Advanced Techniques for Dispute Resolving and Authorship Proofs on Digital Works

André Adelsbach[a] and Ahmad-Reza Sadeghi[a]

[a]Department of Computer Science, Saarland University, Saarbrücken, Germany *

ABSTRACT

Digital watermarking is a promising technology for protecting intellectual property rights on digital content. Resolving authorship-disputes was one of the first and most propelling applications of robust digital watermarks, and much research effort has gone into protocols for resolving authorship-disputes by means of digital watermarks. Unfortunately, previous proposals lack formal definitions of their trust model, their assumptions, and requirements they should fulfill. This lack of formal definitions makes security proofs for such protocols impossible and many dispute resolving protocols, claimed to be secure, can be shown to be insecure.

In this paper we set off to rigorously defining dispute resolving schemes based on a reasonable formal definition of "authorship". Building on this formal fundament, we analyze the most important proposals for dispute resolving, and discuss their connection to our authorship model. We show that existing proposals suffer from two major problems: First, they require an unnecessary high level of trust in the dispute resolving party. The second and even more serious is that the winner of the dispute is not guaranteed to be the rightful author of the disputed work (conclusiveness problem). As solutions, we propose dispute resolving schemes based on zero-knowledge watermark detection and asymmetric watermarking schemes.

Keywords: Copyright protection, authorship-dispute resolving, direct authorship proofs, zero-knowledge watermark detection, asymmetric watermarks

1. INTRODUCTION

The need for copyright protection solutions has increased steadily with the rapid development in digital processing and distribution techniques, since they can be used to violate the intellectual property rights of parties involved in the whole distribution chain of digital content such as authors and content distributors.

Robust digital watermarks are a promising technique in the context of copyright protection. Besides copy protection and identification of pirates (fingerprinting), *proving authorship* for digital works is the most prominent and most basic application of digital watermarks. Today, we distinguish two kinds of authorship proofs which have very different properties: *authorship-proofs in dispute scenarios* and *direct authorship-proofs*.

In an authorship-dispute[†] two or more parties, the so called *authorship claimants* claim to be the rightful author of a *disputed work* W_{dis}. Loosely speaking, the goal of an *authorship-dispute resolving scheme* is to allow the trusted *dispute resolver* to resolve authorship-disputes in a *"fair"* way[‡] by comparing the proofs presented by the disputants. *Direct authorship proofs* consider a two-party proof scenario of mutually distrusting parties: an authorship claimant claims to be the rightful author of a work and another party D, e.g., a customer, wants to verify this claim. Such scenarios are quite common when trading digital works electronically and are an important prerequisite for faithful commerce with digital works, especially for non-famous authors and works.

Further author information: (Send correspondence to André Adelsbach)

André Adelsbach (adelsbach@cs.uni-sb.de) Ahmad-Reza Sadeghi (sadeghi@cs.uni-sb.de), Address: Department of Computer Science, Security and Cryptography Group, Postfach 151150, D-66041 Saarbrücken, Germany

[†]Note that authorship-disputes exist as long as copyrights (or even longer) and resolving such disputes in court is common practice. Dispute resolving, as discussed in this paper, investigates the machine-aided resolving of authorship-disputes, among other things by means of digital watermarks.

[‡]Note that usually, one would expect that disputes, if resolvable, should be resolved in favor of the *real author*. As traditional schemes do not achieve this ideal goal *(conclusiveness-problem)* (see "soundness" in Section 3.3), we chose the phrase "fair".

Previous Work Since the invention of digital watermarks the problem of proving authorship for digital works has been the subject of intense research and has undergone considerable development. Early publications, such as [1, 2, 3], focused on developing robust watermarking schemes and treated authorship proofs in a rather informal and simplistic way. The common belief was that embedding the author's identity as a watermark into all his works prior to publication and proving the presence of this watermark later in some work would be sufficient for the author to prove rightful authorship for it. Further details on the process of dispute resolving were left open.

Later, Craver et al.[4] investigated the process of watermark-based dispute resolving in more detail. They demonstrated the insufficiency of previous proposals for dispute resolving by showing that many watermarking schemes are *invertible*: An adversary can exploit this property to generate fake pieces of evidence (watermark, watermarking key and original work), which results in an *authorship-deadlock*, i.e., prevents authorship-disputes from being resolved (see Section 3.2). As a solution Craver et al. proposed constructions for making watermarking schemes *non-invertible* and argued that such non-invertible schemes would be sufficient for dispute resolving. However, neither a proof for non-invertibility for their constructions nor was a proof given that non-invertibility is sufficient for resolving authorship-disputes. In fact [5] showed that a non-invertibility construction of Craver et al. is insecure.

Subsequent research[6, 7, 8, 9, 10, 5] mainly focused on countermeasures against the attacks proposed by Craver et al. and tried to apply cryptographic techniques, such as digital signatures, one-way functions and time-stamping to the authorship-deadlock problem.

In [11] Adelsbach et al. introduced a first formal definition of authorship and they realized that all previous proposals achieved only dispute resolving. Furthermore, proposed dispute resolving schemes do not even guarantee that the winner of a dispute is the real author. Therefore, they introduced the concept of direct authorship proofs, which are suitable for e-commerce scenarios, e.g., when trading digital works. The authors introduced the first generic schemes for direct proofs of authorship, proved their schemes secure, and showed concrete instantiations of their schemes based on robust watermarks and robust features. Later, Adelsbach and Sadeghi[12] introduced an improved protocol for direct authorship proof which uses provably secure zero-knowledge watermark detection protocols.

Thus, previous proposals for authorship-dispute resolving suffer from following deficiencies:

1. The proposals lack formal and explicit definitions, especially, of their trust model and their security requirements. As a consequence thereof, previous proposals lack a proper security analysis.
2. The proposed schemes require unnecessary high trust in the dispute-resolver.
3. The proposed schemes do not guarantee that the winner of a dispute is the real author.

Our Contribution We start in Section 2 by introducing basic notations and definitions. Then, we formally define authorship-dispute resolving schemes, i.e., their protocols and the basic requirements these protocols have to fulfill (Section 3.1). Guided by our authorship model, we review and discuss means of proving authorship in disputes in Section 3.2. Following this discussion, we introduce classification criteria and reasonable optional requirements in Section 3.3, which previous proposals for dispute resolving do not fulfill. In Section 4, we introduce two advanced dispute resolving schemes: The first scheme uses zero-knowledge watermark detection to reduce the trust required in the dispute resolving party. The second scheme is a general construction based on a special class of asymmetric watermarking schemes, which guarantees that if disputes are resolved, then the result is in favor of the real author (*soundness*). The latter is an important prerequisite for the winner of a dispute being able to enforce legal actions against the other disputant or to claim compensation for losses.

