) supports the distribution of authentic public keys between all parties needed for mutual authentication, non-repudiation, and integrity. The communication channels between the different entities are shown in Fig 3.
Our approach enables the secure generation and transmission of watermarked data using an asymmetric key pair applied in public-key cryptography. The cover-data set is watermarked, while the wa-

[96] Any modification of this certificate must be detected.

termark is encoded using the secret key of the asymmetric key pair. Derived information of the resulting stego-data set such as the data set identifier, a textual description and related information is transmitted authentically to a registration party, the CCC, while the same keys are used for establishing a secure transmission between the parties. The CCC permanently stores after content verification the transmitted data and issues a copyright certificate which is stored locally and then transmitted to the CH. During the trading process between the CH and the B, the involved parties use their asymmetric key pairs to send authentic information. In addition, they apply an image ciphering and deciphering process for content protection and retrieval. This content protection may be applied for on-line and off-line communication.

4.1 Symbols

H, C, B, I	Distinguished (unique) name of the Copyright Holder, the Copyright Certificate Center, the Buyer B and the Public Key Infrastructure I.
Cert H , Cert C, Cert B	Entity H's public key certificate from I, entity C's public key certificate from I and entity B's public key certificate from I.
(ps_X, vs_X)	The asymmetric signature and verification key pair of an entity with the distinguished name X [19].
(pc_X, vc_X)	The asymmetric decipherment and encipherment key pair of an entity with the distinguished name X [19].
CC	A copyright certificate
$DSSMR_G$ (X,Y,Z)	A digital signature generation scheme with message recovery, where X denotes the private key, Y the input data, and Z the resulting signature.
$DSSMR_V$(X,Y,Z)	A signature verification scheme with message recovery, where X denotes the public key, Y the input data, and Z the resulting output data.
$DSSAP_G$(X,Y,Z)	A digital signature generation scheme with appendix, where X denotes the private key, Y the input data, and Z the resulting signature.
$DSSAP_V$(X,Y,Z)	A signature verification scheme with appendix, where X denotes the public key, Y the input data, and Z the resulting output data.
crh	A collision resistant hash function
OWEA(X,Y,CD, SD)	The oblivious, spread spectrum based watermark embedding algorithm with the seed X, the payload Y, the cover data CD, and the resulting stego-data SD.
OWVA(X,SD,Y)	The oblivious, spread spectrum based watermark verification algorithm with the seed X, the stego-data SD, and the resulting payload Y).
TVP	Time variant parameter, such as a sequence number or a time stamp.
RPMG(X,Y)	A random phase mask generator, where X denotes the cryptographic key as input data and Y denotes the resulting phase mask as output data.
DIES(PM,OI,CI)	A symmetric digital image encryption scheme, which is based on the Fourier transform of the image, phase modification (random mask encoding by multiplication on the complex exponential component $e^{j\varphi(m,n)}$), inverse Fourier transform, and quantization, where PM denotes the phase mask and ID denotes the original image as input data and OI denotes the ciphered image as output data.
||	Concatenation of two data elements.
CD	Cover-Data
SD	Stego-Data

4.2 Registration Based Copyright And Content Protection

Depending on the proof-level to be provided for the protection, our approach provides three different protection levels, which are based on each other, namely individual copyright protection, copyright protection with registered cryptographic keys, and

copyright protection with a CCC on the basis of registered cryptographic keys. Since the first two cases are special cases of the third one, we present only the approach for the registration based copyright protection.

The system for the CCC based watermark protection is partitioned into four processes, namely the CH with the name H, the B process with the name B, the PKI process with the name I, and the CCC process with the name C. Suppose (ps_H ,vs_H), (pc_H ,vc_H), (ps_B ,vs_B), (pc_B ,vc_B), (ps_I ,vs_I), (pc_I ,vc_I), (ps_C ,vs_C), and (pc_C ,vc_C) are the asymmetric key pairs of H, B, I and C, respectively and all the involved parties would like to exchange information by on-line communication.[97] H has an authentic copy of $Cert_B$ and $Cert_C$ whose signatures were verified with the authentic copy of vs_I . B has an authentic copy of $Cert_H$ and $Cert_C$ whose signatures were verified with the authentic copy of vs_I . C has an authentic copy of $Cert_H$ and $Cert_B$ whose signatures were verified with the authentic copy of vs_I . The following phases are then applied:

Phase 1: H retrieves the cover data CD, generates a unique identifier ID_{CD} := crh(H||SN), where SN is a serial number, stores ID $_{CD}$, and retrieves the key pair (ps_H, vs_H).

Phase 2: Detection watermark embedding
H generates the stego-data SD applying the transformation: OWEA(crh(ps_H), SN||SN, CD, SD).

Phase 3: Private watermark embedding
1. H generates the private Owner Authentication Data OAD_{CD} applying $DSSMR_G(ps_H, ID_{CD}, OAD_{CD})$.
2. H generates the stego-data SD applying the transformation: OWEA(crh(OAD_{CD}), ID_{CD}, CD, SD), where CD is the SD of the last phase.

Phase 4: H stores the resulting stego-data SD.

Phase 5: H and C execute the following steps for the secure registration or validation of copyright requests, and the generation of copyright certificates.
1. H generates first the copyright request data CRD, CRD := crh(SD||SN) and then the copyright request CR, CR := <TD||SigTD>, with TD := CRD|| TVP||• H||C, and $DSSAP_G(ps_H, TD, SigTD)$. H then transmits CR to C.
2. C receives CR and verifies TD, applying $DSSAP_V$ (vs_H, SigTD, IVR), where IVR denotes the intermediate verification result. If IVR = crh(TD), with TD := CRD||TVP||H||• C, then TD has been successfully verified and the next step shall be executed. In any other case, the processing and communication between the H and C is stopped. C verifies the CRD content. If the data has been successfully verified then the next step shall be executed. In any other case, the processing and communication between the H and C is stopped.
3. If verification was successful, C generates the corresponding digital copyright certificate executing $DSSAP_G(ps_C$, CCD||TVP, SigCCD). C then stores the copyright certificate CC := CCD||TVP• • SigCC and generates then the Copyright Confirmation Reply CCR, CCR := <TD||SigTD>, with TD := CC||TVP||C||H, and $DSSAP_G(ps_C$, TD, SigTD). C then transmits CCR to H.
4. H receives CCR and verifies TD, applying $DSSAP_V(vs_H$, SigTD, IVR), where IVR denotes the intermediate verification result. If IVR = crh(TD), with TD := CC||TVP||C||H, then TD has been successfully verified. H then verifies and stores the CC. The following phase can now be executed repeatedly, if necessary, without repetition of the previous phases.

Phase 6: H and B execute the following steps for the trading of copyright protected digital images:
1. B generates the trading transaction T1, T1 := <TD||SigTD>, with TD := ID_{CD}||TVP||B||H, and $DSSAP_G(ps_B$, TD, SigTD). B then transmits T1 to H.
2. H receives T1, verifies TD, applying $DSSAP_V(vs_B$, SigTD, IVR) where IVR denotes the intermediate verification result. If IVR = crh(TD), with D := ID_{CD}||TVP||B||H, then TD has been successfully verified and the next step shall be executed. In any other case, the processing and communication between the H and B is stopped.

[97] In the case of off-line communication, the security mechanisms to be provided for the communication are covered by operational means.

3. If the verification was successful, H retrieves with the ID_{CD} information the corresponding stego-data SD and generates the trading transaction T2 := <TD||SigTD>, with TD := CI||TVP||H||B, DIES(PM, SD, CI) with RPMG($DSSMR_G(ps_H$, B||SN), PM), and $DSSAP_G(ps_H$, TD, SigTD). H then stores $DSSMR_G(ps_H$, B||SN) and transmits T2 to B.

Phase 7: B receives T2 and verifies TD, applying $DSSAP_V(vs_H$, SigTD, IVR), where IVR denotes the intermediate verification result. If IVR = crh(TD), with TD := CI||TVP||H||B, then TD has been successfully verified and CI is locally stored.

Phase 8: After B has paid, H sends $vc_B[DSSMR_G(ps_H$,B||SN)]. H deciphers it and generate the random phase mask PM. This random phase mask is then used for deciphering CI to get the original stego-data SD.

4.3 Content Protection

The following three Figures illustrate the image encryption and decryption process based on the generated phase mask derived from the cryptographic key.

Figure 4: The original image

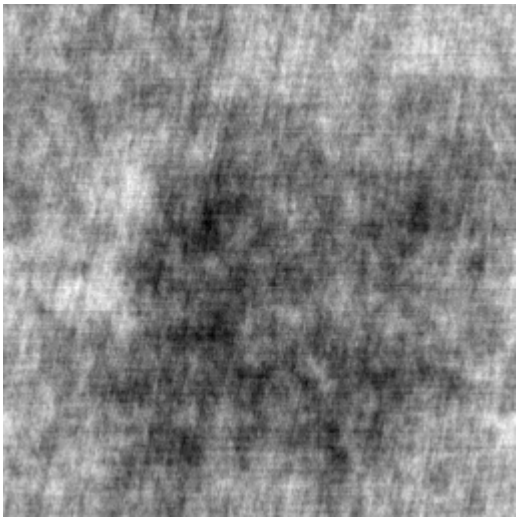


Figure 5: The ciphered image

Figure 6: The deciphered image

4.4 Remarks

1. Depending on the applied asymmetric scheme the private decipherment key may be identical to the private signature key and the public encipherment key may be identical with the public verification key.
2. Since the generated asymmetric key pairs are unique, the CH can be uniquely identified on the basis of the digital copyright certificate.
3. B may check the copyright certificate requesting C (or H) to transfer an authentic copy of the copyright certificate for a given identifier ID CD. Except the data transferred, the applied protocol is the same as described above.
4. If H would like to transfer a specific copyright of a CD set to another legal party, he may initiate a copyright revocation request with C. The

different phases of this request are analogue to the copyright request.

5. For copyright verification process, the CH first verifies the detection watermark and then the private watermark with the extracted SN.

5 Implementation

To demonstrate and verify in detail the feasibility of the new approach, a distributed software system, running on one or more computers, has been implemented. The overall system is partitioned into four different server application processes, namely the Public Key Infrastructure (PKI) application process, the Copyright Holder (CH) application process, the Copyright Certificate Center (CCC) application process, and the Buyer (B) application process. These processes support a Graphical User Interface and dedicated cryptographic communication protocols based on a socket interface. In addition, the CH can embed and verify the digital watermarks.

5.1 The Copyright Holder Application Process

The CH application process can load images, enter copyright information, embed a watermark into a copy of the cover-image, called stego-image, contact the CCC, save all data encrypted in a database and sell the digital stego- image. In addition, he can verify for any stego- image the watermark which was previously embedded.

