IST-2002-507932

ECRYPT

European Network of Excellence in Cryptology

Network of Excellence

Information Society Technologies

D.WVL.5
First Summary Report on Hybrid Systems

Due date of deliverable: 31. January 2005
Actual submission date: 31. January 2005

Start date of project: 1 February 2004 Duration: 4 years

Lead contractor: Otto-von-Guericke Universität Magdeburg (GAUSS)

Revision 1.0

Project co-funded by the European Commission within the 6th Framework Programme		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission services)	
RE	Restricted to a group specified by the consortium (including the Commission services)	
CO	Confidential, only for members of the consortium (including the Commission services)	

First Summary Report on Hybrid Systems

Editor
Stefan Katzenbeisser (GAUSS)

Contributors
Jordi Herrera-Joancomartí (UOC)
Stefan Katzenbeisser (GAUSS)
David Megías (UOC)
Julià Minguillón (UOC)
Andreas Pommer (GAUSS)
Martin Steinebach (FHG)
Andreas Uhl (GAUSS)

31. January 2005
Revision 1.0

The work described in this report has in part been supported by the Commission of the European Communities through the IST program under contract IST-2002-507932.

Abstract

In this report we give an overview of three specific problems in the intersection between cryptography and watermarking. In the first chapter we discuss the security of dispute resolving protocols, which were one of the first applications of watermarking technology; we will conclude that the security of this application depends heavily on the properties of the underlying watermark detector. In the second chapter we discuss the feasibility of watermarking encrypted digital images in such a way that the decrypted document carries a watermark. The existence of such a watermarking scheme that commutes with an encryption function allows to construct secure Digital Rights Management (DRM) systems and asymmetric fingerprinting schemes. Finally, we give an overview of selective (partial) encryption in the last chapter. Partial encryption schemes allow to encrypt media data in a format-compliant way.

Contents

Chapter 1

Dispute Resolving Protocols

Digital watermarks were proposed as main building block of dispute resolving schemes. Dispute resolving tries to sort out ownership disputes over digital objects by embedding ownership information as a watermark.

1.1 Application scenario and requirements

In general, the process of delivering multimedia services to customers can be broken down to the scenario depicted in Figure 1.1. A content provider $\mathcal{P}$ (who owns the copyright on multimedia objects) forwards these objects to a seller $\mathcal{S}$ who in turn sells them to n different customers $\mathcal{C}_1, \ldots, \mathcal{C}_n$. Examples for this kind of application include electronic-commerce systems (using the Internet as transfer medium), pay-per-view digital television or information systems based on multimedia databases.

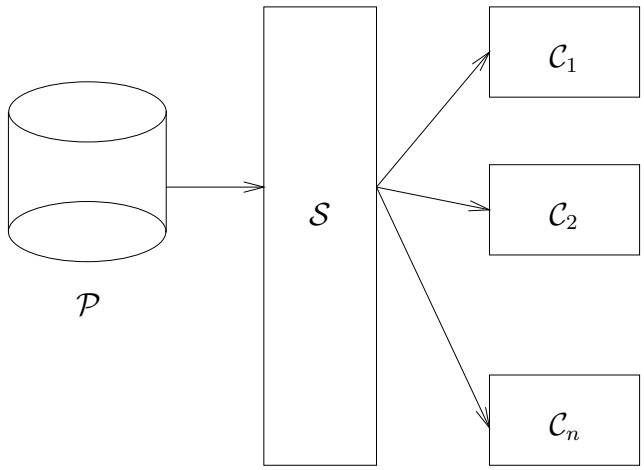

Figure 1.1: Application scenario for multimedia distribution containing a content provider $\mathcal{P}$, a seller $\mathcal{S}$ and n customers $\mathcal{C}_1, \ldots, \mathcal{C}_n$.

The main goal of copyright protection is to construct a framework that is able to resolve

the copyright situation after an infringement occurred. In particular, a copyright protection protocol allows to identify the copyright holder ($\mathcal{P}$) of a multimedia object distributed illegally on the Internet; optionally, also the buyer ($\mathcal{B}_i$) of the illegally copied object can be traced. For unambiguous resolving, at least the following two requirements must be satisfied:

- Assume that an object O is property of $\mathcal{P}$. In case a dispute arises over an object O_D that is *derived* from O, the dispute resolving process should be able to identify the copyright holder $\mathcal{P}$. We will argue later that the copyright holder must be a protocol participant in the resolving process in order to fulfill this requirement.
- In the same scenario, the copyright protection scheme should uniquely identify the customer $\mathcal{C}_i$ who bought the object O that was used to create O_D.

These requirements naturally translate into the following security properties:

- **Resolvability.** The dispute resolving mechanism should uniquely identify the copyright holder of the multimedia object. It should not be possible by an attacker to fake an ownership proof (that either identifies the attacker or any third party as copyright holder).
- **Robustness.** The mechanism should "work" also with highly modified objects. In other words, it should not be possible to deliberately modify a multimedia object such that the resolving process fails.
- **Fairness.** The mechanism should not falsely identify innocent customers as copyright infringers.
- **Nonrepudiation.** Copyright infringers should not be able to plausibly deny that they illegally copied material.

1.2 Early attempts to copyright protection

Early papers on watermarking such as [121, 65] claimed the above problems as solved, once a sufficiently robust watermarking scheme is found. In early copyright protection protocols, the watermark was seen as a transfer medium that stores information about the copyright holder and the customer who bought a specific object.

More specifically, the generic construction principle of [121, 65] is as follows. The content provider $\mathcal{P}$ adds his watermark W_C to all objects he forwards to the seller $\mathcal{S}$, who in turn sells them to a customer $\mathcal{C}_i$. Before delivering the object to $\mathcal{C}_i$, $\mathcal{S}$ adds another watermark W_S whose payload identifies $\mathcal{C}_i$. Due to the robust nature of the watermarks, both embedded marks will be detectable. Call the watermarked object $\overline{O}$. The protocol implicitly assumes that the copyright holder (and thus also the content provider) keeps the unmarked original objects locked away. Once an illegal copy of $\overline{O}$ is found, the content provider extracts both watermarks W_C and W_S. The former will be used to prove to a third party that the content provider actually owns the copyright on the object in question, whereas the latter watermark identifies the traitor.

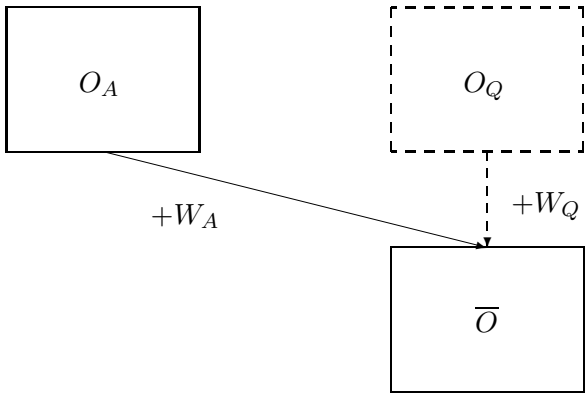

Figure 1.2: Inversion attack: Bob comes up with a fake watermark W_B and a fake original O_B and claims O_B to be the true original.

Unfortunately, this generic construction fails for several reasons:

- **Invertibility.** In case the watermarking scheme is susceptible to an inversion attack [31], inversion can be used to subvert the goal of *resolvability*. In fact, a different person $\mathcal{Q}$ could use an inversion attack to compute a detectable mark W_Q in $\overline{O}$ and challenge the true owner $\mathcal{P}$. By the construction of an invertibility attack (see Figure 1.2), no third person can decide whether the claims of $\mathcal{P}$ or $\mathcal{Q}$ are correct, as $\mathcal{P}$'s claimed "original" contains $\mathcal{Q}$'s watermark and vice versa.

  In order to prevent this attack, several authors proposed the use of non-invertible watermarks in the generic construction. However, the construction of provably-secure watermarks seems to be difficult and is—up to now—an open problem [3]. Alternative methods use central time-stamping services or a trusted registration authority [5, 4].

- **Other protocol attacks.** An attacker could use other protocol attacks; for example using a copy attack, he is able to claim that an object is courtesy of a different person by transferring a watermark between two objects [57]. Again, this contradicts the *resolvability* property discussed above.

- **Robustness of watermarks.** If it is possible to remove watermarks from a multimedia object without visual degradation, an attacker can hinder the resolving process by removing W_C and W_S. Thus, the system may violate the *robustness* requirement.

- **Public versus private information.** When the content provider is asked to publicly prove the ownership of his works (e.g., in front of a judge), he has to reveal the watermark and private key. Since in most watermarking schemes the key is coupled with the location of the watermark in the digital media, it is then possible to remove the mark once the key is public. There are several approaches to solve the problem, such as the use of zero-knowledge and asymmetric watermarking schemes [2, 50] or special purpose tamper-resistant hardware [100, 101]. Another alternative is the use of different watermarks W_C in each sold copy, such that the publication of one watermark/key pair allows an attacker only to remove a watermark in one single copy of the digital work without putting all other copies at risk.

- **Customer's rights problem.** The protocol implicitly assumes that the seller $\mathcal{S}$ is trustworthy. Specifically, it is assumed that the seller sticks to the protocol and correctly embeds W_S. However, this cannot be guaranteed in general. A cheating seller might deviate from the protocol specification and embed the identity of a different, innocent customer instead of W_S, who will subsequently be blamed for all copyright infringements (see Section 1.3). This violates the *fairness* requirement.

1.3 Customer's rights problem

Most copyright protection protocols are designed entirely to protect the copyright holder of digital objects; the implicit assumption of those proposals is that the seller behaves trustworthy, whereas the customers do not. Qiao and Nahrstedt [79] first noted that this asymmetry does not hold in general. More specifically, a cheating seller can fake fingerprints or illegally distribute fingerprinted objects himself. The following two scenarios illustrate these problems:

- **Scenario A.** Suppose that a seller $\mathcal{S}$ sells (by mistake) the same fingerprinted work $\overline{O}$ to two customers $\mathcal{A}$ and $\mathcal{B}$. Suppose further that $\mathcal{S}$ fingerprinted $\overline{O}$ with the identity of $\mathcal{A}$. Now, one of the customers forwards the content illegally to a party $\mathcal{C}$. Based on the fingerprint, $\mathcal{S}$ will identify $\mathcal{A}$ as the traitor. However, from the setup it is unclear whether $\mathcal{A}$ or $\mathcal{B}$ is the traitor. Even if $\mathcal{S}$ did not make any mistake by selling any object twice, customers can always claim this and therefore plausibly deny their delinquency.
- **Scenario B.** Suppose a copyright holder $\mathcal{P}$ forwards his objects to a seller $\mathcal{S}$, who is not trustworthy. Now, $\mathcal{S}$ sells the object to customer $\mathcal{C}_1$; the sold object $\overline{O}$ will carry a fingerprint identifying $\mathcal{C}_1$. Now, $\mathcal{S}$ makes $n-1$ illegal copies of $\overline{O}$ and sells them to customers $\mathcal{C}_2, \ldots, \mathcal{C}_n$ without records. Clearly, $\mathcal{C}_1$ gets blamed for making illegal copies, although the seller is the traitor in this case.

The problem stems from the fact that the seller $\mathcal{S}$ has access to the fingerprinted works $\overline{O}$; he is therefore able to distribute $\overline{O}$ without conforming to any copyright protection protocol. Qiao and Nahrstedt [79] proposed a partial solution to this problem by "binding" the fingerprinted object to an identification token produced by a legal customer. Unfortunately, this proposal does not solve all of the above problems, as the seller still has access to $\overline{O}$. Memon and Wong [61, 62] presented a solution by making the watermark insertion process an interactive protocol between the buyer and the seller. The protocol is constructed in such a way that the seller does not get to know the fingerprinted copy of the work, and the customer has no complete knowledge of the embedded watermark. In this setup, a cheating seller cannot make illegal copies of the fingerprinted work, whereas customers cannot remove fingerprints, as they have no knowledge of the embedded watermarks. Unfortunately a trusted third party is necessary to certify watermarks before embedding. Another drawback is that the security of the proposed scheme cannot be established in a formal manner.

Fortunately, the customers' rights problem can be avoided entirely by using asymmetric fingerprinting protocols [71]. Such schemes can guarantee (with provable security) that the seller does not get to know the fingerprinted work.

1.4 Copyright protection schemes

For simplicity, we assume in the following sections that the party $\mathcal{P}$ acts both as a content provider and a seller. However, the definitions can easily be adapted to the case that the seller and content provider are different persons. A copyright protection scheme can be defined as tuple of three probabilistic polynomial time algorithms and one multiparty cryptographic protocol $\langle$Initialize, Prepare, Trace, Resolve$\rangle$.

The algorithm Initialize initializes the copyright protection scheme. Among other tasks, Initialize chooses cryptographic keys and watermarking keys that will be used in the other parts of the system. Initialize is run by the copyright holder $\mathcal{P}$; he inputs the length of the desired keys (n_S) and an identity string $ID_{\mathcal{P}}$. The output of Initialize is a tuple of keys $\langle K_E, K_D\rangle$. Here, K_E contains the part of the key that is used privately by the copyright holder $\mathcal{P}$, whereas K_D contains all keys that will be disclosed in the dispute resolving protocol (when using a symmetric watermarking scheme, K_E and K_D could be identical or K_D could be a substring of K_E).