2. BASIC NOTATIONS AND MODEL

This section defines the basic notions such as watermarking schemes, zero-knowledge watermark detection, and recapitulates the authorship model, as introduced in [11]. This builds the basis of our formal definition and formal analysis of dispute resolving schemes. Due to lack of space, we omit success-probabilities in the following definitions. Let $\mathcal{ID}$ be the set of unique identifiers of parties (authors, disputants, etc.). The objects that authorship, and hence authorship-disputes, refer to are *works* W. Let $\mathcal{W}$ be the set of all works of a certain data type, e.g., images, video-clips or music.

2.1. Digital Watermarking Schemes

Robust digital watermarking schemes embed additional information in digital objects (cover-data) such that this information can later be detected or extracted again and cannot be removed by an adversary. The following definition states this intuition of an ideal robust digital watermarking scheme formally.

DEFINITION 2.1 (IDEAL ROBUST WATERMARKING SCHEMES). *Let $\mathcal{W}$ be the set of all cover-data and $\mathcal{WM}$ be the set of all watermarks. A* detecting/[extracting] ***watermarking scheme*** *consists of three probabilistic polynomial-time algorithms* **GenKey**, **Embed**, **Detect**/[**Extract**]*, such that for arbitrary cover-data $W \in \mathcal{W}$, arbitrary watermarks $WM \in \mathcal{WM}$ and watermarking keys $(K^{Emb}, K^{Det/Ext}) \xleftarrow{\mathcal{R}} \mathsf{GenKey}$ the following holds:*

$$W' = \mathsf{Embed}(W, WM, K^{Emb}) \text{ and } W' \text{ is perceptually similar to } W, \tag{1}$$

$$\mathsf{Detect}(W', W, K^{Det/Ext}, WM) = \mathsf{true} \qquad [\text{resp. } \mathsf{Extract}(W', W, K^{Det/Ext}) = WM] \tag{2}$$

With "perception" of digital data we mean the perception of its usual interpretation.

We call such keys $(K^{Emb}, K^{Det/Ext})$ ***matching keys*** *and we call a watermarking scheme* ***symmetric*** *iff $K^{Det/Ext} = K^{Emb}$. In this case we usually denote this embedding/detection key as K^{WM}.* ***Blind*** *watermarking schemes do not require the cover-data W as an input to* Detect/Extract*. A blind watermarking scheme with $K^{Det/Ext} \neq K^{Emb}$ is called* ***asymmetric****.*

A symmetric watermarking scheme is called ***robust****, iff it is computationally infeasible for an adversary, given a watermarked work W' and the watermark WM, to produce a perceptually similar work, in which the watermark cannot be detected/extracted anymore. An asymmetric watermarking scheme is called* ***robust****, iff it is computationally infeasible for an adversary, given a watermarked work W', the watermark WM* and the public detection/extraction key *$K^{Det/Ext}$, to produce a perceptually similar work, in which the watermark cannot be detected/extracted anymore.*

2.2. Zero-Knowledge Watermark Detection

Basically, zero-knowledge watermark detection is a zero-knowledge proof system (see [13]) and the zero-knowledge property guarantees that a run of the protocol does not give any "new" knowledge to the verifier. The usual definition of "new knowledge" covers everything, which the verifier cannot efficiently compute from the common inputs on his own, i.e., without interacting with the prover. Requiring the zero-knowledge watermark detection to be a zero-knowledge protocol is necessary, but not sufficient, because it does not exclude trivial "protocols" that give all critical detection inputs directly and unconcealed to the verifier and let him perform detection with the usual Detect-algorithm. In this case there is simply no "new knowledge" about these inputs, which the verifier may obtain during the protocol run.

Therefore, the common detection inputs have to be concealed by suitable cryptographic measures, such as commitment schemes:

DEFINITION 2.2 (COMMITMENT SCHEMES). *A* commitment scheme *$(\mathsf{Commit}(), \mathsf{Open}())$ for the message space M and commitment space C consists of a two-party protocol* Commit() *to commit to a value $m \in M$ and a protocol* Open() *that opens a commitment. A commitment to a value m is denoted by $com(m) = \mathsf{Commit}(m, par_{com})$ where par_{com} stands for all public parameters needed to compute the commitment value. To open a commitment com to the verifier, the committer runs the protocol $\mathsf{Open}(com, par_{com}, sk_{com})$ where sk_{com} is the secret opening information of the committer. For brevity we sometimes omit par_{com} and sk_{com} in the notation of* Commit() *and* Open()*. Furthermore, we use* Commit() *and* Open() *on tuples over M, with the meaning of component-wise application of* Commit() *or* Open()*.*

The security requirements are the binding (committing) *and* hiding (secrecy) *properties. The first one requires that a dishonest committer cannot open a commitment to another message $m' \neq m$ than the one to which he committed and the second one requires that the commitment does not reveal any information about the message m to the verifier.*

A suitable commitment scheme has been introduced in [14]. Now we can formally define zero-knowledge watermark detection as follows:

DEFINITION 2.3 (ZERO-KNOWLEDGE WATERMARK DETECTION). *Let $(\mathsf{Commit}(), \mathsf{Open}())$ be a secure commitment scheme. A* zero-knowledge watermark detection protocol ZKDetect() *for the watermarking scheme $(\mathsf{GenKey}, \mathsf{Embed}, \mathsf{Detect})$ is a* zero-knowledge proof of knowledge protocol[13] *between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$: The common protocol input of $\mathcal{P}$ and $\mathcal{V}$ is the stego-data W'', $com(WM)$, $com(W)$, $com(K^{WM})$, i.e., commitments on the watermark, the reference data and the detection key respectively, as well as the public parameters*

$par_{com} = (par_{com}^{WM}, par_{com}^{W}, par_{com}^{K^{WM}})$ *of these commitments. The private input of the prover is the secret opening information of these commitments* $sk_{com} = (sk_{com}^{WM}, sk_{com}^{W}, sk_{com}^{K^{WM}})$.
$\mathcal{P}$ *proves knowledge of a tuple* $(WM, W, K^{WM}, sk_{com}^{WM}, sk_{com}^{W}, sk_{com}^{K^{WM}})$ *such that:*

$$\begin{aligned}[(\mathsf{Open}(com(WM), par_{com}^{WM}, sk_{com}^{WM}) = WM) \wedge \\ (\mathsf{Open}(com(W), par_{com}^{W}, sk_{com}^{W}) = W) \wedge \\ (\mathsf{Open}(com(K^{WM}), par_{com}^{K^{WM}}, sk_{com}^{K^{WM}}) = K^{WM}) \wedge \\ \mathsf{Detect}(W'', W, K^{WM}, WM)] = \quad \mathsf{true}\end{aligned}$$

The protocol outputs a boolean value to the verifier, stating whether to accept the proof or not.

