A preliminary version appears in Information Hiding 2007, Lecture Notes in Computer Science, Springer-Verlag, 2007.

Security of Invertible Media Authentication Schemes Revisited

Daniel Dönigus Stefan Endler Marc Fischlin Andreas Hülsing
Patrick Jäger Anja Lehmann Sergey Podrazhansky
Sebastian Schipp Erik Tews Sven Vowe Matthias Walthart
Frederik Weidemann

Darmstadt University of Technology, Germany
marc.fischlin@gmail.com www.minicrypt.de

Abstract. Dittmann, Katzenbeisser, Schallhart and Veith (IACR ePrint 2004) introduced the notion of invertible media authentication schemes, embedding authentication data in media objects via invertible watermarks. These invertible watermarks allow to recover the original media object (given a secret encryption key), as required for example in some medical applications where the distortion must be removable.

Here we revisit the approach of Dittmann et al. from a cryptographic viewpoint, clarifying some important aspects of their security definitions. Namely, we first discuss that their notion of unforgeability may not suffice in all settings, and we therefore propose a strictly stronger notion. We then show that the basic scheme suggested by Dittmann et al. achieves our notion if instantiated with the right cryptographic primitives. Our proof also repairs a flaw in the original scheme, pointed out by Hopper, Molnar and Wagner (TCC 2007).

We finally address the issue of secrecy of media authentication schemes, basically preventing unauthorized recovering of the original media object without the encryption key. We give a rigorous security statement (that is, the best security guarantee we can achieve) and prove again that the scheme by Dittmann et al. meets this security level if the right cryptographic building blocks are deployed. Together our notions of unforgeability and of secrecy therefore give very strong security guarantees for such media authentication schemes.

1 Introduction

The transition from analog to digital media facilitates many tasks but also comes along with continually improved manipulation tools, which allow various modifications of media objects. Thus, it becomes increasingly difficult to distinguish authentic from altered objects. To enable a better distinction it is therefore necessary to apply techniques that guarantee authenticity, integrity and possibly secrecy of data.

The straightforward use of digital signatures is not always a satisfying solution to provide authenticity and integrity, because an object and its signature have to be stored separately.

This, however, may not be convenient in the area of multimedia data. To counter this problem fragile watermarks were proposed, which can be used to embed a signature directly into an object, such that any (significant) modification will destroy the watermark and thereby invalidates the signature. Unfortunately, this approach comes with the disadvantage that it always leads to irrevocable distortions in the authenticated object, which may not be acceptable in all applications, e.g., X-ray imaging objects are extremely sensitive to modifications. One solution is to use invertible watermarking schemes, which are special fragile watermarks addressing the need to re-obtain the original media object by allowing a complete removal of the embedded data.

Media Authentication Schemes. Using invertible watermarking schemes in combination with encryption and digital signatures, Dittmann, Katzenbeisser, Schallhart and Veith (DKSV) [DKSV04] introduced the notion of an invertible media authentication scheme that allows reconstruction of the original object. They also propose a framework to build such authentication schemes: To protect a media object O the $\mathcal{MAS}_{\mathrm{DKSV}}$ scheme first applies an invertible watermarking scheme as proposed by Fridrich et al. [FGD02], dividing O into two parts A_O, B_O by running the watermarking algorithm Separate. See Figure 1. The part B_O next gets compressed and encrypted to a ciphertext X that is stored as the first part of the watermark. To achieve an appropriate compressibility level, B_O has to be chosen accordingly. The second part of the watermark contains the digital signature s of the encrypted part X and A_O, the public part of the object. Finally, the watermark (X, s) is joined with A_O to a single protected object $\overline{O}$ by using the watermarking algorithm Join.

Reconstruction of the original object from $\overline{O}$ is done by decrypting to recover the part Compress(B_O) and uncompressing this value to get the part B_O. A simple join operation with A_O merges the parts together again. As for integrity and secrecy, as long as the object is not altered the signature can be verified by using the public verification key, while the reconstruction of the original object is protected by the secret reconstruction key.

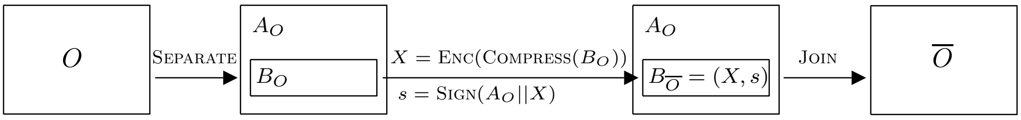
O
Separate
A_O
B_O
$X = \text{Enc}(\text{Compress}(B_O))$
$s = \text{Sign}(A_O||X)$
A_O
$B_{\overline{O}} = (X, s)$
Join
$\overline{O}$

Figure 1: Protection of media objects in the $\mathcal{MAS}_{\mathrm{DKSV}}$ Scheme

In contrast to most known watermarking schemes where the security is only analyzed by ad-hoc methods, the media authentication scheme of Dittmann et al. comes with a formal model against malicious modification attempts, following well-known approaches for signature schemes. In [DKSV04] a media authentication scheme is called secure against forgeability if for every adversary it is infeasible to produce an object O and its protected version $\overline{O}$ for a given verification key. This should even hold if the adversary may ask for protected versions of other objects before.

Our Results (Integrity). Demanding from the adversary to output a pair $(O, \overline{O})$ seems to be overly restrictive, since the authentication system should be already considered broken if an adversary merely creates an arbitrary authenticated object $\overline{O}$ (without knowing a corresponding original object O). Consider for example a governmental organization publishing satellite data $\overline{O}$ of which parts may be classified as confidential for issues of national security (contained in the encrypted B_O part), but which should still allow public verification of authenticity. In this case, the adversary's goal could be to produce other partially protected satellite data bearing a correct signature of the governmental authority, but without any need of being able to generate a matching unprotected object. In this case, the unforgeability definition of Dittmann et al. would provide no security guarantee.

Therefore we propose a stronger definition of unforgeability, which we call strong unforgeability and which prevents attacks like the one above. To show that our definition is indeed strictly stronger than the definition of Dittmann et al., we first give a proof that strong unforgeability implies (basic) unforgeability. After that, we present two examples of media authentication schemes which are secure according to the basic notion, but not according to our enhanced definition.

Before proving that the original scheme of Dittmann et al. [DKSV04] can be lifted to satisfy the notion of strong unforgeability, we need to tweak the signing process. Hopper et al. revealed in [HMW07] that, in the original scheme, an adversary can easily find different objects that generate the same input $A_O||X$ to the signing resp. verification process and thus straightforwardly constitute a forgery. We show that those attacks can be prevented by using an appropriate encoding for computing the signature, where A_O and X are clearly separated. Together with a strongly unforgeable signature scheme, this also provides a sufficient condition for a strongly unforgeable media authentication scheme.

Our Results (Secrecy). Another security aspect considered in our paper is secrecy of the original data contained in the protected object. In order to achieve a secure protection of the B_O part, Dittmann et al. [DKSV04] propose to use a symmetric encryption scheme. Unfortunately, they neither provide any rigorous security model, nor make any conclusions about the secrecy of their scheme.

In a companion paper, Katzenbeisser and Dittmann [KD04] discuss a desirable secrecy requirement, resembling semantic security of encryption schemes [GM84] where a ciphertext should not reveal anything about the original message. In [KD04] the authors conclude that a similar notion for media authentication schemes "might not be possible to satisfy" because, due to the requirement of good compressibility, the protected part B_O is typically not completely random and may statistically depend on the public part A_O. Therefore, an adversary may be able to derive some information about the encrypted part from the public part A_O. In [KD04] the authors thus outline an alternative (and somewhat non-standard) security definition, but remain rather informal and do not prove that the $\mathcal{MAS}_{\mathrm{DKSV}}$ scheme achieves the desired level of secrecy.

Our starting point is to note that the fact that A_O may reveal some information about B_O does not obviate similar claims about the secrecy for the media authentication scheme. The reason, originating in the context of encryption, is that the precise idea of semantic security is that one should not be able to learn anything about a message m from a ciphertext X *than what is known about m anyway.* For instance, if m is a credit card number sent encrypted, but the card type is transmitted in clear, then the first digit is usually deducible from the type. Secrecy with respect to such side information is therefore the highest security level we can achieve and should aim for.

Adapting the notion of semantic security with side information we give a formal definition of secrecy for media authentication schemes. Our definition basically says that an authentication scheme is considered secure if whatever can be computed from a protected object $\overline{O} = (A_O, B_{\overline{O}})$ could also be derived from the public part A_O alone. We can even strengthen our notion to a more realistic scenario where the adversary is able to obtain protected and reconstructed objects of his choice. Based on the formal definition we then consider the secrecy of the media authentication scheme by Dittmann et al. and show that semantic security of the used encryption function is a sufficient condition for the authentication scheme to be semantically secure as well.

