Zero-Knowledge Watermark Detection and Proof of Ownership

André Adelsbach[*] and Ahmad-Reza Sadeghi

Universität des Saarlandes, FR 6.2 Informatik
D-66123 Saarbrücken, Germany
{adelsbach,sadeghi}@cs.uni-sb.de

Abstract. The goal of zero-knowledge watermark detection is to allow a prover to soundly convince a verifier of the presence of a watermark in certain stego-data without revealing any information which the verifier can use to remove the watermark. Existing proposals do not achieve this goal in terms of definition (not formally zero-knowledge), security (unproven) and coverage (handle only blind watermark detection).

In this paper we define zero-knowledge watermark detection precisely. We then propose efficient and provably secure zero-knowledge protocols for blind and non-blind versions of a well-known class of watermarking schemes. Using these protocols the security and efficiency of many watermark based applications can be significantly improved.

As an example of use we propose concrete protocols for direct proof of ownership which enable offline ownership proofs, i.e., copyright holders can prove their rightful ownership to anyone without involving a trusted third party in the actual proof protocol.

Keywords: Zero-Knowledge Watermark Detection, Ownership Proofs

1 Introduction

Protection of digital works against misuse and illegal distribution has become a challenging task in the information society and there has been intensive research in this area in the last years. As the total prevention of misuse does not seem to be achievable at reasonable cost, most technical copyright protection schemes aim to deter illegal usage or redistribution of digital content by making misuse detectable. For this, identifying information is imperceptibly embedded into the original work by means of watermarking techniques, e.g., [8,24,20,19]. This information can be used later as evidence to identify the owner of the digital work [23,9,20,24,19] or the source of its illegal redistribution (*fingerprinting schemes*, [4,22]). However, a conceptual problem of these schemes is that showing the presence of the watermark as evidence discloses sensitive information which can be used to remove the watermark. Thus it is desirable to convince a verifier of the presence of a watermark without revealing any information helping the

[*] This author is supported by the European IST project MAFTIA (IST-1999-11583).

I. S. Moskowitz (Ed.): IH 2001, LNCS 2137, pp. 273–288, 2001.

verifier to remove the watermark. There are two approaches trying to tackle this problem:

One possible approach are *asymmetric watermarking schemes* [14,12,11]. Here a secret/public key pair is generated and used to embed/detect the watermark. However, asymmetric watermarking schemes have the conceptual drawback that the public detection key makes oracle/sensitivity attacks [10,21] even more serious, since an attacker who knows the public detection key can carry out this attack on his own, i.e., without any interaction with the holder of the secret key.

Another approach is to use zero-knowledge proof protocols [16]. In such protocols a prover convinces a verifier that she knows a secret or that a value has a certain property and the verifier learns nothing new from a protocol-run about the secret inputs of the prover. Zero-knowledge proof protocols are powerful tools and are applied as building blocks in many cryptographic applications. Recently, they also have been applied in the context of *blind watermark detection*[1] [7,18].

In the most secure protocol of [7] a prover constructs a *legal watermark* to be embedded into given cover-data, i.e., a watermark for which the prover knows a secret, e.g., a hard to compute pre-image.[2] For the zero-knowledge detection, she hides the legal watermark in a long list of fake watermarks[3] and lets the verifier detect them all in the stego-data. Then she proves that at least one of the watermarks in the list is a legal one without disclosing which one. The security of this scheme is strongly based on the fact that the number of watermarks in the list must be so large that they could not be removed all without severely degrading the underlying stego-data. Furthermore, besides the fact that the published list reveals information about the watermark it is not clear why a cheating prover cannot generate fake watermarks from which she knows the discrete logarithms.

A further proposal based on a cryptographic protocol is discussed in [18] and called *watermarking decision problem*: Given certain stego-data, decide whether an RSA encrypted watermark is present in this stego-data. The authors propose a protocol for solving this problem for the blind version of the well-known watermarking scheme from Cox et al. [8]. The basic idea is to secretly and verifiably compute the correlation between the watermark and the underlying stego-data. For this, the prover sends to the verifier an RSA-encryption of the watermark and an RSA-encryption of the blinded stego-data. In a challenge-response manner the prover should convince the verifier that the watermark correlates with the stego-data. However, no proof of soundness is given and it is not really zero-knowledge since the verifier obtains a good estimation of the correlation value enabling oracle attacks [10].

In this paper we first give a formal definition of zero-knowledge watermark detection protocols based on the definitions known from cryptography. We propose

[1] Blind watermarking schemes do not require the original cover-data as an input for the detection process.

[2] Craver proposes the discrete logarithm of the embedded watermark.

[3] This is achieved by *invertibility attacks* introduced in [9].

provably secure zero-knowledge detection protocols for a blind and a non-blind version of a well-known class of watermarking schemes as introduced in [8]. The definition of zero-knowledge watermark detection and the corresponding protocols are the subject of the Sections 3 and 4.

