Provably Secure Authentication of Digital Media Through Invertible Watermarks*

Jana Dittmann[1], Stefan Katzenbeisser[2], Christian Schallhart[2], Helmut Veith[2]

[1] Otto-von-Guericke Universität Magdeburg, Germany,
jana.dittmann@iti.cs.uni-magdeburg.de

[2] Technische Universität München, Germany
katzenbe,schallha,veith@in.tum.de

November 7, 2004

Abstract

The recent advances in multimedia technology have made the manipulation of digital images, videos or audio files easy. On the one hand the broad availability of these new capabilities enabled numerous new applications. On the other hand, for the same reasons, digital media can easily be forged by almost anyone. To counteract this risk, fragile watermarks were proposed to protect the integrity and authenticity of digital multimedia objects. Traditional watermarking schemes employ non-cryptographic and signal processing oriented techniques, which fail to provide any provable security guarantee against malicious modification attempts. In this paper, we give for the first time a provably secure authentication mechanism for digital multimedia files that is based on both cryptographic signatures and invertible watermarks. While traditional watermarking schemes introduce some small irreversible distortion in the digital content, invertible watermarks can be completely removed from a watermarked work.

1 Introduction

The recent advances in multimedia technology brought powerful tools for manipulating digital images, videos or audio files to everybody's desktop. While this enables numerous new applications, the authenticity and integrity of digital artefacts cannot be readily asserted—the origin and integrity of almost every digital object must be doubted. For example, a picture displaying a car accident cannot be readily trusted as evidence, since it is possible to modify the location of the cars on the picture only using a common personal computer

*The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the author's views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

running digital image processing tools. Similar problems apply to digital sound clips or video files, where sets of samples can be removed or replaced.

This problem was first noted by Friedman [5], who proposed to sign digital images using a cryptographic signature in order to assert their authenticity and integrity. The apparent drawback of this proposal was that the signature and the image had to be stored separately. The direct encoding of signatures into digital images was made possible by the availability of sophisticated fragile watermarking schemes. A fragile watermark is a digital watermark [9] that is *not* robust against common signal processing tools—if a watermarked object is modified, the watermark cannot be detected any more. Fragile watermarks were proposed as tools to assure the integrity of image files [11, 13]. In these approaches, non-cryptographic signatures are encoded as fragile watermarks in digital images. An image is deemed authentic if and only if it is possible to recover and verify its embedded signature. If a file with such a watermark is modified, then either the watermark cannot be detected any more or the recovered non-cryptographic signature does not match the image. In both cases, the image is considered to be tampered. Unfortunately, this approach has the apparent drawback that it is not possible to formally prove its security in a cryptographically precise way.

In high security applications, like medical imaging, there is another concern, namely that the watermark embedding process induces some distortion that interferes with the contents of the digital media files. For example, X-ray images are extremely sensitive to blurring operations, which limits the use of watermarking schemes in medical applications. To address these concerns, invertible (or reversible) watermarking schemes were proposed [8, 3, 4, 12, 10, 1]. Invertible watermarking schemes allow to insert a watermark into an object as usual, but facilitate the lossless removal of the watermark from an untampered watermarked object. More precisely, if a watermark is successfully detected, the information contained in the recovered watermark, together with the watermark key, suffices to remove the watermark completely from the object. Most invertible watermarks are also fragile and therefore suitable to implement authentication schemes.

In this paper we provide the first construction for a provably secure authentication scheme for digital media files that relies on watermarking technology. Technically, we use invertible fragile watermarks to embed a digital signature of the media. After reviewing the necessary watermarking technology in Section 2, we introduce media authentication schemes in Section 3. Finally, we give two provably secure constructions for media authentication schemes in Sections 4 and 5; the second construction can be used for large media files or in streaming applications.

2 Invertible Watermarks

While virtually all previous watermarking schemes introduced some small amount of irreversible distortion in the data during the embedding process, *invertible watermarking* schemes were first introduced by Honsinger et. al. [8]. They were able to construct a watermarking scheme where a watermark can be completely removed from an untampered watermarked object, thereby recovering the original object. However, their construction is

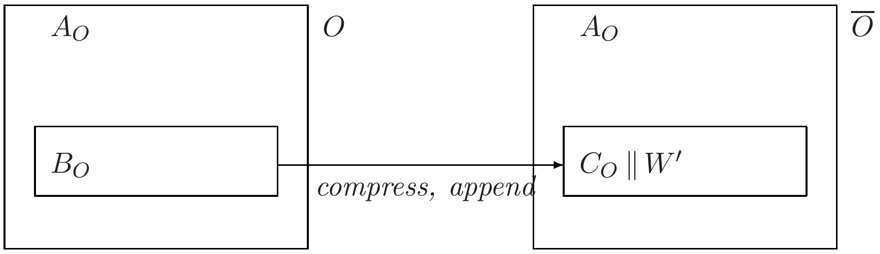

Figure 1: Invertible watermarking. An object O is divided into two parts A_O and B_O. The watermark consists of the compressed part B_O, denoted by C_O, and the watermark payload W'.

not practical, as it introduces (small, but visible) distortions in the watermarked objects.