Before a content provider releases an object O to a customer ($\mathcal{B}$), he runs the algorithm Prepare on it. On input K_E, an identity string of the customer ($ID_{\mathcal{B}}$) and O, the algorithm outputs a prepared object $\overline{O}$, together with a transaction token (t) and a proof token ($proof_P$). The transaction token uniquely identifies both the transaction and the customer $\mathcal{B}$; it will be used to decide who bought an unidentified copy $\overline{O}'$ allegedly derived from O. The proof token $proof_P$ acts as "proof of ownership" in future dispute resolving steps; in watermark-based dispute resolving schemes, the proof token may consist of a watermark and the original object O. Both tokens are generated for the content provider $\mathcal{P}$, whereas $\overline{O}$ will be forwarded to the customer $\mathcal{B}$.

The algorithm Trace may be used privately by the content provider in order to discover the original buyer of an object $\overline{O}$ that is apparently derived from $\mathcal{P}$'s object O. For this purpose, the algorithm Trace takes the key K_E of the copyright owner, the object $\overline{O}$ and a list of transaction tokens $\langle t_i\rangle_{i\in\mathbf{T}}$, where each t_i is the output of a run of the algorithm Prepare. The detection algorithm either returns an index $i \in \mathbf{T}$ (in case $\overline{O}$ was originally bought by the user identified in the token t_i) or the special symbol fail (in case no conclusion about the original buyer could be drawn). Note that Trace does not attempt to resolve any disputes; it just enables the copyright holder to determine the buyer who might be a possible traitor. Trace may, for example, contain the detection algorithm of a fingerprinting scheme.

Finally, the dispute resolving protocol Resolve is a multiparty protocol between $n+1$ participants, namely n (possibly cheating) parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$ who claim to be the true owner of an object $\overline{O}$ and a dispute resolver $\mathcal{R}$, who acts as a trusted judge. During the protocol, every party $\mathcal{P}_i$ inputs his detection key K_{D_i} together with a proof token $proof_{P_i}$. Using this input, the dispute resolver $\mathcal{R}$ decides who owns the copyright on the object $\overline{O}$. The output of the protocol is either the identity of a party $\mathcal{P}_i$ or the special symbol fail in case no conclusion about the owner could be drawn. We write the decision of the dispute resolver as Resolve$(\overline{O}, proof_{P_1}, K_{D_1}, \ldots, proof_{P_n}, K_{D_n})$.

More formally:

- The probabilistic algorithm Initialize initializes the whole system for the copyright

holder $\mathcal{P}$; on input n_S and $ID_{\mathcal{P}}$, INITIALIZE computes a tuple of strings $\langle K_E, K_D \rangle$ with $|K_D| = n_S$.

- The algorithm PREPARE prepares a multimedia object O for delivery to a party $\mathcal{B}$. On input K_E, $ID_{\mathcal{B}}$ and O, the protocol outputs a prepared object $\overline{O}$ along with a transaction token t identifying the transaction and a proof string $proof_P$.
- The algorithm TRACE takes an object $\overline{O}$, the key K_E and a list of transaction tokens $\langle t_i \rangle_{i \in \mathbf{T}}$ and outputs either an index $j \in \mathbf{T}$ or the special symbol FAIL.
- Finally, the protocol RESOLVE performs dispute resolving among n disputants; the protocol involves the parties $\mathcal{P}_1, \ldots, \mathcal{P}_n$ and $\mathcal{R}$. During the protocol, $\mathcal{R}$ inputs an object $\overline{O}$ and each $\mathcal{P}_i$ inputs his detection key K_{D_i} and proof string $proof_{P_i}$. At the end of the protocol, $\mathcal{R}$ outputs either the identity of a party $\mathcal{P}_j$ for $1 \leq j \leq n$ or the special symbol FAIL.

The basic security property of the dispute resolving process is the *weak resolvability*. Note that this definition does *not* attempt to formally define the "ownership" of a work, as this cannot be done without looking at the specific requirements of an application.

Let $\overline{O}$ be an arbitrary object and $\mathcal{P}$ its rightful owner. Furthermore, let $proof_P$ be the output of algorithm PREPARE for $\mathcal{P}$, n_K be a security parameter and $\langle K_{E_P}, K_{D_P} \rangle$ be the output of a run of algorithm INITIALIZE for $\mathcal{P}$. Then we call a dispute resolving scheme for n disputants $\mathcal{P}, \mathcal{P}_1, \ldots, \mathcal{P}_{n-1}$ weakly resolving, if

$$\mathbf{P}[\text{RESOLVE}(\overline{O}, proof_P, K_{D_P}, proof_{P_1}, K_{D_1}, \ldots proof_{P_{n-1}}, K_{D_{n-1}}) \neq \mathcal{P}]$$

is negligible in n_K for all strings $proof_{P_1}, K_{D_1}, \ldots, proof_{P_{n-1}}$ and $K_{D_{n-1}}$. Note that this must hold for *all* strings $proof_{P_1}, K_{D_1}, \ldots, proof_{P_{n-1}}$ and $K_{D_{n-1}}$, not only ones that were correctly generated by the algorithm INITIALIZE, as attackers may compute their proof strings in an arbitrary (unfair) way. The only restriction we make is that the *true* author has to behave according to the protocol specification by generating his proof string in a valid way.

In other words, a dispute resolving scheme is weakly resolving, if all disputes are resolved in favor of the rightful owner, *in case this party participates in the protocol*. This restriction has been commonly neglected in the literature in the past. As such protocols cannot guarantee any answer in case the true author is not present, we call the security property "weak" (see Section 1.4.1).

It is easy to see that one can restrict the discussion of dispute resolving schemes to disputes between two participants, as every weakly resolving dispute resolving scheme between two disputants can be extended to a weakly resolving dispute resolving scheme between n disputants $\mathcal{P}_1, \ldots, \mathcal{P}_n$. To see this, let RESOLVE2 be the weakly resolving dispute resolving scheme for two disputants. Define a $n+1$ party protocol RESOLVE between $\mathcal{P}_1, \ldots, \mathcal{P}_n$ and $\mathcal{R}$ in the following manner: $\mathcal{R}$ runs RESOLVE2 between each pair of participants $\langle \mathcal{P}_i, \mathcal{P}_j \rangle$, $i \neq j$, $1 \leq i, j \leq n$. In case there exists a party $\mathcal{P}_k$ that wins all two-party disputes, $\mathcal{R}$ outputs the identity of $\mathcal{P}_k$, otherwise FAIL. Suppose that the true author is among $\mathcal{P}_1, \ldots, \mathcal{P}_n$. Then, by the assumption of RESOLVE2 being weakly resolving, he wins all two-party disputes resolved by RESOLVE2 (except with a negligible probability) and thus he wins also RESOLVE. This shows that RESOLVE is weakly resolving. In the following we will therefore only consider two-party disputes.

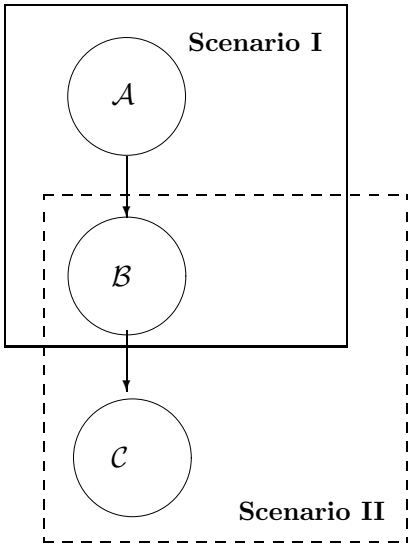

Figure 1.3: Conclusiveness problem.

1.4.1 Conclusiveness problem

As noted in [6], a difficult problem arises in dispute resolving schemes. As argued above, the security of a dispute resolving scheme is only guaranteed if the copyright holder is a protocol participant. If this is not the case there are no restrictions on the result of the dispute resolving protocol, i.e., after a protocol run any answer (that is, the identity of any participating party or the symbol FAIL) may be obtained.

Consider the two scenarios depicted in Figure 1.3. In the first scenario, the true author $\mathcal{A}$ and a traitor $\mathcal{B}$ dispute. Clearly, $\mathcal{A}$ wins the dispute using a weakly resolving scheme. In the second scenario, two traitors $\mathcal{B}$ and $\mathcal{C}$ dispute. Here, the output of the dispute resolving scheme is unclear: it can be either $\mathcal{B}$, $\mathcal{C}$ or FAIL. Assume for a moment that the output is $\mathcal{B}$. Now, in both scenarios the dispute resolver received the identity of a participant as output. However, he cannot distinguish between these two scenarios. In fact, he cannot decide whether the output of the dispute resolving scheme identified the correct copyright owner or a traitor (just because the true owner did not participate). This problem was termed *conclusiveness problem* [6, 51].

Although it is possible to avoid the problem with special watermarking schemes (whose existence is open) [6], it is an open problem whether dispute resolving schemes—without certification by a trusted clearing center—can be used for the identification of a copyright owner.

1.5 Dispute resolving using watermarks

As noted in Section 1.2, early watermarking papers proposed the following general framework for dispute resolving. During object preparation (PREPARE), $\mathcal{P}$ adds a watermark W

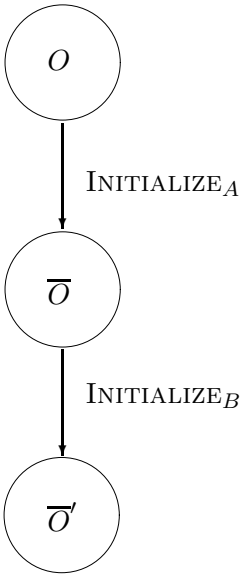

Figure 1.4: Doubly watermarked object $\overline{O}'$.

identifying himself, along with a watermark identifying the buyer ($\mathcal{B}$). Both watermarks are inserted with $\mathcal{P}$'s key K_E. The proof string $proof_P$ consists of the watermark W together with the unmarked original object O, i.e., $proof_P = \langle O, W \rangle$.

During the dispute resolving protocol, the dispute resolver $\mathcal{R}$ determines which evidence $proof_{P_i}$ matches the object $\overline{O}$ in dispute, i.e., which watermarks contained in the proof strings are actually detectable. If only one protocol participant can present a detectable watermark to the dispute resolver $\mathcal{R}$, the dispute is resolved in his favor. In case more than one participant is able to present a detectable watermark, the dispute resolver $\mathcal{R}$ has to decide which party actually inserted the watermark (i.e., gained its knowledge of a detectable watermark not through attacks).

The easiest attack that a dispute resolving protocol has to cope with is the addition of a second watermark, as indicated in Figure 1.4. Here, $\mathcal{A}$ is the true author of an object O. $\mathcal{A}$ runs INITIALIZE on O to obtain the watermarked object $\overline{O}$ and his proof string $proof_A$. Afterwards, $\mathcal{A}$ publishes $\overline{O}$. Now, the attacker $\mathcal{B}$ runs INITIALIZE on $\overline{O}$ to obtain a doubly-watermarked object $\overline{O}'$ and his proof token $proof_B$. The robustness assumption implies that both $\mathcal{A}$ and $\mathcal{B}$ obtain a proof string that is valid for $\overline{O}'$.

To defeat this attack, some authors proposed a resolving process that makes an ownership decision based on the original objects. More precisely, both $\mathcal{A}$ and $\mathcal{B}$ provide to the dispute resolver an alleged original as part of their proof strings. In case $\mathcal{B}$ engineered his "original" object from the watermarked object $\overline{O}$ by running INITIALIZE, then (assuming a sufficiently robust watermarking scheme) $\mathcal{B}$'s claimed original should contain $\mathcal{A}$'s watermark. Therefore, one might be tempted to declare the party as owner, who is able to present an "original" that does not contain the watermark from the other party. In fact, this criterion was sometimes used as formal definition of "ownership" (the ability to present an original that does not contain the other disputant's watermarks).

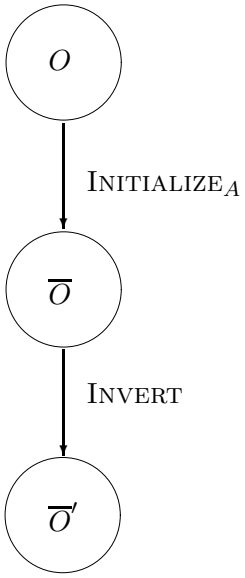

Figure 1.5: Inversion attack.

Unfortunately, the discovery of the inversion attacks showed that such systems are insecure; see Figure 1.5. In fact, the attacker is not required to use the initialization algorithm during his attack; he is free to choose any probabilistic polynomial attack algorithm that works in his favor. For example, the attacker $\mathcal{B}$ may use an inversion attack INVERT instead of INITIALIZE to engineer his proof string. According to the definition of an inversion attack, his computed watermark will also be detectable in the original O of $\mathcal{A}$, whereas $\mathcal{A}$'s watermark will be present in $\mathcal{B}$'s original by the robustness assumption. Thus, no party is able to provide an original that does not contain the other person's watermark and the resolving process necessarily fails.

One crucial assumption for the security of a dispute resolving scheme is that the true copyright holder keeps his "original" unmarked object secret. In other words, the attacker can only access a prepared version of the copyrighted object, i.e., the output of algorithm PREPARE. In general, no conclusion can be drawn in case this condition is violated. For example, consider Figure 1.6. Here, two parties independently take an object O and run the algorithm PREPARE on it to obtain prepared objects O_A and O_B. As both parties act independently, a watermarking-based dispute resolver cannot be expected to obtain any useful results.