Summary. Overall, this paper here complements the work of Dittmann et al. by giving precise security models that describe the guarantees in terms of integrity and secrecy. We introduce the notion of strong unforgeability to strengthen the security against malicious modification attempts and provide the sufficient requirements for an authentication scheme to achieve this security goal. Furthermore we show that secrecy in the sense of semantic security for media authentication schemes can be defined, which is completed by proving secrecy for the construction of Dittmann et al. under reasonable assumptions about the encryption scheme.

Organization. In Section 2 we recall the definition of an invertible media authentication scheme by Dittmann et al. [DKSV04]. In Section 3 we introduce the scheme (or, to be precise, the framework) by Dittmann et al. and the underlying tools (watermarking, encryption and signatures). Section 4 deals with our refinement of integrity of media authentication schemes and relates the notions, whereas Section 5 covers the secrecy aspects of such schemes. We note that, following the terminology of [DKSV04], in this paper here we deal with *offline* media authentication only. It is easy to adapt our notions and proofs to the case of *online* media authentication; we refer to Appendix A for details.

2 Media Authentication Schemes

An invertible media authentication scheme ($\mathcal{MAS}$), defined by Dittmann et al. [DKSV04], consists of a set of algorithms allowing to protect a media object. More precisely, an invertible $\mathcal{MAS}$ is able to produce a protected media object using the algorithm Protect while retaining the ability to losslessly reconstruct the original media object using algorithm Reconstruct. The ability for lossless reconstruction of protected media objects is typically achieved by using invertible watermarking schemes as introduced by Honsinger et al. [HJRS99]. If a media object has been previously protected, its integrity can be unambiguously verified using algorithm Verify.

Usage of the above algorithms necessitates cryptographic keys for protection as well as reconstruction of media objects, which have to be kept private. However, verification of the integrity of a protected media object assumes a public verification key, thus enabling integrity checks by third parties. The generation of all necessary keys is summarized in a single algorithm GenKey, which takes as input a security parameter and selects keys of the corresponding strength.

Definition 2.1 *An* invertible media authentication scheme *is a tuple of probabilistic polynomial-time algorithms*

$$\mathcal{MAS} = (\textsc{GenKey}, \textsc{Protect}, \textsc{Verify}, \textsc{Reconstruct})$$

with the following properties:

- $\textsc{GenKey}$ *takes as input a security parameter* n *(in unary, as* 1^n*) and outputs a triple of keys* (K_P, K_V, K_R)*, where* K_P *is the secret protection key,* K_V *is the public verification key and* K_R *is the secret reconstruction key.*

- $\textsc{Protect}$ *takes as input a media object* O *and a protection key* K_P*, and outputs a protected media object* $\overline{O}$ *or* FAIL*, if protection is not possible.*

- $\textsc{Verify}$ *accepts as input a protected media object* $\overline{O}$ *and a verification key* K_V*, and outputs either* TRUE *or* FALSE*.*

- $\textsc{Reconstruct}$ *takes a protected media object* $\overline{O}$ *and a reconstruction key* K_R*, and outputs a media object* O *or* FAIL*.*

Furthermore, we require that verification and reconstruction for valid protected objects always succeeds, i.e., for any media object O*, for all keys* $(K_P, K_V, K_R) \leftarrow \textsc{GenKey}(1^n)$ *and any* $\overline{O} \leftarrow \textsc{Protect}(O, K_P)$*, we have*

$$\begin{aligned} \mathrm{Prob}\left[\textsc{Verify}(\overline{O}, K_V) = \mathsf{TRUE} \;\middle|\; \overline{O} \neq \mathsf{FAIL}\right] &= 1, \\ \mathrm{Prob}\left[\textsc{Reconstruct}(\overline{O}, K_R) = O \;\middle|\; \overline{O} \neq \mathsf{FAIL}\right] &= 1. \end{aligned}$$

3 The DKSV Media Authentication Scheme

In this section we first recall the basic ingredients of the media authentication scheme by Dittmann et al. [DKSV04], before presenting the actual $\mathcal{MAS}_{\mathrm{DKSV}}$ scheme.

3.1 Tools

Recall that the basic idea of the $\mathcal{MAS}_{\mathrm{DKSV}}$ scheme is to divide the object O into a public part A_O and a part B_O which should be protected. This splitting (and its inverse process) are performed via an invertible watermarking scheme, as described in Section 3.1.1. The B_O part is then compressed, encrypted and signed. Encryption and Signatures are therefore described formally in Sections 3.1.2 and 3.1.3.

3.1.1 Watermarking

Watermarking schemes are an alternative to the concept of cryptographic signatures, designed specifically to embed authentication and integrity data within media objects, thus eliminating the need for separate storage. They occur in different security scenarios:

- *Fragile* watermarking schemes [YM98] may be employed if the integrity of a media object needs to be proven to deem its contents authentic. They are designed to be instantly destroyed when the media object is tampered with.

- *Robust* watermarking schemes [SBT96] may be employed if for instance the origin of a media object needs to be determined to trace illicit reproduction. Robust watermarks withstand most digital processing operations in video clips and digital images and can be recognized even after several alterations. However, in order to provide such a tamper-resistant method, straightforward usage of cryptographic signatures is all but impossible.

Since robust watermarking schemes are inherently unable to retain cryptographic signatures in a coherent manner, their usuage is infeasible in our scenario of media authentication schemes providing secrecy and integrity in a provable way. On the other hand, when employing fragile watermarking schemes, the embedding process induces distortions into the original media object, thus inevitably altering the original. Although sophisticated embedding algorithms induce a barely visible distortion into the media object, a lossless reconstruction may be desirable.

Invertible watermarking schemes are often a special case of fragile watermarks and have been introduced by Honsinger et al. [HJRS99] to address the need to re-obtain the original media object. Fridrich et al. [FGD02] later proposed a general framework for invertible watermarking schemes that uses lossless compression to allow the reversion of the embedding process. Thereby, the ability to embed data into a media object O is accomplished by two polynomial-time algorithms $\textsc{Join}$ and $\textsc{Separate}$:[1]

- $\textsc{Separate}$ takes a media object O as input and produces a tuple (A_O, B_O) (or the output FAIL),
- $\textsc{Join}$ takes a pair (A'_O, B'_O) as input and returns a media object O' (or the output FAIL).

If the following equalities hold, $\textsc{Join}(\textsc{Separate}(O)) = O$ (given $\textsc{Separate}(O) \neq \mathsf{FAIL}$) for any object O, and $\textsc{Separate}(\textsc{Join}(A_O, B_O)) = (A_O, B_O)$ (given that $\textsc{Join}(A_O, B_O) \neq \mathsf{FAIL}$) for all A_O, B_O, then we call the pair $(\textsc{Join}, \textsc{Separate})$ an *invertible watermarking scheme*.

Note that the completeness condition above also provides some sort of collision-resistance for the $\textsc{Separate}$ algorithm. Namely, for any objects $O \neq O'$ with $\textsc{Separate}(O) \neq \mathsf{FAIL}$, $\textsc{Separate}(O') \neq \mathsf{FAIL}$ we must have $\textsc{Separate}(O) \neq \textsc{Separate}(O')$. Otherwise, if $\textsc{Separate}$ returned the same output for some $O \neq O'$, then $\textsc{Join}$ would sometimes fail to recover the right object O or O' from these identical outputs. The analogous argument applies to $\textsc{Join}$. We note that we could also use a relaxed version in which "bad" objects $O \neq O'$ may exist, but then they are hard to find in reasonable time (similar to collision-resistance of hash functions). Our results remain valid under this relaxed version.

When using $\textsc{Separate}$ in invertible watermarking schemes, visually insignificant data is usually assigned to the second part B_O, which will afterwards be losslessly compressed. The newly gained free space is then used to embed secrecy and integrity data. This enhanced part of a media object will be denoted with $B_{\overline{O}}$. After the embedding process, A_O and B_O are reassembled using $\textsc{Join}$ to form the protected media object $\overline{O}$.

However, since a compression algorithm is necessary during protection, we have to keep in mind that the media object is expected to contain a certain amount of redundancy to be compressed densely enough since a minimum set of secrecy and integrity data needs to be embedded within the freed space. With negligible possibility, a media object O may fail to meet the minimum requirement. In such a case, the invertible watermarking scheme fails.