It is straightforward to see that the execution of such a zero-knowledge watermark detection protocol does not reduce the security (robustness) of a watermark. Currently, we are only aware of one scheme, which fulfills this strong definition. It was introduced in [12], where we already showed its usefulness for authorship proofs. Here, we will make use of zero-knowledge watermark detection to reduce the trust necessary in the dispute resolver.

2.3. Works and Similarity

Generally speaking, authorship does not refer to a single work only, but rather to a set of closely related *perceptually similar or derived* works. Consider for example an artist A who created a digital image W. Then, naturally and legally, a rotated or compressed version W' of W is also considered to be a creation of this artist and, vice versa, A is considered to be the rightful author of W' too. This is because rotating or compressing a digital image is no creative achievement and does not lead to new original works of authorship.

In the following, we assume a *similarity relation* $\rightarrow_{sim}$ on works to be given. Here, "$W \rightarrow_{sim} W'$" denotes the fact that "W' is similar to W". Works similar to a work W are called "the *similarity-set* of W" and are denoted as $\mathcal{W}_W^{sim} := \{ W' \in \mathcal{W} |\ W \rightarrow_{sim} W' \}$.

For judging whether a work W' is similar to a work W, i.e., lies in the similarity-set $\mathcal{W}_W^{sim}$, we assume the existence of an *automatic similarity-test* resp. a protocol for proving similarity of works. There are different reasonable ways for testing/proving similarity between works and we note that conformance with copyright-laws strongly depends on a suitable definition of similarity. We will discuss this issue in more depth in Section 3.2.

2.4. Authorship Model

We model *authorship* as a *family of relations* $(\sim^t)_{t \in \mathbb{N}}$, which is indexed by a discrete time-parameter t. Each relation $\sim^t$ of this family is a binary relation between an author A and works $W^\star$. The family of authorship relations $\sim^t$ is defined using a family of ternary auxiliary authorship relation $(\approx^t)_{t \in \mathbb{N}}$:

Definition 2.4 (Auxiliary Authorship Relation). *Let* $A \in \mathcal{ID}$ *and* $W, W^\star \in \mathcal{W}$. *Then, the* auxiliary authorship-relation $\approx_W^t \subseteq \mathcal{ID} \times \mathcal{W} \times \mathcal{W}$ *is defined as:*

$$\begin{aligned}A \approx_W^t W^\star \quad :\Longleftrightarrow \quad & \underbrace{(A \text{ created } W \text{ at time } t_W \leq t)}_{1} \wedge \underbrace{(W \rightarrow_{sim} W^\star)}_{2} \\ & \wedge \underbrace{(\nexists \hat{W} \neq W : (\hat{W} \rightarrow_{sim} W^\star) \wedge (\hat{W} \text{ was created at time } t_{\hat{W}} < t_W))}_{3}.\end{aligned} \tag{3}$$

$A \approx_W^t W^\star$ means that "A is the rightful author of $W^\star$ at time t due to the creation of the original work W".

By $\mathcal{W}_W^{\approx} := \{ W'' \in \mathcal{W} |\ A \approx_W^t W'' \}$ we denote the *creation-set* which is considered as being a creation of A due to the creation of W.

Definition 2.5 (Authorship Relation). *Let* $A \in \mathcal{ID}$ *and* $W^\star \in \mathcal{W}$. *Then, the family of authorship relations* $\sim^t \subseteq \mathcal{ID} \times \mathcal{W}$ *is defined as:* $\quad A \sim^t W^\star \quad :\Longleftrightarrow \quad \exists W \in \mathcal{W} : A \approx_W^t W^\star$

The rationales behind this definition of authorship are explained in the following, where the numbering resembles the numbering in Definition 2.4.:

1. Prerequisite for A being the author of a work $W^\star$ at time t is that A previously (at time t_W) created a certain original work W.

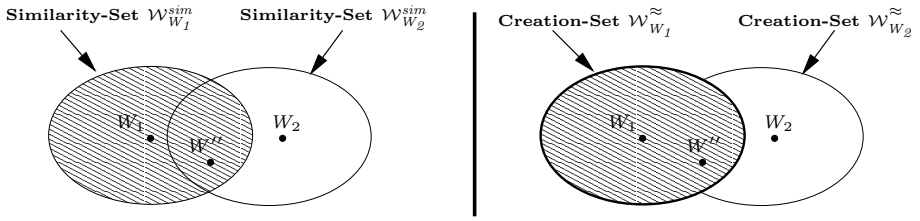

Figure 1. Left side: Intersecting *similarity sets* and the resulting authorship-ambiguity. Right side: *Authorship sets* as given by the authorship relation.

2. Authorship not only concerns the original W which has been created, but extends to similar, trivially derivable works, such as compressions or scalings, etc. of images. Part 2 of equation 3 captures that, because it only requires W'' to be similar to the original creation W and not to be the original creation itself.
3. Realistic similarity relations are no equivalence relations[§]. Thus, a scenario as shown on the left side of Figure 1 may be possible, where the similarity-sets of two independent creations W_1, W_2 by authors A_1 resp. A_2 intersect. Assume that W'' is an element of this intersection. For making the authorship relation unique an additional criterion is used to uniquely identify the author: the *creation time* of the original work. In case A_1 created her original W_1 before A_2 created her original W_2, our authorship model considers only A_1 to be the author.[¶] The resulting authorship partitioning is illustrated on the right side of Figure 1.

In Section 3.2 we will discuss measures of how to prove these authorship-conditions.

3. DISPUTE RESOLVING

To be able to analyze the proposed schemes, identify their shortcomings or prove such schemes correct, it is essential to have a formal definition of authorship-dispute resolving schemes and the corresponding requirements. We start with a formal definition of dispute resolving schemes in Section 3.1. Most dispute resolving schemes we are aware of fit into our definitional framework. Definition 3.2 formally states the basic requirements a secure dispute resolving scheme should fulfill. In Section 3.2 we discuss various aspects of proving rightful authorship in disputes and review corresponding techniques. Finally, Section 3.3 discusses variants of dispute resolving schemes and optional requirements of these variants.

3.1. Scheme

In schemes for dispute resolving, the following main roles can be distinguished: Two disputants A_1, A_2, a dispute resolver D (e.g., a judge), and eventually a further party T (e.g., a time-stamping service or a registration center). Depending on the scheme, D and T may be fully or partially trusted.

In the following, we assume that every correct party X knows its own identity, the identities of all other participants of a protocol run and all necessary keys (i.e., his key pair (sk_X, pk_X) and the public-keys of the other participants) and provides them as input to the following protocols. Furthermore, we assume that each party X has a local memory (state) mem_X whose content is a further implicit input to these protocols, and which is possibly changed to mem'_X during a protocol run. These standard in-/outputs will not be explicitly mentioned in the following in order to keep the list of in-/outputs lucid.