Fridrich et. al. [3] introduced a general framework that allows to construct an invertible (fragile) watermarking scheme out of a fragile one. The general idea is to divide the object O, dependent on a public key K_W, into two parts A_O and B_O. The latter part contains perceptually insignificant portions of the object that can be overwritten by a watermark without lowering the object quality, whereas A_O contains perceptually visible parts that must be preserved. To provide invertibility, the original part B_O is compressed and stored in the watermark; denote the compressed part B_O with C_O. The watermark W consists of the watermark payload W' and C_O, thus $W = C_O \,\|\, W'$. W replaces the part B_O in the watermarked object $\overline{O}$. This general framework is depicted in Figure 1.

The distortion of the watermark can easily be removed by separating the marked object $\overline{O}$ into the two parts $A_{\overline{O}}$ and $B_{\overline{O}}$. During the watermark insertion process, only $B_{\overline{O}}$ was modified, so $A_{\overline{O}} = A_O$. Now, $B_{\overline{O}}$ has the form $W = C_O \,\|\, W'$; decompressing C_O yields to the part B_O of the original object O. By overwriting $B_{\overline{O}}$ with B_O in the object $\overline{O}$, O can be completely recovered. This procedure works only if $\overline{O}$ was not altered; it is therefore a fragile watermarking scheme.

In the rest of the paper, we denote an invertible watermarking scheme as a tuple of two probabilistic polynomial algorithms $\langle \textsc{Separate}, \textsc{Join} \rangle$. On input O and K_W, $\textsc{Separate}$ produces the tuple $\langle A_O, B_O \rangle$. $\textsc{Join}$ inverts the algorithm $\textsc{Separate}$, i.e., on input $\langle A_{\overline{O}}, B_{\overline{O}} \rangle$ and K_W it outputs $\overline{O}$. Except with negligible probability, we require that

$$\textsc{Join}(K_W, \textsc{Separate}(K_W, O)) = O,$$

for all objects O and keys K_W with $\textsc{Separate}(K_W, O) \neq \textsc{fail}$.

From the previous description it is obvious that it is not possible to embed an invertible watermark in every object. In case the part B_O cannot be sufficiently compressed, there is not enough room to store both the watermark payload and the compressed part C_O. However, typical multimedia files (such as images or audio files) contain enough redundant, compressible information so that the watermarking operation works for virtually all relevant objects. In this paper, we do not detail the insertion and recovery operations of invertible watermarking schemes and rather use them as black-box primitives. For specific implementation details of invertible watermarking schemes, we refer to [3, 4, 2].

3 Media Authentication Schemes

Similar to cryptographic signatures, media authentication schemes based on invertible watermarks can be described in terms of four probabilistic polynomial time algorithms $\langle\textsc{GenKey}, \textsc{Protect}, \textsc{Verify}, \textsc{Reconstruct}\rangle$. The algorithm GenKey denotes the key-generation process; by using a private key, Protect authenticates an object O and outputs its signed version $\overline{O}$. Signed objects can be verified by the algorithm Verify and a public key; Verify either outputs TRUE or FALSE. In the first case, the object is deemed authentic; in the latter case, the object is considered modified. Finally, Reconstruct reverses the protection mechanism and losslessly reconstructs O out of $\overline{O}$.

3.1 Definition

More formally, an *invertible media authentication scheme* is defined as follows:

- Algorithm GenKey generates the necessary keys for the application. On input 1^n, GenKey produces a triple of strings $\langle K_P, K_V, K_R\rangle$ with $|K_P \,\|\, K_V \,\|\, K_R| = n$; the operation $\|$ denotes string concatenation. The key K_P will be used in the protection step, whereas K_V and K_R are used for verification and recovery. The verification key K_V is a public key, whereas K_P and K_R are private keys.
- Algorithm Protect takes K_P, K_R and an object O. The output of the algorithm consists of an authenticated object $\overline{O}$.
- Algorithm Verify takes the verification key K_V and an object $\overline{O}$ and outputs a boolean variable.
- Algorithm Reconstruct takes the keys K_R and K_V and an object $\overline{O}$ and restores the original object O.

Note that we have defined all algorithms as probabilistic, which implies that they can fail on certain instances (for example it may not be possible to embed a watermark in an invertible manner); in this case, the algorithms output a special symbol FAIL. We require that the media authentication scheme "works" for almost all objects that can be authenticated. In particular,

$$\textsc{Verify}(\textsc{Protect}(O, K_P, K_R), K_V) = \textsc{true}$$

and

$$\textsc{Reconstruct}(\textsc{Protect}(O, K_P, K_R), K_R, K_V) = O$$

must hold except for a negligible fraction of all objects O with $\textsc{Protect}(O, K_P, K_R) \neq$ FAIL.