3.1.2 Encryption

A symmetric encryption scheme $\mathcal{E} = (\textsc{GenEnc}, \textsc{Enc}, \textsc{Dec})$ consists of three probabilistic polynomial-time algorithms, where algorithm $\textsc{GenEnc}$ on input 1^n generates a key K_E,

[1] These algorithms are often defined to be initialized with a watermarking key K_W. Here we presume for simplicity that this key is "hardwired" into the description of the algorithms, or that the key is available to all parties as a system parameter. The key K_W may also contain randomness for both algorithms (if required).

algorithm $\textsc{Enc}$ on input K_E and message $m \in \{0,1\}^*$ outputs a ciphertext X, and algorithm $\textsc{Dec}$ also takes K_E and a ciphertext X and returns $m \in \{0,1\}^*$ or FAIL. Furthermore, for all keys K_E produced by $\textsc{GenEnc}(1^n)$, all messages $m \in \{0,1\}^*$ and ciphertexts $X \leftarrow \textsc{Enc}(K_E, m)$, we have $m = \textsc{Dec}(K_E, X)$.

As for security of encryption schemes we follow the idea of semantic security, as defined by Goldwasser and Micali [GM84]. Informally, the idea of semantic security for encryption schemes is that any information $f_{\text{enc}}(m)$ an efficient adversary could learn about a message m from a ciphertext X could also be computed efficiently without X. All this holds of course relative to any side information about m. This extra knowledge about m is typically formalized by having some side information hist_m about the message m.

For notational convenience we denote by $(m, \text{hist}_m) \leftarrow (\mathcal{M}, \text{hist}_{\text{enc}})(1^n)$ the joint sampling process in which the message m is picked according to distribution $\mathcal{M}(1^n)$ and, at the same time, side information hist_m is generated according to algorithm $\text{hist}_{\text{enc}}(1^n)$. Note that in this process both algorithms $\mathcal{M}$ and hist_{enc} may share state.

Definition 3.1 *A symmetric encryption scheme $\mathcal{E} = (\textsc{GenEnc}, \textsc{Enc}, \textsc{Dec})$ is called* semantically secure (with respect to side information hist_{enc}) *if for every probabilistic polynomial-time algorithm $\mathcal{A}_{enc}$ there is a probabilistic polynomial-time algorithms $\mathcal{S}_{enc}$, the simulator, such that for every polynomial-time distribution $\mathcal{M}$ and any function f_{enc} the difference*

$$\text{Prob}\left[\boldsymbol{Exp}_{\mathcal{E},\mathcal{A}_{enc}}^{sem,\mathcal{M},f_{enc},hist_{enc}}(n) = 1\right] - \text{Prob}\left[\boldsymbol{Exp}_{\mathcal{E},\mathcal{S}_{enc}}^{sem,\mathcal{M},f_{enc},hist_{enc}}(n) = 1\right]$$

is negligible, where

```
Experiment $\boldsymbol{Exp}_{\mathcal{E},\mathcal{A}_{enc}}^{sem,\mathcal{M},f_{enc},hist_{enc}}(n)$
    $K_E \leftarrow \textsc{GenEnc}(1^n)$
    $(m, hist_m) \leftarrow (\mathcal{M}, hist_{enc})(1^n)$
    $X \leftarrow \textsc{Enc}(K_E, m)$
    $a \leftarrow \mathcal{A}_{enc}(1^n, X, hist_m)$
    output 1 if and only if
        $a = f_{enc}(m, hist_m)$
```

```
Experiment $\boldsymbol{Exp}_{\mathcal{E},\mathcal{S}_{enc}}^{sem,\mathcal{M},f_{enc},hist_{enc}}(n)$
    $K_E \leftarrow \textsc{GenEnc}(1^n)$
    $(m, hist_m) \leftarrow (\mathcal{M}, hist_{enc})(1^n)$

    $a \leftarrow \mathcal{S}_{enc}(1^n, hist_m)$
    output 1 if and only if
        $a = f_{enc}(m, hist_m)$
```

We note that Dittmann et al. [DKSV04] do not make any security claim about the underlying encryption scheme in their $\mathcal{MAS}$. See also the discussion in Section 5. Finally, we remark that semantic security (with respect to any side information) is a very common property of modern encryption schemes, and is usually met by all practical and theoretical solutions (cf. [Gol04]).

3.1.3 Signature Schemes

A signature scheme $\mathcal{S} = (\textsc{GenSign}, \textsc{Sign}, \textsc{SigVerify})$ consists of probabilistic polynomial-time algorithms such that algorithm $\textsc{GenSign}$ on input 1^n generates a key pair $(K_{VS}, K_{SS}) \leftarrow \textsc{GenSign}(1^n)$, algorithm $\textsc{Sign}$ for input K_{SS} and a message $m \in \{0,1\}^*$ outputs a signature $s \leftarrow \textsc{Sign}(K_{SS}, m)$, and algorithm $\textsc{SigVerify}$ for input K_{VS}, m and s returns a decision $d \leftarrow \textsc{SigVerify}(K_{VS}, m, s)$ which is either TRUE or FALSE. Additionally, for all security parameters n, all keys $(K_{VS}, K_{SS}) \leftarrow \textsc{GenSign}(1^n)$, all messages $m \in \{0,1\}^*$ and all signatures $s \leftarrow \textsc{Sign}(K_{SS}, m)$ it holds $\textsc{SigVerify}(K_{VS}, m, s) = \mathsf{TRUE}$.

Below we define a strong notion of security for signature schemes, called strong unforgeability, which supersedes the common notion of unforgeability for signatures (cf. [Gol04]).

Roughly, strong unforgeability also prevents the adversary from producing new signatures for previously signed messages (even if the adversary can see other signatures for chosen message through a signature oracle $\textsc{Sign}(K_{SS},\cdot)$):

Definition 3.2 *A signature scheme $\mathcal{S} = (\textsc{GenSign}, \textsc{Sign}, \textsc{SigVerify})$ is called strongly unforgeable if for any probabilistic polynomial-time algorithm $\mathcal{A}_{sig}$,*

$$\text{Prob}\left[\boldsymbol{Exp}^{StUnf}_{\mathcal{S},\mathcal{A}_{sig}}(n) = 1\right]$$

is negligible, where

```
**Experiment** $\boldsymbol{Exp}^{StUnf}_{\mathcal{S},\mathcal{A}_{sig}}(n)$
    $(K_{VS}, K_{SS}) \leftarrow \textsc{GenSign}(1^n)$
    $(m^*, s^*) \leftarrow \mathcal{A}_{sig}^{\textsc{Sign}(K_{SS},\cdot)}(K_{VS})$,
        where we let $m_i$ denote the $i$-th query to oracle $\textsc{Sign}(K_{SS},\cdot)$
        and $s_i$ the oracle's answer to this query
    output 1 if and only if
        $\textsc{SigVerify}(K_{VS}, m^*, s^*) = \mathsf{TRUE}$ and
        $(m^*, s^*) \neq (m_i, s_i)$ for all $i$.
```

Note that in the regular notion of unforgeability we strengthen the requirement on (m^*, s^*) in the experiment above, and demand that $m^* \neq m_i$ for all i (such that finding another signature s^* to a given pair m_i, s_i is no longer considered a successful attack). In particular, if a scheme is strongly unforgeable, then it is also unforgeable in the basic sense. Yet, it is also easy to construct an unforgeable signature scheme which does not achieve the stronger notion, e.g., if for each signature the signing algorithm appends a redundant bit which the verification algorithm simply ignores.

Efficient strongly unforgeable signature schemes exist both in the random oracle model [BR96, BLS04] and in the standard model [CS00, Fis03, BSW06]. Existentially they can be derived from any one-way function [NY89, Rom90, Gol04] and are thus based on the same complexity assumption as signature schemes which are unforgeable in the ordinary sense.

3.2 The $\mathcal{MAS}_{\textbf{DKSV}}$ Scheme

With the tools of the previous sections we can now recapture the $\mathcal{MAS}_{\text{DKSV}}$ scheme. To protect a media object O the $\mathcal{MAS}_{\text{DKSV}}$ scheme first uses the watermarking scheme to determine the parts A_O and B_O. Then the B_O part is first compressed to C_O and, together with a hash value $H(O)$ of the object, encrypted to a ciphertext X.[2] The resulting ciphertext and the public part A_O of the original media object O are signed together with the signature algorithm, $s \leftarrow \textsc{Sign}(K_{SS}, (A_O, X))$. The values X and s are finally joined with A_O into a single media object $\overline{O}$.

The integrity of a protected object $\overline{O}$ can be verified by anyone by recovering A_O, X, s from the protected object and verifying the signature s for (A_O, X). This can be done without decrypting X and recovering B_O. Reconstruction then can easily be achieved by first verifying $\overline{O}$ and then decrypting with K_E. After uncompressing C_O to B'_O algorithm

[2] The role of $H(O)$ concerning the security of the scheme remains somewhat unclear, i.e., Dittmann et al. [DKSV04] never specify any security requirements on H. It appears that security-wise H does not serve any purpose. We include H here only for sake of completentess; the reader may simply think of H as the function with empty output.