We therefore only consider attackers that take an already prepared object $\overline{O}$ as input. The attacker is able to perform any probabilistic polynomial time attack on the object $\overline{O}$ and is expected to output a proof string and detection key such that the dispute resolving process fails or identifies the attacker as copyright owner. We also allow the attacker to fail with a certain bounded probability. Again, as we are only attempting to protect *honest* copyright holders, we require that the true copyright holder computes his proof token and keys according to the protocol specification.

Dispute resolving schemes should work also with objects that are slightly modified by an

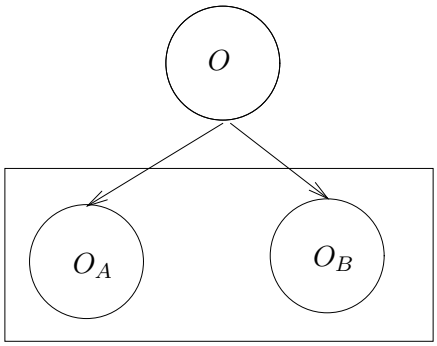

Figure 1.6: Attack scenario where two objects O_A and O_B are prepared independently.

attacker. That is, the dispute resolving step should be robust enough to work on an object $\overline{O}'$ that was derived out of the prepared object $\overline{O}$ of the copyright holder. To capture this, we allow an attacker to change the disputed object $\overline{O}$ invisibly during the attack and output an attacked object $\overline{O}'$, on which the dispute resolving process is started. However, an attacker is only allowed to perform invisible changes so that the two objects $\overline{O}$ and $\overline{O}'$ stay perceptually similar. Following this approach, we can see an attack on a dispute resolving scheme a probabilistic polynomial time algorithm that outputs, on input $\overline{O}$, the tuple $\langle \overline{O}', K_{D_A}, \mathit{proof}_A \rangle$ so that $\textsc{Resolve}(\overline{O}', K_{D_P}, \mathit{proof}_P, K_{D_A}, \mathit{proof}_A) \neq \mathcal{P}$ and $\textsc{Similar}(\overline{O}, \overline{O}') = \textsc{true}$ with non-negligible probability. Here, proof_P denotes the output of algorithm $\textsc{Prepare}$ for $\mathcal{P}$ and $\langle K_{E_P}, K_{D_P} \rangle$ be the output of a run of algorithm $\textsc{Initialize}$ for $\mathcal{P}$. It is easy to see that a dispute resolving protocol cannot be weakly resolving unless each such attack has only negligible success probability. We say that a dispute resolving scheme is secure, if every such polynomial attack has only negligible success probability.

Using this approach, one can analyze the security of a watermark-based dispute resolving process that bases its decisions on watermarks that are detectable in the claimed original objects, as described above. One can show that for its security it is necessary and sufficient that the underlying digital watermarking scheme has a sufficiently small false positives probability (i.e., a low probability that an arbitrary attacker finds a watermark in a digital object although this mark was never inserted). For a formal analysis we refer to [3].

Chapter 2

Watermarking Encrypted Data

In this chapter we discuss the feasibility of directly watermarking encrypted data, i.e., inserting a watermark into an encrypted multimedia object such that the corresponding decrypted object is watermarked. Such a watermarking operation that commutes with encryption can be a central building block of Digital Rights Management (DRM) and asymmetric fingerprinting systems.

2.1 Homomorphic Encryption

A first approach towards computing in the encrypted domain is to investigate constructions of cryptosystems that allow some basic algebraic operations in the encrypted domain which are "translated" onto the corresponding secret messages. Privacy homomorphisms were first proposed by Rivest et. al. [82], who defined them as encryption functions which permit encrypted data to be operated on without decryption of the operands. In his seminal paper, he presented five simple examples of privacy homomorphisms; the most relevant one was RSA, which is multiplicative:

$$\mathcal{E}(a \cdot b) = (a \cdot b)^e \bmod n = (a^e \bmod n) \cdot (b^e \bmod n) = \mathcal{E}(a) \cdot \mathcal{E}(b)$$

In [39] an additive and multiplicative privacy homomorphism was presented. The proposal allows addition, subtraction and multiplication to be carried out directly on encrypted data, but it does not allow computation of multiplicative inverses; in other words, it is a ring privacy homomorphism, but not a field privacy homomorphism. Another well known privacy homomorphism is the one proposed by Paillier [67], which is now used as a building block of different applications.

Security of privacy homomorphisms has been extensively studied. It is known that if a privacy homomorphism preserves order when encrypting, it is insecure against a ciphertext-only attack; if it is additive, it is insecure against a chosen-cleartext attack [8]. With the exception of RSA, all the examples given in [82] were subsequently broken in [17] using ciphertext-only or known-cleartext attacks. On the other hand, the privacy homomorphism presented in [39] is secure against a known-cleartext attack.

Computing with encrypted values can be a convenient tool in the construction of secure protocols. In particular, homomorphic encryption has been used successfully in different applications as a central building block, such as verifiable signature sharing [18], secure multiparty computation [30], electronic voting systems [69] and auctions [117, 19, 12]. Homomorphic encryption is even used in the context of databases [27, 47, 40] in order to allow database computation while maintaining a certain level of privacy of the records. In mobile agent technology, homomorphic encryption is applied to solve the problem of protecting agents against malicious hosts [84]; the solution is based on computing with encrypted functions, an extension of computing with encrypted data.

Homomorphic encryption was also proposed in watermarking scenarios: [43] use the homomorphic properties of RSA to define a publicly verifiable watermarking scheme in which the verification process does not reveal the embedded mark.

In addition to the basic algebraic operations described in the previous section, more complex computations—for example, encryption or the computation of signatures—can be performed on encrypted data.

An encryption function $\mathcal{E}_k(\cdot)$ is called commutative, if

$$\mathcal{E}_{k_1}(\mathcal{E}_{k_2}(m)) = \mathcal{E}_{k_2}(\mathcal{E}_{k_1}(m))$$

for all keys k_1,k_2. Obviously, most symmetric encryption functions are not commutative while RSA is a commutative encryption scheme, since

$$\mathcal{E}_{k_1}(\mathcal{E}_{k_2}(m)) = \mathcal{E}_{k_1}(m^{k_2} \bmod n) = m^{k_2^{k_1}} \bmod n = m^{k_1^{k_2}} \bmod n = \mathcal{E}_{k_2}(\mathcal{E}_{k_1}(m))$$

Commutative encryption schemes are useful in different areas. For instance, a fair coin flipping protocol and a bit commitment scheme can be defined using commutative encryption [87].

The possibility of digitally signing encrypted data has been discussed in the literature. In [1, 11] the authors argue that it makes no sense to sign encrypted data since non-repudiation cannot be ensured because the signer does not know what he/she signs. However, as it is pointed out in [97], there are some applications where privacy is more important than non-repudiation. The concept of blind signatures was introduced by Chaum in [21]. For example, using RSA signatures the blinding function can be implemented as

$$\mathcal{E}(m) = m \cdot r^e \bmod n = c,$$

where $k = r^e$ is the private key with $r \in_R \mathbb{Z}_n$ and e the public key of the signature function $\mathcal{S}$ defined by

$$\mathcal{S}(c) = c^d \bmod n = m^d \cdot r^{ed} \bmod n = m^d \cdot r \bmod n = c',$$

since $d \cdot e \equiv 1 \bmod \phi(n)$. The unblinding function is defined by

$$\mathcal{D}(c') = c' \cdot r^{-1} = m^d = s$$

and the signature verification function is

$$\mathcal{V}(s) = s^e \bmod n = (m^d)^e \bmod n = m.$$

Using this approach it is possible to sign in the blinded domain and verify in the clear domain, that is

$$\mathcal{V}(\mathcal{D}(\mathcal{S}(\mathcal{E}(m)))) = m.$$

The first application of blind signatures was an anonymous payment system [22], also proposed by D. Chaum.

2.2 Watermarking in the encrypted domain

In this section we want to pay attention to the possibility of watermarking in the encrypted domain, that is to say, we want to study when or how encryption and decryption functions commute with the watermarking operations (typically, mark and verify).

As mentioned above, computing in the encrypted domain can be seen as central building block for the construction of cryptographic protocols. In the same way, watermarking encrypted data can be a convenient tool for secure watermarking protocols and enables publicly verifiable watermarking schemes or asymmetric fingerprinting schemes.

A watermarking scheme can be defined through two functions, the embedding or marking function $\mathcal{M}$ and the extraction or verification function $\mathcal{V}$. Roughly speaking, the marking function takes an image[1] I as an input together with the mark m to be embedded and produces the marked image $I^* = \mathcal{M}(I, m)$. On the other hand, the verification function takes the marked (and possibly distorted) image $\widehat{I}$ and reconstructs the mark embedded in the image $\mathcal{V}(\widehat{I}) = m$.

Again, we denote an encryption function by $\mathcal{E}$; using an encryption key k, $\mathcal{E}$ produces an encrypted image $I_k = \mathcal{E}_k(I)$. The decryption function $\mathcal{D}$ reconstructs the image $I = \mathcal{D}_{k^{-1}}(I_k)$, given the decryption key k^{-1} and the encrypted image I_k.

Depending on the application, four properties of watermarking schemes and encryption functions can be convenient in the context of watermarking encrypted data:

Property 1. The marking function $\mathcal{M}$ can be performed on an encrypted image I_k to embed a mark m.

Property 2. The verification function $\mathcal{V}$ is able to reconstruct a mark in the encrypted domain when it has been embedded in the encrypted domain.

Property 3. The verification function $\mathcal{V}$ is able to reconstruct a mark in the encrypted domain when it has been embedded in the clear domain.

[1]For simplicity, we speak about images. However the main ideas can be applied in a similar manner to other digital objects (such as audio files).

Property 4. The decryption function does not affect the integrity of the watermark.

Property 1 ensures that when the marking function is performed on an encrypted image, the result $I_k^* = \mathcal{M}(I_k, m)$ will be an encrypted image. The second property ensures that the marking and verification processes can be performed entirely in the encrypted domain

$$\mathcal{V}(\mathcal{M}(\mathcal{E}_k(I), m)) = m.$$

The third property ensures that the encryption function does not affect the mark integrity in terms of mark reconstruction process; this property holds if

$$\mathcal{D}_{k^{-1}}(\mathcal{V}(\mathcal{E}_k(\mathcal{M}(I, m))) = m.$$

Note that properties 2 and 3 are equivalent in case the marking function and the encryption function commute

$$\mathcal{M}(\mathcal{E}_k(I), m) = \mathcal{E}_k(\mathcal{M}(I, m)) = I_k^*.$$

The last property implies that

$$\mathcal{V}(\mathcal{D}(\mathcal{M}(\mathcal{E}_k(I), m))) = m.$$

At first glance, it could be difficult to construct encryption/decryption functions and watermarking algorithms that satisfy all the proposed properties. For instance, Property 1 fails for the vast majority of marking schemes in the literature, as to achieve imperceptibility they are based on image characteristics that disappear when the image is encrypted.

However, in some scenarios, a secure watermarking protocol can be defined only if one of the above properties is satisfied. An example is the verification protocol proposed in [43] that allows to demonstrate the presence of a watermarking in an image without revealing the mark, thus producing the first approach to a zero-knowledge watermarking verification protocol. In this case, the watermarking verification protocol is based on any linear and additive watermarking algorithm in which the watermarking verification can be performed by mark correlation (for instance the spread spectrum technique proposed by Cox et. al. [29]). The encryption algorithm used is the RSA. When using such watermarking techniques together with this encryption algorithm Property 3 holds due to the homomorphic properties of the RSA algorithm together with the multiplicative operations performed in the verification process. This example shows the possibilities of watermarking in the encrypted domain, even if only one of the properties holds.

If there exist functions $\mathcal{M}$, $\mathcal{V}$, $\mathcal{E}$, and $\mathcal{D}$ that satisfy the above properties, we can define new watermarking protocols, as shown in the next subsection.

2.2.1 An asymmetric fingerprinting scheme

In contrast to watermarking (which allows ownership protection), fingerprinting is a technique which allows to track redistributors of electronic information. In a fingerprinting scheme, each image copy is marked with a different mark that allows buyer identification. Usually, it is

assumed that two or more dishonest buyers can only locate and delete those watermark parts that differ between two or more copies (*marking assumption* [16]).

Classical fingerprinting schemes [14, 16] are symmetric in the sense that both the seller and the buyer know the fingerprinted copy. Even if the seller succeeds in identifying a dishonest buyer, her previous knowledge of the fingerprinted copies prevents her from using them as a proof of redistribution in front of third parties. In [72], the concept of asymmetric fingerprinting was introduced, where only the buyer knows the fingerprinted copy. Different asymmetric fingerprinting proposals can be found in the literature [72, 73, 37, 38, 70, 83, 26].

In [37, 72] the authors use secure multiparty computation [23] to ensure that the seller cannot obtain knowledge about the marked copy. Since multiparty computation is computationally complex, some proposals are focused on avoiding such a technique. For instance, in [38] Committed Oblivious Transfer [32] is used instead of multiparty computation. These schemes have been improved in [83] and [26].

Watermarking in the encrypted domain can offer new possibilities towards the construction of new asymmetric fingerprinting schemes. As an example of a possible application, consider the following scenario.