As usual, we will denote a cryptographic *signature scheme* as triple of probabilistic polynomial time algorithms $\mathbb{S} = \langle\textsc{GenSign}, \textsc{Sign}, \textsc{SigVerify}\rangle$, where GenSign denotes the key generation, SIGN the signing and SIGVERIFY the signature verification algorithm. A signature scheme is said to be secure, if it is secure against existential forgery of signatures under a chosen-message attack [7]; that is, if the attacker is unable (even with access to a signing oracle) to forge a valid pair of a message and a corresponding signature.

3.2 Attacker Model

Sticking to Kerckhoffs' principle, we assume that an attacker possesses complete knowledge of the system; furthermore, the attacker has access to the public verification key K_V. Similar to attacks against cryptographic signature schemes, we can distinguish several types of attacks against media authentication schemes according to the possibilities for an attacker to interfere with the system. It seems natural to assume that an attacker will know several protected media files under one verification key K_V, as such objects might be freely available on the Internet. A more powerful attacker may even launch a *chosen message attack*. In this setup, an attacker is able to obtain protected objects of his own choice. That is, he can obtain a signed object $\overline{O}$ corresponding to an object O chosen during the attack. In imaging applications, such an attack is particularly realistic, as long as the attacker has physical access to the imaging device and can take pictures of his own choice.

For this reason, we adopt the notion of existential forgery under chosen message attacks for the present scenario. In particular, an attacker can query an oracle for authenticated objects of his own choice and perform any polynomially bounded computation. We say that an attack is successful, if the attacker manages to output an object $\overline{O}$ together with an alleged original O such that $\textsc{Verify}(\overline{O}, K_V) = \textsc{true}$ and the original object O was *not* presented to the oracle previously.

Definition 1 *Let* $\langle \textsc{GenKey}, \textsc{Protect}, \textsc{Verify}, \textsc{Reconstruct} \rangle$ *a media authentication scheme and* $\textsc{Query}_{K_P}$ *be an oracle that computes* $\overline{O} \leftarrow \textsc{Protect}(O, K_P)$ *on input* O. *Furthermore, let* $\langle K_P, K_V, K_R \rangle \in [\textsc{GenKey}(1^{n_K})]$.

An attack is a probabilistic algorithm $\textsc{Attack}$ *with oracle access to* $\textsc{Query}_{K_P}$ *and success probability* $\varepsilon_{\textsc{Attack}}$ *such that*

$$\textsc{Attack}(1^n, K_V) = \begin{cases} \langle O, \overline{O} \rangle & \textit{such that } \textsc{Verify}(\overline{O}, K_V) = \textsc{true},\ |O| = n, \\ & \overline{O} \in [\textsc{Protect}(O, K_P)] \textit{ and } O \neq O_i \textit{ for all } 1 \leq i \leq l, \\ & \textit{with probability } \varepsilon_{\textsc{Attack}} \\ \textsc{fail} & \textit{with probability } 1 - \varepsilon_{\textsc{Attack}}, \end{cases}$$

where O_i *denotes the input to the* i*-th oracle query* $\textsc{Query}_{K_P}$. *The probability is taken over all coin tosses of* $\textsc{Attack}$ *and all keys* $\langle K_P, K_V, K_R \rangle$.

We say that a media authentication scheme is secure, if the success probability of *every* probabilistic polynomial time attack is negligible:

Definition 2 *A media authentication scheme is secure against existential forgery of authenticated objects, if every probabilistic polynomial time attack* $\textsc{Attack}$ *has negligible success probability.*

4 Offline Media Authentication

In this section, we describe an offline media authentication scheme. We call a scheme offline, if the protection algorithm needs access to the whole media file at once.

Let $\mathbb{S} = \langle \textsc{GenSign}, \textsc{Sign}, \textsc{SigVerify} \rangle$ be a cryptographic signature scheme producing signatures of length k, Encrypt and Decrypt be the encryption and decryption operation of a symmetric cipher and Compress be the compression algorithm of a lossless compression scheme. Furthermore, we fix an invertible watermarking scheme $\langle \textsc{Separate}, \textsc{Join} \rangle$ that can embed watermark strings of length k.

Loosely speaking, the media authentication scheme stores a cryptographic signature of the unmodified portion of the object (the part A_O) and the encrypted, compressed part B_O as an invertible watermark. The construction is as follows:

- GenKey runs GenSign to obtain a key pair $\langle K_{SS}, K_{VS} \rangle$; furthermore, it computes a key K_E for the symmetric cipher and a random string K_W. Let $K_P = K_{SS} \| K_W$, $K_V = K_{VS} \| K_W$ and $K_R = K_E \| K_W$.