Based on these protocols and the model of [1] we propose protocols for *proof of ownership* where participation of the registration center is not required in the actual ownership proof. The concept of direct proof of ownership has been introduced and formally considered for the first time in [1] to overcome the following shortcomings of the existing watermark-based solutions for identifying the rightful owner of digital works [9,20,23,19]: First, the common watermark-based schemes focus only on resolving ownership disputes between two disputants, each claiming to be the rightful owner of a certain work. However, in real-life electronic market places, buyers want to directly ensure that they are purchasing digital items from the real copyright holder. Second, resolving ownership disputes in favor of a party does not necessarily mean at all that this party is the rightful owner. This is because the real copyright holder may not even know about a dispute taking place on her digital work and thus may not be able to show the presence of her identifying information (watermark) in the work. Note that the judge can not notice the presence of watermarks without knowing the corresponding detection key. The protocols for proof of ownership are presented in Section 5.

We start our discussion by introducing some required building blocks.

2 Cryptographic Building Blocks

2.1 Commitment Schemes

A *commitment scheme* $(com, open)$ for the message space M and commitment space C consists of a two-party protocol com 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 $com(m, par_{com})$ where par_{com} stands for all public parameters needed to compute the commitment value. To open a commitment com the committer runs the protocol $open(com, par_{com}, sk_{com})$ 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 $com()$ and $open()$. Furthermore, we use $com()$ and $open()$ on tuples over M, with the meaning of component-wise application of $com()$ 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.

Furthermore we require that the commitment scheme has following *homomorphic property*: Let $com(m_1)$ and $com(m_2)$ be commitments to arbitrary messages $m_1, m_2 \in M$. Then the committer can open $com(m_1)*com(m_2)$ to m_1+m_2

without revealing additional information about the contents of $com(m_1)$ and $com(m_2)$.

In the following we use a commitment scheme of [15]: Let n be a product of two safe primes p and q, let g and h be two generators of the cyclic subgroup G of $\mathbb{Z}_n^*$ with order $\frac{p-1}{2}\frac{q-1}{2}$ and let $par_{com} = (n, g, h)$. Furthermore the factorization of n and the discrete logarithms $\log_g(h)$ and $\log_h(g)$ must be unknown to the committer. The committer commits to a value $m \in M = \{0, \cdots n-1\}$ by computing $com(m, par_{com}) := g^m h^r \bmod n$ where $sk_{com} = r$ is a randomly selected natural number from $[0, 2^l n)$ and l is in the order of the bit-length of n. This scheme is statistically hiding and computationally binding under the factoring assumption.

2.2 Proving Relations for Committed Numbers

To ensure the correctness of the committed values used in our protocols, we need to prove that certain relations hold for committed numbers, i.e., a committed number lies in an interval or a committed number is the product of two other committed numbers. In [5] efficient protocols are described for proving in zero-knowledge that a committed number lies in an exact interval.

In [6] efficient and secure techniques for proving relations in modular arithmetic (addition, multiplication, exponentiation) between committed[4] numbers in zero-knowledge are proposed: Given commitments to the values $a, b, c, m \in M$ one can prove that $a + b \equiv c \bmod m$, $a * b \equiv c \bmod m$ or $a^b \equiv c \bmod m$. These protocols are statistically zero-knowledge in the general model.

Remark 1.1. The protocols for proving the relations mentioned above are interactive in general. Using Fiat-Shamir heuristics [13] these protocols can be made non-interactive, however with the limitation that the zero-knowledge property can only be proven in the random oracle model [3]. ○

3 Definitions and Notations

In this section, we first introduce our basic definitions and notations of watermarking schemes. Following this, we give a formal definition of *zero-knowledge watermark detection* and discuss some important issues.

3.1 Watermarking Schemes

Watermarking is a very lively area of research, with an exploding variety of different schemes. The following definitions do not aim at providing a complete framework which fits all known watermarking schemes. We rather introduce the basic notations, which are needed in the following sections.

[4] Although not mentioned explicitly in [6], these protocols work also for the commitments from [15] (private communications with Jan Camenisch).

A *watermarking scheme with detection* consists of four polynomial-time algorithms $\mathit{GEN\_KEY}$, $\mathit{GEN\_WM}$, EMBED, and DETECT. $\mathit{GEN\_KEY}$ and $\mathit{GEN\_WM}$ are probabilistic and generate a key-pair (k_{emb}, k_{det}) resp. a watermark WM. The algorithm $\mathit{EMBED}(W, \mathit{WM}, k_{emb})$ imperceptibly embeds the watermark WM into the cover-data W, using the key k_{emb}. This results in stego-data W' (watermarked version of W). The algorithm $\mathit{DETECT}(W'', \mathit{WM}, W, k_{det})$ returns a boolean value, which states whether the data W'' contains the watermark WM relative to the reference data W, using key k_{det}.

A *symmetric* watermarking scheme needs the same key k_{wm} for detection as for embedding. *Unkeyed* watermarking schemes need no key for embedding or detection. Watermarking schemes whose DETECT algorithms do not require the input of reference data W are called *blind*, in contrast to *non-blind* schemes.