The CH supports the following functions:

1. Generate public/private key-pair, register it at the PKI and save it persistently
2. Designate important regions in the image (rectangles)
3. Entry and management of copyright information
4. Embed/Verify watermark
5. Display and verify digital copyright certificate
6. Request and receive public key certificates from PKI
7. Contact Copyright Certificate Center providing mutual authentication, integrity and non-repudiation to receive a copyright certificate which is then locally stored
8. Save the cover-image encrypted and delete the unencrypted version
9. Adjust the security or quality factor of the watermark
10. Provide commercial image trading features offering a secure socket connection, which supports mutual authentication, integrity, and non-repudiation

5.2 The Copyright Certificate Center Application Process

The CCC application process is an important entity in the copyright protection process to resolve possible conflicts if multiple digital watermarks are embedded. In addition, this entity provides the digital evidence needed by the copyright holder for a court case. The CCC application process is contacted by the CH application process and generates on request a digital copyright certificate, which is then sent back to the CH application process. The current implementation covers the following functions:

1. Request and receive public key certificates from the PKI
2. Provide a listener port to answer incoming copyright requests
3. Verify content consistency of all received copyright requests
4. Save all information persistently and securely, because the data has to be recovered after a server restart
5. Provide display of the actual sequence number to any client system
6. Information display of registered images

Since the submitted information to the CCC application process is based only on derived data from the stego- image, there is no need for the copyright holder to put additional substantial trust in this process. Only the generation of the actual sequence number must satisfy additional constraints.

5.3 The Buyer Application Process

The Buyer application process requests a list of registered images from the CCC application process. If an image is selected, the associated classification, the text description, and the thumbnail is transferred to the Buyer's system. If the user would like to buy the image on the given pricing information, he just needs to execute the corresponding Buy-request. This request activates the cryptographic protocol, which supports mutual authentication, integrity, and non-repudiation (for billing purpose). The selected stego- image is then transferred from the CH system to the Buyer system.

5.4 THE PUBLIC KEY INFRASTRUCTURE

The PKI may be applied as a Trusted Third Party (TTP) or Certification Authority (CA) server with an on-line registration service issuing all user key certificates. On the basis of the authentically distributed public key from the PKI and the mode of operation (TTP or CA), a public/private key-pair along with its associated certificate may be generated and then passed back as ciphered information to the client. These operations are based on a specific re-quest from the CH. The request includes a password for online key generation, key distribution, and certificate handling. The request is protected on the basis of the TTP's authentically distributed public key. In the case of the CA mode, a PKCS #10 type of request is issued by the CH and a certificate is then generated and transmitted by the PKI to the CH.

5.5 Example

The complete commercial process, which is covered by our technology, is illustrated by the following example. Suppose a digital image has been prepared for commercial exploitation. After a given number of image processing steps, the responsible authorized manager has controlled and approved the quality of the digital image shown in Figure 7 (see Annex). This image should now be entered in the commercial business process of the service provider.

The manager then executes the CH application process, shown in Figure 8 (see Annex). Based on a password based authentication scheme, this application process checks the identification of the manager. The CH application process retrieves the corresponding protected asymmetric key pair of the company, after the manager has been authorized as a valid user. The authorized manager then identifies the digital image of Figure 7 (see Annex) in the file system and loads the file into the graphical display of the application process. Depending on the classification he chooses – in our case Politic – and the adequate robustness level of the watermark (see entries WM Strength and Image Quality) he embeds, selecting the Add button, the digital watermark in the image with a payload he has entered in the Secret Signature field. After the digital watermark has been embedded and the resulting digital image locally stored in a new file which is called the stego-image file, the application process then displays a second panel which facilitates an easy registration procedure as shown in Figure 9 (see Annex). Most of the data are automatically determined and displayed by the application process. Only the countries copyright ownership is claimed for and a short description concerning the image context has to be entered by the user. After this information has been entered, a secured digital copyright request is then generated and send to the CCC application process, if the Register button was selected. Figure 10 displays the main panel of the CCC application process. This process waits for incoming digital copyright requests. If a request has been sent, the associated cryptographic attributes are checked. A digital copyright certificate is then generated and locally stored, after the information content has been validated. The generated digital copyright certificate is then sent back to the CH application process. The CCC application process stores for every client the corresponding digital copyrights, along with the copyright request data as shown in Figure 10 (see Annex). Figure 11 (see Annex) shows in detail an example of a digital copyright certificate. Referring to the copyright forms often applied in former days for registration purposes (example USA), the digital copyright certificate has all important data needed to prove the copyright ownership. Since the certificate is protected by the digital signature of the CCC application process, which refers to the asymmetric key pair of the company running this service, it is not possible for the CH to change some information of the digital copyright certificate. For the user's convenience, this digital certificate can also be printed for the preparation of an investigation or a court case. If the authorized manager or any other persons detects, that the image is used by unauthorized third parties, they can load the image into the CH application process and apply the watermark verification procedure. Since the coding depends on the signature generated by the private key of the applied asymmetric key pair, the authorized manager is the only one who can detect the watermark. In addition, by verifying the signature he has used as the seed for the coding process, a third party can validate if he has embedded the watermark. Figure 12 (see Annex) shows the graphic output of the verification procedure. The original classification and the payload are retrieved. Any buyer can access the Internet and execute the Buyer application process Figure 13 (see Annex) shows the panel of this process which contacts the CCC application process and retrieves a list of images, the buyer is potentially interested in. This list is displayed in a specific window. Selecting one image in the list initiates a request to the CCC application process to transfer the text description and the thumbnail representation of the registered digital image with a watermark.

If the user would like to buy one image, he has only to select the Buy button, which activates a security protocol between the CCC application process and the CH application process. The ciphered stego-image is then transferred from the CH system to the Buyer's system. After having paid the buyer can decipher the image and use it in his production process.

6 Conclusions and Future Work

We have presented in this paper a new approach for the copyright protection of multimedia data sets. This approach is based on asymmetric cryptographic protocols and techniques. In contrast to any other scheme, the combination of any spread spectrum based technique in conjunction with an asymmetric cryptographic technique allows the construction of a one-way watermark function, since only the copyright holder is able to verify the private watermark.

In addition, the CH may prove that he has invoked the protection procedure, if a third party, such as the court of law, verifies the signature applied for the seed generation of the spread spectrum technique. Even if the different phases of the approach are known in the public, the security of our approach is not compromised. Compared to other ap-

proaches the following new properties have been identified:

1. The use of an asymmetric cryptographic key pair for the seed generation enables the execution of asymmetric key agreement protocols with message recovery or appendix and the protection of the communication between the involved parties. Different security services for the communication, such as mutual authentication, integrity, and non-repudiation are supported along with the protection against copyright infringement by the system with one asymmetric cryptographic key pair[98].
2. The present technique enables a strong binding relation between the image ID, the image, and the CH if the CH registers his copyright at the CCC. If an image is watermarked later by an unauthorized person, the unique sequence number[99] and image hash value in the copyright certificates resolves the copyright ownership.
3. The CH does not have to reveal his private cryptographic key if ownership verification has to be applied by a different legal party.
4. The present technique supports the transferal of copyrights. If copyright is transferred to another legal party, corresponding copyright revocation certificates may be generated (Rights Management Information).
5. Due to the fact that the copyright certificate is linked uniquely to the copyright protected multimedia data, the CH is in a position to submit third party backed evidence in infringement proceedings for the fact that he claimed copyright in the work at a specific time. The secure communication supported by the derived security architecture is not only in the interest of the CH but also of the potential buyer/licensor. This is an important feature since the bona fide acquisition of copyright is not protected by the courts and therefore it is up to the buyer/licensee to ensure that the alleged seller/licensee is entitled to grant or transfer the rights in question. In addition, the buyer/licensee has the confidence of buying authentic content.

The approach presented is based on a spread spectrum technique developed by [9,10,15,17]. We would like to investigate enhancements such that StirMark [19] and other attacks do not affect the robustness of the embedded authentication data. The security architecture presented may also cover video, audio, or binary data. We are actually investigating new spread spectrum techniques for the copyright protection of video, audio and binary data.

[98] In our implementation the private decipherment key is identical with the private signature key and the public encipherment key is identical with the public verification key.

[99] Generated by the CCC.

7 Acknowledgments

We would like to thank Dr. Thomas Mittelhozer and Nazanin Baumgärtner from DCT for many stimulating discussions. We would also like to thank Professor Thierry Pun and Dr. J. K. Ó Ruanaidh of the University of Geneva, CUI, Computer Vision Group, for the technical discussions, the support, and the ongoing collaboration.

8 References

[1] Freier, P. Karlton and P. Kocher, "SSL Version 3.0", Netscape Communications", Version 3.0, November 1996.

[2] Cox, J. Killian, T. Leighton, and T. Shamoon, "Secure spread spectrum communication for multimedia", Technical report, N.E.C. Research Institute, 1995.

[3] G. Caronni "Assuring Ownership Rights for Digital Images" in H. H. Brueggemann and W. Gerhardt Haeckl, editors, Reliable IT Systems VIS '95, Vieweg Publishing Company, Germany, 1995.

[4] Z. Tirkel, G. A. Rankin, R. G. van Schyndel, W. J. Ho, N. R. A. Mee, and C. F. Osborne, "Electronic watermark", in Dicta-93, pages 666-672, Macquarie University, Sydney, December 1993.

[5] Z. Tirkel, R. G. van Schyndel, and C. F. Osborne, "A two-dimensional digital watermark", in ACCV'95, pages 378-383, University of Queensland, Brisbane, December 6-8, 1995.

[6] K. Matsui and K. Tanaka, "Video-Steganography : How to secretly embed a signature in a picture", in IMA Intellectual Property Project Proceedings, pages 187-206, January 1994.

[7] J. Smith and B. Comiskey, "Modulation and information hiding in images", in Ross Anderson, editor, Proceedings of the First International Workshop in Information Hiding, Lecture Notes in Computer Science, pages 207-226, Cambridge, UK, May/June 1996. Springer.

[8] 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.

[9] J. K. Ó Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of images", IEEE Inter-national Conference on Image Processing, Lausanne, Switzerland, September 1996.

[10] J. J. K. Ó Ruanaidh, W. J. Dowling, and F. M. Boland, "Watermarking digital images for copyright protection", IEEE Proceedings on Vision, Image and Signal Processing, 143(4): pp. 250-256, August 1996.

[11] P. Davern and M. Scott, "Fractal based image steganography", in Ross Anderson, ed., Proceedings of the First International Workshop in Information Hiding, Lecture Notes in Computer Science, pp. 279-294, Cambridge, UK, May/June 1996, Springer Verlag.

[12] J. Zhao and E. Koch, "Embedding robust labels into images for copyright protection", Technical report, Fraunhofer Institute for Computer Graphics, Darmstadt, Germany, 1994.