- Protect, on input O, $K_P = K_{SS} \| K_W$ and $K_R = K_E \| K_W$, separates O, using algorithm Separate and key K_W, into two parts A_O and B_O. The latter part is compressed to obtain C_O. Denote with W' the string $W' = X \| s$, where

  $$X \leftarrow \textsc{Encrypt}(K_E,\ C_O \| H(O))$$

  and

  $$s \leftarrow \textsc{Sign}(K_{SS},\ A_O \| X).$$

  Protect runs Join on K_W and $\langle A_O, W' \rangle$ to obtain the authenticated object $\overline{O}$ or fail. If Join fails, Protect outputs fail, otherwise $\overline{O}$.

- Verify, on input $\overline{O}$ and $K_V = K_{VS} \| K_W$, runs Separate on K_W and $\overline{O}$ to obtain the two parts $A_{\overline{O}}$ and $B_{\overline{O}}$ of $\overline{O}$. The latter part has the form $B_{\overline{O}} = X \| s$, where X is an arbitrary string and s is a cryptographic signature. Verify outputs the Boolean value

  $$\textsc{SigVerify}(K_{VS},\ A_{\overline{O}} \| X,\ s).$$

- Reconstruct, on input $\overline{O}$, $K_R = K_E \| K_W$ and $K_V = K_{VS} \| K_W$, first runs Verify to assure the integrity of $\overline{O}$; in case Verify outputs false, Reconstruct exits with fail. Otherwise, it separates $\overline{O}$ (using Separate and key K_W) into the two parts $A_{\overline{O}}$ and $B_{\overline{O}}$. The latter part has the form $B_{\overline{O}} = X \| s$. By using K_E, Reconstruct decrypts X to obtain $C_O \| h$, where h denotes a hash; the part C_O is decompressed to obtain B_O. Finally, the part $B_{\overline{O}}$ of $\overline{O}$ is overwritten with B_O to obtain an object O. If $H(O) = h$, Reconstruct outputs O, otherwise fail.

Intuitively, the scheme is secure because of the following argument: in case an attacker modified the part $A_{\overline{O}}$ of $\overline{O}$, the embedded cryptographic signature s is matched against a modified string. On the other hand, if any bit in $B_{\overline{O}}$ is modified, then the embedded fragile watermark (containing either the signature s or the compressed part B_O) is destroyed. In all cases, the tampering will be detected during the verification step. Formally, we can state this result as a theorem:

Theorem 1 *If $\mathbb{S}$ is a cryptographic signature scheme secure against existential forgery of messages under a chosen message attack, then the above scheme is a secure media authentication scheme.*

Proof. Suppose, for the sake of contradiction, that there exists an attack $\textsc{Attack}$ (with access to the media authentication oracle $\textsc{Query}_{K_P}$) against the scheme, which succeeds with non-negligible probability. We show that in this case there exists also an attack $\textsc{Forge}$ (with access to a signing oracle $\textsc{SignQuery}_{K_{SS}}$) against $\mathbb{S}$, which contradicts the assumption.

We construct the signature forging algorithm $\textsc{Forge}$ (for the public signature key K_{VS}) in the following manner. On input K_{VS}, $\textsc{Forge}$ first chooses random keys K_E and K_W. Finally, $\textsc{Forge}$ simulates $\textsc{Attack}$. Whenever $\textsc{Attack}$ makes an oracle query $\textsc{Query}_{K_P}(O_i)$, this query is replaced by the following probabilistic algorithm, which utilizes the signing oracle $\textsc{SignQuery}_{K_{SS}}$; here, K_{SS} denotes the corresponding secret signature key:

$\langle A_{O_i}, B_{O_i}\rangle \leftarrow \textsc{Separate}(K_W, O_i)$
compress B_{O_i} to obtain C_{O_i}
$X_i \leftarrow \textsc{Encrypt}(K_E,\ C_{O_i} \,\|\, H(O_i))$
query $\textsc{SignQuery}_{K_{SS}}(A_{O_i} \,\|\, X_i)$ for signature s
$W'_i = X_i \,\|\, s$
output $\textsc{Join}(K_W, \langle A_{O_i}, W'\rangle)$

Note that $\textsc{Join}$ either outputs $\textsc{fail}$ or the watermarked version $\overline{O}_i$ of O_i.

When the simulation of $\textsc{Attack}$ is finished, $\textsc{Attack}$ either outputs $\textsc{fail}$ or obtains a tuple $\langle O, \overline{O}\rangle$. In the first case, $\textsc{Forge}$ exits with $\textsc{fail}$. Otherwise, $\textsc{Forge}$ runs $\textsc{Separate}$ on $\overline{O}$ and K_W, resulting in the tuple $\langle A_{\overline{O}}, B_{\overline{O}}\rangle$; $B_{\overline{O}}$ has the form $B_{\overline{O}} = X \,\|\, s$. Finally, $\textsc{Forge}$ outputs the pair $\langle A_{\overline{O}} \,\|\, X, s\rangle$. It is easy to see that $\textsc{Forge}$ perfectly simulates $\textsc{Attack}$ so that a valid pair of a message and a signature is produced if and only if $\textsc{Attack}$ succeeded.