$\textsc{Join}$ can be applied to (A_O, B'_O). The resulting object O' is hashed to $H(O')$ which is compared to the embedded hash. If this is successful the restored object is returned as O, otherwise the reconstruction algorithm fails.

We note that, in the original scheme, Dittmann et al. use the signature algorithm to sign the concatenation $A_O||X$ of the values A_O and X. But this introduces a weaknesses which the attack by Hopper et al. [HMW07] exploits. Here we therefore tweak the signature process by signing (A_O, X) instead, with the usual meaning that this string (A_O, X) contains a separator between the two values. For instance, we can encode the bit length of A_O into a starting block of fixed length (say, into the first n bits for security parameter n) and then append $A_O||X$. Other choices are possible, of course.

Construction 3.3 (DKSV-MAS) *Let* $(\textsc{Join}, \textsc{Separate})$ *be an invertible watermarking scheme, $\mathcal{E}$ be a symmetric encryption scheme and $\mathcal{S}$ be a signature scheme. Furthermore, let* $(\textsc{Compress}, \textsc{Uncompress})$ *be a lossless compression scheme and H be some function (with fixed output length). Then the DKSV media authentication scheme $\mathcal{MAS}_{DKSV}$ is defined by the following algorithms:*

- *Algorithm* $\textsc{GenKey}$ *on input 1^n runs the key generation algorithms of the signature scheme and the encryption scheme,* $(K_{SS}, K_{VS}) \leftarrow \textsc{GenSign}(1^n)$ *and* $K_E \leftarrow \textsc{GenEnc}(1^n)$*, and outputs $K_V = K_{VS}$, $K_R = (K_{VS}, K_E)$ and $K_P = (K_{SS}, K_E)$.*

- *Algorithm* $\textsc{Protect}$ *on input K_P and object O first splits the object by computing* $(A_O, B_O) \leftarrow \textsc{Separate}(O)$*, then compresses* $C_O \leftarrow \textsc{Compress}(B_O)$ *and computes a ciphertext* $X \leftarrow \textsc{Enc}(K_E, C_O||H(O))$*. It computes a signature* $s \leftarrow \textsc{Sign}(K_{SS}, (A_O, X))$ *and joins it together with A_O and X into the protected object* $\overline{O} \leftarrow \textsc{Join}(A_O, (X, s))$*. It outputs $\overline{O}$ (or* FAIL *if any of the deployed algorithms returns* FAIL*).*

- *Algorithm* $\textsc{Verify}$ *on input K_V and a protected object $\overline{O}$ splits the protected object into* $(A_O, (X, s)) \leftarrow \textsc{Separate}(\overline{O})$ *and returns the output of the signature verification algorithm for these data,* $\textsc{SigVerify}(K_{VS}, (A_O, X), s)$ *(which equals* FAIL *in the special case that* $\textsc{Separate}$ *returned* FAIL *before).*

- *Algorithm* $\textsc{Reconstruct}$ *takes as input K_R and a protected object $\overline{O}$, and only continues reconstruction if verification of $\overline{O}$ works. If so, then it recovers* $(A_O, (X, s)) \leftarrow \textsc{Separate}(\overline{O})$ *and decrypts X to $C_O||h$ and re-computes* $B_O = \textsc{Uncompress}(C_O)$ *and* $O \leftarrow \textsc{Join}(A_O, B_O)$*. If $H(O) = h$ then it outputs O; in any other case the algorithm returns* FAIL*.*

4 Integrity of Media Authentication Schemes

In this section we address integrity protection of media authentication schemes. We first review the definition of Dittmann et al. [DKSV04] about unforgeability of $\mathcal{MAS}$[3] and then present our improved security guarantee, denoted by *strong unforgeability*. We show that strong unforgeability is strictly stronger than the notion of Dittmann et al., and finally prove that the $\mathcal{MAS}_{\text{DKSV}}$ scheme achieves the stronger notion if instantiated with the right primitives.

[3] Dittmann et al. call the property in their paper "security against existential unforgeability" but, for sake of better distinction with other security notions such as secrecy, we rename the property here to "unforgeability".

4.1 Definitions

The original unforgeability requirement of Dittmann et al. [DKSV04] demands that, without the protection key, it is infeasible to find an object O and its protected version $\overline{O}$, even after having seen other protected objects:

Definition 4.1 *Let* $\mathcal{MAS} = (\textsc{GenKey}, \textsc{Protect}, \textsc{Verify}, \textsc{Reconstruct})$ *be an invertible media authentication scheme. It is called* unforgeable *if for every probabilistic polynomial-time algorithm* $\mathcal{A}_{DKSV}$ *the value*

$$\mathrm{Prob}\left[\boldsymbol{Exp}^{mas\text{-}unf}_{\mathcal{MAS},\mathcal{A}_{DKSV}}(n) = 1\right]$$

is negligible, where

```
$\boldsymbol{Experiment}\ \boldsymbol{Exp}^{mas\text{-}unf}_{\mathcal{MAS},\mathcal{A}_{DKSV}}(n)$
    $(K_P, K_V, K_R) \leftarrow \textsc{GenKey}(1^n)$
    $(O, \overline{O}) \leftarrow \mathcal{A}_{DKSV}^{\textsc{Protect}(\cdot, K_P)}(1^n, K_V)$
        where $O_i$ denotes the $i$-th query to oracle $\textsc{Protect}(\cdot, K_P)$
        and $\overline{O}_i$ the oracle's answer to this query
    output $1$ if and only if
        $\textsc{Verify}(\overline{O}, K_V) = \mathsf{TRUE}$ and
        $\overline{O} \in [\textsc{Protect}(O, K_P)]$ and
        $O \neq O_i$ for all $i$.
```

We note that Dittmann et al. [DKSV04] claim their scheme to be secure under this definition. However, as mentioned before, Hopper et al. [HMW07] point out a gap in this proof, exploiting a weak encoding for the signing algorithm. Patching the signature and verification process as described in Construction 3.3 gives a version which is indeed secure according to this definition here (if the signature scheme achieves basic unforgeability). This can be easily inferred from the security proof for our stronger notion in the next section, and we therefore omit a formal proof for this simpler fact.

Our first definitional strengthening concerns the adversary's task to find a protected object $\overline{O}$ together with its original counter part O. Recall the satellite data example from the introduction, where the adversary's goal is only to produce another valid protected object without knowing a matching object in clear. Then the previous definition would provide no security guarantee in this case. In fact, as we will discuss later, there are even schemes satisfying the unforgeability notion above but which fail to meet the stronger requirement in the example. In our refinement below we therefore reduce the requirement on the adversary's output and merely demand that the attacker outputs a new protected object $\overline{O}$.

The other strengthening refers to availability of other components of a system. Since the algorithms may operate in a highly interactive setting, we follow the conservative approach in cryptography and allow our algorithm $\mathcal{A}_{\text{strong}}$ to also communicate with a $\textsc{Reconstruct}$ oracle, enabling him to reconstruct objects of his choice. Note that verification can be carried out locally by the adversary with the help of the public key anyway. With these two refinements we obtain the following definition:

Definition 4.2 *Let* $\mathcal{MAS} = (\textsc{GenKey}, \textsc{Protect}, \textsc{Verify}, \textsc{Reconstruct})$ *be an invertible media authentication scheme. It is called* strongly unforgeable *if for every probabilistic* *polynomial-time algorithm $\mathcal{A}_{strong}$ the value*

$$\text{Prob}\left[\boldsymbol{Exp}^{\textit{mas-stunf}}_{\mathcal{MAS},\mathcal{A}_{strong}}(n) = 1\right]$$

is negligible, where

```
Experiment $\boldsymbol{Exp}^{\textit{mas-stunf}}_{\mathcal{MAS},\mathcal{A}_{strong}}(n)$
    $(K_P, K_V, K_R) \leftarrow \textsc{GenKey}(1^n)$
    $\overline{O} \leftarrow \mathcal{A}_{strong}^{\textsc{Protect}(\cdot,K_P),\textsc{Reconstruct}(\cdot,K_R)}(1^n, K_V)$
        where $O_i$ denotes the $i$-th query to oracle $\textsc{Protect}(\cdot, K_P)$
        and $\overline{O}_i$ the oracle's answer to this query
    output 1 if and only if
        $\textsc{Verify}(\overline{O}, K_V) = \mathsf{TRUE}$ and
        $\overline{O} \neq \overline{O}_i$ for all $i$.
```

4.2 On the Relationship of the Notions

In this section we show that security according to our definition of strong unforgeability is strictly stronger than the one for the definition by Dittmann et al. This is done in two steps. First we will show that our definition implies the definition of Dittmann et al. After that, we provide two examples of schemes which are secure according to the basic notion but not to the enhanced definition. We remark that the separating examples even hold if we augment the DKSV definition by giving $\mathcal{A}_{\text{DKSV}}$ access to a $\textsc{Reconstruct}$ oracle. This difference merely stems from the fact that $\mathcal{A}_{\text{DKSV}}$ has to output a pair $(O, \overline{O})$, compared to $\overline{O}$ as in our definition.