[13] J. Zhao, "A WWW Service To Embed And Prove Digital Copyright Watermarks", Proc. Of the European Conference on Multimedia Application, Services and Techniques, Louvain-La-Neuve, Belgium, May 1996.

[14] J.-F. Delaigle, J.-M. Boucqueau, J.-J. Quisquater & B. Macq, "Digital Images protection techniques in a broadcast framework: An overview", Laboratoire de Télécommunications et de Télédéction, Université Catholique de Louvain.

[15] Joseph J. K. Ó Ruanaidh and Shelby Pereira, "A Secure Robust Digital Watermark", Europto'98.

[16] Birgit Pfitzmann, Ross Anderson (Ed.), "Information Hiding", "Information Hiding Terminology", First International Workshop, Cambridge, UK, May/June, 1996, Proceedings, Springer, Lecture Notes in Computer Science, 1174.

[17] Alexander Herrigel, Adrian Perrig, and Joseph J. K. Ó Ruanaidh, "A Copyright Protection Environment for Digital Images" Gesellschaft für Informatik e.V., Fachgruppe 2.5.3 "Verläßliche IT-Systeme", Institut für Informatik und Gesellschaft, Universität Freiburg, "Verläßliche IT-Systeme: Zwischen Key Escrow Und Elektronischem Geld", VIS'97, Freiburg, Germany.

[18] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996, ISBN 0-8493-8523-7.

[19] Fabien Petitcolas, "Weakness of Existing Watermarking Schemes", http://www.cl.cam.ac.uk/~fapp2/watermarking/image_watermarking/stirmark/

[20] Berne Convention for the Protection of Literary and Artistic Work, Paris, 1971.

[21] General Agreement on Tariffs and Trade of 1994 and Art. 9 of its addendum on Trade Related Aspects of Intellectual Property Rights, TRIPS, AS 19995, p. 2357 and seq.

[22] World Intellectual Property Organization (WIPO), WIPO Copyright Treaty, Geneva, December 1996.

[23] WIPO Performances and Phonograms Treaty, Geneva, December 1996.

9 Annex

Figure 7: Approved digital image.

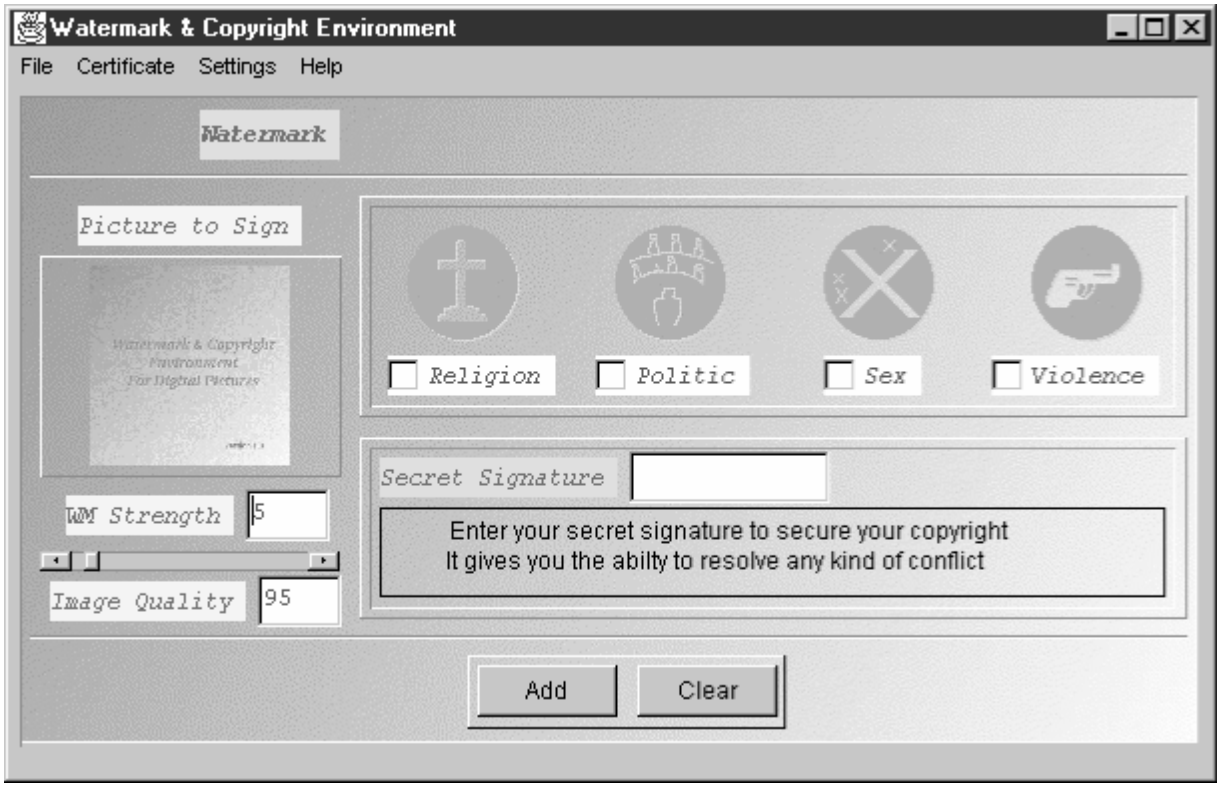


Figure 8: The main panel of the CH application process.

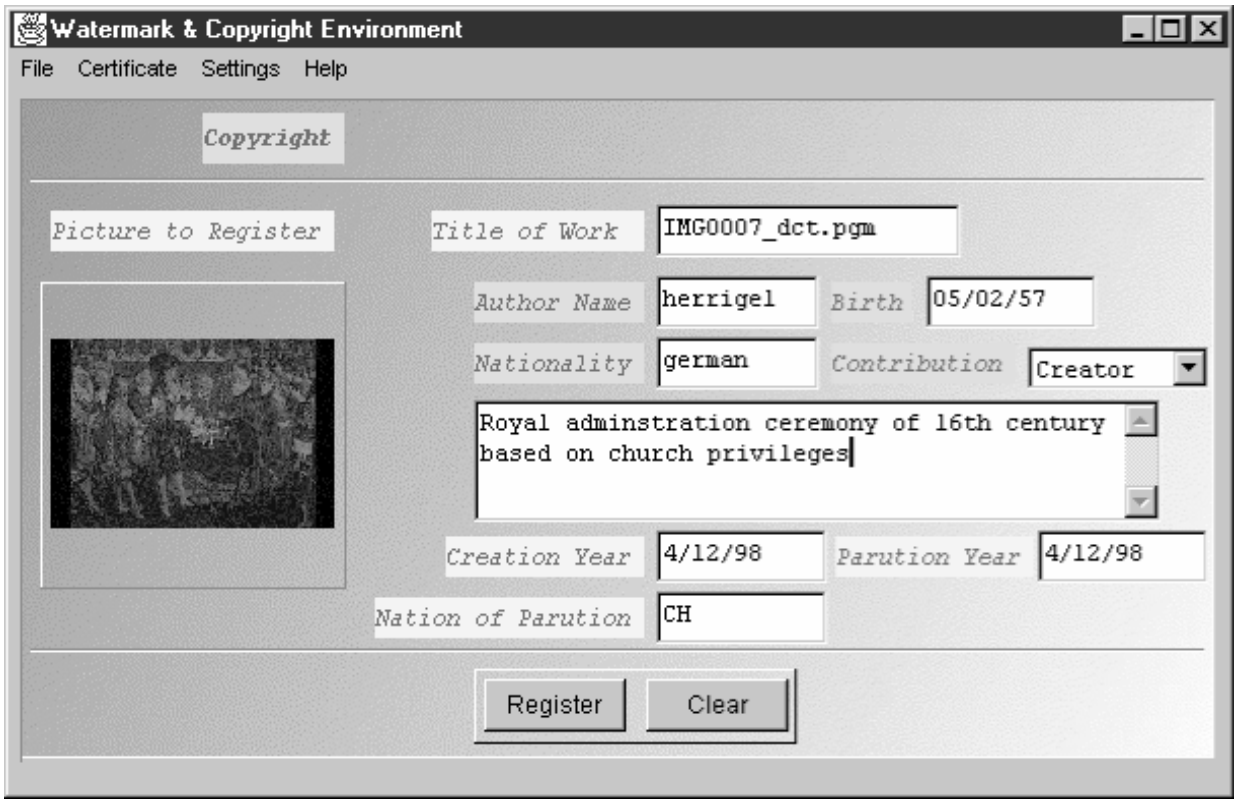


Figure 9: The panel for the registration of a watermarked image.

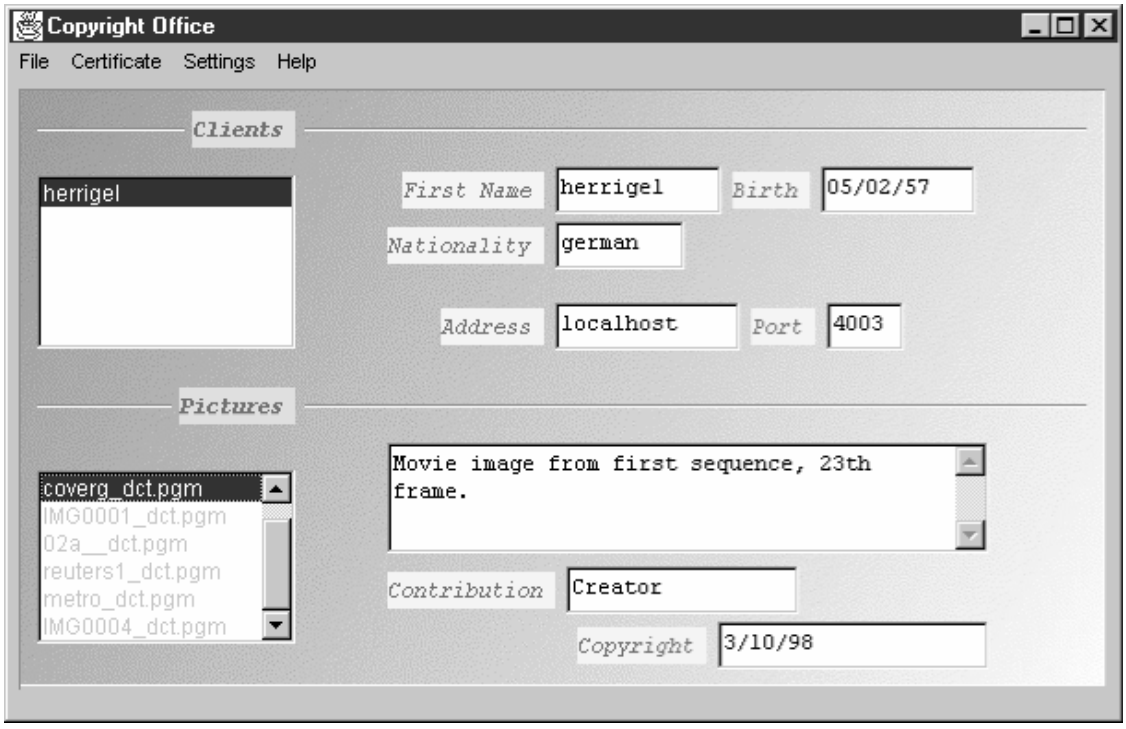


Figure 10: The panel of the CCC application process.