[§]They are not symmetric in general (e.g., a rough map is a trivial derivation of a detailed map of the same geographic area, but not vice versa).

[¶]The rationale behind using the time of creation as an indication for authorship is that A_2 could have obtained A_1's original and derived her faked original from it. In practice one could still think of additional measures that allow A_2 to prove that she created W_2 independently from W_1, but this is not manageable in an automatic process. Thus, we do not take it into account in this authorship-model.

The following definition gives a formal and detailed specification of these protocols.

DEFINITION 3.1 (DISPUTE RESOLVING SCHEME). *An* authorship-dispute resolving scheme *consists of three protocols* **Initialize()**, **Prepare()** *and* **Resolve()**, *which fulfill the requirements from definition 3.2:*

- *The initialization protocol* **Initialize()** *sets up the system and covers actions, which are necessary for the following protocols. The output to participant $X \in \mathcal{ID}$ is a key pair (sk_X, pk_X) and the authentic and integer public-keys of all other participants. (These keys may comprise several keys, depending on which cryptographic techniques are used in the following protocols. Key distribution can be achieved by means of a "public-key infrastructure" (PKI) and could also be performed on demand.)*
- *The preparation protocol* **Prepare()** *prepares a newly created work W in such a way, that the rightful author A prove his rightful authorship in subsequent disputes, which may arise for this work (and all similar works). It is either a local algorithm for A or a two-party protocol between A and a party T. The input of A is the original work W, which he created and which should be protected in future disputes.*

  The protocol output to A is a proof token $proof_A$, a possibly modified[‖] version W' of the original work and a boolean value $result_A$. The latter indicates to A whether the protocol-run was successful or not. In case party T participates, this party inputs its standard inputs and obtains a boolean value $result_T$ which indicates whether the protocol run was successful for T.
- *Up to four parties participate in the dispute resolving protocol* **Resolve()**. *At least the two disputants A_1, A_2 and the dispute resolver D are participants. In some schemes participation of T may be necessary in this protocol as well. In addition to their standard inputs, every disputant inputs the disputed work W_{dis} and its corresponding proof tokens $proof_{A_1}$ and $proof_{A_2}$ respectively. D and T input their standard inputs.*

  The output $result_D$ for D is the identity of the party in whose favor the dispute is resolved or failed *if the dispute could not be resolved. Note that disputes may be resolved in favor of a party, which is not one of the initiating disputants (see "*soundness *of* Resolve()*" in Section 3.3).*

The following definition summarizes the basic requirements on dispute resolving schemes:

DEFINITION 3.2 (BASIC REQUIREMENTS ON DISPUTE RESOLVING SCHEMES). *Let A_1, A_2 be mutually mistrusting disputants. Furthermore, let D be a trusted dispute resolver and let T be a trusted third party. Moreover, assume that* Initialize() *has already been performed.*
For the rightful author A_i of a work W_i:

- ***Completeness of* Prepare()***: If $A_i \sim^t W_i$ holds, then a run* Prepare(W_i)$_t$ *at time t ends with (W_i', $proof_{A_i}$,* true*) to A_i, where $W_i \rightarrow_{sim} W_i'$ holds.*
- ***Completeness of* Resolve()***: If $A_i \approx^t_{W_i} W_{dis}$ $(i \in \{1,2\})$ and A_i has successfully performed (W_i', $proof_{A_i}$,* true*) =* Prepare(W_i) *beforehand then a protocol run* Resolve(W_{dis}, $proof_{A_i}$, •)$_t$ *at time t ends with output $result_D = A_i$ to D.*

It is straightforward to see that the restriction of dispute resolving schemes to two-party disputes is no real restriction[**].

3.2. Proving Authorship in Disputes

From a legal point of view, creation of the original work is completely sufficient for authorship to be established and for the authorship relation to be well defined. However, for authorship to be *provable and verifiable*, the author has to be able to prove that the conditions required by the authorship relation hold.

Furthermore, authors have to arrange themselves for proving the required authorship-conditions: creation of an original work, similarity of the disputed work to this original work, and time/order of creation of the alleged originals (see Definition 2.5). Measures for proving these facts are discussed below and most of them require the author to perform certain preparations before publishing his work (Prepare()-protocol).

[‖] This is necessary to capture schemes which embed watermarks, e.g., as evidence for the similarity of one work to another. In this case, A should distribute only versions of his work, which have been derived from W'.

[**] We can easily construct n-party dispute resolving schemes for n disputants which satisfies the above requirements by performing the two-party dispute resolving protocol pairwise for all pairs of disputants and resolving the dispute in favor of the disputant who won all pairwise disputes or end with failed otherwise.

Suppose an authorship-dispute, where two disputants A_1 and A_2 claim to be the rightful author of a work W_{dis} and justify their claims by means of their proof tokens $proof_{A_1}$ and $proof_{A_2}$ respectively. In this situation, the dispute-resolver D compares the presented proof tokens and resolves the dispute in favor of the disputant whose proof token complies most with the authorship-conditions:

1. **Disputant A_i created his alleged original work W_i (ProveCreation()):** Obviously, proving creation of an (original) work is highly dependent on the type of work and how such works are typically created. For instance, for photographies one may use a special tamper-resistant trustworthy digital camera which produces a digital signature $sig_{Cam}(A_i, W_i)$, which certifies that A_i created photography W_i.

   For digital works produced by a certain application, proving creation of W_i may be possible by providing the *sequence of instructions*, which leads from an empty work to W_i and documents the creation of W_i. Such instruction sequences may be automatically produced by the application. Alternatively, the *source files* may be considered as a proof of creation. Commonly, weaker conditions are being proved in dispute resolving schemes: disputants only prove knowledge/existence of an alleged original work, for which the following two conditions hold.[††]

2. **Disputed work W_{dis} is similar to the original work W_i (ProveSim()):** In [11] various similarity tests/proofs, e.g., by means of robust digital watermarks and robust features/hashes, have been discussed. Here, we focus on proving similarity of W_{dis} to W_i ($W_i \rightarrow_{sim} W_{dis}$) by using robust non-blind watermarking schemes: In Prepare() the rightful author A_i generates a watermark WM_{A_i}, a watermarking key $K_{A_i}^{WM}$ and computes the marked version $W_i' := \mathsf{Embed}(W_i, WM_{A_i}, K_{A_i}^{WM})$. Later, in runs $\mathsf{Resolve}(W_{dis}, \bullet)$, A_i can prove $W_i \rightarrow_{sim} W_{dis}$ to D, by showing that $\mathsf{Detect}(W_{dis}, W_i, K_{A_i}^{WM}, WM_{A_i}) = \mathsf{true}$. To prove the latter, current proposals require A_i to give all information necessary for detection, i.e., W_i, WM_{A_i} and $K_{A_i}^{WM}$, to D. Note that this requires a unnecessary high level of trust in D and we will discuss possible improvements in the following Sections.