4.2.1 Strong Unforgeability Implies Unforgeability

Proposition 4.3 *If an invertible $\mathcal{MAS}$ scheme is strongly unforgeable then it is also unforgeable.*

Proof. An attack according to the definition by Dittmann et al. can be easily transferred to an attack according to our definition of strong unforgeability. Note that from the adversary's viewpoint in our definition the attacker $\mathcal{A}_{\text{strong}}$ only has to output the protected object $\overline{O}$, whereas in the model of Dittmann et al. he also needs to output the original object O. We can therefore transfer an attacker $\mathcal{A}_{\text{DKSV}}$ according to the definition of Dittmann et al. to an attacker $\mathcal{A}_{\text{strong}}$ according to our definition by just omitting O from $\mathcal{A}_{\text{DKSV}}$'s output $(O, \overline{O})$.

Suppose an attacker $\mathcal{A}_{\text{DKSV}}$ is successful regarding the definition by Dittmann et al. Then, according to the requirements of a successful attack, the adversary's object must be distinct from all previous objects, $O \neq O_i$ for all i, and it must be valid, $\textsc{Verify}(\overline{O}, K_V) = \mathsf{TRUE}$. Assume towards contradiction that $\overline{O} = \overline{O}_i$ for some i (in particular, $\overline{O} = \overline{O}_i \neq \mathsf{FAIL}$). But since $\overline{O}_i$ was the reply of $\textsc{Protect}(O_i, K_P)$ and we have $O_i = \textsc{Reconstruct}(\overline{O}_i, K_R) = \textsc{Reconstruct}(\overline{O}, K_R) = O$ by the completeness property of media authentication schemes, this would contradict the assumption $O \neq O_i$. This means that none of the $\overline{O}_i$'s received from $\textsc{Protect}$ can be used for a successful forgery by the attacker $\mathcal{A}_{\text{DKSV}}$. Since the main difference between the two definitions lies in the condition on the adversary's output, the attacker $\mathcal{A}_{\text{DKSV}}$ is also successful according to our definition when dropping O from the output. □

4.2.2 Separating Example #1

In this example we show that for specific instantiation of the $\mathcal{MAS}_{\mathrm{DKSV}}$ there exists a successful adversary $\mathcal{A}_{\mathrm{strong}}$ against the strong unforgeability, while any attacker $\mathcal{A}_{\mathrm{DKSV}}$ against the basic notion of unforgeability fails. This shows that strong unforgeability is a strictly stronger notion.

Recall that in the (patched) $\mathcal{MAS}_{\mathrm{DKSV}}$ scheme the PROTECT algorithm computes

$$X \leftarrow \textsc{Enc}(K_E, \textsc{Compress}(B_O)||H(O)) \quad \text{and} \quad s \leftarrow \textsc{Sign}(K_{SS}, (A_O, X))$$

and outputs the protected object

$$\overline{O} = \textsc{Join}(A_O, (X, s)).$$

Now suppose that we use a trivial signature scheme where verification succeeds all the time, independently of the input, that is $\textsc{SigVerify}(K_{VS}, m, s) = \mathsf{TRUE}$ for all inputs m, s. Then we also have

$$\textsc{Verify}(K_{VS}, \overline{O}) = \mathsf{TRUE} \qquad \text{for all protected objects } \overline{O}.$$

Now an attacker $\mathcal{A}_{\mathrm{strong}}$ against strong unforgeability can output any protected object $\overline{O}$, without having queried the PROTECT oracle (but such that $\textsc{Separate}(\overline{O}) \neq \mathsf{FAIL}$). With our assumption about the signature scheme, the attacker succeeds with probability 1, because verification $\textsc{Verify}(K_{VS}, \overline{O})$ always returns TRUE.

Note that the adversary $\mathcal{A}_{\mathrm{strong}}$ wins for any choice for the symmetric encryption scheme and the function H. We can therefore use an encryption scheme which satisfies a security notion called INT-CTXT [BN00], i.e., if one sees ciphertexts for adaptively chosen messages and decryptions for chosen ciphertexts, then without the secret encryption key it is still infeasible to create a valid ciphertext for a new message. We also select H to be collision-resistant. With these choices we can next prove that there is no adversary against basic unforgeability of this instantiation of the $\mathcal{MAS}_{\mathrm{DKSV}}$ scheme.

Consider any attacker $\mathcal{A}_{\mathrm{DKSV}}$ against basic unforgeability for our instantiation, i.e., with the trivial signature verification algorithms, the INT-CTXT encryption scheme and the collision-resistant hash function. Adversary $\mathcal{A}_{\mathrm{DKSV}}$ may ask the PROTECT oracle several times (as well as the RECONSTRUCT oracle if we augment the DKSV definition accordingly), before outputting a pair $(O, \overline{O})$. Note that in contrast to strong unforgeability, here $\overline{O}$ additionally has to be from the set of protected objects of O:

$$\overline{O} \in [\textsc{Protect}(O, K_P)].$$

In particular, $\overline{O}$ must contain a valid ciphertext X of $\textsc{Compress}(B_O)||H(O)$. Then, because the final output of $\mathcal{A}_{\mathrm{DKSV}}$ satisfies $O \neq O_i$ and H is collision-resistant, we may assume that $\textsc{Compress}(B_O)||H(O) \neq \textsc{Compress}(B_{O_i})||H(O_i)$ for all i. In this case, a success of $\mathcal{A}_{\mathrm{DKSV}}$ implies that he has produced a new valid ciphertext for a new message, contradicting the INT-CTXT property. Hence, $\mathcal{A}_{\mathrm{DKSV}}$ fails unless he breaks the INT-CTXT property of the encryption scheme or the collision-resistance of H.

4.2.3 Separating Example #2

We give another separating example for the patched $\mathcal{MAS}_{\mathrm{DKSV}}$ framework where, in contrast to the previous example, we make no further assumptions about the encryption scheme and the hash function. Here, we merely assume that the signature scheme is *not* strongly unforgeable, i.e., where one can easily transform a signature s to a message m into another valid signature $s^* \neq s$. With this instantiation choice there exists a successful attack against the strong unforgeability, but which does not constitute a break against basic unforgeability.

The adversary against the strong unforgeability calls the PROTECT oracle only once about an object O to derive a protected object $\overline{O} = \textsc{Join}(A_{\overline{O}}, (X, s))$. The attacker next runs $\textsc{Separate}(\overline{O})$ to obtain $A_{\overline{O}} = A_O$ and (X, s). Since the signature scheme is not strongly unforgeable the attacker can now compute another valid signature $s^* \neq s$ for (A_O, X). He finally outputs $\overline{O}^* = \textsc{Join}(A_O, (X, s^*))$ as the forgery attempt.

The attack succeeds according to the strong unforgeability, because $s^* \neq s$ and thus $\overline{O}^*$ was never received from the PROTECT oracle before, and VERIFY evaluates to TRUE. In the DKSV definition of an attack, however, an attacker must output $(O, \overline{O})$. So in our case, prepending O to $\overline{O}^*$ would not constitute a successful attack as O has been sent to the PROTECT oracle before. In fact, it is easy to see from our proof in the next section that any attacker fails according to the DKSV definition if the underlying signature scheme achieves basic unforgeability.

4.3 Strong Unforgeability of the $\mathcal{MAS}_{\text{DKSV}}$-Scheme

We next prove that the $\mathcal{MAS}_{\text{DKSV}}$ scheme achieves strong unforgeability if the underlying signature scheme is strong enough. Note again that this statement necessitates the patch of the signature and verification algorithm; else the attack by Hopper er al. would still apply.

Theorem 4.4 (Strong Unforgeability) *If the signature scheme $\mathcal{S}$ is strongly unforgeable then the $\mathcal{MAS}_{DKSV}$ media authentication scheme in Construction 3.3 is strongly unforgeable.*

Proof. If there would be a successful attacker $\mathcal{A}_{\text{strong}}$ on the $\mathcal{MAS}_{\text{DKSV}}$ according to our strong definition, then by using the prerequisites we could use this attacker to construct a successful attacker $\mathcal{A}_{\text{sig}}$ against the strong unforgeability of the deployed signature scheme. In the following we will show the construction of such an attacker $\mathcal{A}_{\text{sig}}$.