It remains to show that the message $A_{\overline{O}} \,\|\, X$ was not presented to the signature oracle previously. For this, assume the contrary, i.e., that there exists an index i such that $A_{\overline{O}} \,\|\, X = A_{O_i} \,\|\, X_i$. This can only be the case if $A_O = A_{\overline{O}} = A_{O_i}$ and $X = X_i$, i.e., $\textsc{Encrypt}(K_E,\ C_O \,\|\, H(O)) = \textsc{Encrypt}(K_E,\ C_{O_i} \,\|\, H(O_i))$. This requires that both O and O_i agree on part A; furthermore, by $\textsc{Encrypt}$ being uniquely decipherable, we have $C_O \,\|\, H(O) = C_{O_i} \,\|\, H(O_i)$. This can only be the case if both O and O_i agree on part C and thus also on part B. We conclude that $O = O_i$, but this contradicts the definition of a successful attack against the media authentication scheme. This completes the proof. □

5 Online Media Authentication

The authentication method of the previous section assumes that the full media O is present when the media file is authenticated. However, for many multimedia applications such a solution is unacceptable, e.g., in audio or video streaming. In this section we present an online authentication scheme that operates only on fixed-length chunks of media at a time, but nevertheless allows the full media object to be authenticated. For this purpose, an object O is considered to consist of n chunks of equal length $O_1, \ldots, O_n$; in abuse of notation, we write $O = O_1 \parallel \cdots \parallel O_n$.

The online media authentication scheme presented in this paper is targeted towards applications where it must be possible to produce authenticated *excerpts*, i.e., small consecutive portions of the media stream. It is crucial that these excerpts can be produced without access to the secret protection key K_P. For example, consider the evidence produced by eavesdropping a telephone, which might be automatically authenticated by future devices; in a court hearing only a small and relevant part of the overall evidence is presented to the public. In order to prevent tampering, this excerpt should be produced *without* access to the secrets of the eavesdropping system. Nevertheless the integrity and authenticity of the excerpt should be publically verifiable.

Given an object O, we call an object O' an *excerpt* of O, if O' may be obtained from O by removing some chunks from the beginning and the end of O. Formally, $O' = O'_1 \parallel \cdots \parallel O'_m$ is an excerpt of $O = O_1 \parallel \cdots \parallel O_n$, written as $O' \preceq O$, if $m \leq n$ and there exists an index $1 \leq i \leq n - m$ so that $O'_1 = O_i, \ldots, O'_m = O_{i+m}$.

Given an original object O, it is possible with the proposed system to generate a signed object $\overline{O}$ such that *each* excerpt of the signed object $\overline{O}' \preceq \overline{O}$ can be checked for its integrity and authenticity. More precisely, the algorithm Verify will detect any modifications in an excerpt and will report the presence of non-consecutive chunks.

Formally, the attacker model we use for online authentication schemes is similar to the one presented in Section 3.2, with the exception that the production of excerpts is not considered an attack. Again, an attacker is forced to perform a selective forgery under a chosen message attack. However, the media object obtained at the end of the attack must not be an excerpt of an object submitted to the signing oracle previously.

Definition 3 *Let* $\langle \textsc{GenKey}, \textsc{Protect}, \textsc{Verify}, \textsc{Reconstruct} \rangle$ *an online authentication scheme and* $\textsc{Query}_{K_P}$ *be an oracle that, on input* O*, computes* $\overline{O} \leftarrow \textsc{Protect}'(O, K_P)$*. Furthermore, let* $\langle K_P, K_V, K_R \rangle \in [\textsc{GenKey}'(1^{n_K})]$*. An attack is a probabilistic algorithm* $\textsc{SAttack}$ *with oracle access to* $\textsc{Query}_{K_P}$ *and success probability* $\varepsilon_{\textsc{SAttack}}$ *such that*

$$\textsc{SAttack}(1^n, K_V) = \begin{cases} \langle O, \overline{O} \rangle & \textit{such that } \textsc{Verify}'(\overline{O}, K_V) = \textsc{true},\ |O| = n, \\ & \overline{O} \in [\textsc{Protect}'(O, K_P)] \textit{ and} \\ & O \not\preceq O^{(i)} \textit{ for all } 1 \leq i \leq l, \\ & \textit{with probability } \varepsilon_{\textsc{SAttack}} \\ \textsc{fail} & \textit{with probability } 1 - \varepsilon_{\textsc{SAttack}}, \end{cases}$$

where $O^{(i)}$ *denotes the input to the i-th oracle query* $\textsc{Query}_{K_P}$*. The probability is taken over all coin tosses of* $\textsc{SAttack}$ *and all keys* $\langle K_P, K_V, K_R \rangle$*.*

Again, we say that an online media authentication scheme is secure, if *every* probabilistic attack has only negligible success probability.