Suppose functions $\mathcal{M}$, $\mathcal{V}$, $\mathcal{E}$, $\mathcal{D}$ satisfy the four properties stated above. Suppose further the waterma rking scheme used is asymmetric in the sense that a different secret information is needed for embedding and verifying the mark [42]. Finally, suppose the function $\mathcal{E}$ is a commutative encryption function.

Under these assumptions we can define an asymmetric fingerprinting scheme based on a protocol between a seller S and a buyer B that allows to embed a mark into an image in a way that:

- only B obtains the marked image, and
- only S knows the original image.

Protocol 1 (The marking protocol)

1. *S encrypts the original image I using an encryption algorithm $\mathcal{E}$ and a secret key k_1. $I_{k_1} = \mathcal{E}_{k_1}(I)$. Then S sends the encrypted image I_{k_1} and the mark m to be embedded to B.*
2. *B embeds the mark m into the encrypted image I_{k_1} using the marking function $\mathcal{M}$. The marked image is denoted by $I^*_{k_1} = \mathcal{M}(I_{k_1}, m)$. Then B encrypts $I^*_{k_1}$ using a key k_2 and sends the result $I^*_{k_1k_2} = \mathcal{E}_{k_2}(I^*_{k_1})$ to S.*
3. *S verifies that B has embedded the mark, that is, using the verification function $\mathcal{V}$, S extracts the mark m and checks that $\mathcal{V}(I^*_{k_1k_2}) = m$.*
4. *S removes the first encryption with k_1^{-1} over the image $\mathcal{D}_{k_1^{-1}}(I^*_{k_1k_2}) = I^*_{k_2}$ and sends the result to B.*

5. *B removes the second encryption with* k_2^{-1} *over the image* $\mathcal{D}_{k_2^{-1}}(I^*_{k_2}) = I^*$ *and obtains the marked image.*

Note that although B is the one that marks the image, he does not obtain the original image since he only obtains the encrypted version I_{k_1} and the marked one I^*. On the other hand, the scheme proposed is an asymmetric fingerprinting scheme since S does not obtain the marked image I^*.

The correctness of the protocol is based on the assumptions stated before. Step 2 uses Property 1: the marking function can be executed in the encrypted domain. Correctness of Step 3 is based on Property 3: the encryption function does not affect the mark integrity in terms of mark reconstruction. Step 4 is based on the assumption that the encryption function is a commutative encryption function and also on Property 4: the decryption function does not affect the mark integrity in terms of mark reconstruction. Finally, Step 5 is also based on Property 4. On the other hand, the asymmetry assumed for the watermarking scheme allows B to embed the mark in a way that S should not be able to reproduce, although S can verify that the mark is embedded into the marked image. Otherwise, the knowledge of the original image and the mark would allow S to obtain the marked image.

Of course, this is a high level overview of the protocol and we have made many assumptions on the properties of functions $\mathcal{M}$, $\mathcal{V}$, $\mathcal{E}$, $\mathcal{D}$. However, this protocol illustrates how watermarking in the encrypted domain can be applied to obtain new protocols with new properties.

2.3 Further Research Directions

Although the work presented in this report shows a possible application of the field of watermarking in the encrypted domain, this is only a first step. Further research should be directed to obtain functions $\mathcal{M}$, $\mathcal{V}$, $\mathcal{E}$, $\mathcal{D}$ that satisfy all of the aforementioned properties. Different strategies can be used to reach this objective. First of all, a deep study of the existing watermarking schemes and cryptosystems must be carried out in order to determine if functions with those properties already exist. A first approach has shown that Property 3 holds for some specific schemes [43]. Secondly, new watermarking schemes and ad-hoc cryptosystems can be proposed. Here we mean by ad-hoc cryptosystem an encryption function that encrypts an object producing a same object type as encrypted output (for instance, a clear image that is encrypted giving an encrypted image). Such ad-hoc encryption for images has been studied in the literature [41, 85, 28, 20], but the objective of such work has been sometimes undervalued since detractors argue that an image can be considered as a file and can be encrypted using any standard cryptosystem. However, if you need to compute in the encrypted domain you may need the encrypted object be an image and then ad-hoc cryptosystems may have a new field of application. On the other hand, combined watermarking/encryption functions can be defined in order to obtain some of the needed properties.

Furthermore, if such functions can be defined, a detailed study about their security must be carried out before they can be used within a secure protocol.

Chapter 3

Selective Encryption

3.1 On the Role of Selective Encryption in DRM

Today two technologies are applied to protect multimedia data in Digital Rights Management (DRM) environments: Encryption and digital watermarking. Encryption renders the data unreadable for those not in the possession of a key enabling decryption. This is especially of interest for access control, as usage of the data is restricted to those owning a key. Digital watermarking adds additional information into a media file without influencing quality or file size. This additional information can be used for inserting copyright information or a customer identity into the data stream. The later method is of special interest for DRM as it is the only protection mechanism enabling tracing illegal usage to a certain customer after the audio data has escaped the secure DRM environment.

Both mechanisms can be applied together in a DRM system. But existing methods create a sequence of protection operations: The audio content is first watermarked and then encrypted. To detect the watermark, the content has to be decrypted again as the watermarking needs to access the audio data. In commercial applications, this leads to a severe drawback: When running an online shop which uses watermarking for embedding customer identity and encryption for secure internet transfer, each time a customer buys and downloads an audio file, both security mechanisms need to be applied. This requires a huge computational power, disabling the strategy for most applications. Today even on the fly watermarking is challenging for applications with a great number of downloads.

To solve this problem, a combined watermarking and encryption scheme is necessary where both mechanisms are transparent to each other. A watermark can be embedded in and detected from an encrypted or unencrypted file. The watermark also does not influence the encryption mechanism. There are certain requirements for such a method:

- A common key must be available to both algorithms, but additional independent keys can be used for watermarking and encryption.
- Encryption must not produce noise-like data, but the process must be predictable within a certain boundary.

One possible solution to these problems is the use of selective (partial) encryption. The most important challenge is to find an embedding method for the watermark which is not influenced by the encryption. On the other hand, the partial encryption method must not be too weak to offer protection in a DRM environment.

In this chapter, we review selective encryption schemes for various media types.

3.2 Selective Encryption of Visual Data

Most modern methods for lossy compression of image and video data rely on either DCT-based techniques (with the JPEG and MPEG-* standards) or on wavelet-based techniques (see the JPEG-2000 standard). Therefore in the following we consider only selective encryption methods which are based on these techniques. In other words we disregard various older AVI-codecs which are based on vector quantisation, or lossless formats like PNG.

In the following we list encryption methods, usually selective encryption methods. First we subdivide the proposals according to the underlying transform domain, DCT or wavelet transform. The DCT-based techniques are further subdivided into approaches which are suitable for still images, and probably for I-frames of videos as well, and into approaches for video data.

3.2.1 DCT-based Techniques

Image Encryption

Compression Oriented Schemes

Zig-zag Permutation Algorithm: The historically first MPEG encryption proposal is due to Tang [99] and is called "zig-zag permutation algorithm". The idea is to substitute the fixed zig-zag quantised DCT coefficient scan pattern by a random permutation list. Additional security measures are proposed as follows:

- The 8 bit DC coefficient is to be split into two 4 bit halves, out of which the MSB part replaces the original DC coefficient and the LSB part replaces the smallest AC coefficient. The reason is that the DC coefficients could be identified immediately by their size thus revealing a low-pass approximation of the image.
- Before DC-splitting, the DC coefficients of eight 8×8-pixels blocks are concatenated, DES encrypted and written back byte-oriented.
- Instead of using one permutation list a cryptographically strong random bit generator selects one out of two permutation list for each 8×8-pixels block.

Shin et al. [94] propose a very similar system except that instead of splitting the DC coefficient the sign bits of the DC coefficients are DES encrypted and the DC coefficients are not subject to permutation.

Frequency-band Coefficient Shuffling: In order to limit the drop in compression efficiency as seen with zig-zag permutation, Zeng and Lei [119, 120] propose not to permute

the coefficients within a single 8×8 pixels block but to group the coefficients from an entire set of such blocks together and perform permutation to DCT coefficients within a frequency band (i.e. with similar frequency location). This strategy reduces the bit overhead significantly, but still a file size increase of 10 - 20% can be observed [120]. Whereas the security problems as induced by the use of permutations remain valid in principle, the situation is improved as compared to the pure zig-zag permutation approach since additional key material may be employed to define which blocks are used to select coefficients from and the described ciphertext only attack is much more difficult since more blocks are involved.

Scalable Coefficient Encryption: Cheng and Li [24] propose to encrypt the low-frequency DCT coefficients only and leave the high-frequency ones unencrypted in the JPEG environment. Their approach is therefore a partial encryption technique. The authors themselves mention that the security of this idea is questionable since when applied to all image blocks edge information remains visible. Kunkelmann and Reinema [56, 55] apply this idea to the MPEG case, use DES or IDEA for encryption, and suggest to use a different amount of coefficients for I and P/B frames. When the technique is applied to the DCT coefficients, care needs to be taken that the encrypted coefficients exhibit an admissible magnitude to be further processed correctly. In their latter paper Kunkelmann and Reinema apply this idea to the MPEG bitstream instead to coefficients. This of course raises the question of bitstream compliance.

Wu and Kuo [112, 113] raise the same security problems with respect to scalable coefficient encryption as Cheng and Li and give visual examples how well edge information can by recovered from material encrypted in the described way. They point out that the concentration of signal energy to a few coefficients as done by most orthogonal transforms does not necessarily imply that the same is true for intelligibility, which is often scattered among all frequency components.

Coefficient Sign Bit Encryption: Zeng and Lei [119, 120] suggest to encrypt the sign bit of each DCT coefficient only (which is a partial encryption approach). The rationale behind this idea is that the sequence of sign bits already exhibits high entropy, consequently, a further increase of entropy caused by encryption (and with it a file size increase after compression) should not be expected. Experimental results involving a H.263 codec even show a small bitrate reduction when applying sign bit encryption. Shin et al. [94] propose to encrypt the DC coefficient sign bit in addition to zig-zag permutation (see above).

Shi and Bhargava [91, 13] propose VEA (**V**ideo **E**ncryption **A**lgorithm) which relies as well on the basic principle to randomly change the sign bits of all DCT coefficients; however, they propose to apply this principle directly on the bitstream (which is possible in principle since coefficient sign bits are separated from the Huffman codewords in the bitstream).

Secret Fourier Transform Domain: An approach significantly different from those discussed so far is to conceal the transform domain into which the image data is transformed by the compression scheme. The underlying idea is that if the transform domain is not known to a hostile attacker, it is not possible to decode the image. Fractional Fourier domains have been used in earlier work to embed watermarks in an unknown

domain [36]. Unnikrishnan and Singh [107] suggest to use this technique to encrypt visual data. In particular, the input plane, the encryption plane, and the output plane of the proposed method are fractional Fourier domains related to each other by a fractional Fourier transform. While the security and the size of the corresponding keyspace is discussed, the complexity is not. The authors discuss an optical implementation, but it seems that an implementation on a digital computer would be very costly due to the transforms and the additional encryption involved. While being an interesting approach in principle, it seems that this technique might be used only in very specific environments and the advantage over a classical full encryption is not obvious.

Secret Entropy Encoding: Based on the observation that both, cipher and entropy coder, turn the original data into redundancy-free bitstreams which cannot be decoded without certain information, Wu and Kuo [112] discuss the possibility to turn an entropy coder into a cipher. The information required for decoding is the key in the cryptographic case and the statistical model in the entropy coder case. The authors refer to earlier work where it is shown that is is extremely difficult to decode a Huffman coded bitstream without the knowledge of the Huffman codewords. Shi and Bhargava [92] (see the section on Zig-zag Permutation) suggest a codeword permutation which has a very limited keyspace.

Wu and Kuo propose to use multiple Huffman coding tables out of which a specific one is selected based on random decisions for encoding a given symbol. To be able to maintain the compression ratio it is suggested to use a different set of training images for each table. Huffman tree mutation can also be used to create a large number of Huffman tables out of 4 initially given tables. In their later work [113], the authors suggest to enhance the security of their scheme by inserting additional bits at random positions and by doing this in different ways for different portions of the data.

Bitstream Oriented Schemes

Header Encryption: The most straightforward to encrypt an image or video bitstream is to encrypt its header. Bitstream compliance is immediately lost and the image or video cannot be displayed any longer using a common player. In case an attack is conducted against this kind of encryption, most header informations can simply be guessed or extracted from the syntax of the bitstream itself. As a consequence, the security of such a scheme depends on the type of header data encrypted – if the protected header data can not be guessed or computed by analysing the bitstream, and if this data is crucial for the decoding of the visual data, this approach could be secure. Header encryption is an interesting approach since it requires minimal encryption effort only.

Permutations: Permutations are a class of cryptographic systems well suited for designing soft encryption schemes and have been proposed to be applied at the bitstream level. All these schemes are extremely vulnerable against a known plaintext attack as described in the context of the Zig-Zag Permutation Algorithm. The entities subject to permutation (*basic shuffling units* [111]) are different when comparing the suggestions made so far.

1. **Bytes** Qiao and Nahrstedt [81] discuss the *Pure Permutation Algorithm* where single bytes of an MPEG video stream are permuted. Depending on the security

requirements the permutation lists in use can be made longer or shorter. Whereas this approach is extremely fast in terms of the actual encryption process and in terms of parsing the bitstream to identify the data subject to encryption, the semantics of the MPEG stream are entirely destroyed and no bitstream compliance is obtained.