3.2 Definition of Zero-Knowledge Watermark Detection

To motivate the need for zero-knowledge watermark detection, we return to a well known application of robust watermarking schemes, namely, resolving ownership disputes on digital works [9,20,19,23]. In this context, the presence of a party's watermark in the disputed work is an indication for the rightfulness of that party's ownership claim.

All these proposals suffer under one problem of the watermark detection process: the disputing parties have to disclose information, which is necessary to detect the watermark, to the dispute-resolver. However, this information is in most cases also sufficient to remove the watermark from the disputed data.

This problem is not symptomatic for dispute resolving only, but a common problem of all applications where the presence of a watermark has to be verified by a not fully trusted party, i.e., also some fingerprinting schemes.

Zero-knowledge watermark detection eliminates this security risk, because it enables a prover to prove to an untrusted verifier that a certain watermark is present in stego-data *without revealing any information about the watermark, the reference data and the detection key.* We now give a formal definition of zero-knowledge watermark detection.

Definition 1 (Zero-Knowledge Watermark Detection).
Let $(com, open)$ *be a secure commitment scheme. A* zero-knowledge watermark detection protocol $\mathit{ZK\_DETECT}$ *for the watermarking scheme* $(\mathit{GEN\_KEY}, \mathit{GEN\_WM}, \mathit{EMBED}, \mathit{DETECT})$ *is a* zero-knowledge proof of knowledge protocol *[17,2] 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'', $com(\mathit{WM})$, $com(W)$, $com(k_{wm})$, *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.*[5] *The private input of the prover is the secret opening information of these commitments* $sk_{com} = (sk_{com}^{WM}, sk_{com}^{W}, sk_{com}^{k_{wm}})$.

[5] One can relax this by allowing $\mathit{Transf}(W'')$ and $com(\mathit{Transf}(W))$ be input instead for certain transformations Transf, e.g., the discrete cosine transformation. We will make use of this convention in later sections.

$\mathcal{P}$ proves knowledge of a tuple $(WM, W, k_{wm}, sk^{WM}_{com}, sk^{W}_{com}, sk^{k_{wm}}_{com})$ such that:

$$\begin{aligned}[(open(com(WM), par^{WM}_{com}, sk^{WM}_{com}) = WM) \wedge \\ (open(com(W), par^{W}_{com}, sk^{W}_{com}) = W) \wedge \\ (open(com(k_{wm}), par^{k_{wm}}_{com}, sk^{k_{wm}}_{com}) = k_{wm}) \wedge \\ DETECT(W'', WM, W, k_{wm})] = true\end{aligned}$$

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

Remark 1.2. The input of $(com(W), par^{W}_{com}, sk^{W}_{com})$ and $(com(k_{wm}), par^{k_{wm}}_{com}, sk^{k_{wm}}_{com})$ is optional, depending on whether the watermarking scheme is blind/non-blind, resp. keyed/unkeyed.

Remark 1.3. One can simply adapt the previous definition to a zero-knowledge proof for showing that a watermark is *not detectable* in data W''. This may be useful in applications where one has to show that a certain watermark is *not* present. Our protocols in Section 4 can be easily adapted to this kind of protocol too.

Remark 1.4. When using a zero-knowledge watermark detection protocol one must take care that the parameters for the commitments are chosen correctly. This can be achieved by running the setup-phase of the commitment scheme between $\mathcal{P}$ and $\mathcal{V}$ or by letting a trusted party choose these parameters.

Remark 1.5. Many applications using watermark detection require that certain properties of the watermark are verifiable by the party which detects the watermark. When using zero-knowledge detection, these verifications have to be carried out on the committed watermark. This may be achieved either by additionally running appropriate zero-knowledge proof protocols or by an appropriate certification by a trusted party (see Section 5.2 for an example of the latter.) ○

4 Zero-Knowledge Watermark Detection

Before presenting our *blind* and *non-blind* zero-knowledge detection protocols, we give an overview of the underlying watermarking scheme.

4.1 Watermarking Scheme of Cox et al.

The robust watermarking scheme of Cox et al. is unkeyed in its basic form and thus quite simple, since it does not involve a pseudorandom selection of the coefficients used for embedding/detecting the watermark. It is based on the spread spectrum principle and has been described originally in terms of image-data, although being a whole methodology of watermarking schemes. Following Cox et al., we also restrict the following discussions to image-data. Using suitable transformations, this technique is applicable to other types of data too and so are our zero-knowledge detection protocols.[6]

[6] They can be easily modified to work on other data-types as well by replacing the *DCT* transformation by any kind of suitable pre-computation.