   This proof of similarity takes advantage of the non-blindness of the detection process: detection of a certain watermark in W_{dis}, using W_i as the reference work, implies that a close relation holds between W_{dis} and W_i, which is considered to be a proof of similarity[‡‡]. Thus, blind watermarking schemes are not as suitable for proving similarity between works as non-blind watermarking schemes. The same holds for asymmetric watermarking schemes, since they are inherently blind.

3. **Order of creation of the alleged original works W_1, W_2 (ProveOrder()):** This condition becomes important only in case both disputants have presented valid alleged originals W_1, W_2 and W_{dis} is similar to both W_1 and W_2.

   Two approaches have been proposed to determine the order of creation of the supposed original works:

   (a) *Use of non-invertible robust watermarks*[4, 9]*:* The idea behind using robust watermarks is that generally one of the disputants, say A_2, must be dishonest and must have derived his alleged original W_2 from the prepared (i.e., watermarked) work W_1' of the rightful author A_1. Hence, by the robustness of the watermarking scheme, the watermark WM_{A_1} of the rightful author is still detectable in A_2's alleged original W_2 ($\mathsf{Detect}(W_2, W_1, K_{A_1}^{WM}, WM_{A_1}) = \mathsf{true}$). In contrast, A_2's watermark WM_{A_2} is *generally not detectable* in the real original W_1, because it is A_1's own creation and has not been derived from a watermarked version W'_2 of W_2. However, as Craver et al.[4] showed, the latter "assumption" does not hold for several watermarking schemes, since they are invertible: Invertibility of the underlying watermarking scheme allows the dishonest disputant A_2 to compute a fake original W_2, a fake watermark WM_{A_2} and a fake watermarking key $K_{A_2}^{WM}$, such that WM_{A_2} is detectable in the rightful author's original W_1 ($\mathsf{Detect}(W_1, W_2, K_{A_2}^{WM}, WM_{A_2}) = \mathsf{true}$). Therefore, D cannot

[††] However, we note that resolving disputes without requiring proofs of creation makes it generally easier for an adversary to present a valid alleged original for the disputed work which may lead to an authorship-deadlock (see below).

[‡‡] In general, this argument only holds if the false-positive probability of the watermarking scheme is reasonably small, i.e., watermarks are not detected by accident, but only if they have been embedded. Therefore, the lower the false-positive probability, the more persuading the similarity proof by means of watermarks is. Thus, we have to guarantee that the false-positive probability is reasonably small.

determine the order of creation based on invertible watermarking schemes and the dispute cannot be resolved (*authorship-deadlock*).

To counter this authorship-deadlock problem, several constructions have been proposed which apply cryptographic techniques, e.g., one-way hash functions[4] and encryption schemes[9] , to prevent this attack and make watermarking schemes non-invertible. A common problem of non-invertibility constructions based on cryptographic primitives (one-way functions) is that they offer only marginal security if the false-positive probability of the watermarking scheme is non-negligible. This is because a non-negligible false-positive probability makes brute-force trial-and-error attacks feasible, which do not require the cryptographic primitive to be broken (inverted). This is how Ramkumar and Akansu[5] broke a non-invertibility construction proposed by Craver et al[4] . The improved scheme proposed in [5] intends to avoid this problem by excluding false-positive watermarks heuristically. For this, they introduce the additional statistical requirement that a watermark should not correlate with original work.

Summarizing the previous discussion, we advocate the use of a cryptographic time-stamping service for proving the time of creation as described below, since their security is better analyzed.

(b) *Certification of creation time*[15, 10]*:* One possibility for certification is to use tamper-resistant hardware, e.g., a camera with a tamper-resistant time-stamping module, which produces a digital signature certifying the *time of creation*. Another possibility is to use a *time-stamping service*[16] which is a well-known cryptographic technique for providing evidence for the existence of a document at a certain point in time. Haber and Stornetta[16] introduce a time-stamping service *with linking*, which reduces the trust required in the time-stamping authority. For this, the rightful author requests a time-stamp TS_{W_1} for his original W_1 from a time-stamping service T as part of the Prepare()-protocol. This time-stamp TS_{W_1} is included as a further component into A_1's proof token $proof_{A_1}$ and can be verified by D to determine the time of creation of W_1. If time-stamps are being used to prove the time of creation to D, non-invertibility is no longer a necessary requirement for watermarking scheme used for proving similarity.

3.3. Classification Criteria and Optional Requirements

Dispute resolving schemes can be classified according to additional attributes among which the most important are:

1. **Trust in the dispute resolver:** Any meaningful dispute resolving scheme must at least assume the dispute resolver D to be *correct in the sense that he behaves as specified by the* Resolve()*-protocol*. Regarding the trust required beyond D's correctness, we can at least distinguish two types of schemes: **disclosing** and **non-disclosing** schemes.

   The Resolve()-protocol of *disclosing* schemes may leak security critical information from $proof_{A_i}$. Hence, the rightful author has to additionally trust that D does not abuse this information. If this additional trust in D cannot be established, the scheme does not guarantee any further protection of the work, once a dispute has been resolved. To our knowledge, all watermarking-based dispute resolving schemes proposed so far (e.g., [6, 7, 9, 4]) use symmetric watermarking schemes and are disclosing: For D being able to detect the presence of the claimant's watermark in the disputed work, the claimant has to reveal $K_{A_i}^{WM}$ and the watermark WM_{A_i} to D. Some schemes even give the original work W_i to D. Thus, D may easily produce versions of the disputed work, which does not contain WM_{A_i} anymore and for which the rightful author cannot prove authorship anymore, because the watermarking-based proof of similarity fails (see Section 3.2 item 2). In contrast to disclosing schemes, *non-disclosing* schemes assume D only to be *correct*, i.e., to resolve disputes according to the protocol. In this case, the following additional requirement must be fulfilled:

   - ***Non-disclosing* Resolve()*:*** A run of $\mathsf{Resolve}(W_{dis}, \bullet)_t$ must not disclose "critical" information, which allows an attacker to make runs $\mathsf{Resolve}(W_A, \bullet)_{t'}$ at time $t' > t$, in which D behaves correct, end with output $result_D = \mathsf{failed}$ for works W_A with $A \sim^t W_A$.

   An analog requirement is possible for Prepare().