2. **VLC codewords** Based on their earlier Frequency-band Coefficient Shuffling idea [119], Wen et al. [111] and Zeng et al. [118] propose to shuffle VLC run-level codewords corresponding to single non-zero DCT coefficients. Codewords from different 8×8 pixels blocks are grouped together according to their codeword index and permuted with one permutation list. The number of groups and the range of codeword indices within one group can be adjusted according to security requirements. A problem with this approach is that different 8×8 pixels blocks usually contain a different number of non-zero coefficients which can be resolved by controlling the "last field" of each block. Kankanhalli and Guan [49] independently propose exactly the same idea, they further increase the security of their system by additionally flipping the last bit of the codewords randomly and apply corrections if the prefix of the subsequent codeword is affected.
3. **Blocks and Macroblocks** In the same papers, the authors also discuss the use of 8×8 pixels blocks and macroblocks as the basic shuffling units. Whereas in the case of macroblocks format compliance is guaranteed, in the case of 8×8 pixels blocks care must be taken about the different VLC tables used to encode inter and intra coded blocks, i.e. these blocks need to be permuted separately or the corresponding flag in the bitstream needs to be adjusted if the type of block was changed by permutation. This approach is equal to a permutation of (smaller or larger) image blocks in the spatial domain and it is widely accepted that such a technique is vulnerable to a ciphertext only attack. It is only necessary to group blocks with corresponding or similar boundaries together to get a good approximation of the frame.

One-time pad VEA: Qiao and Nahrstedt [80, 81] propose another partial encryption VEA (**V**ideo **E**ncryption **A**lgorithm) which operates on MPEG streams at the byte level. Odd-numbered and even-numbered bytes form two new byte streams, the *Odd List* and the *Even List*. These two streams are XORed, subsequently the Even List is encrypted with a strong cipher. The result of the XOR operation and the encrypted Even List are the resulting cipher text. As a consequence the DES encrypted half of the byte stream serves as a one-time pad for the other half which makes the system fairly secure, because there exists low correlation between bytes and pairs of bytes in MPEG streams (this is confirmed experimentally [80, 81]). This exploits the fact that both compression and encryption decorrelate the data. In order to further increase the security the following improvements are suggested:

- The fixed odd-even pattern is replaced by two randomly generated byte lists (where this is controlled by a 128 – 256 bit key).
- Each set of 32 bytes is permuted, 8 different permutation lists are employed which are used in fixed order.
- The generation of the two byte lists is changed for every frame (of a video).

Byte-Encryption: Griwodz et al. [44, 45] propose to randomly destroy bytes in an MPEG stream for free distribution, while the original bytes at the corresponding positions are transferred in encrypted form to legitimate users. This is actually equivalent to encrypting bytes at random positions. The authors find that encrypting 1% of the data is sufficient to make a video undecodable or at least unwatchable. The cryptanalysis given is entirely insufficient. Consider the worst case where only MPEG header data is encrypted by chance using this approach. It is well known that header data may be reconstructed easily provided the encoder in use is known. To guarantee a certain level of security a higher amount of bytes needs to be encrypted and care needs to be taken about which bytes are encrypted. Wen et al. [111] describe a more general approach as part of the MPEG-4 IPMP standard, named *Syntax Unaware Runlength-based Selective Encryption*. This algorithm encrypts X bits, the next Y bits are left in plain-text, the next Z bits encrypted again, and so on. In addition to the abovementioned security problems, both approaches partially destroy the MPEG bitstream syntax (which is the main security approach of these schemes) and potentially emulate important MPEG markers causing a decoder to crash (which is again desired).

Alattar et al. [10] consider a somehow related approach by encrypting only every other basic encryption unit as opposed to other techniques where all such units are protected, although this is done in the context of a inter-frame encryption approach.

VLC Codeword Encryption: Whereas Byte Encryption does not take the syntax of the video into account, VLC codeword encryption does. Contrasting to the permutation of such codewords, strong encryption should be applied in this case. In case bitstream compliance after encryption is not the aim, Byte Encryption applied to a significant fraction of the bitstream is a better choice since VLC codewords need not be identified and therefore bitstream parsing may be avoided to a large extent. In case bitstream compliance after encryption is the aim, when encrypting VLC codewords we face the problem the the encryption of a concatenation of VLC codewords leads not necessarily to a concatenation of valid codewords. Wen et al. [110, 111] propose a solution to this problem, which has also been adopted for the MPEG-4 IPMP standard. While this scheme guarantees a standard compliant bitstream it does not preserve the size of the bitstream. In general, the original and "encrypted" concatenation of codewords will not have the same size (although the number of codewords is equal)—the examples in [111] exhibited an overhead of about 9% of the original filesize.

Video Encryption

Video encryption based on DCT methods is focused on standardised formats like MPEG-1,2,4 or H.26X, therefore all these techniques try to take advantage of the corresponding data formats and bitstreams. Whereas all the techniques discussed subsequently could as well be applied during the compression stage, they are mostly discussed in the context of directly manipulating the bitstream data (after compression has taken place).

Bitstream Oriented Schemes Most schemes for video encryption combine various ideas and are neither purely frame-based nor purely motion-based. Note that all techniques de-

scribed in the section on image encryption may be applied to single frames or the bitstream of videos as well.

Header Encryption: As the second-lowest security level in their SECMPEG scheme (see below), Meyer and Gadegast [63] propose to encrypt all (MPEG) header data of the MPEG sequence layer, group of picture layer, picture layer, and slice layer. Lookabaugh et al. [59] discuss in detail which types of header data are interesting candidates for being encrypted. The data suggested by Meyer and Gadegast to be selectively encrypted turns out to be hardly suited for that purpose (except for the `quantizer_scale_code` field in the slice header). They suggest to encrypt the `macroblock_type` field in the macroblock header. Wen et al. [111] investigate the encryption of the *Dquant* parameter (difference of quantisation step size QP between current and previous macroblock), which is a very simple approach since many macroblocks simply use the default settings which makes this parameter easy to attack.

Encryption of I-Frames: Spanos and Maples [96] and Li et al. [58] independently propose to encrypt I-frames only. Since P and B-frames are reconstructed based on predictions obtained from I-frames, the main assumption is that if these are encrypted, P and B-frames are expected to be protected as well. This very simple idea has still been suggested in 2003 for a combined DVD watermarking-encryption scheme by Simitopoulos et al. [95] and for a wireless multimedia home network by Taesombut et al. [98]. There are several problems associated with this approach. First, the percentage of the bitstream which is comprised of encoded I-frames is about 25-50%. Second, the motion in the video remains visible, especially when replacing the encrypted I-frames by uniform frames. Third, and even more severe, the I-blocks contained in P and B-frames resulting from poor prediction results even deliver texture information of I-frames when collected over several frames. This especially affects high motion sequences since in this case P and B-frames contain many I-blocks. Agi and Gong [7] noted this problem in 1996 and suggested to encrypt also these I-blocks. This technique is found as well in one mode of the SECMPEG scheme, in one mode of the combined watermarking encryption scheme of Wu and Wu [114], and has been also suggested in 1999 by Alattar and Al-Regib [9]. To reduce to computational overhead Alattar et al. [10] suggest to encrypt every other I-block (reducing the encryption percentage to 20 - 40% which is still a lot), however, still the entire approach remains quite insecure.

Encryption of Motion Vectors: Motion vectors comprise about 10% of the entire data of an MPEG video [59], therefore, restricting the encryption to motion vectors might be an interesting idea. From a security point of view encryption of motion vectors alone can never be sufficient since all texture information from I-frames remains in plain text.

SECMPEG: SECMPEG defines a new bitstream including a header structure which makes it incompatible with respect to common MPEG players. Besides its data integrity and authentication functionality (which we do not discuss here), five levels of security are defined:

1. No encryption.
2. Header data from the sequence layer down to the slice layer is encrypted.

3. Encrypt same data as in level 2 and the low frequency DCT coefficients of all blocks in I-frames.
4. Encrypt all I-blocks (also those in P and B-frames).
5. Encrypt the entire video.

As can be seen, three different basis techniques are combined into SECMPEG and all their properties and restrictions apply correspondingly: Header encryption, I-frame encryption, and Scalable Coefficient encryption.

Alattar et al. [10] propose as well a scheme with scalable complexity and security using three levels:

1. Encrypt all data associated with every n-th I-macroblock.
2. Encrypt all data associated with every n-th I-macroblock and all header data of every n-th P and B-encoded macroblocks.
3. Encrypt all data associated with every n-th I-macroblock and all header data of P and B-encoded macroblocks.

Again, Header encryption and I-frame encryption is combined. As a third example for an explicitly scalable scheme we describe the security levels of the combined watermarking encryption scheme by Wu and Wu [114]:

1. Encrypt the DC coefficient of the luminance component of all I frames.
2. Encrypt the luminance and chrominance DC coefficients of all I frames.
3. Encrypt all luminance and chrominance coefficients of all I frames.
4. Encrypt the DC coefficient of the luminance component of all I macroblocks.
5. Encrypt the luminance and chrominance DC coefficients of all I macroblocks.
6. Encrypt all luminance and chrominance coefficients of all I macroblocks.
7. Encrypt the data of all frames (but no header data).

Contrasting to the other two suggestions no header data is encrypted which enables the scheme in principle to deliver compliant bitstreams. An explicit distinction between luminance and colour component is made, where securing the luminance component is of course more important from a perceptual viewpoint. The scheme by Wu and Wu again combines I-frame encryption and Scalable Coefficient encryption.

MVEA and RVEA: Shi and Bhargava [13, 93, 90] suggest to improve their former algorithm VEA by encrypting the sign bits of the motion vectors in addition to the sign bits of the DC coefficient of I-blocks. This technique is denoted MVEA (if sign bits are randomly changed or RVEA (if sign bits are encrypted). Both, DC coefficients and motion vectors are differentially encoded which causes significant impact when the corresponding sign bits are changed. The authors give a fixed scan order through the data of a macroblock, at most 64 sign bits are encrypted per macroblock. For I-macroblocks, first the luminance and chrominance DC coefficient sign bits are encrypted, subsequently the lowest frequency AC coefficient sign bits and so on. For P and B-macroblocks, the first sign bits to be encrypted are the sign bits of motion vector data, then the scan proceeds as in the case of I-macroblocks.

Techniques in MPEG-4 IPMP: In their simulations under the MPEG-4 IPMP framework Wen et al. [111] investigate three different configurations with increasing security:

1. Encrypt the FLC coded DCT sign bit, the parameter DQUANT (two bits determining the difference between the quantisation parameter used for the previous macroblock and the current one), and the I-macroblock DC value.
2. Encrypt the VLC motion vector field.
3. Encrypt the data of both previous suggestions.

This proposal combines several techniques as well: I-frame encryption, Coefficient Sign Bit encryption, Header encryption, Motion vector encryption, and Scalable Coefficient encryption. Whereas the first two options are said to be useful for entertainment purposes only, the third configuration provides satisfying results. The authors therefore propose to combine their suggested encryption configurations with permutations to achieve a higher security level.

Implementation and Assessment of Selective MPEG-encryption

General Information For our assessment on the effects of the various encryption method proposed in the literature we recreated a number of them, following their description to a higher or lesser degree. We integrated these encryption schemes into an open-source MPEG encoder called `mpeg2enc` from the `mjpegtools` package available at `http://sourceforge.net/projects/mjpgtools/`. This enables us to compare them in the same environment with the same sequences and settings, like the target bitrate. These experiments may be performed online at `http://www.ganesh.org/book/` (see also [106]).

VLC Table Codeword Permutation A possible way to encrypt a bitstream is to exchange the entries in the VLC table. When a value/runlength pair is encoded a different codeword is retrieved from the table than it would be in the unencrypted case. Several consequences follow: when random codewords are used they are in general not prefix-free. So the easiest way to generate another set of valid codewords is to use the existing ones and to permute them. In this scenario the encryption key is the seed value for a PRNG which generates to permutation.

Macroblock Permutation Each frame within a video consists of the same number of macroblocks, each macroblock contains such data as quantised coefficients from the Y, U and V bands, or motion vectors. A variant to encrypt videos is to exchange the macroblocks within a frame. The key for this encryption method is used as a seed value for a PRNG which generates a permutation.

DCT Block Permutation Similar to the approach where complete macroblocks are exchanged it is possible to exchange the individual 8*8 DCT coefficient blocks. By permuting the smaller blocks the confusion being created is bigger, and reconstruction becomes more difficult but not impossible. Additionally the algorithm does not discriminate between DCT blocks from the Y plane and DCT blocks from the U and V planes.

Motion Vector Permutation Similar to the macroblock permutation and the DCT block permutation it is possible to permute the motion vectors which are assigned to distinctive macroblocks. Within a predicted frame each macroblock can be either an I-block or a predictive block, motion vectors are just assigned to the latter. These vectors can be permuted according to an order provided by a PRNG, where the seed is the key.

Motion Vector Sign Change Motion vectors are signed values, and therefore another possibility for light encryption is to change to sign bits of these vectors. The actual motion vectors are not stored in the bitstream, they are predicted based upon previous motion vectors, and just the prediction error (residual) is stored. So we have two variants for sign encryption: change the signs of the prediction, and change the signs of the residual. Again the decision which signs are changed and which are left as they were is based on the output from a PRNG, again with the key as its seed value.