The watermark generation algorithm: GEN_WM generates watermarks $WM = (WM_1, \ldots, WM_k)$ that are sequences of real numbers, each chosen independently according to a certain probability distribution, e.g., a $N(0,1)$ normal distribution with mean 0 and variance 1. Its length k influences to which degree the watermark is spread over the stego-data and how large the modifications for embedding the watermark have to be.
The embedding algorithm: A given watermark WM is embedded by modifying the k highest magnitude DCT AC coefficients $DCT(W,k) = (DCT(W,k)_1, \ldots, DCT(W,k)_k)$. Cox et al. proposed several formulas for embedding, e.g.,

$$DCT(W',k)_i := DCT(W,k)_i * (1 + \alpha * WM_i).$$

Here, the value α denotes a scaling parameter and its choice may depend on the cover-data, offering a tradeoff between the robustness and the non-perceptibility of the watermark in the stego-data.
The detection algorithm: The detection algorithm does not necessarily need the original data W as an input. However, it may be used in the detection process to improve the robustness and reduce the number/probability of false positives (see Section 6).

Detection works by computing a correlation value. As a measure of confidence in the presence of WM in W'' (relative to W in case of non-blind detection), the detector tests whether $corr \geq \delta$ holds for a predefined *detection-threshold* δ.

In blind detection the correlation value

$$corr = \frac{< DCT(W'',k), WM >}{\sqrt{< DCT(W'',k), DCT(W'',k) >}} \tag{1}$$

between the watermark WM and the DCT-coefficients $DCT(W'',k)$ is used. In non-blind detection the correlation value

$$corr = \frac{< \Delta, WM >}{\sqrt{< \Delta, \Delta >}} \tag{2}$$

between the watermark WM and $\Delta = DCT(W'',k) - DCT(W,k)$ is used. Here $< x, y >$ denotes the scalar product of the two vectors x and y.

Zero-knowledge watermark detection for non-blind watermarking schemes seems to be inherently more difficult than for blind ones. The reason for this is that the reference data W is additionally needed for the detection-relevant computation without being disclosed to the verifier. On the other hand, non-blind detection is more robust and its zero-knowledge version is quite elegantly applicable for *offline* proof of ownership, as shown in Section 5.2.

Before going into the details of our zero-knowledge watermark detection protocols, we have to discuss some technicalities first.

4.2 Some Technicalities

In contrast to Cox et al., we assume that the watermark and DCT-coefficients are *integers* and not real numbers.[7]

The parameters par_{com} of the commitment scheme must be chosen sufficiently large so that the resulting values do not exceed the order of the commitment base g when doing computations with the committed values.[8]

For efficiency reasons we do not use the correlation values as computed in Formula 1 and 2 for detection. We use the equivalent detection criteria

$$C := (\underbrace{< DCT(W'', k), WM >}_{A})^2 - \underbrace{< DCT(W'', k), DCT(W'', k) > * \delta^2}_{B} \geq 0 \tag{3}$$

in the blind case and

$$F := (\underbrace{< \Delta, WM >}_{D})^2 - \underbrace{< \Delta, \Delta > * \delta^2}_{E} \geq 0 \tag{4}$$

in the non-blind case instead.

4.3 Blind Zero-Knowledge Detection Protocol

Let par_{com}, W'', $com(WM)$, δ be the common inputs of $\mathcal{P}$ and $\mathcal{V}$ and let sk_{com} be the private input of $\mathcal{P}$. In the zero-knowledge version of the blind detection algorithm a prover $\mathcal{P}$ proves to a verifier $\mathcal{V}$ that the watermark contained in $com(WM)$ is present in the image W'', without revealing any information about WM. The blind zero-knowledge detection protocol $ZK\_DETECT(W'', WM, -, -)$ is shown in Figure 1.

$\mathcal{P}$ and $\mathcal{V}$ compute the DCT of W'', especially $DCT(W'', k)$. Then $\mathcal{P}$ and $\mathcal{V}$ can compute the value B from Equation 3, $\mathcal{P}$ sends a commitment $com(B)$ to $\mathcal{V}$ and opens it immediately to $\mathcal{V}$. $\mathcal{V}$ verifies that the opened commitment contains the same value B which he computed himself. Now $\mathcal{V}$ computes the commitment

$$com(A) := \prod_{i=1}^{k} com(WM_i)^{DCT(W'',k)_i} \bmod n, [9]$$

taking advantage of the homomorphic property of the commitment scheme. $\mathcal{P}$ computes the value A^2, sends a commitment $com(A^2)$ to $\mathcal{V}$ and gives $\mathcal{V}$ a zero-knowledge proof that it really contains the square of the value contained in $com(A)$. We refer to this sub-protocol as **ZKP**$(com(A^2)$ contains $A^2)$ (see [6]).

[7] Note that this is no real constraint, because we can scale the real valued coefficients appropriately.

[8] Alternatively, we may choose smaller parameters and prove for each operation in zero-knowledge that no over overflow occurred.

[9] Note that the modulus n is contained in the public commitment parameters par_{com}.