In this context, recent research directions like asymmetric watermarks[17] and zero-knowledge watermark detection[12] offer a significant advantage over the use of traditional symmetric watermarks: In an asymmetric watermarking scheme, the information necessary for watermark detection/extraction does not jeopardize the robustness of the watermark. Zero-knowledge watermark detection proves the presence of a symmetric watermark without leaking any new knowledge to the verifier (dispute resolver) at all. Therefore, in principle, both techniques can be helpful in achieving non-disclosing dispute resolving.

However, asymmetric watermarking schemes are not as suitable for proving similarity between works as non-blind symmetric watermarking schemes (see item 2 in Section 3.2). Therefore, in Section 4.1, we pursue the approach based on zero-knowledge watermark detection for non-blind symmetric watermarking schemes.

2. **Soundness of Resolve():** The basic requirements of Definition 3.2 only guarantee that a dispute is resolved in favor of the rightful author, if he *is one of the participating disputants*. Thus, such schemes are only of limited practical use: the disputant who lost the dispute may decline paying compensation to the winner by claiming that the winner himself is not the real author but a cheater as well (we call this problem the *"conclusiveness problem"*). In Section 4.2, we investigate an advanced dispute resolving scheme, which fulfills the following additional *soundness requirement*:

   - ***Soundness of* Resolve():** If a run $\mathsf{Resolve}(W_{dis}, \bullet)_t$ ends with output $result_D = A_i$ then $A_i \sim^t W_{dis}$ holds, i.e., any dispute, which is resolved, is resolved in favor of the rightful author.

4. ADVANCED DISPUTE RESOLVING SCHEMES

In this section, we investigate advanced schemes for dispute resolving: in Section 4.1 we introduce a non-disclosing dispute resolving scheme and in Section 4.2 we introduce a sound dispute resolving scheme. For readability, we make following assumptions and simplifications in the presentation of our protocols: Firstly, we assume secure reliable channels between all participants. Secondly, we omit details of the message formats. In particular, we assume that techniques of robust protocol design like protocol- and message-type tags are used.

4.1. Non-Disclosing Dispute Resolving

We reviewed various means and techniques of proving the different conditions for authorship in dispute situations in Section 3.2. In this section we focus on the applicability of zero-knowledge watermark detection for achieving *non-disclosing* dispute resolution. Non-blind zero-knowledge watermark detection is used for proving the similarity of the disputed work W_{dis} to the rightful author's original work. For proving the order of creation we use a time-stamping service (see discussion in Section 3.2).

Let $(\mathsf{Commit}(), \mathsf{Open}())$ be a commitment scheme as defined in Definition 2.2, $(\mathsf{GenKey}, \mathsf{Embed}, \mathsf{Detect})$ be a robust symmetric non-blind watermarking scheme, and $\mathsf{ZKDetect}()$ be a corresponding zero-knowledge watermark detection protocol. Furthermore, let T be a linking time-stamping service as introduced in [16]. We define our non-disclosing dispute resolving scheme as follows:

1. The **Initialize()**-protocol initializes the time-stamping authority T and authentically distributes the public-key of T. Furthermore, D generates and authentically distributes the public parameters par_{com} of the commitment scheme as described in [14].
2. In **Prepare()**, the rightful author A selects a random watermark $WM_{A,W}$, a random watermarking key $K^{WM}_{A,W}$ and computes commitments $com(W) := \mathsf{Commit}(W, par_{com})$, $com(WM_{A,W}) := \mathsf{Commit}(WM_{A,W}, par_{com})$ and $com(K^{WM}_{A,W}) := \mathsf{Commit}(K^{WM}_{A,W}, par_{com})$. Then, A generates the marked version $W' := \mathsf{Embed}(W, WM_{A,W}, K^{WM}_{A,W})$, and requests a timestamp $TS_{W,A}$ for $(com(W), com(WM_{A,W}), com(WM_{A,W}), W')$ from the time-stamping authority T. The output of $\mathsf{Prepare}()$ is $(W', proof_{A,W}, \mathsf{true})$, where $proof_{A,W} = (com(W), com(WM_{A,W}), com(WM_{A,W}), TS_{W,A})$.
3. The **Resolve()**-protocol starts with both disputants sending their commitments $(com(W_i), com(WM_{A_i,W_i}), com(K^{WM}_{A_i,W_i})$ and their time-stamps TS_{W_i,A_i} corresponding to W_{dis} to D. Then, D runs a sub-protocol $\mathsf{ZKDetect}(W_{dis}, com(W_i), com(WM_{A_i,W_i}), com(K^{WM}_{A_i,W_i}))$ with each disputant A_i, where A_i is the prover and D is the verifier. Depending on the boolean-valued outputs B_1, B_2 of these sub-protocols, D resolves the dispute as follows: if both sub-protocols ended with false, i.e., $W_i \not\rightarrow_{sim} W_{dis}$, then neither A_1, nor A_2

is the author of W_{dis} and Resolve() ends with $result_D$ = failed. If one of the sub-protocols ended with true, W_{dis} is similar to exactly one of the committed originals and the dispute is resolved in favor of the corresponding disputant. In case W_{dis} is similar to both alleged originals, i.e., both sub-protocols ended with true, the time-stamps are compared and the dispute is resolved in favor of the disputant with the older time-stamp. The Resolve()-protocol is summarized in Figure 2.

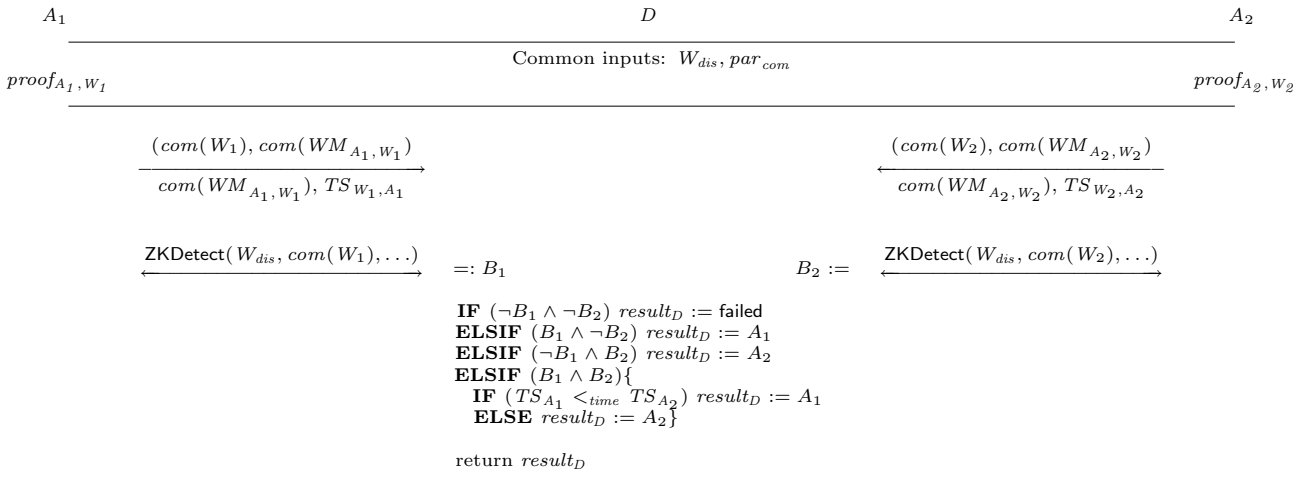

Figure 2. The non-disclosing Resolve()-protocol based on zero-knowledge watermark detection.