The attacker $\mathcal{A}_{\text{sig}}$ on the signature scheme gets the signature public key K_{VS} as input. He chooses an encryption key K_E and passes the key $K_V = K_{VS}$ to $\mathcal{A}_{\text{strong}}$ to start a black-box simulation. In this simulation of $\mathcal{A}_{\text{strong}}$, adversary $\mathcal{A}_{\text{sig}}$ can easily answer queries of $\mathcal{A}_{\text{strong}}$ to oracle RECONSTRUCT with the help of the key $K_R = (K_E, K_{VS})$. For any query O_i of $\mathcal{A}_{\text{strong}}$ to the PROTECT oracle, $\mathcal{A}_{\text{sig}}$ calculates $(A_{O_i}, B_{O_i}) = \textsc{Separate}(O_i)$, $C_{O_i} = \textsc{Compress}(B_{O_i})$ and $X_i \leftarrow \textsc{Enc}(K_E, C_{O_i} || H(O_i))$. If any of the algorithms returns FAIL then $\mathcal{A}_{\text{sig}}$ immediately returns FAIL to $\mathcal{A}_{\text{strong}}$, else $\mathcal{A}_{\text{sig}}$ passes $m_i = (A_{O_i}, X_i)$ to his SIGN-oracle to get a signature s_i. Thereafter he returns $\overline{O}_i = \textsc{Join}(A_{O_i}, (X_i, s_i))$ to attacker $\mathcal{A}_{\text{strong}}$. Once $\mathcal{A}_{\text{strong}}$ outputs a protected object $\overline{O}$ and stops, adversary $\mathcal{A}_{\text{sig}}$ runs SEPARATE on $\overline{O}$ to obtain A_O and (X, s). Now $\mathcal{A}_{\text{sig}}$ outputs $m^* = (A_O, X)$ and $s^* = s$.

It is obvious that $\mathcal{A}_{\text{sig}}$ perfectly mimics the PROTECT oracle as well as the RECONSTRUCT oracle in $\mathcal{A}_{\text{strong}}$'s emulation. It remains to show that $\mathcal{A}_{\text{sig}}$ succeeds in his attack whenever $\mathcal{A}_{\text{strong}}$ wins. If $\mathcal{A}_{\text{strong}}$'s output $\overline{O}$ satisfies $\textsc{Verify}(\overline{O}, K_V) =$ TRUE then in particular $\textsc{SigVerify}(K_{VS}, m^*, s^*)$ for $\mathcal{A}_{\text{sig}}$'s output will also be TRUE and $\textsc{Separate}(\overline{O}) = (A_O, (X, s)) \neq$ FAIL. Furthermore $\overline{O} \neq \overline{O}_i$ for all i.

We have to show that the pair $(m^*, s^*) = ((A_O, X), s)$ has not appeared in $\mathcal{A}_{\text{sig}}$'s interactions with the signature oracle. This is clearly true if, in the i-th request, $\mathcal{A}_{\text{sig}}$ returned $s_i =$ FAIL before even querying the signature oracle, namely, if separation, compression or encryption failed. If, on the other hand, $\overline{O}_i =$ FAIL for the i-th interaction, because the final JOIN in the simulation of the protection query returned FAIL, but a message $m_i = (A_{O_i}, X_i)$ was still signed with s_i, then we must have $(m^*, s^*) \neq (m_i, s_i)$. Else, for equality $(m^*, s^*) = (m_i, s_i)$ we would have $\mathsf{FAIL} = \textsc{Join}(A_{O_i}, (X_i, s_i)) = \textsc{Join}(A_O, (X, s)) = \textsc{Join}(\textsc{Separate}(\overline{O}))$ for $\textsc{Separate}(\overline{O}) \neq \mathsf{FAIL}$, contradicting the completeness of the watermarking scheme. Finally, if $\overline{O}_i \neq \mathsf{FAIL}$, then because $\overline{O} \neq \overline{O}_i$ and the $\textsc{Separate}$-function is collision-resistant (see Section 3.1.1) we have $(A_O, (X, s)) \neq (A_{O_i}, (X_i, s_i))$.

Hence, if attacker $\mathcal{A}_{\text{strong}}$ on the media authentication scheme is successful, attacker $\mathcal{A}_{\text{Sig}}$ will also succeed with the same probability, because (m^*, s^*) was never received from the $\textsc{Sign}$-oracle and $\textsc{SigVerify}(K_{VS}, m^*, s^*) = \mathsf{TRUE}$. □

5 Secrecy of Media Authentication Schemes

Recall that the scheme by Dittmann et al. [DKSV04] introduces an encryption scheme in order to protect the B_O-part of an object O. However, in their paper they do not provide any claim about the secrecy under reasonable conditions about the encryption scheme, not to mention a rigorous security model. In a companion paper, though, Katzenbeisser and Dittmann [KD04] discuss a desirable secrecy requirement, resembling semantic security of encryption schemes (as defined in Section 3.1.2). Yet, their proposal advocates a somewhat elliptical mixture between semantic security and indistinguishability of encryption schemes (cf. [Gol04]), and remains rather sketchy. It also remains unclear if, or under which conditions, the $\mathcal{MAS}_{\text{DKSV}}$ scheme meets this goal.

Recall that the idea behind semantic security of an encryption scheme was that anything an efficient adversary could learn about a message m from a ciphertext X could also be computed efficiently without X. Here we discuss that, by using appropriate notions of secrecy with side information, we can indeed define secrecy for media authentication schemes in the sense of semantic security. Our definition basically says that an $\mathcal{MAS}$ provides secrecy if whatever one can compute from a protected object $\overline{O}$ (including the public part A_O) could also be derived from A_O alone. We then continue to show that semantic security of the encryption function (with respect to side information) also guarantees secrecy of the $\mathcal{MAS}_{\text{DKSV}}$ scheme.

5.1 Definition

The definition below follows the one for semantic security of encryption (with respect to side information) closely. Namely, we again compare the success probability of an adversary predicting some information $f_{\text{MAS}}(O)$ of an object O from the protected version $\overline{O}$ (and hist_O) with the prediction success of a simulator given only hist_O. For a secure $\mathcal{MAS}$ these probabilities should be close.

We write $\mathcal{O}$ for the distribution of the objects and hist_{MAS} for the algorithm computing the side information. For notational convenience we again denote by $(O, \text{hist}_O) \leftarrow (\mathcal{O}, \text{hist}_{\text{MAS}})(1^n)$ the joint sampling process, possibly sharing state between the two algorithms.

Definition 5.1 *An invertible media authentication scheme $\mathcal{MAS}$ is called* semantically secure with respect to side information hist_{MAS} *if for every probabilistic polynomial-time algorithm $\mathcal{A}_{MAS}$, there is a probabilistic polynomial-time algorithm $\mathcal{S}_{MAS}$, the simulator, such that for every polynomial-time distribution $\mathcal{O}$ of objects and for every function f_{MAS}, the* *difference*

$$\mathrm{Prob}\left[\boldsymbol{Exp}^{\textit{mas-sem},\mathcal{O},f_{MAS},hist_{MAS}}_{\mathcal{MAS},\mathcal{A}_{MAS}}(n)=1\right]-\mathrm{Prob}\left[\boldsymbol{Exp}^{\textit{mas-sem},\mathcal{O},f_{MAS},hist_{MAS}}_{\mathcal{MAS},\mathcal{S}_{MAS}}(n)=1\right]$$

is negligible, where

```
$\textbf{\textit{Experiment}}\ \boldsymbol{Exp}^{\textit{mas-sem},\mathcal{O},f_{MAS},hist_{MAS}}_{\mathcal{MAS},\mathcal{A}_{MAS}}(n)$
    $(K_P, K_V, K_R) \leftarrow \textsc{GenKey}(1^n)$
    $(O, hist_O) \leftarrow (\mathcal{O}, hist_{MAS})(1^n)$
    $\overline{O} \leftarrow \textsc{Protect}(K_P, O)$
    $a \leftarrow \mathcal{A}_{MAS}(K_V, \overline{O}, hist_O)$
    output 1 if and only if
        $a = f_{MAS}(O, hist_O)$
```

```
$\textbf{\textit{Experiment}}\ \boldsymbol{Exp}^{\textit{mas-sem},\mathcal{O},f_{MAS},hist_{MAS}}_{\mathcal{MAS},\mathcal{S}_{MAS}}(n)$
    $(K_P, K_V, K_R) \leftarrow \textsc{GenKey}(1^n)$
    $(O, hist_O) \leftarrow (\mathcal{O}, hist_{MAS})(1^n)$

    $a \leftarrow \mathcal{S}_{MAS}(K_V, hist_O)$
    output 1 if and only if
        $a = f_{MAS}(O, hist_O)$
```

We remark that we can even strengthen the notion above by granting $\mathcal{A}_{\mathrm{MAS}}$ access to oracles $\textsc{Protect}(\cdot, K_P)$ and $\textsc{Reconstruct}(\cdot, K_R)$ (with the restriction that the adversary never queries the reconstruct oracle about the challenge $\overline{O}$, enabling a trivial attack otherwise). Assuming chosen-plaintext security of the underlying encryption scheme (where the adversary is also allowed to see ciphertexts of arbitrary messages via an oracle $\textsc{Enc}(K_E, \cdot)$), our result also holds under this more advanced attack model, as we will discuss after the proof for the basic case. Interestingly, the proof for this extension also takes advantage of our notion of strong unforgeability.