DCT Coefficient Sign Bit Encryption part from the motion vectors the transform coefficients are signed values as well. So it is possible to change their signs in a pseudo-random manner as well. There are several variation possible: the most simple variation is to change the sign bits of all blocks. Other variations are to change just the sign bits of I-blocks, either just the I-blocks which are located in I-frames, or additionally the I-blocks in predicted frames. The latter can be seen as a complementary option to the various motion vector encryption methods, as they do not change any I-block.

DCT Coefficient Mangling Before the transformed values in an 8*8 block are encoded it is possible to modify them. Individual bits are XOR-encoded with PRNG values. This is prone to generate longer bitstreams since some significant bits which were originally 0 are now changed to 1. To minimise this effect our approach is first to change just values which are likely to be non-zero anyway and second to change just some of the least significant bits of these values, so that they change just within their order of magnitude.

Zigzag Order Permutation After the quantisation in a 8*8 DCT coefficient block the coefficients are encoded together with runs of zeros. The order starts with the DC coefficient, continues with the low frequency AC coefficients and ends with the highest frequency coefficient, the order is a zig-zag curve in the 8*8 block. A method to encrypt the data is to use a different order to encode the data. Based on a seed for a PRNG the coefficients are permuted before the zig-zag scan is performed.

3.2.2 Wavelet-based Techniques

Wavelet-based techniques devoted to encryption of visual data have not been discussed very extensive in literature so far [74], although all proposals made for image encryption may be applied to each frame of a video independently of course. The lack of a wavelet-based video coding standard explains this situation. As in the section on DCT-based techniques, we distinguish between schemes operating during the compression stage ("compression oriented") and schemes applicable to a given bitstream ("bitstream oriented').

Compression Oriented Schemes

Coefficient Selective Bit Encryption: Similar to their proposals for DCT-based systems, Zeng and Lei [119, 120] suggest to encrypt selected parts of the transform coefficients' binary representation. They compare refinement, significance, and sign bits with respect to their entropy and compressibility. Based on this analysis, it is suggested to encrypt bits that are not highly compressible due to their high entropy and low intercorrelation.

Coefficient Permutation: In the context of DCT-based compression systems, coefficient permutation has been proposed as a means to provide confidentiality within the compression pipeline. In the context of wavelet-based compression schemes, random permutation lists have been proposed by Uehara et al. as well to secure wavelet-subbands [102].

In the following we use random permutation lists to secure wavelet-coded visual data (compare also [66]). We show that a system based on randomly permuting wavelet-subbands incorporated in the JPEG 2000 or the SPIHT coder generally delivers much worse results in terms of compression performance as given in [102]. The comparison of JPEG 2000 and SPIHT in this context provides interesting insights with respect to the correctness of the zerotree hypothesis.

The basic approach is to permute the wavelet coefficients of different wavelet subbands with dedicated permutation keys. A permutation key is defined as a vector of length n, and n wavelet coefficients can be encrypted using this key. In case permutation keys have to be transmitted along with the compressed image data (and not generated on the fly as proposed in [111]) the used keys have to be protected and therefore be encrypted with a standard encryption scheme like AES. For example, the key data itself can be inserted conveniently into the JPEG 2000 bitstream taking advantage of the so-called termination markers.

Coefficient Block Permutation and Rotation: Zeng and Lei [119, 120] also propose a generalisation of the coefficient permutation approach. They suggest to divide each subband into a number of blocks of the same size. The size of these blocks can vary from subband to subband. Within each subband, blocks of coefficients are permuted according to a permutation key which should also differ from one subband to another. Since the local statistics of the wavelet coefficients are preserved, the expected impact on coding performance is smaller as compared to the pure coefficient permutation case (the degradation is the smaller, the larger the block size is selected). On the other hand, using large blocks threatens security due to two reasons:

- The permutation key is small.
- A possible attacker might try exploit edge continuity in the high pass subbands and a smoothness constraint in the low pass subband to invert the permutation.

Secret Transform Domains: Similar to the idea of using a secret Fourier transform domain it is also possible to use secret wavelet transforms for an encryption application. The idea of using secret wavelet domains has also been used to increase the security of watermarking schemes. In this context, filter parameterisations [35] and wavelet packet subband structures [34] have been used to conceal the embedding domain.

Contrasting to the Fourier case, all proposals concealing the wavelet transform domain for encryption are integrated into a compression pipeline. Vorwerk et al. [108] propose to encrypt the filter choice used for wavelet decomposition, however, this suggestion remains vague and is not supported by any experiments. In the following we discuss two ways of generating a large variety of wavelet filters out of which a secret one may be chosen for actual decomposition. All these techniques have a significant advantage: the amount of data subject to encryption is minimal since only information about the transform in use needs to be encrypted. Several variations of this approach can be distinguished:

Codebook Approach: Generalised wavelet decompositions (where different filters are used at different decomposition levels) may be employed and the structure of these decompositions may be kept as key: here non-stationary multiresolution analysis NSMRA [33, 60, 103, 75] or subband variant decompositions (which use the same idea applied for wavelet packet decompositions at the subband level) [104] are possible candidates.

Parametrisation Approach: For the construction of compactly supported (orthonormal) wavelets, solutions for the dilation equation have to be derived, satisfying two conditions on the coefficients c_k ($\phi(t) = \sum_{k\in\mathbb{Z}} c_k \phi(2t-k)$, with $c_k \in \mathbb{R}$). One approach is to use parameterised filter generated according to an algorithm proposed by Schneid and Pittner [86]. Related to this parameterisation is [54] by Köckerbauer et al. for the JPEG-2000 case. Note that similar parameterisations are available for biorthogonal filterbanks [48] and for the lifting scheme in the context of JPEG 2000 [89]. For an description of some problems associated with parameterised biorthogonal wavelet filters see Uhl [105].

Secret Subband Structures: With this approach we use wavelet packet based compression instead of pyramidal compression schemes in order to provide confidentiality (compare also [75, 76, 77, 78]). Header information of a wavelet packet (WP) image coding scheme based either on a uniform quantiser or on zerotrees is protected, in particular we use AES to encrypt the subband structure used by the encoder only. In our approach the encoder uses different decomposition schemes with respect to the wavelet packet subband structure for each image (in fact, the subband tree is chosen randomly or determined by some pseudo-random algorithm). These decomposition trees are encrypted and have to be present at the decoder to be able to reconstruct the image data properly.

Bitstream Oriented Schemes

SPIHT Encryption: Cheng and Li [25] discuss a partial encryption scheme for SPIHT which can be applied to any zerotree-based wavelet coding scheme. The basic observation is as follows: the compression algorithm produces many different types of bits – sign bits, refinement bits, and significance bits of pixels and sets. The decompression algorithm has to interpret each bit under the correct context. Incorrect significance bits may lead to an incorrect interpretation of subsequent bits, this is not the case when sign bits or refinement bits are decoded incorrectly. As a consequence it is suggested to encrypt the significance information of sets and pixels of the two lowest resolution

pyramid levels. The reason for not encrypting all significance information is as follows: the significance information of the low resolution levels is used to initialise the different lists used by the algorithm. If the states of these lists are incorrect right from the start of the decoding, it is hardly possible for the algorithm to recover from the error. The information left unencrypted is of low value for an attacker since without the significance information the type of bits can not be distinguished from another. Basically the argumentation is quite similar to the case of encrypting the wavelet packet subband structure only (see above).

JPEG 2000 Encryption: For selectively encrypting the JPEG 2000 bitstream we have two general options. First, we do not care about the structure of the bitstream and simply encrypt a part, e.g. the first 10% of the bitstream. In this case, the main header and a couple of packets including packet header and packet data are encrypted. Since basic information necessary for reconstruction usually located in the main header is not available at the decoder, encrypted data of this type can not be reconstructed using a JPEG 2000 decoder. Although this seems to be desirable at first sight, an attacker could reconstruct the missing header data using the unencrypted parts, and, additionally, no control over the quality of the remaining unencrypted data is possible. Therefore, the second option is to design a JPEG 2000 bitstream format compliant encryption scheme which does not encrypt main and packet header but only packet data. This option is investigated further.

Grosbois et al. [46] propose the first partial encryption scheme for JPEG 2000 bitstreams. A pseudo random inversion of the bits in certain layers is suggested, but no further details with respect to amount and position of the encrypted data are given. Also, no attacks are demonstrated. Wee and Apostopoulos [109] integrate Motion JPEG 2000 into their secure scalable streaming concept (SSS) by exploiting the different ways of achieving scalability in JPEG 2000. Of course, JPEG 2000 suits much better in the SSS context as compared to other codecs since scalability is an inherent property, different mixtures of quality and resolution levels are experimentally evaluated. Triple-DES and AES in CBC mode are used for encryption which means that data has to be padded to suit the block-size specification of these algorithms. A very interesting issue is discussed by Kiya et al. [53] in the context of encrypting packet data of JPEG 2000 streams. Straightforward encryption of this data may lead to the emulation of marker codes which cause the resulting bitstream to be non-compliant and would cause a decoder to crash. They suggest to perform the encryption process based on half bytes in a specific marker aware mode which uses the hexadecimal notation of the markers. Wu and Deng [115] also address the problem of compliant encryption and suggest to check the compliance during the encryption process and to change the process accordingly in case of marker generation. The same authors also discuss an access control scheme using a key generation scheme for parts of the codescheme [116] relying on their former work on hash trees for authenticating JPEG 2000 streams [68].

3.3 Standards and Commercial Applications

Several standards as well as products based on these standards exist which provide functionality which can be subsumed under selective encryption. They provide means to modify the

bitstream (which may contain image, audio, video and/or other data) to achieve several goals. Most of the time this is confidentiality which requires encryption.

3.3.1 JPSEC — Secure JPEG 2000

JPSEC will be a standard also known as ISO/IEC 15444-8, it is an extension to the JPEG-2000 standard. According to the current timeline [88, resolution 42] we can expect the FDIS (Final Draft International Standard) in December 2004, and the final standard in February 2005. Since JPSEC is not finalised at the time of this writing the following information might change.

JPSEC allows the content creators and providers to protect parts (called "zone of influence" ZOI) of a JPSEC file. It distinguishes between image data and non-image data (e.g. headers), and it allows manifold protection schemes: fragile integrity verification (using cryptographic hashes), semi-fragile verification (usually with the help of watermarks), source authentication, conditional access, secure scalable streaming and transcoding, registered content identification and of course confidentiality (by encryption or selective encryption). JPSEC allows multiple applications of the above protection methods on the same data, for example it will be possible during creation to first authenticate the image, watermark it, and then to encrypt a part of it. On the side of the recipient the process is then reversed. Some protection methods are predefined (such as encryption using AES), but others can be attached. The standardisation committee decided to set up a registry for such additional protection methods, such a registry allows the unique identification of the protection methods in any JPSEC bitstream.

3.3.2 IPMP — Intellectual Property Management and Protection

IPMP is a standard within the MPEG family which has been developed at first for MPEG-2 and MPEG-4. During the time some problems with this version of the standard have been found (there were interoperability conflicts with security and with flexibility). A second attempt, now part of MPEG-21 ("Multimedia Framework") tries to address the wishes of consumers and manufacturers, this new version was "back-ported" to MPEG-2 and -4, and can be found there as IPMP-X.

IPMP tries to create a way of interoperability for the deployment of content and applications, it distinguishes between 5 different communities: end-users or consumers, content providers, device manufacturers, service providers, and content authors. IPMP tries to meet the goals of all these groups by the creation of an extensive framework. One important part of this framework is the concept of the "IPMP tools": they are modules that perform one or more functions like authentication, decryption or watermarking on an IPMP terminal, such modules are identified by an ID, they can be embedded in a bitstream, downloaded or acquired by other means. When a user requests a specific content, then the following steps are executed: the IPMP tools description is accessed, the relevant IPMP tools are retrieved, instantiated, initialised and updated during the content consumption [52, 64].

3.3.3 MPEG, DVB and CSA

The MPEG standards include an encryption mechanism which allows vendor-specific plugins, it is called "Common Scrambling Algorithm" CSA. CSA is built as a combination of a block cipher and a stream cipher. The block cipher uses a 64-bit key to generate 56 different 64-bit keys for the individual rounds to encrypt a 64-bit block. The stream cipher uses several LFSRs in parallel, each 10 bits long, the output is fed back via an S-box permutation. During decryption the data is first decrypted using the stream cipher and then by the block cipher.

The idea is that every provider uses the same algorithm to encrypt the transmitted MPEG stream. Each provider can use its own algorithm to calculate the seed value for the CSA. On the customer side the MPEG-receiver needs a "Conditional Access Module" from the respective provider which enables the decryption. Modern receivers contain an interface following the "Common Interface Standard", such an interface is basically a PCMCIA slot, and the access module (also called "CI module") is a smartcard within a PCMCIA adapter. The chip on the smartcard is responsible for the correct generation of the CSA seed values. This seed is also called "Control Word" or "Common Key".

3.3.4 DVD and CSS

DVDs can be protected using an encryption method called CSS, which was developed in 1996 by Matsushita. The sectors of such a DVD are encrypted using a chain of keys:

Title keys: these keys are used to protect the actual contents on the DVD.

Disc keys: these keys are used to encrypt/decrypt the title keys on the DVD.