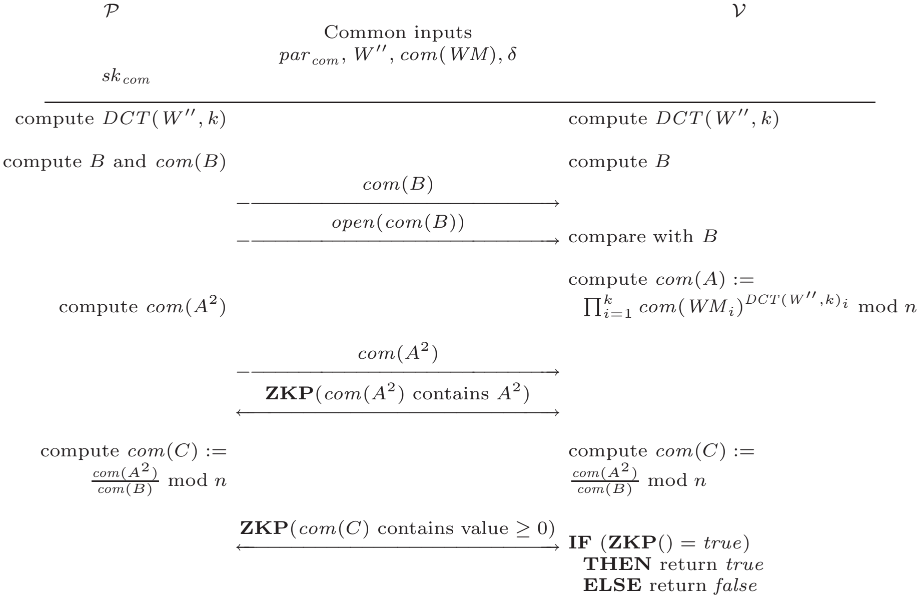
$\mathcal{P}$
$\mathcal{V}$
Common inputs
par_{com}, W'', $com(WM)$, δ
sk_{com}
compute $DCT(W'', k)$
compute $DCT(W'', k)$
compute B and $com(B)$
compute B
$com(B)$
$open(com(B))$
compare with B
compute $com(A) := \prod_{i=1}^{k} com(WM_i)^{DCT(W'',k)_i} \bmod n$
compute $com(A^2)$
$com(A^2)$
ZKP$(com(A^2)$ contains $A^2)$
compute $com(C) := \frac{com(A^2)}{com(B)} \bmod n$
compute $com(C) := \frac{com(A^2)}{com(B)} \bmod n$
ZKP$(com(C)$ contains value $\geq 0)$
IF (**ZKP**$() = true$)
THEN return *true*
ELSE return *false*

Fig. 1. The blind zero-knowledge detection protocol *ZK_DETECT*$(W''$, *WM*, $-$, $-)$

Being convinced that $com(A^2)$ really contains the correctly computed value A^2, $\mathcal{V}$ and $\mathcal{P}$ compute the commitment

$$com(C) := \frac{com(A^2)}{com(B)} \bmod n$$

on the value C. Finally $\mathcal{P}$ proves to $\mathcal{V}$ in zero-knowledge, that the value contained in $com(C)$ is ≥ 0 using protocols from [5]. We refer to this sub-protocol as **ZKP**$(com(C)$ contains value $\geq 0)$. If $\mathcal{V}$ accepts this proof then *ZK_DETECT*() ends with *true*, otherwise with *false*.[10]

The protocol above satisfies the requirements of Definition 1: The completeness requirement is easy to verify by inspection. Soundness holds, because $\mathcal{P}$ can only cheat in *ZK_DETECT*() by cheating in the computation of $com(C)$ or by cheating $\mathcal{V}$ in proving that $com(C)$ contains a value ≥ 0. However, for this $\mathcal{P}$ has to either break the soundness of one of the **ZKP**() sub-protocols or the binding property of the commitment scheme which is assumed to be computationally infeasible. The protocol is zero-knowledge proof of knowledge in sense of Definition 1 since the sub-protocols **ZKP**() are zero-knowledge proof of knowledge

[10] Note that one has just to substitute the last zero-knowledge sub-protocol by **ZKP**$(com(C)$ contains value $< 0)$ to get a zero-knowledge protocol for *proving the absence* of a certain watermark.

(see [6] and [5]) and WM and all intermediary results involving WM, i.e., A and C, are perfectly hidden in the commitments.

Remark 1.6. If one relaxes the zero-knowledge requirement, then the efficiency of the protocol can be further improved by letting $\mathcal{P}$ *open* the commitment $com(C)$ instead of running **ZKP**($com(C)$ contains value ≥ 0). The information which is disclosed by C may be uncritical for certain applications. However, if carried out several times (for different W''), oracle attacks become possible [10]. ∘

4.4 Non-Blind Zero-Knowledge Detection Protocol

The protocol for non-blind zero-knowledge detection is quite similar to the previous one. However, one must take into account that Δ (in contrast to $DCT(W'', k)$) must not be disclosed to $\mathcal{V}$, because $\mathcal{V}$ would learn $DCT(W, k)$ otherwise. Thus one cannot directly use the homomorphic property of the commitment scheme to let $\mathcal{V}$ compute the value E in Equation 4 on his own, as it was the case for B in the blind zero-knowledge detection protocol.