5.2 Secrecy of the $\mathcal{MAS}_{\mathbf{DKSV}}$-Scheme

The following theorem shows that semantic security of the encryption scheme carries over to the secrecy of the $\mathcal{MAS}_{\mathrm{DKSV}}$ scheme:

Theorem 5.2 *Let $hist_{MAS}(1^n)$ be the function which takes an object O and outputs A_O where $(A_O, B_O) \leftarrow \textsc{Separate}(O)$. Let $\mathcal{E}$ be a semantically secure encryption scheme (with respect to side information $hist_{\mathrm{enc}} = hist_{MAS}$). Then the invertible media authentication scheme $\mathcal{MAS}_{DKSV}$ in Construction 3.3 is semantically secure with respect to side information $hist_{MAS}$.*

Proof. The proof is by contradiction. Assume that $\mathcal{MAS}_{\mathrm{DKSV}}$ is not semantically secure (with respect to the given side information $\mathrm{hist}_{\mathrm{MAS}}$). We will show that this allows to construct a successful attacker $\mathcal{A}_{\mathrm{enc}}$ on the semantic security encryption function (for side information $\mathrm{hist}_{\mathrm{enc}}$), which will contradict the initial assumption about the security of the encryption scheme.

Recall that a successful attacker $\mathcal{A}_{\mathrm{MAS}}$ is able to compute information $f_{\mathrm{MAS}}(O, \mathrm{hist}_O)$ for an object O distributed according to $\mathcal{O}$ from $\overline{O}$, such that no simulator can approximate this success probability close enough (here f_{MAS} and $\mathcal{O}$ may depend on the simulator). We now construct our adversary $\mathcal{A}_{\mathrm{enc}}$ against the encryption scheme, having a non-negligible advantage over any simulator on predicting the results of some function f_{enc} and some message distribution $\mathcal{M}$.

The specific information and distribution for adversary $\mathcal{A}_{\mathrm{enc}}$ are defined in terms of $\mathcal{O}$ and f_{MAS} used by $\mathcal{A}_{\mathrm{MAS}}$:

```
distribution $(\mathcal{M}_{\mathcal{O}}, \text{hist}_{\text{enc}})(1^n)$:
    sample $O \leftarrow \mathcal{O}(1^n)$
    compute $(A_O, B_O) \leftarrow \textsc{Separate}(O)$
    let $m = \textsc{Compress}(B_O)||H(O)$
    let $\text{hist}_m = A_O$
```

```
function $f_{\text{enc}}(m, \text{hist}_m)$:
    let $m = C_O||H(O)$
    compute $B_O = \textsc{Uncompress}(C_O)$
    let $O \leftarrow \textsc{Join}(\text{hist}_m, B_O)$
    return $f_{\text{MAS}}(O, \text{hist}_m)$
```

Next we define our attacker $\mathcal{A}_{\text{enc}}$ against the encryption scheme. This attacker gets a ciphertext $X \leftarrow \textsc{Enc}(K_E, m)$ together with side information $\text{hist}_m = A_O$ as input (where the values m and hist_m are sampled according to $\mathcal{M}_{\mathcal{O}}$ and hist_{enc}). Adversary $\mathcal{A}_{\text{enc}}$ next "wraps" the ciphertext X into a protected object and then uses $\mathcal{A}_{\text{MAS}}$ to derive some information about the original message m. More precisely, $\mathcal{A}_{\text{enc}}$ first generates signature keys $(K_{SS}, K_{VS}) \leftarrow \textsc{GenSign}(1^n)$ and then uses these keys to compute a protected object $\overline{O}$:

$$\text{compute } \; s \leftarrow \textsc{Sign}(K_{SS}, (A_O, X)) \; \text{ and set } \; \overline{O} \leftarrow \textsc{Join}(A_O, (X, s)).$$

Then $\mathcal{A}_{\text{enc}}$ invokes algorithm $\mathcal{A}_{\text{MAS}}$ on inputs $K_V = K_{VS}$ and $\overline{O}$ as created above, and on $\text{hist}_O = A_O$. It waits to receive some answer a and stops with output a, too.

Note that the input to $\mathcal{A}_{\text{MAS}}$ in the simulation above is identically distributed to genuine data in experiment $\mathbf{Exp}^{\text{mas-sem},\mathcal{O},f_{\text{MAS}},\text{hist}_{\text{MAS}}}_{\mathcal{MAS},\mathcal{A}_{\text{MAS}}}(n)$. Therefore, if the output a of the adversary $\mathcal{A}_{\text{MAS}}(K_V, \overline{O}, \text{hist}_O)$ satisfies $f_{\text{MAS}}(O, A_O)$, then it also matches the value $f_{\text{enc}}(m, \text{hist}_m)$ by construction. Hence, $\mathcal{A}_{\text{enc}}$ succeeds with the same probability in the attack on the encryption scheme, as $\mathcal{A}_{\text{MAS}}$ does in the attack on the media authentication scheme.

It remains to show that attacker $\mathcal{A}_{\text{enc}}$ has a non-negligible advantage over any simulator in predicting the function value f_{enc}. That is, we still need to prove that there is no simulator $\mathcal{S}_{\text{enc}}$ being as successful as the attacker $\mathcal{A}_{\text{enc}}$. This will follow because, if there was an almost equally successful simulator $\mathcal{S}_{\text{enc}}$, then we can construct a simulator $\mathcal{S}_{\text{MAS}}$ approximating $\mathcal{A}_{\text{MAS}}$ close enough, contradicting the assumption about the insecurity of the media authentication scheme.

Suppose that there was an appropriate simulator $\mathcal{S}_{\text{enc}}$ for the encryption case (for all message distributions and all functions f_{enc} and thus for all distributions and functions as constructed above). Then we build a simulator $\mathcal{S}_{\text{MAS}}$ for the $\mathcal{MAS}$ as follows. $\mathcal{S}_{\text{MAS}}$ gets $K_V = K_{VS}$ and A_O as inputs. To run a black-box simulation of $\mathcal{S}_{\text{enc}}$ the $\mathcal{MAS}$ simulator defines $\text{hist}_m = A_O$ and invokes $\mathcal{S}_{\text{enc}}$ on hist_m to derive some information a. Simulator $\mathcal{S}_{\text{MAS}}$ copies a and stops.

We observe that the data in the black-box simulation has the same distribution as in the experiment of $\mathcal{S}_{\text{enc}}$. Thus, by construction $\mathcal{S}_{\text{MAS}}$ successfully predicts $f_{\text{MAS}}(O, A_O)$ whenever $\mathcal{S}_{\text{enc}}$'s output satisfies the value $f_{\text{enc}}(m, \text{hist}_m)$ of the previously defined function f_{enc}. But then the success probability of $\mathcal{S}_{\text{MAS}}$ is identical to the one of $\mathcal{S}_{\text{enc}}$, which in turn approximates $\mathcal{A}_{\text{enc}}$'s probability of winning the encryption experiment closely by assumption. Since $\mathcal{A}_{\text{enc}}$'s success probability is identical to the one of $\mathcal{A}_{\text{MAS}}$ attacking the secrecy of the media authentication scheme, we conclude that there would be a good simulator $\mathcal{S}_{\text{MAS}}$ for $\mathcal{A}_{\text{MAS}}$ (for all $\mathcal{O}$ and f_{MAS}).

Overall we have shown that we can construct an successful attacker on the encryption if $\mathcal{MAS}_{\text{DKSV}}$ is not semantically secure, contradicting the assumption that the encryption function used within the scheme is semantically secure. This completes the proof. □

Finally, we discuss that our proof extends to the case that we grant the adversary $\mathcal{A}_{\text{MAS}}$ access to a PROTECT-oracle and to a RECONSTRUCT-oracle. Recall that our proof idea is to turn an adversary $\mathcal{A}_{\text{MAS}}$ against the MAS into one against the encryption scheme. To this end, our attacker $\mathcal{A}_{\text{enc}}$ against the encryption scheme has simulated a PROTECT-oracle with the help of the given ciphertext. Here, $\mathcal{A}_{\text{enc}}$ now has to answer additional PROTECT queries of $\mathcal{A}_{\text{MAS}}$, and we assume that $\mathcal{A}_{\text{enc}}$ is also given access to an encryption oracle $\text{ENC}(K_E, \cdot)$ initialized with the same key with which the input ciphertext X has been produced (chosen-plaintext security).