Player keys: these keys are used to protect the access to the disc keys. The disc key is stored 400-fold in encrypted form on the DVD, each time encrypted with a different player key. Each DVD player manufacturer gets its own player key, the manufacturer must take care because it must not be compromised.

To access a title on a DVD the player has to use its own key to decrypt the disc key. This disc key is used to obtain the title key for a specific title on the DVD. Prior to this encryption sequence the DVD drive and the unit performing the CSS decryption have to authenticate to each other, to verify that the partner module complies with the DVD standards. DVD copy protection mechanisms are described in more detail in [15].

Bibliography

[1] Martin Abadi and Roger Needham. Prudent engineering practice for cryptographic protocols. *IEEE Trans. Softw. Eng.*, 22(1):6–15, 1996.

[2] A. Adelsbach, S. Katzenbeisser, and A.-R. Sadeghi. Watermark detection with zero-knowledge disclosure. *ACM Multimedia Systems Journal*, 9(3):266–278, 2003.

[3] A. Adelsbach, S. Katzenbeisser, and A.-R. Sadeghi. On the insecurity of non-invertible watermarking schemes for dispute resolving. In *International Workshop on Digital Watermarking (IWDW'03)*, volume 2939 of *Lecture Notes in Computer Science*, pages 355–369. Springer Verlag, 2004.

[4] A. Adelsbach, S. Katzenbeisser, and H. Veith. Watermarking schemes provably secure against copy and ambiguity attacks. In *ACM Workshop on Digital Rights Management (DRM'2003), Proceedings*, pages 111–119, 2003.

[5] A. Adelsbach, B. Pfitzmann, and A. Sadeghi. Proving ownership of digital content. In *Proceedings of the Third International Workshop on Information Hiding*, volume 1768 of *Lecture Notes in Computer Science*, pages 117–133. Springer, 2000.

[6] A. Adelsbach and A.-R. Sadeghi. Advanced techniques for dispute resolving and authorship proofs on digital works. In *Proceedings of the SPIE vol. 5020, Security and Watermarking of Multimedia Contents V*, pages 677–688, 2003.

[7] I. Agi and L. Gong. An empirical study of secure MPEG video transmissions. In *ISOC Symposium on Network and Distributed Systems Security*, pages 137–144, San Diego, California, 1996.

[8] Niv Ahituv, Yeheskel Lapid, and Seev Neumann. Processing encrypted data. *Commun. ACM*, 30(9):777–780, 1987.

[9] A. Alattar G. Al-Regib. Evaluation of selective encryption techniques for secure transmission of MPEG-compressed bit-streams. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages IV–340–IV–343, 1999.

[10] A. M. Alattar, G. I. Al-Regib, and S. A. Al-Semari. Improved selective encryption techniques for secure transmission of MPEG video bit-streams. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'99)*, volume 4, pages 256–260, Kobe, Japan, October 1999. IEEE Signal Processing Society.

[11] R. Anderson and R. Needham. Robustness principles for public key protocols. In Don Coppersmith, editor, *Advances in Cryptology - Crypto '95*, pages 236–247, Berlin, 1995. Springer-Verlag. Lecture Notes in Computer Science Volume 963.

[12] Olivier Baudron and Jacques Stern. Non-interactive private auctions. In *FC '01: Proceedings of the 5th International Conference on Financial Cryptography*, pages 364–378. Springer-Verlag, 2002.

[13] B. Bhargava, C. Shi, and Y. Wang. MPEG video encryption algorithms. *Multimedia Tools and Applications*, 24(1):57–79, 2004.

[14] G. R. Blakley, C. Meadows, and G. B. Purdy. Fingerprinting long forgiving messages. In *Advances in Cryptology-CRYPTO'85*, LNCS 218, pages 180–189. Springer-Verlag, 1986.

[15] Jeffrey A. Bloom, Ingemar J. Cox, Ton Kalker, Jean-Paul Linnartz, Matthew L. Miller, and B. Traw. Copy protection for DVD video. *Proceedings of the IEEE Special issue on Identification and Protection of Multimedia Information*, 87(7):1267–1276, 1999.

[16] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. In *Advances in Cryptology-CRYPTO'95*, LNCS 963, pages 452–465. Springer-Verlag, 1995.

[17] E. F. Brickell and Y. Yacobi. On privacy homomorphisms. In David Chaum and Wyn L. Price, editors, *Advances in Cryptology - EuroCrypt '87*, pages 117–126, Berlin, 1987. Springer-Verlag. Lecture Notes in Computer Science Volume 304.

[18] M. V. D. Burmester. Homomorphisms of secret sharing schemes: a tool for verifiable signature sharing. In Ueli Maurer, editor, *Advances in Cryptology - EuroCrypt '96*, pages 96–106, Berlin, 1996. Springer-Verlag. Lecture Notes in Computer Science Volume 1070.

[19] C. Cachin. Efficient private bidding and auctions with an oblivious third party. In *Proceedings of 6th ACM Conference on Computer and Communications Security*, pages 120–127. ACM Press, 1999.

[20] Chin-Chen Chang, Min-Shian Hwang, and Tung-Shou Chen. A new encryption algorithm for image cryptosystems. *J. Syst. Softw.*, 58(2):83–91, 2001.

[21] D. Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, , and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of Crypto '82*, pages 199–204, New York, USA, 1982. Plenum Publishing.

[22] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(5):1030–1044, October 1985.

[23] D. Chaum, I. B. Damgaard, and J. van de Graaf. Multiparty computations ensuring privacy of each party's input and correctness of the result. In *Advances in Cryptology - CRYPTO'87*, LNCS 293, pages 87–119. Springer-Verlag, 1988.

[24] H. Cheng and X. Li. On the application of image decomposition to image compression and encryption. In P. Horster, editor, *Communications and Multimedia Security II,*

IFIP TC6/TC11 Second Joint Working Conference on Communications and Multimedia Security, CMS '96, pages 116–127, Essen, Germany, September 1996. Chapman & Hall.

[25] H. Cheng and X. Li. Partial encryption of compressed images and videos. *IEEE Transactions on Signal Processing*, 48(8):2439–2451, 2000.

[26] Jae-Gwi Choi, Ji-Hwan Park, and Ki-Ryong Kwon. Analysis of cot-based fingerprinting schemes: New approach to design practical and secure fingerprinting scheme. In J.Fridrich, editor, *Information Hiding - IH'04*, pages 253–265, Berlin, 2004. Springer-Verlag. Lecture Notes in Computer Science Volume 3200.

[27] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

[28] T.J. Chuang and J.C. Lin. New approach to image encryption. *JEI*, 7(2):350–356, April 1998.

[29] Ingemar J. Cox, Joe Kilian, Tom Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.

[30] Ronald Cramer, Ivan Damgaard, and Jesper Buus Nielsen. Multiparty computation from threshold homomorphic encryption. In *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 280–299. Springer-Verlag, 2001.

[31] S. Craver, N. Memon, B. L. Yeo, and M. M. Yeung. Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks and implications. *IEEE Journal on Selected Areas in Communications*, 16(4):573–586, 1998.

[32] C. Crépeau, J. van de Graaf, and A. Tapp. Committed oblivious transfer and private multi-party computation. In *Advances in Cryptology-CRYPTO'95*, LNCS 963, pages 110–123. Springer-Verlag, 1995.

[33] Philippe Desarte, Benoit M. Macq, and Dirk T. M. Slock. Signal-adapted multiresolution transform for image coding. *IEEE Transactions on Information Theory*, 38(2):897–904, March 1992.

[34] W. M. Dietl and A. Uhl. Robustness against unauthorized watermark removal attacks via key-dependent wavelet packet subband structures. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '04*, Taipei, Taiwan, June 2004.

[35] Werner Dietl, Peter Meerwald, and Andreas Uhl. Protection of wavelet-based watermarking systems using filter parametrization. *Signal Processing (Special Issue on Security of Data Hiding Technologies)*, 83:2095–2116, 2003.

[36] Igor Djurovic, Srdjan Stankovic, and Ioannis Pitas. Digital watermarking in the fractional fourier transformation domain. *Journal of Network and Computer Applications*, 24:167–173, 2001.

[37] J. Domingo-Ferrer. Anonymous fingerprinting of electronic information with automatic identification of redistributors. *Electronics Letters*, 34(13):1303–1304, June 1998.

[38] J. Domingo-Ferrer. Anonymous fingerprinting based on committed oblivious transfer. In H. Imai and Y. Zheng, editors, *Public key cryptography, PKC'99*, LNCS 1560, pages 43–52. Springer-Verlag, 1999.

[39] Josep Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism. In A. Chan and V. Gligor, editors, *Information Security*, pages 471–483, Berlin, 2002. Springer-Verlag. Lecture Notes in Computer Science Volume 2433.

[40] M. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Advances in Cryptology - Eurocrypt '04*, pages 1–19, Berlin, 2004. Springer-Verlag. Lecture Notes in Computer Science Volume 3027.

[41] J. Fridrich. Image encryption based on chaotic maps. In *Proceedings of the IEEE Conf. on Systems, Man, and Cybernetics*, volume 2, pages 1105–1110. IEEE Press, 1997.

[42] Teddy Furon and Pierre Duhamel. An asymmetric public detection watermarking technique. In *IH '99: Proceedings of the Third International Workshop on Information Hiding*, pages 88–100. Springer-Verlag, 2000.

[43] K. Gopalakrishnan, Nasir Memon, and Poorvi L. Vora. Protocols for watermark verification. *IEEE MultiMedia*, 8(4):66–70, 2001.

[44] C. Griwotz. Video protection by partial content corruption. In *Multimedia and Security Workshop at the 6th ACM International Multimedia Conference*, pages 37–39, Bristol, England, 1998.

[45] C. Griwotz, O. Merkel, J. Dittmann, and R. Steinmetz. Protecting VOD the easier way. In *Proceedings of the 6th ACM Multimedia Conference*, pages 21–28, Bristol, England, 1998.

[46] Raphaël Grosbois, Pierre Gerbelot, and Touradj Ebrahimi. Authentication and access control in the JPEG 2000 compressed domain. In A.G. Tescher, editor, *Applications of Digital Image Processing XXIV*, volume 4472 of *Proceedings of SPIE*, pages 95–104, San Diego, CA, USA, July 2001.

[47] Hakan Hacigumus, Balakrishna R. Iyer, Chen Li, and Sharad Mehrotra. Executing SQL over encrypted data in the database service provider model. In *SIGMOD Conference*, 2002.

[48] F. Hartenstein. Parametrization of discrete finite biorthogonal wavelets with linear phase. In *Proceedings of the 1997 International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, April 1997.

[49] M. S. Kankanhalli and K. F. Hau. Watermarking of electronic text documents. *Electronic Commerce Research*, 2(1):169–187, 2002.

[50] S. Katzenbeisser. On the design of copyright protection protocols for multimedia distribution using symmetric and public-key watermarking. In *Proceedings of the DEXA*

2001, Fifth International Query Processing and Multimedia Issues in Distributed Systems Workshop, pages 815–819. IEEE Press, 2001.

[51] S. Katzenbeisser. On the integration of watermarks and cryptography. In *International Workshop on Digital Watermarking (IWDW'03)*, volume 2939 of *Lecture Notes in Computer Science*, pages 50–60. Springer Verlag, 2004.

[52] Jeong Hyun Kim, Seong Oun Hwang, Ki Song Yoon, and Chang Soon Park. MPEG-21 IPMP. In *First International Conference on Information Technology and Applications — ICITA 2002*, Bathurst, Australia, November 2002.

[53] H. Kiya, D. Imaizumi, and O. Watanabe. Partial-scrambling of image encoded using JPEG2000 without generating marker codes. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'03)*, volume III, pages 205–208, Barcelona, Spain, September 2003.

[54] T. Köckerbauer, M. Kumar, and A. Uhl. Lightweight JPEG 2000 confidentiality for mobile environments. In *Proceedings of the IEEE International Conference on Multimedia and Expo, ICME '04*, Taipei, Taiwan, June 2004.

[55] Thomas Kunkelmann. Applying encryption to video communication. In *Proceedings of the Multimedia and Security Workshop at ACM Multimedia '98*, pages 41–47, Bristol, England, September 1998.

[56] Thomas Kunkelmann and Rolf Reinema. A scalable security architecture for multimedia communication standards. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS'97)*, pages 660–661, Ottawa, Canada, June 1997.

[57] M. Kutter, S. Voloshynovskiy, and A. Herrigel. The watermark copy attack. In *Proceedings of the SPIE 3971, Security and Watermarking of Multimedia Contents II*, pages 371–380, 2000.

[58] Yongcheng Li, Zhigang Chen, See-Mong Tan, and Roy H. Campbell. Security enhanced MPEG player. In *Proceedings of IEEE First International Workshop on Multimedia Software Development (MMSD'96)*, pages 169–175, Berlin, Germany, 1996.

[59] T. D. Lookabaugh, D. C. Sicker, D. M. Keaton, W. Y. Guo, and I. Vedula. Security analysis of selectiveley encrypted MPEG-2 streams. In *Multimedia Systems and Applications VI*, volume 5241 of *Proceedings of SPIE*, pages 10–21, September 2003.

[60] B. Macq and J.Y. Mertes. Optimization of linear multiresolution transforms for scene adaptive coding. *IEEE Trans. on Signal Process.*, 41(12):3568–3572, 1993.