5.1 Construction

In this section, we provide the construction of an online media authentication scheme that operates blockwise on the media content. Essentially, we apply the authentication scheme described in the previous section on each chunk O_i, with the exception that the there is some linkage (computed by a hash function) between the chunks. Technically, we rely on the concept of hash chains [6].

Fix any collection of hash functions $\mathbb{H} = \left\langle H_h : \{0,1\}^* \to \{0,1\}^{\ell(|h|)} \mid h \in \{0,1\}^* \right\rangle$ for any super-logarithmically growing function $\ell : \mathbb{N} \mapsto \mathbb{N}$. Denote with k_h an index to $\mathbb{H}$; furthermore, let k be the length of the cryptographic signatures. We assume that both k_h and k are polynomial in the security parameter. For the construction we use an invertible watermarking scheme that is capable of storing $k+\ell(k_h)$ bits. The construction is as follows:

- GenKey runs GenSign to obtain a tuple of keys $\langle K_{SS}, K_{VS}\rangle$; furthermore it computes a key K_E for a symmetric cipher and a random string K_W. GenKey′ outputs the keys $K_P = K_{SS} \,\|\, K_W$, $K_V = K_{VS} \,\|\, K_W$ and $K_R = K_E \,\|\, K_W$.

- Protect, on input $O = O_1 \,\|\, \cdots \,\|\, O_n$, K_P and K_R, performs the following steps:

  $h_0 \leftarrow \textsc{Random}(\ell(k_h))$
  for $i = 1, \ldots, n$ **do**
    $\langle A_{O_i}, B_{O_i}\rangle \leftarrow \textsc{Separate}(K_W, O_i)$
    compress B_{O_i} to obtain C_{O_i}
    $X_i \leftarrow \textsc{Encrypt}(K_E,\ C_{O_i} \,\|\, H_h(O_i))$
    $s_i \leftarrow \textsc{Sign}(K_{SS},\ A_{O_i} \,\|\, X_i \,\|\, h_{i-1})$
    $h_i \leftarrow H(A_{O_i} \,\|\, X_i \,\|\, h_{i-1})$
    let $W_i = X_i \,\|\, h_{i-1} \,\|\, s_i$
    $\overline{O}_i \leftarrow \textsc{Join}(K_W, \langle A_{O_i}, W_i\rangle)$
    if $\overline{O}_i = $ FAIL, **exit with** FAIL
  end for
  output $\overline{O} = \overline{O}_1 \,\|\, \cdots \,\|\, \overline{O}_n$

- Verify, on input $\overline{O} = \overline{O}_1 \,\|\, \cdots \,\|\, \overline{O}_n$ and K_V, performs the following steps:

  for $i = 1, \ldots, n$ **do**
    $\left\langle A_{\overline{O}_i}, B_{\overline{O}_i} \right\rangle \leftarrow \textsc{Separate}(K_W, \overline{O}_i)$
    $B_{\overline{O}_i}$ has the form $X_i \,\|\, h_{i-1} \,\|\, s_i$
    if $i > 1$ and $h_{i-1} \neq \tilde{h}$ **exit with** FAIL
    let $\tilde{h} = H_h(A_{\overline{O}_i} \,\|\, X_i \,\|\, h_{i-1})$
    $b_i \leftarrow \textsc{SigVerify}(K_{VS},\ A_{\overline{O}_i} \,\|\, X_i \,\|\, h_{i-1},\ s_i)$
    if $b_i = $ FALSE, **exit with** FALSE
  end for
  exit with TRUE

- Reconstruct applies the reconstruction algorithm of Section 4 on the chunks of $\overline{O}$.

5.2 Security Against Forgeries

In a similar way as in Theorem 1, the security of the above scheme can be established:

Theorem 2 *If $\mathbb{S}$ is a cryptographic signature scheme secure against existential forgery of messages under a chosen message attack and if $\mathbb{H}$ is a collection of preimage- and collision-resistant hash functions, then the above scheme is a secure online media authentication scheme.*

Proof. Suppose, for the sake of contradiction, that there exists an attack SAttack against the above scheme, which succeeds with a non-negligible probability. We show that in this case there exists also an attack Forge against $\mathbb{S}$, which contradicts the assumption.

We construct the signature forging algorithm Forge (for the public signature verification key K_{VS}) in the following manner. On input K_{VS}, Forge first chooses random keys K_E and K_W. Finally, Forge invokes SAttack. In the rest of the proof, denote with $O^{(i)}$ the input to the i-th query to the oracle $\textsc{Query}_{K_P}$, whereas $O_j^{(i)}$ denotes the j-th chunk of $O^{(i)}$; the number of chunks in $O^{(i)}$ is given by n_i.