Therefore we let $\mathcal{P}$ *stepwise* compute E and D and commit to the intermediary results. Now $\mathcal{P}$ can prove to $\mathcal{V}$ in zero-knowledge that the modular relations hold for the committed intermediary results as required by Equation 4. All these proofs together imply that the commitments $com(E)$ and $com(F)$ were computed correctly based on $com(\Delta)$ and $com(WM)$.

Having convinced $\mathcal{V}$, that $com(F)$ was computed correctly, $\mathcal{P}$ uses the same zero-knowledge proof protocol as in the blind case to prove to $\mathcal{V}$ that $com(F)$ contains a value ≥ 0. The whole protocol is illustrated in Figure 2. The proof sketch for completeness, soundness and the zero-knowledge property is similar to that of the blind version.

Remark 1.7. Note that zero-knowledge detection protocols for other watermarking schemes can be developed analogously, as long as their detection statistics are computable solely by the operations for which the correct computation on commitments is provable in zero-knowledge. When using protocols from [6] these operations are addition, multiplication and exponentiation. ∘

5 Proof of Ownership

In this section we show how the non-blind zero-knowledge watermark detection protocol can be applied in the context of proofs of ownership. We start by informally summing up some basics of ownership proof schemes. A complete formal treatment can be found in [1].

5.1 Ownership Proof Model and Scheme

The main parties involved in an ownership proof scheme are: a (supposed) copyright holder $\mathcal{H}$, a registration center $\mathcal{RC}$ and another third party $\mathcal{T}$. Here we

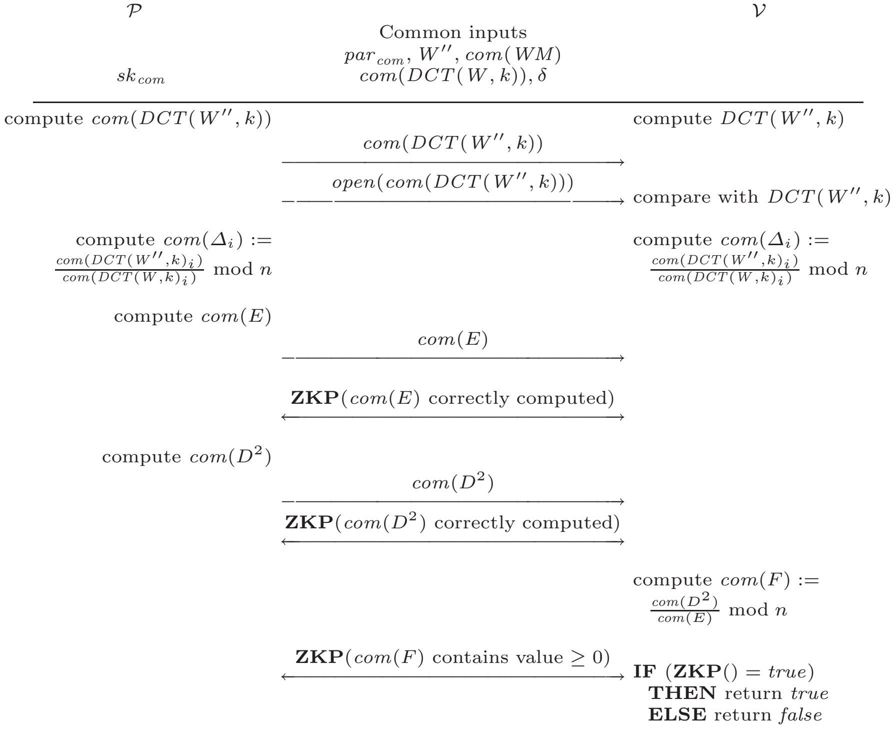
$\mathcal{P}$
$\mathcal{V}$
Common inputs
par_{com}, W'', $com(WM)$
$com(DCT(W,k)),\delta$
sk_{com}
compute $com(DCT(W'',k))$
compute $DCT(W'',k)$
$com(DCT(W'',k))$
$open(com(DCT(W'',k)))$
compare with $DCT(W'',k)$
compute $com(\Delta_i) := \frac{com(DCT(W'',k)_i)}{com(DCT(W,k)_i)} \bmod n$
compute $com(\Delta_i) := \frac{com(DCT(W'',k)_i)}{com(DCT(W,k)_i)} \bmod n$
compute $com(E)$
$com(E)$
ZKP($com(E)$ correctly computed)
compute $com(D^2)$
$com(D^2)$
ZKP($com(D^2)$ correctly computed)
compute $com(F) := \frac{com(D^2)}{com(E)} \bmod n$
ZKP($com(F)$ contains value ≥ 0)
IF (**ZKP**() = *true*)
THEN return *true*
ELSE return *false*

Fig. 2. The non-blind zero-knowledge detection protocol *ZK_DETECT*(W'', WM, W, $-$)

will assume that $\mathcal{RC}$ is trusted by all parties. However, using additional cryptographic techniques, the necessary trust in $\mathcal{RC}$ can be reduced by making $\mathcal{RC}$ accountable (see [1] for more details).