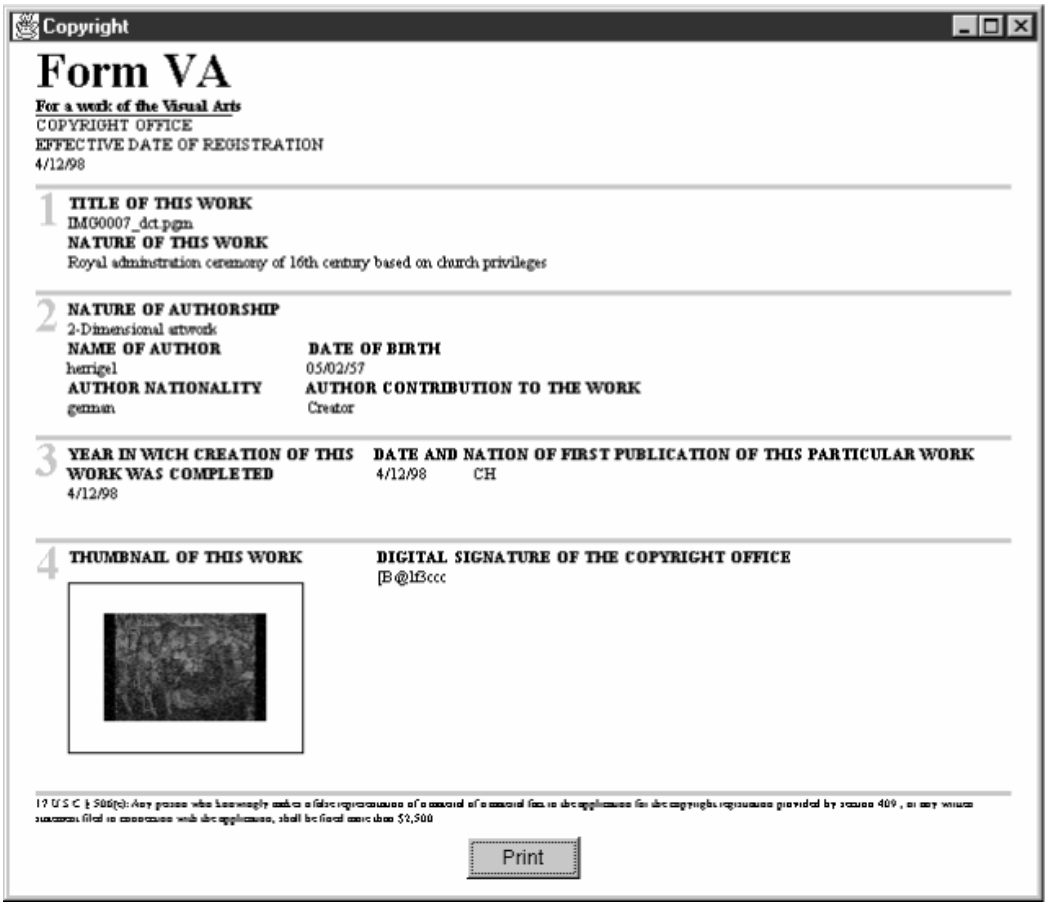


Figure 11: The digital copyright certificate.

Figure 12: The watermark verification process.

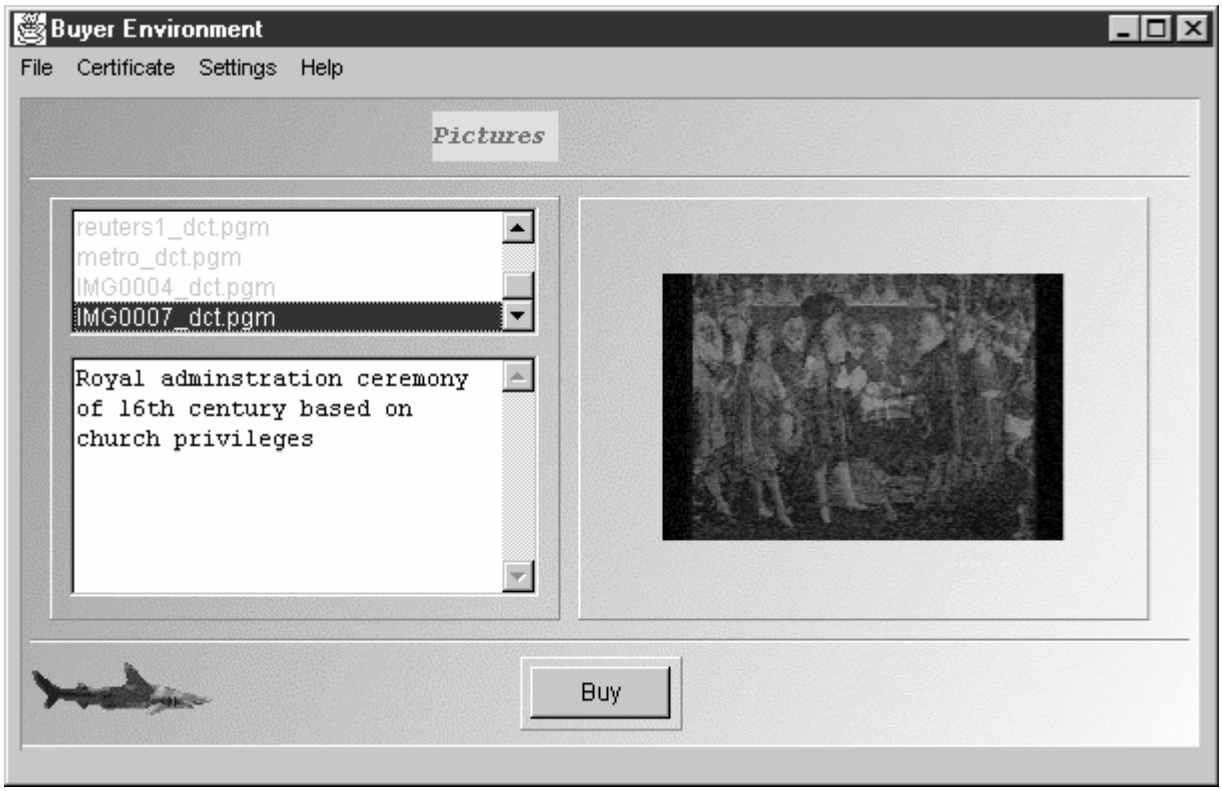


Figure 13: The panel of the Buyer application process.

Robust MPEG Video Watermarking Technologies

Jana Dittmann
GMD - German National Research Center for Information Technology, Institute (IPSI)
Dolivostraße 15,
D-64293 Darmstadt, Germany
+49-6151-869-845
dittmann@darmstadt.gmd.de

Mark Stabenau
GMD - German National Research Center for Information Technology, Institute (IPSI)
Dolivostraße 15,
D-64293 Darmstadt, Germany
+49-6151-869-845
stabenau@darmstadt.gmd.de

Ralf Steinmetz
Darmstadt University of Technology, Industrial Process and System Communications
Merckstr. 25,
D-64283 Darmstadt, Germany
+49-6151-166151
Ralf.Steinmetz@KOM.tu-darmstadt.de

ABSTRACT

The development of new multimedia services and environments requires new concepts both to support the new working process on distributed computers and to protect the multimedia data during the production and the distribution in digital marketplaces. This article addresses copyright protection as a major security demand in digital marketplaces. We propose and compare two watermarking techniques for MPEG video with the intention to show the advantages and the possible weakness in the schemes working in the frequency domain and in the spatial domain. To improve the view to the distortion of the watermarked frames we generate a 3D-difference view measuring the changes which were made during watermarking process and/or caused by several damaging attacks.

KEYWORDS

Security and the media, digital watermarking for MPEG video, copyright protection

1 Motivation

In the A4SM project, a new production environment currently under development at GMD-IPSI, a variety of security features are being developed to increase the users' acceptance. A detailed security risk model can be found in [4]. One major aspect is copyright protection. The video production process leads to a final product based on individual ideas, and the result is an unique intellectual creation. With the digital representation of the video, the procuders run the risk of suffering disadvantages like direct financial loss, legal problems and image loss. Problems include unauthorized taping, reading, manipulating or removing data. Designers, producers and publishers of video or multimedia material are therefore seeking technical solutions to the problems associated with copyright protection of multimedia data.

In this paper we propose and compare two watermarking techniques for MPEG video with the intention to show the advantages and the possible weakness in the schemes working in the frequency domain and in the spatial domain. The main concept of the approaches is to provide environments where digital videos can be signed by authors or producers as their intellectual property to ensure and prove ownership rights on the produced video material during its distribution. The most existing systems are mainly used for still images, [1], [2], [3], [10], or the copyright system SysCoP developed by [8]. The rarely existing video based technologies mostly lack robust continuous signing of all video frames. The following projects are related to digital video copyright protection, the Watermarking DataBlade™ from NEC (1996) or the digital watermarking of MPEG-2 Coded Video in the Bitstream Domain from the University of Erlangen, Germany,[7]. Our work focuses on the implementation and evaluation of robust watermarking technologies for MPEG video with the intention to embed watermarks in every encoded video frame. In the first section we describe two existing watermarking techniques for images and our main intentions to adapt these algorithms to design a video based algorithm based on the existing schemes. We continue with the description of the experimental system, our improvements and tests results. Our tests mainly based on compression, format conversions and StirMark attacks, [8]. Finally, we assess our

achievements so far, and provide an overview of further work.

2 Digital Watermarking

Digital watermarking is the enabling technology to prove of ownership on copyrighted material, detect the originator of illegally made copies, monitor the usage of the copyrighted multimedia data and analyse the spread spectrum of the data over networks and servers. Our goal is to design an algorithm which can be used for all these features and embeds every kind of coded information typically binary coded words.

Basically, watermarks, labels or codes embedded into multimedia data for enforcing a copyright must uniquely identify the data as property of the copyright holder, and must be difficult to be removed, even after various media transformation processes. Thus the goal of a label is to always remain present in the data. Today the existing labelling techniques have different security problems regarding robustness and visual artefacts, [2], [8]. In order to prevent any copyright forgery, misuse or violation, the key to the copyright labelling technique is to provide security and robustness of the embedded label against a variety of threats which include:

- Detecting embedding locations by comparing differently labelled versions of the same original material.
- Finding and altering the embedded label through visual or statistical analysis.
- "The IBM-attack": Instead of introducing a new watermark with an own algorithm and claiming the authorship, a counterfeit original of a watermarked picture is produced by removing a watermark, thus claiming that the original of the real owner contains the watermark which we removed.
- Damaging or removing the embedded label using common multimedia processing.