[61] N. Memon and P. W. Wong. Buyer-seller watermarking protocol based on amplitude modulation and the El Gamal public key crypto system. In *Proceedings of the SPIE 3657, Security and Watermarking of Multimedia Contents*, pages 189–294, 1999.

[62] N. Memon and P. W. Wong. A buyer–seller watermarking protocol. *IEEE Transactions on Image Processing*, 10(4):643–649, 2001.

[63] J. Meyer and F. Gadegast. Securitymechanisms for mulimedia-data with the example MPEG-I-video, 1995. unpublished, available at http://www.gadegast.de/frank/doc/secmeng.pdf.

[64] Ji Ming and SM Shen. MPEG IPMP extensions overview. ISO/IEC JTC1/SC29/WG11 N6338, March 2004.

[65] Nikos Nikolaidis and Ioannis Pitas. Copyright protection of images using robust digital signatures. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-96)*, volume 4, pages 2168–2171, 1996.

[66] R. Norcen and A. Uhl. Encryption of wavelet-coded imagery using random permutations. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, Singapure, October 2004. IEEE Signal Processing Society.

[67] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology - Eurocrypt '99*, pages 223–238, Berlin, 1999. Springer-Verlag. Lecture Notes in Computer Science Volume 1592.

[68] Cheng Peng, Robert Deng, Yongdong Wu, and Weizhong Shao. A flexible and scalable authentication scheme for JPEG2000 codestreams. In *Proceedings of ACM Multimedia 2003*, pages 433–441, San Francisco, CA, USA, November 2003.

[69] Kun Peng, Riza Aditya, Colin Boyd, Ed Dawson, and Byoungcheon Lee. Multiplicative homomorphic e-voting. In *Advances in Cryptology - Indocrypt '04*, pages 61–72, Berlin, 2004. Springer-Verlag. Lecture Notes in Computer Science Volume 3348.

[70] B. Pfitzmann and A. Sadeghi. Coin-based anonymous fingerprinting. In *Advances in Cryptology - EUROCRYPT'99*, pages 150–164, Berlin, 1999. Springer-Verlag. Lecture Notes in Computer Science Volume 1592.

[71] B. Pfitzmann and M. Schunter. Asymmetric fingerprinting. In *Advances in Cryptology—EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 84–95. Springer, 1996.

[72] B. Pfitzmann and M. Schunter. Asymmetric fingerprinting. In *Advances in Cryptology-EUROCRYPT'96*, LNCS 1070, pages 85–95. Springer-Verlag, 1996.

[73] B. Pfitzmann and M. Waidner. Asymmetric fingerprinting for larger collusions. In *Proceedings of the 4th ACM Conference on Computer and Communicatios Security*, pages 151–160, 1997.

[74] A. Pommer. *Selective Encryption of Wavelet-compressed Visual Data*. PhD thesis, University of Salzburg, Austria, June 2003.

[75] A. Pommer and A. Uhl. Wavelet packet methods for multimedia compression and encryption. In *Proceedings of the 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 1–4, Victoria, Canada, August 2001. IEEE Signal Processing Society.

[76] A. Pommer and A. Uhl. Selective encryption of wavelet packet subband structures for obscured transmission of visual data. In *Proceedings of the 3rd IEEE Benelux Signal Processing Symposium (SPS 2002)*, pages 25–28, Leuven, Belgium, March 2002. IEEE Benelux Signal Processing Chapter.

[77] A. Pommer and A. Uhl. Selective encryption of wavelet packet subband structures for secure transmission of visual data. In J. Dittmann, J. Fridrich, and P. Wohlmacher, editors, *Multimedia and Security Workshop, ACM Multimedia*, pages 67–70, Juan-les-Pins, France, December 2002.

[78] A. Pommer and A. Uhl. Selective encryption of wavelet-packet encoded image data — efficiency and security. *ACM Multimedia Systems (Special issue on Multimedia Security)*, 9(3):279–287, 2003.

[79] L. Qiao and K. Nahrstedt. Watermarking schemes and protocols for protecting rightful ownerships and customer's rights. *Journal of Visual Communication and Image Representation*, 9(3):194–210, 1998.

[80] Lintian Qiao and Klara Nahrstedt. A new algorithm for MPEG video encryption. In *Proceedings of the International Conference on Imaging Science, Systems, and Technology, CISST '97*, pages 21–29, Las Vegas, NV, USA, June 1997.

[81] Lintian Qiao and Klara Nahrstedt. Comparison of MPEG encryption algorithms. *International Journal on Computers and Graphics (Special Issue on Data Security in Image Communication and Networks)*, 22(3):437–444, 1998.

[82] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. In R.A. DeMillo et al., editor, *Foundations of Secure Computation*, pages 169–179, New-York, 1978. Academic Press.

[83] Ahmad-Reza Sadeghi. How to break a semi-anonymous fingerprinting scheme. In I.S. Moskowitz, editor, *Information Hiding - IH'01*, pages 384–394, Berlin, 2001. Springer-Verlag. Lecture Notes in Computer Science Volume 2137.

[84] T. Sander and C.F. Tschudin. Protecting mobile agent against malicious hosts. In *Mobile Agents and Security*, LNCS 1419, pages 44–60. Springer-Verlag, 1998.

[85] J. Scharinger. Fast encryption of image data using chaotic kolmogorov flows. *JEI*, 7(2):318–325, April 1998.

[86] J. Schneid and S. Pittner. On the parametrization of the coefficients of dilation equations for compactly supported wavelets. *Computing*, 51:165–173, May 1993.

[87] B. Schneier. *Applied Cryptography*. John Wiley & Sons, Inc., 1996.

[88] WG1, SC29 Secretariat. Resolutions of the 32^{nd} ISO/IEC JTC 1/SC 29/WG1 meeting, Madrid, Spain, 2004-03-29 to 2004-04-02. URL: `http://www.itscj.ipsj.or.jp/sc29/open/29view/29n59691.pdf`, 2004-07-14.

[89] Yong-Seok Seo, Min-Su Kim, Ha-Joong Park, Ho-Youl Jung, Hyun-Yeol Chung, Young Huh, and Jae-Duck Lee. A secure watermarking for JPEG-2000. In *Proceedings of the*

IEEE International Conference on Image Processing (ICIP'01), Thessaloniki, Greece, October 2001.

[90] C. Shi and B. Bhargava. An efficent MPEG video encryption algorithm. In *Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems*, pages 381–386, West Lafayette, Indiana, USA, 1998.

[91] C. Shi and B. Bhargava. A fast MPEG video encryption algorithm. In *Proceedings of the Sixth ACM International Multimedia Conference*, pages 81–88, Bristol, UK, September 1998.

[92] C. Shi and B. Bhargava. Light-weight MPEG video encryption algorithm. In *Proceedings of the International Conference on Multimedia (Multimedia98 Shaping the Future)*, pages 55–61, New Delhi, India, January 1998.

[93] Changgui Shi, Sheng-Yih Wang, and Bharat Bhargava. MPEG video encryption in real-time using secret key cryptography. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, pages 2822–2829, Las Vegas, Nevada, USA, 1999.

[94] S.U. Shin, K.S. Sim, and K.H. Rhee. A secrecy scheme for MPEG video data using the joint of compression and encryption. In *Proceedings of the 1999 Information Security Workshop (ISW'99)*, volume 1729 of *Lecture Notes on Computer Science*, pages 191–201, Kuala Lumpur, November 1999. Springer-Verlag.

[95] D. Simitopoulos, N. Zissis, P. Georgiadis, V. Emmanouilidis, and M. G. Strintzis. Encryption and watermarking for the secure distribution of copyrighted MPEG video on DVD. *ACM Multimedia Systems (Special issue on Multimedia Security)*, 9(3):217–227, 2003.

[96] G. Spanos and T. Maples. Performance study of a selective encryption scheme for the security of networked real-time video. In *Proceedings of the 4th International Conference on Computer Communications and Networks (ICCCN'95)*, Las Vegas, NV, 1995.

[97] Paul Syverson. Limitations on design principles for public key protocols. In *SP '96: Proceedings of the 1996 IEEE Symposium on Security and Privacy*, page 62. IEEE Computer Society, 1996.

[98] N. Taesombut, R. Huang, and V. P. Rangan. A secure multimedia system in emerging wireless home networks. In A. Lioy and D. Mazzocchi, editors, *Communications and Multimedia Security. Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security, CMS '03*, volume 2828 of *Lecture Notes on Computer Science*, pages 76–88, Turin, Italy, October 2003. Springer Verlag.

[99] L. Tang. Methods for encrypting and decrypting MPEG video data efficiently. In *Proceedings of the ACM Multimedia 1996*, pages 219–229, Boston, USA, November 1996.

[100] P. Tomsich and S. Katzenbeisser. Copyright protection protocols for multimedia distribution based on trusted hardware. In *Protocols for Multimedia Systems (PROMS 2000)*, pages 249–256, Cracow (Poland), 2000.

[101] P. Tomsich and S. Katzenbeisser. Towards a robust and de-centralized digital watermarking infrastructure for the protection of intellectual property. In *Electronic Commerce and Web Technologies, First International Conference (ECWEB 2000)*, volume 1875 of *Lecture Notes in Computer Science*, pages 38–47. Springer, 2000.

[102] T. Uehara, R. Safavi-Naini, and P. Ogunbona. Securing wavelet compression with random permutations. In *Proceedings of the 2000 IEEE Pacific Rim Conference on Multimedia*, pages 332–335, Sydney, December 2000. IEEE Signal Processing Society.

[103] A. Uhl. Image compression using non-stationary and inhomogeneous multiresolution analyses. *Image and Vision Computing*, 14(5):365–371, 1996.

[104] A. Uhl. Generalized wavelet decompositions in image compression: arbitrary subbands and parallel algorithms. *Optical Engineering*, 36(5):1480–1487, 1997.

[105] A. Uhl and A. Pommer. Are parameterised biorthogonal wavelet filters suited (better) for selective encryption? In Jana Dittmann and Jessica Fridrich, editors, *Multimedia and Security Workshop 2004*, pages 100–106, Magdeburg, Germany, September 2004.

[106] A. Uhl and A. Pommer. *Image and Video Encryption. From Digital Rights Management to Secured Personal Communication*, volume 15 of *Advances in Information Security*. Springer-Verlag, 2005.

[107] G. Unnikrishnan and Kehar Singh. Double random fractional fourier-domain encoding for optical security. *Optical Engineering*, 39(11):2853–2859, November 2000.

[108] L. Vorwerk, T. Engel, and C. Meinel. A proposal for a combination of compression and encryption. In *Visual Communications and Image Processing 2000*, volume 4067 of *Proceedings of SPIE*, pages 694–702, Perth, Australia, June 2000.

[109] S.J. Wee and J.G. Apostolopoulos. Secure scalable streaming and secure transcoding with JPEG2000. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'03)*, volume I, pages 547–551, Barcelona, Spain, September 2003.

[110] Jiangtao Wen, Mike Severa, Wenjun Zeng, Max Luttrell, and Weiyin Jin. A format-compliant configurable encryption framework for access control of multimedia. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing, MMSP '01*, pages 435–440, Cannes, France, October 2001.

[111] Jiangtao Wen, Mike Severa, Wenjun Zeng, Max Luttrell, and Weiyin Jin. A format-compliant configurable encryption framework for access control of video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6):545–557, June 2002.

[112] Chung-Ping Wu and C.-C. Jay Kuo. Fast encryption methods for audiovisual data confidentiality. In *SPIE Photonics East - Symposium on Voice, Video, and Data Communications*, volume 4209, pages 284–295, Boston, MA, USA, November 2000.

[113] Chung-Ping Wu and C.-C. Jay Kuo. Efficient multimedia encryption via entropy codec design. In *Proceedings of SPIE, Security and Watermarking of Multimedia Contents III*, volume 4314, San Jose, CA, USA, January 2001.

[114] Tsung-Li Wu and S. Felix Wu. Selective encryption and watermarking of MPEG video (extended abstract). In Hamid R. Arabnia, editor, *Proceedings of the International Conference on Image Science, Systems, and Technology, CISST '97*, Las Vegas, USA, February 1997.

[115] Yongdong Wu and Robert H. Deng. Compliant encryption of JPEG2000 codestreams. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, Singapure, October 2004. IEEE Signal Processing Society.

[116] Yongdong Wu and Robert H. Deng. Progressive protection of JPEG2000 codestreams. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, Singapure, October 2004. IEEE Signal Processing Society.

[117] Makoto Yokoo and Koutarou Suzuki. Secure multi-agent dynamic programming based on homomorphic encryption and its application to combinatorial auctions. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 112–119. ACM Press, 2002.

[118] W. Zeng, J. Wen, and M. Severa. Fast self-synchronous content scrambling by spatially shuffling codewords of compressed bitstreams. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'02)*, September 2002.

[119] Wenjun Zeng and Shawmin Lei. Efficient frequency domain video scrambling for content access control. In *Proceedings of the seventh ACM International Multimedia Conference 1999*, pages 285–293, Orlando, FL, USA, November 1999.

[120] Wenjun Zeng and Shawmin Lei. Efficient frequency domain selective scrambling of digital video. *IEEE Transactions on Multimedia*, 5(1):118–129, March 2003.

[121] Jian Zhao and Eckard Koch. Embedding robust labels into images for copyright protection. In *Proceedings of the International Conference on Intellectual Property Rights for Information, Knowledge and New Techniques*, pages 242–251, München, Wien, 1995. Oldenbourg Verlag.