We restrict our discussion to the following main protocols: *REGISTER* and *PROVE*. Using the *REGISTER* protocol, $\mathcal{H}$ registers a new work at $\mathcal{RC}$ and receives an ownership certificate *cert* for this work. Afterwards, $\mathcal{H}$ can run the *PROVE* protocol with an arbitrary third party $\mathcal{T}$ to prove her rightful ownership for any work W'' for which she holds the copyrights.

Informally speaking, our model of copyright ownership considers $\mathcal{H}$ to be the copyright holder of a work W'' iff the following conditions hold:

1. $\mathcal{H}$ has previously registered a *new* work W,
2. W'' is similar to W and
3. W is the first registered work to which W'' is similar.

The last condition is necessary only if the similarity relation is no equivalence relation. This is to resolve collisions/ambiguities of the ownership relation on the basis of the registration time of works.

It was shown that if the similarity relation is an equivalence relation and *public* testable, i.e., it can be tested without any secret information of $\mathcal{RC}$, $\mathcal{RC}$ does not need to participate in the *PROVE* protocol. Using non-blind zero-knowledge watermark detection yields a public similarity test quite naturally: W'' is said to be similar to W, iff $\mathit{ZK\_DETECT}(W'', \mathit{WM}, W, k_{wm}) = \mathit{true}$ for a certain *WM*. This similarity test defines no equivalence relation, thus making it useless for offline ownership proofs in the *theoretical model* of [1]. However, for the above similarity test and for practical purposes one can drop this requirement. This is because the ambiguities only happen by chance with a very small probability or for works which are most likely degraded and worthless. This issue is discussed in more detail in Section 6.

5.2 Proof of Ownership Using ZK Watermark Detection

In the presentation of the following protocols we assume secure communication channels and we omit details of the message formats. In particular, where a signature is sent we assume that all message parts that are not known a priori are also sent and that techniques of robust protocol design like protocol- and message-type tags are used.

Registration: $\mathcal{H}$ starts the protocol by sending a registration request $\mathit{sign}_{\mathcal{H}}(W, \mathit{id}_{\mathcal{H}})$. $\mathcal{RC}$ first checks (using *registered?*) if W is a "new" work, i.e., that it is not similar to a previously registered work. If it is not new, then $\mathcal{RC}$ rejects the registration request and aborts the protocol. Otherwise, $\mathcal{RC}$ continues with the registration process: $\mathcal{RC}$ generates a new watermark *WM* and embeds it into W using the *EMBED* algorithm as described in Section 4.1. Now $\mathcal{RC}$ commits to *WM* and to $\mathit{DCT}(W, k)$. Then it generates an ownership certificate *cert* by signing $\mathcal{H}$'s identity, the public commitment parameters, the commitments to *WM* and $\mathit{DCT}(W, k)$ and the detection threshold δ. Thus an ownership certificate binds the identity of the copyright holder to the common inputs of a non-blind zero-knowledge detection protocol. $\mathcal{RC}$ stores the registration relevant data, especially those data which are necessary to test whether arbitrary works are similar to W, i.e., *WM* and $\mathit{DCT}(W, k)$. Finally $\mathcal{RC}$ returns the watermarked work W', the ownership certificate *cert* and the secret opening information sk_{com} for the commitments in *cert* to $\mathcal{H}$. The latter enables $\mathcal{H}$ to run *ZK_DETECT* protocols with the common inputs contained in *cert* for arbitrary works. Finally, $\mathcal{H}$ verifies *cert* and whether sk_{com} is the correct opening information for the commitments in *cert*. Note that $\mathcal{H}$ has to keep W secret and only publish W' or works derived from it.

Ownership proof: To prove ownership for a work W'' which has been derived from W', $\mathcal{H}$ just sends W'' together with *cert* to $\mathcal{T}$. $\mathcal{T}$ verifies the signature of *cert* and verifies whether the certificate contains the identity of $\mathcal{H}$. Then both run the non-blind zero-knowledge detection protocol as introduced in Section 4.4 with the common inputs W'' and $(\mathit{par}_{com}, \mathit{com}(\mathit{WM}), \mathit{com}(\mathit{DCT}(W, k)), \delta)$ as contained in *cert*. If this run of $\mathit{ZK\_DETECT}()$ ends with *true*, $\mathcal{T}$ is convinced that *cert* matches W'' and thus that $\mathcal{H}$ is the rightful copyright holder of W''. Note that the use of *blind* zero-knowledge watermark detection is not possible