THEOREM 4.1 (NON-DISCLOSING DISPUTE RESOLVING SCHEME). *The scheme* (Initialize(), Prepare(), Resolve()), *as defined above, is a* non-disclosing *dispute resolving scheme.*

Proof. [Sketch] *Completeness of* Prepare() follows trivially by the properties of the underlying watermarking scheme and the correctness of the time-stamping service T. *Completeness of* Resolve() holds, since the ZKDetect()-protocols are a proof of similarity and the order of creation of the alleged originals can be determined using the time-stamps. *Non-disclosing property* of Resolve() follows from the hiding-property of the commitment scheme and the zero-knowledge property of the zero-knowledge watermark detection protocol. □

4.2. Sound Dispute Resolving

In this Section, we propose a general construction to make arbitrary dispute resolving schemes *sound*. The basic idea of our construction is to resolve authorship disputes in a two-step process: In the first step, we run a protocol, which determines the set of *all potential authors* (*PA*) of the disputed work and guarantees that *the rightful author is among the identified potential authors.* In the second step, we run a multi-party dispute resolving scheme (*without soundness*) to resolve an "extended dispute" *between all potential authors, which have been identified in the first step.* As the real author is guaranteed to be one of the disputants in the extended dispute, soundness of the overall two-step protocol follows directly from the completeness of Resolve().

When using the *non-disclosing* scheme, as presented in Section 4.1, for the second step, the overall two-step scheme is *sound* and *non-disclosing*. The general open question is how to achieve the first step, i.e., identification of all potential authors of the disputed work. In the context of direct proofs of authorship,[11] this was basically achieved by means of a registration center. In principle, this approach would work for dispute resolving as well.

Here, we propose another way based on a special class of robust asymmetric watermarking schemes, which has what we call the *"multiple secret-key property"*. Informally speaking, this property guarantees that given a public detection/extraction-key anyone can efficiently compute a random matching secret embedding-key, such that watermarks embedded using such an embedding-key can be detected/extracted with the given public-key, but cannot be removed using the public-key or other matching secret-keys. Using such watermarking schemes, identification of all potential authors can be achieved as follows: The dispute resolver generates a public detection key and the rightful author embeds his identity in his work, using a newly generated secret-key matching D's

public-key. Using his public extraction key, D can extract the identities of all possible authors from a disputed work. The properties of the watermarking scheme guarantee that the author's identity will be one of the extracted identities. The multiple secret-key property is formalized in the following definition:

DEFINITION 4.2 (MULTIPLE SECRET-KEY PROPERTY). *A robust asymmetric watermarking scheme* (GenKey, Embed, Detect/[Extract]) *fulfills the* ***multiple secret-key*** *property* iff *the following holds*

1. *there exists a probabilistic polynomial-time algorithm* GenMatchingSK(), *which, on input of a public detection/extracting key* $K^{Det/Ext}$, *outputs a random embedding key* K^{Emb}, *which matches* $K^{Det/Ext}$ *in the sense of Definition 2.1 and*
2. (GenMatchingSK, Embed, Detect/[Extract]) *is a robust asymmetric watermarking scheme, i.e., using random matching secret-keys produced by* GenMatchingSK *does not reduce the robustness.*
3. *Let* $\mathcal{WM}_{emb}$ *be the set of all watermarks, which have been embedded into* W', *using a secret-key generated by* GenMatchingSK *for a given public detection/extraction key* $K^{Det/Ext}$. *Then, for detecting schemes* Detect(W', $K^{Det/Ext}$, WM) = true *holds for all* $WM \in \mathcal{WM}_{emb}$ *and for extracting schemes* Extract(W', $K^{Det/Ext}$) = $\mathcal{WM}_{emb}$ *holds.*

The asymmetric detecting watermarking scheme proposed by Furon and Duhamel[17] provides the multiple secret-key property, but its robustness has been shown to be limited. Thus, this property is not completely unreasonable. However, we stress that we are currently not aware of other watermarking schemes, which possess this property. Achieving this property for extracting schemes may be even harder than for detecting ones. In the following we assume the existence of an extracting asymmetric watermarking scheme (GenKey, Embed, Extract), which provides the multiple secret-key property.

Given a multi-party dispute resolving scheme (Initialize(), Prepare(), Resolve()), which fulfills the basic requirements as given in Definition 3.2, we define our sound dispute resolving scheme ($\text{Initialize}_\text{s}()$, $\text{Prepare}_\text{s}()$, $\text{Resolve}_\text{s}()$) as follows:

1. **$\text{Initialize}_\text{s}()$** first performs the same actions as Initialize(). Additionally, the dispute resolver D generates a random key-pair $(K_D^{Emb}, K_D^{Det/Ext}) \xleftarrow{R}$ GenKey and distributes $K_D^{Det/Ext}$ authentically.

2. **$\text{Prepare}_\text{s}()$:** First, the author A performs $K_{A,W}^{Emb} \xleftarrow{R}$ GenMatchingSK($K_D^{Det/Ext}$) to generate a new secret embedding key, which matches D's public key $K_D^{Det/Ext}$. Using this secret key, A embeds his identity into his original W, yielding $W' :=$ Embed($W, A, K_{A,W}^{Emb}$). Then, A performs the preparation algorithm/protocol of the multi-party dispute resolving scheme for W': (W'', $proof_{A,W'}$, $result_A$) := Prepare(W'). If $result_A$ = true, $\text{Prepare}_\text{s}()$ returns ($W'', proof_{A,W}$, true) to A, where $proof_{A,W} := (proof_{A,W'}, K_{A,W}^{Emb})$. Otherwise, if Prepare($W'$) returned $result_A$ = false, $\text{Prepare}_\text{s}()$ ends with output false to A.

3. **$\text{Resolve}_\text{s}()$** The sound resolve protocol $\text{Resolve}_\text{s}()$ works as informally described above. Given the disputed work W_{dis}, D extracts the identities of all possible authors as $PA :=$ Extract($W_{dis}, K_D^{Det/Ext}$). Then, the "extended dispute" between all authors in PA is resolved, using the multi-party resolving protocol $result_D :=$ Resolve(W_{dis}, PA, $proof_{A_1}$, $proof_{A_2}$, ...) and $\text{Resolve}_\text{s}()$ outputs this value $result_D$.