Mainly we want to address the last point because MPEG compression itself performs multimedia processing like lossy compression or scaling. All necessary transformations on the frames can lead to a distortion of the embedded information and the label cannot be retrieved by the owner. The two presented techniques are adapted versions of algorithms by [8] and [6] for still images previously published. Our work seeks to address MPEG format specifications to provide robust digital watermarking mechanisms for distributed video production systems.

2.1 Requirements For MPEG Video Watermarking

In this paper we address the video part, MPEG standard 1993, [11]. The MPEG compression algorithms employ Discrete Cosine Transform (DCT) coding techniques on image blocks of 8x8, prediction and motion compensation. The resulting output stream contents a sequence of I-, P- and B-frames.

Following requirements are considered important for MPEG video watermarking:

- Robustness against high compression rates of the DCT compression, motion compensation and prediction (very important)
- Robustness against scaling (very important)
- Labelling of every single video frame (I-, P- and B-frames) to provide continuous watermarking and avoid attacks of cutting single frames
- Ensuring correct decoding of the frame sequences without visual artefacts (remark: changes of an I-frame influences the following coding of B- and P-frames)
- Runtime, performance for streaming video or stored video (because today our environment do not need streaming video watermarking we do not regard performance in our first implementations)

Before we are describing the experimental system we describe the two basic algorithm, their advantages and disadvantages and our new implementation strategy:

3 The Zhao Koch Algorithm

The Zhao-Koch algorithm, [8], embeds the copyright label in the frequency domain. Originally, the luminance information Y in the spatial domain is discrete cosine transformed (DCT) into the frequency domain and then quantized. The algorithm pseudo randomly chooses three coefficients from the quantized DCT encoded block and manipulates them to store a single bit information of the copyright label (like binary coded name or address of the owner) using a secret key. For embedding the 1 or the 0 bit Zhao and Koch define different patterns with High, Middle and Low as manipulation rule, see [8]. If storing a bit of information requires a significant change in the coefficients of a block, then the coefficients are manipulated to form an invalid pattern to tell the retrieval there is no information embedded in that block. Generally the invalid pattern requires less of changes in the coefficients than encoding a 0 or 1.

During extraction process, the same coefficients are pseudo randomly selected using the secret key and the relationship between the coefficients are analysed. Depending on the relationship a 0 or 1 is extracted.

The algorithm does not need the original image for retrieval. An advantage is, that the watermark information is embedded in the compressed domain and can be easily applied to MPEG compressed video with minimal operations.

Despite these advantages the algorithm has a few shortcomings: every block is modified and artefacts are common especially in smooth blocks or in sharp edges. The algorithm is not robust against scaling or rotation because the image dimension is used to generate a appropriate pseudo random sequence. **Our goal** is to evaluate the behaviour with MPEG com-

pression and the coding in P- and B-frames. Our improvements address the visual distortion mainly to keep the high quality of the video and to prevent selective attacks on the watermark using efficient error correcting codes. We use smooth block and edge recognition schemes to avoid artefacts.

4 The Fridrich-Algorithm

The method is based on overlaying a pattern with its power concentrated mostly in low frequencies. The pattern is created using a pseudo random number generator and a cellular automaton with voting rules. The robustness of the method has been shown by Fridrich, [6]. The method also overcomes a possible weakness of the method of [2] which can be attacked by usíng the fact that areas of the image which are almost uniform or have an almost constant brightness gradient may show a portion of the watermark pattern. To overcome this weakness, Fridrich uses a watermarking method based on pattern overlaying. Since the pattern will be formed in a sensitive way based on the watermark sequence, even if the watermark pattern shows in uniform areas, it is not possible to mount an attack. The watermark bit sequence is used for initialising a pseudo-random generator to create a random black and white initial pattern of the same size as the image. A cellular automaton with voting rules is applied till a convergence to a fixed point is obtained. The voting rule coalesces random patches into connected areas. The pattern is further filtered by a smoothing filter to move the main portion of the power to low frequencies. The gray levels of the final pattern are scaled to a small range and the pattern is finally added to the image. The watermarked image shows no visible degradation caused by the overlaid pattern, yet the pattern is embedded in a robust sense. It is possible to prove the presence of the pattern in images after filtering, JPEG compression with as low as 5% quality factor, cropping, re-sampling, blurring, down-sampling, and noise adding. The watermark also appears to be resistant with respect to the collusion attack (averaging several watermarked images to remove the watermark).

The first main disadvantage is, that the retrieval process requires the original, un-watermarked image. For videos this is not acceptable because we would need the whole video to prove the watermark. Our algorithm fixes this shortcoming using plain statistical techniques to retrieve the label without the origin.

The second main disadvantage is that the watermarking algorithm embeds only one information: the pattern created using a pseudo-random number generator and a cellular automaton with voting rules. There is no detailed information about the author or producer embedded. The retrieval process provides only true or false if the pattern was retrieved successfully. Our goal is to extend the algorithm in a way that we can embed code words for detailed information like author name or address.

5 Experimental System - MPEG Watermarking

The experimental systems adapt the strength of the embedded watermark to the HVS-properties using two parameters instead of uniformly modulate the luminance values: smoothness and edge character of the block. The edge characteristics are mainly based on the analysis of DCT values so that we can mostly detect vertical and horizontal edges. Smoothness and edge character are the two main parameters. These are concerned in both algorithms and the expected visibility of the watermark can be calculated (as described later), resulting in a value which can be interpreted as the capability of the block to incorporate the watermark without visual distortions. Before the watermarking starts, the MPEG video is traversed with the decoding and single frames (frame data) are produced. The information to be embedded (label data) is encrypted with a secret user key and then embedded into the image data with the same user key used as seed for a pseudo random number generation. The general embedding scheme of both implementations is shown in the next figure:

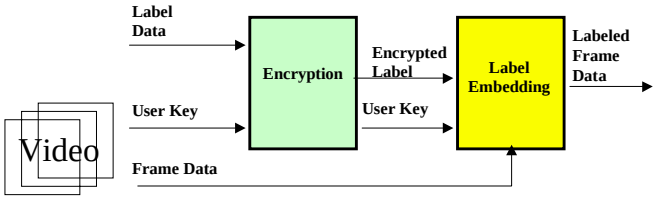


Figure 1: General Embedding Scheme

Before the watermarking starts, the MPEG video is traversed with the decoding and single frames (frame data) are produced. The information to be embedded (label data) is encrypted with a secret user key and then embedded into the image data with the same user key used as seed for a pseudo random number generation. The retrieval is performed with the inverse steps shown in the following figure:

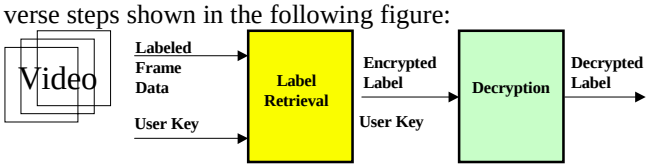


Figure 2: General Retrieval Scheme

Before the retrieval starts, the video is traversed again with the decoder and single frames are produced. In the next chapter we describe the detailed embedding and retrieval steps separated into the two used algorithms.

5.1 Approach I in the DCT Domain

5.1.1 Embedding method

The embedding of the label data is performed in three steps. Originally Zhao and Koch have used only two steps, the first and the third one. We have integrated a second step to improve the visual quality

of the watermarked frame and integrate an error correcting code.

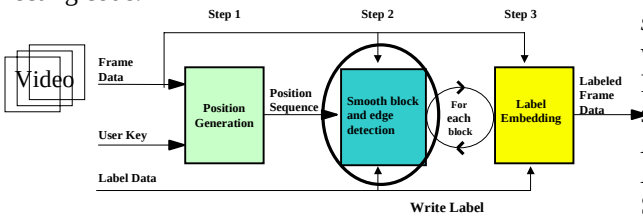


Figure 3: Improved Embedding Scheme

In the first step a position sequence is generated from the user key as a seed with a secure random number generator. This is necessary to hide the watermark in the frame. In the order of the generated position sequence every block is now discrete cosine transformed. The second step consists of the smooth block and edge detection as mentioned earlier. Although sophisticated techniques were developed to check for HVS-characteristics the calculation of the smoothness and the edge character of the block is kept quite simple. This is due to the fact that in future versions this calculation has to be done in the MPEG-stream domain and hopefully in real-time. The parameter *smooth* is simple the number of DCT-coefficients which are not zero after quantization with the quantization matrix Q_m, seen in the following matrix. Thus, high values of smooth indicate many frequency components and therefore a great visual tolerance against additional distortions through the watermark.

low	16	11	10	16	24	40	51	61	
	12	12	14	19	26	58	60	55	
	14	13	16	24	40	57	69	56	
	14	17	22	29	51	87	80	62	
	18	22	37	56	68	109	103	77	
	24	35	55	64	81	104	113	92	
	49	64	78	87	103	121	120	101	
	72	92	95	98	112	100	103	99	High

Unfortunately blocks with edge characteristics often have a lot of frequency components, too. Thus a second parameter *edge* is introduced additionally to reduce artefacts.

The parameter *edge* is calculated as simple as *smooth*: *edge* is the sum of the absolute values of the DCT-coefficients 1, 2, 8, 9, 10, 16, 17 as marked in Q_m, which represents the lower DCT frequencies. High values in these components indicate that the block could have edge characteristics. To determine the level of tolerance against distortions through the watermark caused by each of the two parameters a linear combination is made: *Level = smoothscale*smooth + edgescale*edge + offset*

The parameter *offset* is needed for a base strength of the watermark. The linear combination can now be imagined as a watermark strength indicated by offset and slight variations in strength in dependence of the block characteristics weighted with the parameters *smoothscale* and *edgescale*.

smoothscale = -10, *edgescale= 0.27* and *offset=50* were evaluated through experiments.

Because *Level* can have negative values, *Level* is restricted to values between 0 and 50:

If Level>50 Level=50

If Level<0 Level = 0

So far the level-estimation is independent from the used watermarking algorithm.

To determine the strength of the watermark in dependence of *Level* an additional quantization-factor Q_f is used in the Zhao-Koch algorithm. Every change to DCT-values are made on the originally DCT-value quantized with Q_m/Q_f. Therefore if a change is made to a quantized DCT-value with Q_f=1 this lead to a 4 times higher change than with Q_f=4. Q_f is calculated from *Level* through a table-look-up, because of the not necessarily linear correlation and the small range of Q_f:

Q_f	1	1	2	3	4	4
Level/10	0	1	2	3	4	>4

Before embedding can start an error correcting code is created in the following way: every watermarking information letter is coded into 5 bits first. The resulting bitstream is then (31, 6, 15)-BCH encoded. To ensure improved redundancy every 31 bit word is inserted repeatedly. The parameter bit redundancy determines the amount of redundancy.