Whenever SAttack makes an oracle query $\textsc{Query}_{K_P}(O^{(i)})$ in order to obtain a signed stream $\overline{O}^{(i)}$, given $O^{(i)} = O_1^{(i)} \parallel \ldots \parallel O_{n_i}^{(i)}$, this query is simulated by the following probabilistic computation that uses a signature oracle $\textsc{SignQuery}_{K_{SS}}$ (essentially, this code is equivalent to that of Protect):

$s_{i,0} \leftarrow \textsc{Random}(\ell(h_k))$
for $j = 1, \ldots, n_i$ **do**
  $\left\langle A_{O_j^{(i)}}, B_{O_j^{(i)}} \right\rangle \leftarrow \textsc{Separate}(K_W, O_j^{(i)})$
  compress $B_{O_j^{(i)}}$ to obtain $C_{O_j^{(i)}}$
  $X_j^{(i)} \leftarrow \textsc{Encrypt}(K_E,\ C_{O_j^{(i)}} \parallel H_h(O_j^{(i)}))$
  $s_j^{(i)} \leftarrow \textsc{SignQuery}_{K_{SS}}(A_{O_j^{(i)}} \parallel X_j^{(i)} \parallel h_{j-1}^{(i)})$
  $h_j^{(i)} \leftarrow H_h(A_{O_j^{(i)}} \parallel X_j^{(i)} \parallel h_{j-1}^{(i)})$
  let $W_j^{(i)} = X_j^{(i)} \parallel h_{j-1}^{(i)} \parallel s_j^{(i)}$
  $\overline{O}_j^{(i)} \leftarrow \textsc{Join}\left(K_W, \left\langle A_{O_j^{(i)}}, W_j^{(i)} \right\rangle\right)$
  if $\overline{O}_j^{(i)} = \textsc{fail}$, **exit with** $\textsc{fail}$
end for
output $\overline{O}^{(i)} = \overline{O}_1^{(i)} \parallel \cdots \parallel \overline{O}_{n_i}^{(i)}$

Up to here, Attack perfectly simulates SAttack. When the simulation of SAttack is finished it obtains (with non-negligible probability) a tuple $\langle O, \overline{O} \rangle$, where $\overline{O}$ is a signed media stream with n chunks and $O \not\precsim O^{(i)}$ for all $1 \leq i \leq l$. If SAttack fails, Attack fails as well.

Denote with

$$\mathbf{Q} = \{A_{O_j^{(i)}} \parallel X_j^{(i)} \parallel h_{j-1}^{(i)} \mid 1 \leq i \leq l,\ 1 \leq j \leq n_i\}$$

the set of oracle queries. For all $1 \le k \le n$, Attack runs Separate on $\overline{O}_k$ and K_W to obtain $A_{\overline{O}_k} = A_{O_k}$ and $B_{\overline{O}_k}$; the latter string has the form $B_{\overline{O}_k} = X_k \,\|\, h_{k-1} \,\|\, s_k$. Consider two cases:

- Case 1: there exists an index $1 \le k \le n$ such that $A_{O_k} \,\|\, X_k \,\|\, h_{k-1} \notin \mathbf{Q}$. Then, Attack outputs the tuple
$$\left\langle A_{\overline{O}_k} \,\|\, X_k \,\|\, h_{k-1},\ s_k \right\rangle$$
as signature forgery. By assumption, this tuple is a valid forgery.

- Case 2: for all indices $1 \le k \le n$ we have $A_{\overline{O}_k} \,\|\, X_k \,\|\, h_{k-1} \in \mathbf{Q}$. In this case, Attack fails. We argue later that this case can happen only with negligible probability.

Attack can distinguish the two cases in polynomial time; furthermore, the success probability of Attack equals the success probability of SAttack, up to a negligible quantity (resulting out of case 2). This contradicts the assumption.

It remains to show that case 2 happens only with negligible probability. Note that, by assumption, O (and thus also $\overline{O}$) contains at least two chunks, as otherwise trivially $O \preceq O^{(i)}$ for some index $1 \le i \le l$. Consider the last chunk $\overline{O}_n$; its decomposition according to Separate is given by $\langle A_{\overline{O}_n}, X_n \,\|\, h_{n-1} \,\|\, s_n \rangle$. By assumption, there exist indices $1 \le i \le l$ and $1 \le j \le n_i$ such that

$$A_{O_j^{(i)}} \,\|\, X_j^{(i)} \,\|\, h_{j-1}^{(i)} = A_{O_n} \,\|\, X_n \,\|\, h_{n-1}.$$

In particular, also $h_{j-1}^{(i)} = h_{n-1}$. Distinguish two cases:

- Case (a): We have $j = 1$. Now, as both $\overline{O}$ and $\overline{O}^{(i)}$ are valid,
$$h_{n-1} = H_h(A_{O_{n-1}} \,\|\, X_{n-1} \,\|\, h_{n-2}).$$
By assumption, $h_{n-1} = h_0^{(i)}$, which shows that $A_{O_{n-1}} \,\|\, X_{n-1} \,\|\, h_{n-2}$ is a pre-image of the random string $h_0^{(i)}$.