For each query O_i of $\mathcal{A}_{\text{MAS}}$ to its putative oracle PROTECT, algorithm $\mathcal{A}_{\text{enc}}$ extracts $(A_{O_i}, B_{O_i}) \leftarrow \text{SEPARATE}(O_i)$, compresses $C_{O_i} \leftarrow \text{COMPRESS}(B_{O_i})$ and computes the hash value $H(O_i)$. It sends $C_{O_i}||H(O_i)$ to its ENC-oracle to receive a ciphertext X_i, and signs this ciphertext together with A_{O_i} using the chosen signature key, $s_i \leftarrow \text{SIGN}(K_{SS}, (A_{O_i}, X_i))$. Adversary $\mathcal{A}_{\text{enc}}$ returns $\overline{O}_i \leftarrow \text{JOIN}(A_{O_i}, (X_i, s_i))$ to $\mathcal{A}_{\text{MAS}}$ as the answer of the PROTECT-oracle.

For each query $\overline{O}_i$ of $\mathcal{A}_{\text{MAS}}$ to its putative RECONSTRUCT-oracle algorithm $\mathcal{A}_{\text{enc}}$ proceeds as follows. If $\overline{O}_i$ has been the answer to one of a previously simulated PROTECT-oracle calls then $\mathcal{A}_{\text{enc}}$ looks up the corresponding query O_i and returns this object. In any other case $\mathcal{A}_{\text{enc}}$ returns FAIL.

We note that the PROTECT-oracle is simulated perfectly by $\mathcal{A}_{\text{enc}}$. The RECONSTRUCT-oracle gives consistent answers, except for the case that $\mathcal{A}_{\text{MAS}}$ submits a valid protected object but which has not been produced by a simulated PROTECT-query. But this would straightforwardly contradict the unforgeability of the MAS scheme (assuming that the signature algorithm is strongly unforgeable and the SEPARATE algorithm is collision-resistance, see Theorem 4.4).[4] Hence, this can only happen with negligible probability. The rest of the proof now carries over easily.

Acknowledgments

We thank the anonymous reviewers of Information Hiding 2007 for valuable comments.

Marc Fischlin and Anja Lehmann are supported by the Emmy Noether Program Fi 940/2-1 of the German Research Foundation (DFG).

References

[BLS04] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. *Journal of Cryptology*, 17(4):297–319, 2004.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In *Advances in Cryptology — Asiacrypt 2000*, volume 1976 of *LNCS*, pages 531–545. Springer-Verlag, 2000.

[BR96] Mihir Bellare and Phil Rogaway. The exact security of digital signatures — How to sign with RSA and Rabin. In *Advances in Cryptology — Eurocrypt'96*, volume 1070 of *LNCS*, pages 399–416. Springer-Verlag, 1996.

[BSW06] Dan Boneh, Emily Shen, and Brent Waters. Strongly Unforgeable Signatures Based on Computational Diffie-Hellman. In *Public-Key Cryptography (PKC) 2006*, volume 3958 of *LNCS*, pages 229–240. Springer-Verlag, 2006.

[4] Here we clearly need the strong unforgeability notion where the adversary merely needs to output a protected object. The basic unforgeability notion of Dittmann et al. would not suffice in this case.

[CS00] Ronald Cramer and Victor Shoup. Signature Schemes Based on the Strong RSA Assumption. *ACM Transactions on Information and System Security*, 3(3):161–185, 2000.

[DKSV04] Jana Dittmann, Stefan Katzenbeisser, Christian Schallhart, and Helmut Veith. Provably Secure Authentication of Digital Media Through Invertible Watermarks. Cryptology ePrint Archive, Report 2004/293, 2004. `http://eprint.iacr.org/`.

[FGD02] J. Fridrich, M. Goljan, and R. Du. Lossless data embedding — new paradigm in digital watermarking. *EURASIP Journal of Applied Signal Processing*, 2:185–196, 2002.

[Fis03] Marc Fischlin. The Cramer-Shoup Strong-RSA Signature Scheme Revisited. In *Public-Key Cryptography (PKC) 2003*, volume 2567 of *LNCS*, pages 116–129. Springer-Verlag, 2003.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption. *Journal of Computer and System Science*, 28(2):270–299, 1984.

[Gol04] Oded Goldreich. *The Foundations of Cryptography*, volume 2. Cambridge University Press, 2004.

[HJRS99] C.W. Honsinger, P. Jones, M. Rabbani, and J.C. Stoffel. Lossless recovery of an original image containing embedded data. US patent application, Docket No: 77102/E/D, 1999.

[HMW07] Nicholas Hopper, David Molnar, and David Wagner. From Weak to Strong Watermarking. In *Theory of Cryptography Conference (TCC) 2007*, volume 4392 of *LNCS*. Springer-Verlag, 2007.

[KD04] Stefan Katzenbeisser and Jana Dittmann. Malicious attacks on media authentication schemes based on invertible watermarks. In *Security, Steganography, and Watermarking of Multimedia Contents*, volume 5306 of *Proceedings of SPIE*, pages 838–847. SPIE, 2004.

[NY89] Moni Naor and Moti Yung. Universal One-Way Hash Functions and Their Cryptographic Applications. In *Proceedings of the Annual Symposium on the Theory of Computing (STOC) 1989*, pages 33–43. ACM Press, 1989.

[Rom90] John Rompel. One-Way Functions are Necessary and Sufficient for Secure Signatures. In *Proceedings of the Annual Symposium on the Theory of Computing (STOC) 1999*, pages 387–394. ACM Press, 1990.

[SBT96] M.D. Swanson, Zhu Bin, and A.H. Tewfik. Transparent robust image watermarking. In *International Conference on Image Processing*, volume 3, pages 211–214. IEEE Computer Society Press, 1996.

[YM98] M. Yeung and F. Mintzer. Invisible watermarking for image verification. *Journal of Electronic Imaging*, 7:578–591, 1998.

A Online Media Authentication

Dittmann et al. [DKSV04] introduce the notion of *online* media authentication schemes where objects O are dividied into chunks $O_1||O_2||\ldots||O_m$ of equal length, and PROTECT produces (in a stateful process) "protected chunks" $\overline{O}_i$ from each O_i for $i = 1, 2, \ldots, m$. For such protected chunks it should be possible to verify authenticity of sub sequences $\overline{O}_i||\ldots||\overline{O}_j$ of the protected chunks (say, if one publishes only parts of recorded satellite data).

Ditmmann et al. extend their definition of unforgeability by allowing the adversary to output a sequence $(O_i, \overline{O}_i), \ldots, (O_j, \overline{O}_j)$ such that $O_i||\ldots||O_j$ has never appeares as a subsequence in one of the PROTECT queries. They also give a slight modification of their basic scheme achieving online authentication, using hash chains.

Here we briefly outline that both our notion of strong unforgeability as well as the strongly unforgeable scheme can be easily extended to capture such online authentication as well (even without further cryptographic assumption). First, our definition of strong unforgeability now asks the adversary to output a sequence $\overline{O}_i||\ldots||\overline{O}_j$ such that this sequence has never been part of a reply of PROTECT. Then, we modify the DKSV scheme with the strong unforgeable signature scheme as described next.

For each new object to be authenticated we pick a unique string ID of some fixed length (either by using a counter or by picking a sufficiently long random string). Then we apply our basic protection algorithm to each chunk O_i, but where we now compute the signature as $s_i \leftarrow \text{SIGN}(K_{SS}, (A_{O_i}, X_i, \mathsf{ID}, \langle i \rangle))$ and build $\overline{O}_i \leftarrow \text{JOIN}(A_{O_i}, (X, s, \mathsf{ID}, i))$, i.e., we include the identifier ID and the chunk number i (encoded by some fixed-length encoding $\langle \cdot \rangle$) into the protected chunk. To verify correctness of a sequence of protected chunks we verify the correctness of each chunk and also check that each chunk carries the identical identifier ID and that chunk numbers i appear in consecutive order. Only if all these tests succeed then we accept the sequence as valid.

It is not hard to see that our security proof carries over to this extension (given that the signature scheme is strongly unforgeable and that the SEPARATE algorithm is collision-resistant), thus providing a solution without further cryptographic assumption.