In the third step the watermark information with the error corrections and redundancy is embedded as described in the Zhao-Koch algorithm. From each quantized DCT-block three locations in the medium frequencies with absolute values Y1, Y2 and Y3 are chosen, where the bit should be inserted. To encode the bit the three values were changed to one of the following patterns:

Bit	1	1	1	1	0	0	0	0
Y1	H	H	M	M	L	L	M	M
Y2	H	M	H	M	L	M	L	M
Y3	L	L	L	L	H	H	H	H

If the changes to embed the Bit are too big, so that visual distortions are common, Y1, Y2 and Y3 are changed to an invalid pattern (H,L,M; L,H,M ; M,M,M). After the changes are made the block is requantized and inverse discrete cosine transformed.

5.1.2 Retrieval Method

The retrieval is performed in the same way of the Zhao and Koch algorithm. We decode the single frame of the video and perform the inverse steps of the embedding: first the position generation and the retrieval of the embedded data.

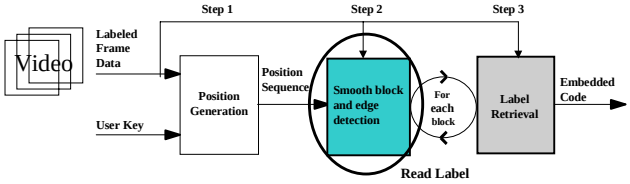


Figure 4: Improved Retrieval Scheme

The first two steps are exactly as in the embedding process. Step 2 is not essential but the information about the strength of the watermark in each block is helpful while using the described error correcting and redundancy code.
In the third step the same three locations from every block must be examined like the ones used in the embedding process. Then the patterns could be checked and the watermark bit could be read out.

5.1.3 Experimental Results

We have tested our adapted implementation of the Zhao-Koch algorithm with different MPEG-Videos. One example should demonstrate the capabilities and the shortcomings of the algorithm. The results of the first 15 frames of the video are shown. The table contains the error rates after MPEG-reencoding, format conversion to QuickTime and after the watermark removing program StirMark is used [9]. StirMark combines various attacks. It simulates distortions caused by a printing and rescanning process. Furthermore it introduces some minor geometric distortions like stretching, shearing, rotations and shifting. It is reported that StirMark is very effective against most even commercial watermarking techniques. However, the distortions introduced by StirMark are unrecognisable.
The chosen video museum.mpg is about a virtual museum. A camera leads from the entrance from the museum through several rooms. The first 30 frames show a short zoom from the entrance of the museum. For a better view the HTML version of the paper can be found in:
www.darmstadt.gmd.de/mobile/watermarking with the images and video sources used in this paper.

Figure 5: Original museum (first frame)

Video characteristics:	Museum.mpg
No. of 8x8 blocks	1320
Compression I-Frame	3.42%
Compression P-Frame	2.65%
Compression B-Frame	1.22%
IPB-order	IBBPBB

Compression-rates are calculated from uncompressed 24 bit images and the algorithm uses the following parameters: smoothscale = -10, edgescale = 0.43 and offset = 40, watermark strength: 2.0 and bit redundancy 4. Conversion table from Level to Quantization factor Q_f:

Q_f	1	1	2	3	4	4
Level/10	0	1	2	3	4	>4

These parameters leads to a watermark strength which cause slight artefacts in uniform regions. To improve the view to the distortion chapter 6.2.3 offers a 3D-difference view measuring the changes. In summary the strength of the algorithm can be assumed as high. The visual quality improvements can be evaluated at www.darmstadt.gmd.de/mobile/media/-watermarking in the HTML version of this paper. The original Zhao algorithm Koch with the default settings produces several artefacts in the smooth regions and throughout brighter frames.

Figure 6: Watermarked museum (first frame)

Our robust test results are dispayed in the next table. We embedded 60 Bits of watermarking information. We performed MPEG-encoding, Quicktime transformation and StirMark-Attack and got following results with the improved Zhao-Koch-algorithm. The table measures the bit errors after the performed transformations with the error correcting code. The numbers show the amount of bit-errors occurred in the first 13 frames after high MPEG compression, QuickTime conversion and Stirmark attack.

Frame No.	**0**	**1**	**2**	**3**	**4**	**5**	**6**
F-Type	I	B	B	P	B	B	I
MPEG BCH	0	0	0	0	0	0	0
QuickTime BCH	0	1	9	1	0	0	0
StirMark BCH	22	29	29	16	17	19	24
Frame No.	**7**	**8**	**9**	**10**	**11**	**12**	**13**
F-Type	B	B	P	B	B	I	B
MPEG BCH	0	0	0	0	0	0	0
QuickTime BCH	0	0	0	1	0	0	5
StirMark BCH	22						

Table 1: Absolut Errors

The following bit error rates can be measured all together of our experiments (about 10 video streams):
museum.mpg:
MPEG: I-Frames <1% P-Frames 1-2% B-Frames 5% QuickTime: I-Frame 1-2% P-Frames 5% B-Frames 7% StirMark: 32%

Table 2: Error Rates

Regarding the error rate table we want to discuss our results: the first apparent thing is that the algorithm shows very good results in MPEG compression and Quicktime conversions. The error rates of the watermark information in I-Frame is excellent. B-Frames have still some problems. StirMark removes the watermark up to 30 percent, without the error correcting code the watermark was destroyed completely. Stirmark destroys the watermark in the original Zhao-Koch completely. Visual artefacts can be avoided.

5.2 Approach II In The Spatial Domain

The proposed watermarking technique of Fridrich is modified in the following way to overcome the two main shortcomings: retrieval without the origin and embedding of binary coded labels. To ensure the last requirement instead of one overlaying pattern over the whole frame we add a 8x8 pattern over every 8x8 Block of the frame. To embed binary code words we define additional modification rules of the overlaying 8x8 pattern described in the following embedding strategy. Furthermore we use statistical properties to find the label in the retrieval procedure without the original frame.

5.2.1 Embedding method

First of all a position sequence like the one used in the Zhao-Koch algorithm is generated to determine the blocks we want to modify. The next figure illustrates the embedding steps. For each block a user key dependent pattern is made in the following manner: We start by creating a 8x8 pseudo random pattern with the user key as a seed, step 1. To eliminate the high frequencies in this pattern a cellular automaton with simple voting rules is used. Every position in the 8x8 random pattern is tested on the number of '1' in the eight co-sited positions. If the number exceeds five the actual position is set to '1' too, if the number is less than 3 it is set to '0', see the marked rectangle for an example. By applying these rules several times on the whole 8x8 block we obtain a pattern M with less high frequencies, steps 2 - 4. Now a correlation between the pattern and the luminance block has to be inserted according to the bit we want to embed, step 5. If we want to embed a '1' we add a value k, which is calculated in the smooth/edge block estimation routine via a table look up from *Level*, in each luminance block position where the corresponding position in the pattern M is '1' and we subtract the value k if the corresponding position is '0'. If we want to embed a '0' we do it vice versa.

Due to the fact that we use much smaller patterns (8x8) than Fridrich (one pattern for the whole frame), we can embed much more information. The disadvantage of this technique is that we have to calculate with high bit errors in the detection process. We can overcome this shortcoming by applying an error-correcting code (we use a (31, 6, 15)-BCH code) and an additional redundancy code on the watermark information before we start the embedding process.

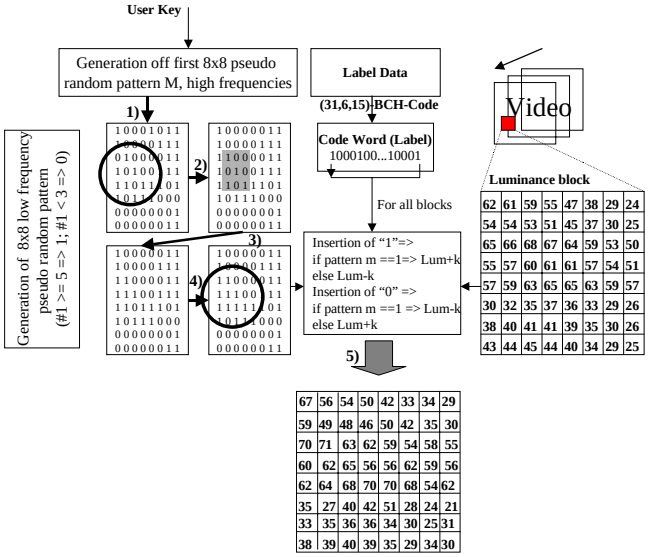


Figure 7: Embedding Process

5.2.2 Retrieval Method

In the retrieval process seen in the next figure the same 8x8 patterns M have to be generated as in the embedding process, step 1 - 4. To test the correlation between the luminance block and the pattern M the average luminance value *av1* (sum1 div #1) of positions with a corresponding '1' and the average luminance value *av0* (sum0 div #0) with a corresponding '0' in the pattern M is produced. If the luminance block and the pattern M would be uncorrelated the difference of both values should be near zero. But due to the embedding process one of these values should be significantly higher (around 2*k) than the other. Thus we estimate an embedded bit '1' if *av1>av0*. Otherwise we estimate an embedded bit '0'. With this statistical analysis we avoid using the original frames. If all bits are retrieved the watermark information is decoded with the same (31,6,15) BCH-Code and the additional redundancy code. The retrieval process is shown in the following diagram:

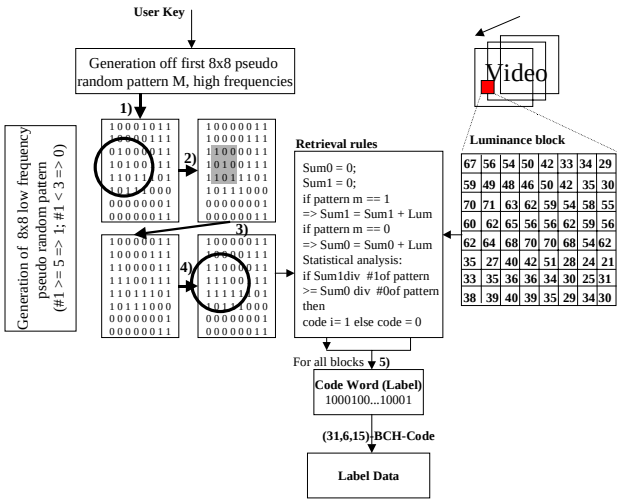


Figure 8: Retrieval Process

5.2.3 Experimental Results

We have tested the improved Fridrich-algorithm with the following parameters: smoothscale = -12, edgescale = 0.27 and offset = 50, watermarking strength 1.1 and bit redundancy 4. Conversion table from Level to k (see description of embedding method):

k	12	12	6	4	3	3
Level/10	0	1	2	3	4	>4

We chose the same video sequences of the museum.mpg. The first watermarked frame can be seen in the following picture. It is hard to evaluate the distortions. A better view can be seen in the 3D-difference view in chapter 6.2.3 or in the HTML version of this paper at www.darmstadt.gmd.de/mobile/media/watermarking. Obviously visual artefacts could be avoided

Figure 9: Watermarked museum (first frame)

We embedded 60 Bits of copyrigth information. We performed the same transformations: MPEG-encoding, Quicktime transformation and StirMark-Attack and got following results with the improved Fridrich-algorithm. The table measures the amount of bit errors after the performed transformations with BCH code and the additional redundancy code.