THEOREM 4.3 (SOUND DISPUTE RESOLVING). *Let* (Initialize(), Prepare(), Resolve()) *be a dispute resolving scheme. Then, the dispute resolving scheme* ($\text{Initialize}_\text{s}()$, $\text{Prepare}_\text{s}()$, $\text{Resolve}_\text{s}()$), *as defined above, is a* sound *dispute resolving scheme.*

Proof. [Sketch] Completeness of $\text{Prepare}_\text{s}()$ follows trivially by the multiple secret-key property of (GenKey, Embed, Extract) and the completeness of Prepare(). Completeness of $\text{Resolve}_\text{s}()$ follows directly by the fact that Prepare() is executed as a sub-protocol of $\text{Prepare}_\text{s}()$ and that Resolve() is complete.

For proving soundness of $\text{Resolve}_\text{s}()$, assume that a run $\text{Resolve}_\text{s}(W_{dis}, \bullet)_t$ ended with output $result_D = A_i$ and that $\boldsymbol{A_i \not\sim^t W_{dis}}$. By construction of $\text{Resolve}_\text{s}()$ the run Resolve($W_{dis}, PA, \ldots$) must have ended with $result_D = A_i$. By the multiple secret-key property we can follow that $A \in PA$, i.e., the rightful author A has been identified as a possible author. Therefore, A participated in the run of the sub-protocol Resolve($W_{dis}, PA, \bullet$). This is a contradiction to the completeness of Resolve(), since it requires this run of Resolve() to end with output $result_D = A$. This completes our soundness argument. □

Note that, as an alternative to resolving the extended dispute, soundness could be also achieved by aborting the resolve protocol with output $result_D = \mathsf{failed}$ if there are other potential authors than the two disputants A_1 and A_2. However, this would enable trivial "denial of service attacks": an adversary could prevent disputes from being resolved by embedding an additional arbitrary identity and using the resulting work as the disputed work. This would violate the completeness requirement of Resolve().

5. CONCLUSION

In this paper we laid the ground for a formal treatment of authorship-dispute resolving. We formally defined dispute resolving schemes and their basic requirements, using a formal definition of authorship. Based on the formal definition of authorship, we discussed which authorship-criteria the rightful author has to prove in disputes as well as possible ways of proving them. Furthermore, we introduced essential optional requirements ("soundness" and "non-disclosure") of dispute resolving schemes, which are important for the practicality of such schemes. Finally, we proposed advanced dispute resolving schemes and proved that they fulfill these requirements. To our knowledge, our schemes are the first, which fulfill the basic requirements as well as the soundness and non-disclosure requirements.

REFERENCES

1. E. Koch and J. Zhao, "Towards robust and hidden image copyright labeling," in *Proceedings of IEEE Workshop on Nonlinear Signal and Image Processing*, pp. 452–455, 1995.
2. N. Nikolaidis and I. Pitas, "Copyright protection of images using robust digital signatures," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-96)*, **4**, pp. 2168–2171, May 1996.
3. I. Pitas and G. Voyatzis, "Applications of toral automorphisms in image watermarking." IEEE Signal Processing Society, 1996.
4. S. Craver, N. Memon, B.-L. Yeo, and M. M. Yeung, "Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications," *IEEE Journal on Selected Areas in Communications* **16**, pp. 573–586, May 1998.
5. M. Ramkumar and A. Akansu, "Image watermarks and counterfeit attacks : Some problems and solutions," in *Content Security and Data Hiding in Digital Media*, 1999.
6. W. Zeng and B. Liu, "On resolving rightful ownerships of digital images by invisible watermarks," in *4th International Conference on Image Processing (ICIP)*, pp. 552–555, IEEE, (Santa Barbara, CA, USA), Oct. 1997.
7. A. Perrig, A. Herrigel, and J. Ruanaidh, "A copyright protection environment for digital images," in *Verlässliche IT-Systeme, GI-Fachtagung VIS '97, DuD Fachbeiträge*, pp. 1–16, Vieweg, 1997.
8. A. Herrigel, J. Ó. Ruanaidh, H. Petersen, S. Pereira, and T. Pun, "Secure copyright protection techniques for digital images," in *Information Hiding—Second International Workshop, IH'98*, D. Aucsmith, ed., *Lecture Notes in Computer Science* **1525**, Springer-Verlag, Berlin Germany, (Portland Oregon, USA), Apr. 1998.
9. L. Qiao and K. Nahrstedt, "Watermarking methods for MPEG encoded video: Towards resolving rightful ownership," in *International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 276–285, IEEE, IEEE, Washington Brussels Tokyo, (Austin, Texas, USA), 1998.
10. M. Kutter and F. Leprevost, "Symbiose von Kryptographie und digitalen Wasserzeichen: Effizienter Schutz des Urheberrechtes digitaler Medien," in *Tagungsband des 6. Deutschen IT-Sicherheitskongress, Bundesamt für Sicherheit in der Informationstechnik*, pp. 1–4, May 1999.
11. A. Adelsbach, B. Pfitzmann, and A.-R. Sadeghi, "Proving ownership of digital content," in Pfitzmann,[18] pp. 126–141.
12. A. Adelsbach and A.-R. Sadeghi, "Zero-knowledge watermark detection and proof of ownership," in *Information Hiding—4th International Workshop, IHW 2001*, I. S. Moskowitz, ed., *Lecture Notes in Computer Science* **2137**, pp. 273–288, Springer-Verlag, Berlin Germany, (Pittsburgh, PA, USA), 2001.
13. O. Goldreich, *Foundations of Cryptography*, vol. Basic Tools, Cambridge University Press, 2001.
14. E. Fujisaki and E. Okamoto, "Statistical zero knowledge protocols to prove modular polynomial relations," in *Advances in Cryptology – CRYPTO '97*, B. S. Kaliski, Jr., ed., *Lecture Notes in Computer Science* **1294**, pp. 16–30, International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1997.

15. R. B. Wolfgang and E. J. Delp, "Overview of image security techniques with applications in multimedia systems," in *Proceedings of the SPIE International Conference on Voice, Video, and Data Communications*, pp. 297–308, 1997.
16. S. Haber and W. S. Stornetta, "How to time-stamp a digital document," *Journal of Cryptology* **3**, 1991.
17. T. Furon and P. Duhamel, "An asymmetric public detection watermarking technique," in Pfitzmann,[18] pp. 88–100.
18. A. Pfitzmann, ed., *Information Hiding—3rd International Workshop, IH'99*, *Lecture Notes in Computer Science* **1768**, (Dresden, Germany), Springer-Verlag, Berlin Germany, Oct. 2000.