- Case (b): We have $j > 1$. Again, as both $\overline{O}$ and $\overline{O}^{(i)}$ are valid,
$$h_{n-1} = H_h(A_{O_{n-1}} \,\|\, X_{n-1} \,\|\, h_{n-2})$$
and
$$h_{j-1}^{(i)} = H_h(A_{O_{j-1}^{(i)}} \,\|\, X_{j-1}^{(i)} \,\|\, h_{j-2}^{(i)}).$$
By assumption, $h_{n-1} = h_{j-1}^{(i)}$. If $A_{O_{n-1}} \,\|\, X_{n-1} \,\|\, h_{n-2} \neq A_{O_{j-1}^{(i)}} \,\|\, X_{j-1}^{(i)} \,\|\, h_{j-2}^{(i)}$, we have found a collision of H_h. Otherwise, $A_{O_{n-1}} = A_{O_{j-1}^{(i)}}$, $h_{n-2} = h_{j-2}^{(i)}$ and $X_{n-1} = X_{j-1}^{(i)}$. The latter equation implies
$$\underbrace{\textsc{Encrypt}(K_E,\ C_{O_{n-1}} \,\|\, H(O_{n-1}))}_{X_{n-1}} = \underbrace{\textsc{Encrypt}(K_E,\ C_{O_{j-1}^{(i)}} \,\|\, H(O_{j-1}^{(i)}))}_{X_{j-1}^{(i)}}.$$

Since Encrypt is uniquely decipherable, $C_{O_{n-1}} = C_{O^{(i)}_{j-1}}$, implying that $B_{O_{n-1}} = B_{O^{(i)}_{j-1}}$. This shows that now O and $O^{(i)}$ also agree on their second-last chunk. By assumption, O must therefore have at least one more chunk (as otherwise trivially $O \preceq O^{(i)}$). Applying this argument inductively, we either find a collision or have $n > j$. In the latter case, as in case (a), $A_{O_{n-j-1}} \,\|\, X_{n-j-1} \,\|\, h_{n-j-2}$ is a pre-image of $h^{(i)}_0$.

In summary, if case 2 happens, then we can either find a pre-image of a random string with respect to H_h or a collision of H_h (a formal proof of this claim uses again a reducibility argument). By the assumptions on $\mathbb{H}$, this can happen only with negligible probability. This completes the proof. □

6 Conclusions

Digital watermarking used to be dominated by signal processing approaches which typically did not provide any formal security guarantees. Currently, there is a trend to substantiate watermarking technology with a cryptographic foundation. In addition, the issue of data authenticity and data integrity for multimedia applications has become an active research topic in the watermarking community.

In this paper, we provide an approach which solves the data integrity problem for multimedia applications by combining methods from cryptography and watermarking. In particular, we present an offline media authentication scheme, an appropriate attacker model, and a security proof with respect to this attacker model. Furthermore, we provide an authentication scheme for online media streaming applications which has the following two properties: First, it is possible to verify the integrity and authenticity of an arbitrary excerpt of the signed object. Second, the generation of an excerpt is possible without access to the secret signing keys.

References

[1] J. Dittmann and O. Benedens. Invertible authentication for 3d-meshes. In *Proceedings of the SPIE vol. 5020, Security and Watermarking of Multimedia Contents V*, pages 653–664, 2003.

[2] J. Dittmann, M. Steinebach, and L. Ferri. Watermarking protocols for authentication and ownership protection based on timestamps and holograms. In *Proceedings of the SPIE vol. 4675, Security and Watermarking of Multimedia Contents IV*, pages 240–251, 2002.

[3] J. Fridrich, M. Goljan, and R. Du. Invertible authentication. In *Proceedings of the SPIE vol. 3971, Security and Watermarking of Multimedia Contents III*, pages 197–208, 2001.

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

[5] G. L. Friedman. The trustworthy digital camera. *IEEE Transactions on Consumer Electronics*, 39(4):905–910, 1993.

[6] R. Gennaro and P. Rohatgi. How to sign digital streams. In *Advances in Cryptology (CRYPTO'97)*, volume 1294 of *Lecture Notes in Computer Science*, pages 180–197. Springer, 1997.

[7] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–302, 1988.

[8] 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.

[9] S. Katzenbeisser and F. A. P. Petitcolas, editors. *Information Hiding Techniques for Steganography and Digital Watermarking.* Artech House, 2000.

[10] D. Maas, T. Kalker, and F. M. Willems. A code construction for recursive reversible data-hiding. In *Proceedings of the ACM Workshop on Multimedia*, pages 15–18, 2002.

[11] M. Schneider and S.-F. Chang. A robust content based digital signature for image authentication. In *IEEE International Conference on Image Processing, Proceedings*, Lausanne, 1996.

[12] M. Steinebach and J. Dittmann. Watermarking-based digital audio data authentication. *EURASIP Journal on Applied Signal Processing*, (10):1001–1015, 2003.

[13] L. Xie and G. R. Arce. A blind wavelet based digital signature for image authentication. In *European Signal Processing Conference, Proceedings*, Rhodes, Greece, 1998.