Frame No.	0	1	2	3	4	5	6
F-Type	I	B	B	P	B	B	I
MPEG BCH	0	0	7	6	0	4	0
QuickTime BCH	0	12	16	11	23	15	5
StirMark BCH	19	27	24	23	26	27	24

Frame No.	7	8	9	10	11	12	13
F-Type	B	B	P	B	B	I	B
MPEG BCH	0	2	0	2	8	0	5
QuickTime BCH	15	18	4	26	33	7	18
StirMark BCH	23						

Table 3: Absolute errors

The following error rates can be measured all together of our experiments (about 10 video streams):

museum.mpg:
MPEG:
I-Frames 1-2% P-Frames 3% B-Frames 7%
QuickTime:
I-Frame 8% P-Frames 15% B-Frames 35%
StirMark: 31%

Table 4: Error rates

Regarding the error rate table we want to discuss our results:

If only I- Frames would be watermarked the error rate after MPEG-encoding are also promising. B- and P-frames are not sufficient watermarked. Compared to our adapted Zhao-Koch implementations the algorithm is less successfully with the used error correction. If the watermark should be robust against QuickTime conversion though, the strength of the watermark must be increased by changing the value k or the parameters of the smooth block and edge detection part. As with the Zhao-Koch algorithm the watermark has error rates up to 30 percent after the StirMark attack. But the advantage of the algorithm is, if we embed only 10 information bits with a higher redundancy we get low error rates about 5 percent. Additionally the algorithm is more flexible to handle StirMark attacks more efficiently and robustly.

Now we want to discuss the visual artefacts in detail. To get a more descriptive view on the distortions introduced in the different steps, we measure the differences of the changes to the original frame. The idea is to transform the difference into a 3D scene, [5]. The absolute difference between the original frame and the watermarked frame, the watermarked areas and the intensity of the watermark, measured by the height of the 3D relief can be seen. Based on these information the quality loss can be measured. It can be seen, if relevant image objects are watermarked and the robustness can be evaluated, regarding the intensity, and different algorithms can be compared. We have created the following 3D-scenes:

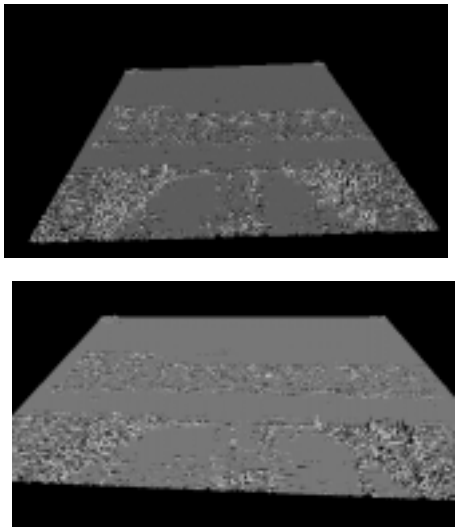


Figure 10: a) Watermarked, Fridrich, b) Watermarked, Zhao-Koch

In the above pictures you see the differences between the original frames and the watermarked versions. The different strengths of the watermark in dependence to the smooth and edge characteristics of the picture can be seen very good. Apparently both algorithms introduce similar changes. This is because of the fact, that both use the same smooth and edge detection algorithm and both introduce changes in the medium frequencies.

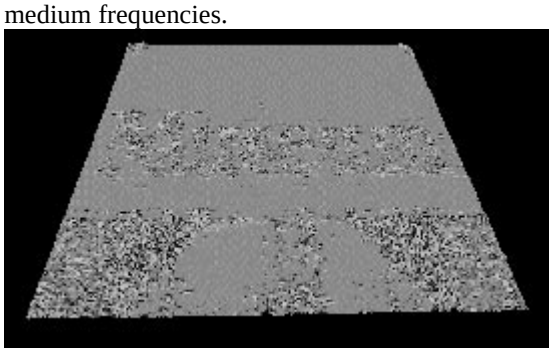


Figure 11: MPEG- reencoded Fridrich

The figure 11 describe the difference of the watermarked-re-encoded frames to the original ones. The difference to the upper two pictures are quite small but leads already to the observed error rates of the Fridrichs approach.

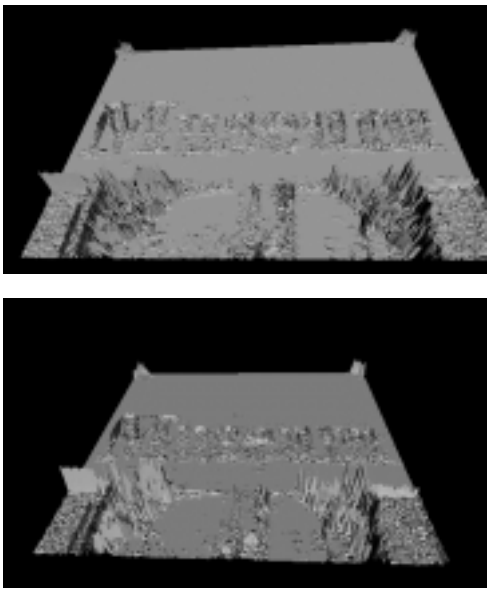


Figure 12: a) StirMark, Fridrich, b) StirMark, Zhao-Koch

The 3D-scenes of figure 12 show the StirMark distortions. Although the distortions seems to be very high they are invisible to the observer when only looking at the "StirMarked" frame. This is due to the fact that the biggest distortions are introduced through slight geometric transformations which are difficult to detect without the original frame. Nevertheless they are not invisible to the watermark detection algorithms as can be seen in the appropriate error rates.

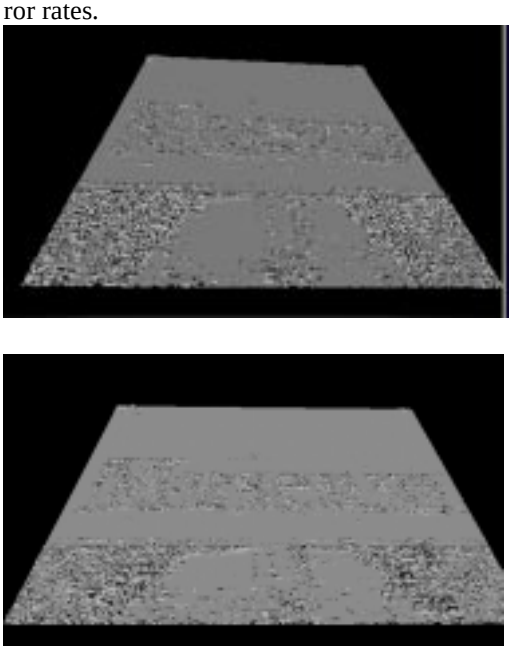


Figure 13: a) QuickTime conversion, Fridrich, b) QuickTime conversion, Zhao-Koch

The last two pictures off figure 13 show the changes to the watermarked frames due to the QuickTime conversion. The distortions of the Zhao-Koch are lower and verify the measured error rates.

5.3 Problems In The Experimental Systems

Is there a way to deal with attacks of the kind StirMark uses? The Fridrich-algorithm offers a nice possibility to handle geometric distortions and clipping. Before checking for correlations the pattern can be geometric transformed in the way the attack is expected. Then the correlation is tested. If the correlation increases the transformation was in the right way and another transformation could be applied. Although this process could be very time consuming it doesn't necessarily need to be applied to each frame, because the attacker would have to apply the same time consuming process. Against clipping the whole pattern consisting of all 8x8 patterns could be shifted over the clipped frame. Again the correlation would be tested and a high value would indicate that the correct part of the pattern matches the clipped frame.

6 Applicability for Object Watermarking

Object Watermarking is one major demand in the MPEG-4 standard to label separate objects of the video or video planes. Both algorithms were analysed if they are useable for inserting and retrieval of labels into regions of video instead of labelling the whole video frames. Our studies are based on traditional MPEG-1 and MPEG-2 videos with an edge detection algorithm based on the Canny algorithm, [5].

The main problem is to retrieve the correct watermarked regions of the whole frame without the knowledge of the position of the objects. Our first approach assumes a minimal region of 64x64 pixel blocks which is watermarked.

The watermarking with the second approach in the spatial domain is simple: a multiple 64x64 pixel pattern which depends on the user key is overlayed across the region which should be watermarked. The retrieval searches for a correlation of the 64x64 patterns in the actual video frame. If the correlation threshold is found an object can be identified. Our experiments have shown that the watermarking strength must be very high to differ from similar primary correlation in other regions of the video frame which were not watermarked. Therefore this practice causes substantial artefacts in the watermarked regions and can be found very easy for an attacker. For tests we have watermarked three regions in the background. The results can be seen in the HTMLversion.

The first approach in the frequency domain provides better results. We embedded an binary sequence of alternating 0 and 1 in 8x8 blocks. The retrieval searches for this alternating sequence in every 64x64 block. The amount of matches of the 0-1 sequence is measured. The visual distortions are less then with the approach II and can be evaluated in the HTML version.

Figure 14: Frame with three watermarked regions approach I

7 Conclusions

In this paper we have discussed MPEG video watermarking techniques, their possibilities and disadvantages to ensure copyright protection. Our robustness tests are mainly based on compression, format conversions and geometrical transformations. We pointed out that our adapted Zhao Koch approach with the error correcting code is appropriate for MPEG video and the visual quality of the watermarked frames could be improved. Format conversions are handled with very low error rates. Especially there are high error rates after StirMark attacks. The attacks are very difficult to handle. The Fridrich approach could also be adapted and improved successfully to embed binary code words and perform a retrieval without the original. The robustness tests are satisfying, but there are still some problems with B- and P-frames after MPEG compression and format conversions. The advantage of the Fridrichs algorithm is to handle StirMark attacks. Therefore our future work is focused on improvements of the Fridrich algorithm to withstand StirMark attacks more efficient. In parallel we ensure multiple watermarking, object watermarking and improve the runtime behaviour by embedding the information into compressed video. The watermarking pattern of the second approach will be first DCT transformed before it is added to the DCT coefficients directly. Afterwards a drift compensation must be performed.

The 3D scenes are very useful to evaluate the visual artefacts and the distortions after several attacks. Our goal is to integrate the watermarking techniques in our distributed video production and distribution environment as an enabling technology for electronic commerce and for digital market places to ensure copyrights.

8 References

[1] Benham D., Memon N., Yeo B.-L., Yeung M.: Fast Watermarking of DCT-based Compressed Images, CISST '97 International Conference

[2] Cox, I.J., Miller, M.L.: A review of watermarking and the importance of perceptual modeling, Proc. Of Electronic Imaging’97, February 1997

[3] Digimarc: Watermarking Technology, PictureMarc™ 1996, http://www.digimarc.com/wt_page.html

[4] Dittmann, J., Steinmetz, A.: Konzeption von Sicherheitsmechanismen für das Projekt DiVidEd, GMD-Studie ’97

[5] Dittmann, J., Steinmetz, A., Nack, F., Steinmetz, R.: Interactive Watermarking Environments, to appear in IEEE Multimedia 1998, Austin Texas

[6] Fridrich, J. :Methods for data hidung, Center for Intelligent Systems & Department of Systems Science and Industrial Engineering, SUNY Binghamton, Methods for Data Hiding", working paper (1997)

[7] Hartung, F., Girod, B.: Copyright Protection in Video Delivery Networks by Watermarking of Pre-Compressed Video, in: S. Fdida, M. Morganti (eds.), "Multimedia [7] Applications, Services and Techniques - ECMAST '97", Springer Lecture Notes in Computer Science, Vol. 1242, pp.423-436, Springer, Heidelberg, 1997

[8] Koch, E. and Zhao, J.: Towards Robust and Hidden Image Copyright Labelling, Proc. of 1995 IEEE Workshop on Nonlinear Signal and Image Processing (Neos Marmaras, Greece, Junu 20-22, 1995)

[9] Kuhn, M.G.: Stirmark, available at http://www.cl.cam.ac.uk/mgk25/strirmark/, Security Group, Computer Lab, Cambridge University, UK (email: mkuhn@acm.org)

[10] Kutter, M., Jordan,F. and Bossen,F.: Digital Signature of Colour Images using Amplitude Modulation, Signal Processing Laboratory, EPFL, Switzerland, 1995

[11] MPEG Internationaler Standard ISO/IEC 11172: Information Technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s, Part1: Systems, Part2: Video, Part3: Audio, 1993

Watermarking: Who Cares? Does it Work?

Edward J. Delp
Purdue University
School of Electrical and Computer Engineering
Purdue Multimedia Testbed
Video and Image Processing Laboratory (*VIPER)*
West Lafayette, IN 47907-1285
+1 765 494 1740
+1 765 494 0880 (fax)

email: ace@ecn.purdue.edu

http://www.ece.purdue.edu/~ace

Watermarking: Who Cares? Does it Work?

Edward J. Delp

Purdue University
School of Electrical and Computer Engineering
Purdue Multimedia Testbed
Video and Image Processing Laboratory (*VIPER)*
West Lafayette, IN 47907-1285
+1 765 494 1740
+1 765 494 0880 (fax)
email: ace@ecn.purdue.edu
http://www.ece.purdue.edu/~ace

Outline

- **Overview of the multimedia security problem**
- **The watermarking problem**
- **Will it work?**

Multimedia Security Applications

- **Privacy**
- **Forgery Detection**
- **Copyright Protection**
- **Proof of Purchase and Delivery**
- **Audit Trails**
- **Intruder Detection**
- **Network Security**
- **Being anonymous**

Security should not be noticed by user

Multimedia Security - Tool Set

- **Encryption**
- **Authentication**
- **Hashing**
- **Time-stamping**

Multimedia Applications

- **Many approaches insert controlled amount of distortion into a multimedia element - "watermarking"**
- **Audio**
- **Video**
- **Documents (including HTML documents)**
- **Images**
- **Graphics**
- **Models (e.g., MPEG4)**
- **Programs (executable code)**
- **Data hiding**

Why is Digital Watermarking Important?

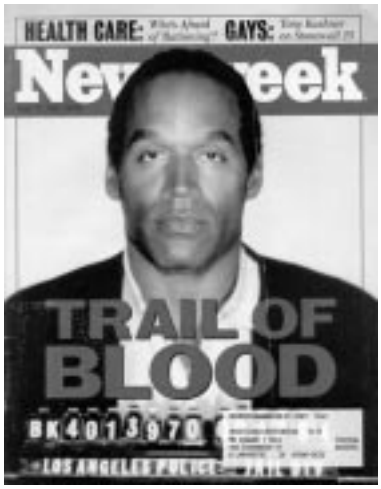
- **Scenario**
  - **an owner places digital images on a network server and wants to detect and track redistribution and/or altered versions**
- **Goals**
  - **verify the owner of a digital image**
  - **detect forgeries of an original image**
  - **identify illegal copies of the image**
  - **prevent unauthorized distribution**

Why Watermarking is Important?

Why is Watermarking Important?

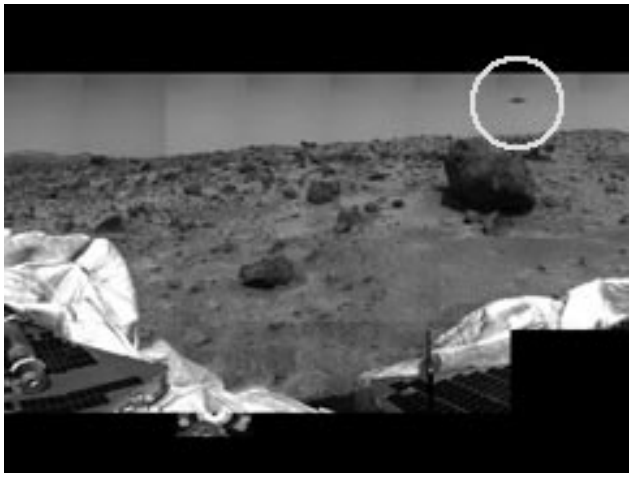
Why is Watermarking Important?

Why is Watermarking Important?



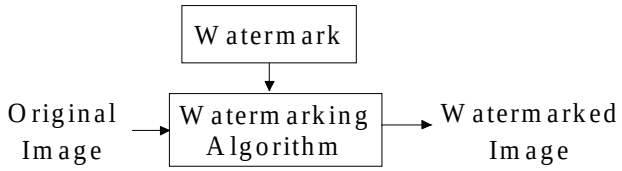
Why is Watermarking Important?

Why is Watermarking Important?



A Review of Watermarking Techniques

- **Spatial watermarking**
- **Sub-band (wavelet) watermarking**
- **DCT coefficient modulation**
- **Visible watermarks**



Components of a Watermarking Technique

- **The watermark, W**
  - **each owner has a unique watermark**
- **The marking algorithm**
  - **incorporates the watermark into the image**
- **Verification algorithm**
  - **an authentication procedure (determines the integrity / ownership of the image)**

Main Principles

- **Transparency - the watermark is not visible in the image under typical viewing conditions**
- **Robustness to attacks - the watermark can still be detected after the image has undergone linear and/or nonlinear operations (this may *not* be a good property - *fragile watermarks*)**
- **Capacity - the technique is capable of allowing multiple watermarks to be inserted into the image with each watermark being independently verifiable**

Attacks

- **Compression**
- **Filtering**
- **Printing and rescanning**
- **Geometric attacks - cropping, resampling, rotation**
- **Collusion - spatial and temporal**
- **Re-watermarking**
- **Conversion to analog signal**

Original Image

Watermarked Image

Watermark

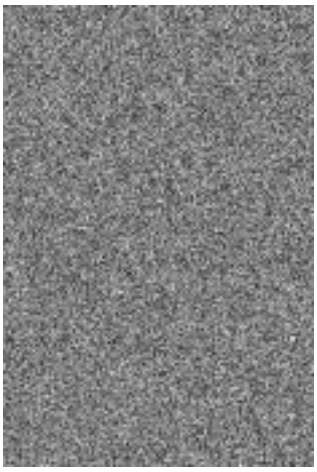
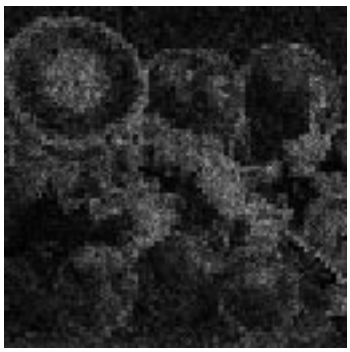


Image Adaptive Watermarks



Video Watermarking

- **Could still image approaches**
  - **may have problem in MPEG with B and P frames**
- **Techniques could be used to prevent multiple viewing, copying, and editing (e. g. inserts)**
- **Can the watermark survive the conversion back to an analog NTSC or PAL signal?**

Watermarking: Who Cares and Will it Work?

- **Watermarking technologies have not been tested in court**
  - **what constitutes a "derived work" from a digital image?**
- **Might one be better off just doing timestamping and/or other forms of authentication**
- **Image encryption will have very limited use BUT encryption techniques need to be used more, particularly timestamping**

Is watermarking the "feel good" technology of multimedia?

Ownership

- **Copyright Ownership - Who did it first? ⇒ timestamp (prevents re-watermarking attacks)**

Forgery Detection

Forgery Detection

- **medical images**
- **forensic images**
- **digital cameras**

It is better to hash the image and time-stamp

Unauthorized Distribution and Illegal Copies

- **Unauthorized Distribution**
  - **You took my image from my web site!**
  - **You are selling my image from the CD-ROM you bought from me!**
- **Who owns it? ⇒ hash and timestamp**
- **Is your image the same as mine? (derived work)**
- **Perhaps the transport mechanism could prevent this**

Unauthorized Distribution and Illegal Copies

- **Unauthorized Distribution**
  - **You took my image from my web site!**
  - **You are selling my image from the CD-ROM you bought from me!**
- **Who owns it? ⇒ hash and timestamp**
- **Is your image the same as mine? (derived work)**
- **Perhaps the transport mechanism could prevent this**

Legal Issues

- **When one says: "My watermarking withstands the X attack!"**
  - **What does it mean? (Has the watermark been damaged?)**
  - **It is legally defensible?**
  - **Nearly all watermarks require statistical tests for verification**

Conclusions

- **Watermarking is an extremely interesting but untested technology!**
- **Much has promised…will it be delivered?**
- **Time stamping (and hashing) is important!**

Watermarking Conference

Conference on Security and Watermarking of Multimedia Contents **(January 23-29, 1999) in San Jose**

http://www.spie.org/web/meetings/calls/pw99/confs/ei25.html