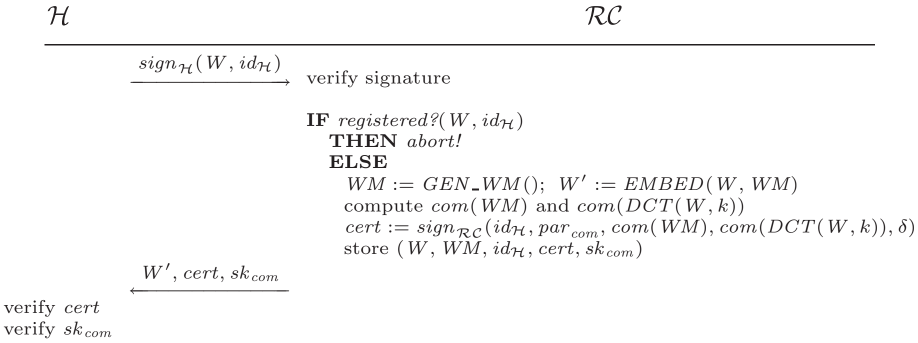
$\mathcal{H}$
$\mathcal{RC}$
$sign_{\mathcal{H}}(W, id_{\mathcal{H}})$
verify signature
IF $registered?(W, id_{\mathcal{H}})$
THEN $abort!$
ELSE
$WM := GEN\_WM();\ W' := EMBED(W, WM)$
compute $com(WM)$ and $com(DCT(W,k))$
$cert := sign_{\mathcal{RC}}(id_{\mathcal{H}}, par_{com}, com(WM), com(DCT(W,k)), \delta)$
store $(W, WM, id_{\mathcal{H}}, cert, sk_{com})$
$W', cert, sk_{com}$
verify $cert$
verify sk_{com}

Fig. 3. The registration protocol for offline proof of ownership

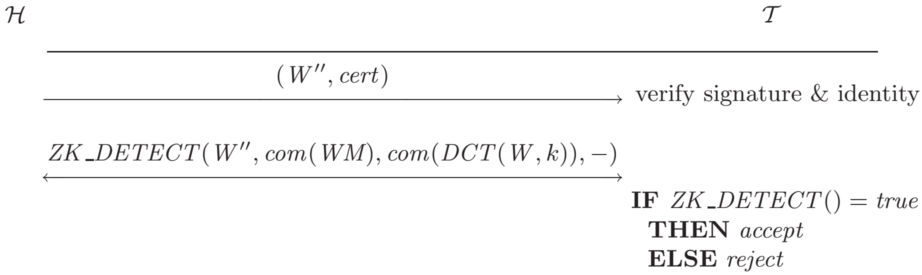

Fig. 4. The offline ownership proof protocol using the non-blind zero-knowledge detection protocol from Section 4.4

for this purpose. This is because it would weaken the link between ownership certificates and corresponding works, allowing attacks by a cheating $\mathcal{H}$: A cheating $\mathcal{H}$ could embed the watermark *WM* of one of her ownership certificates into any work and prove her ownership for the resulting work by showing the presence of *WM* in it.

6 Security

The security of the previously introduced protocols follows mainly from the security proof in [1], because in principle they are instantiations of the generic protocols. The only difference to the proofs of the generic protocols is that although *ZK_DETECT* does not test an equivalence relation, it is used in the offline ownership proof protocol. In the remainder of this section we will show that our protocols fulfill even those security requirements whose generic proofs make use of the fact that the similarity test is an equivalence relation:

Uniqueness for $\mathcal{H}$: *No other party apart from $\mathcal{H}$ can prove its ownership for a work W'' that is similar to W, i.e., was derived from W' by an operation against which the watermarking scheme is robust.*

Suppose that an attacker can successfully prove his ownership for a work W''. For this he needs an ownership certificate containing his identity and he must be able to run *ZK_DETECT*() successfully with W'' and the common inputs contained in the certificate. $\mathcal{RC}$'s test "*registered?*" prevents, that the attacker simply registers a work derived from W'.[11] The only chance of the attacker is to register a new work W^* so that the corresponding certificate *cert* matches W'' in the sense, that running *ZK_DETECT*() for W'' with the common inputs in *cert* ends with *true*. However, this happens only with a small probability, since $\mathcal{RC}$ chooses the watermark that it embeds into W^* independently from W'' and thus W'' would be a false positive detection.

Correctness for $\mathcal{T}$: *$\mathcal{T}$ accepts only correct ownership proofs, i.e., he cannot be cheated by a dishonest $\mathcal{H}$.*

$\mathcal{H}$ knows the reference data $DCT(W, k)$ and the watermark *WM* that is used in the run of *ZK_DETECT*() as part of the *PROVE* protocol. This is because $\mathcal{H}$ knows W itself and the secret opening information for the commitments contained in the ownership certificate. Using this information a dishonest $\mathcal{H}$ may be able to compute false positive data W^* for which he can prove his ownership to $\mathcal{T}$ by using the ownership certificate. However, such a "constructed" data item is with high probability randomly looking or strongly degraded (and nobody would ask for an ownership proof anyway).

To even prevent the possibility of such an attack we may require the prover to give additional zero-knowledge proofs that the differences $DCT(W^*, k)_i - DCT(W, k)_i$ lie in a certain range.

7 Conclusions

We presented the first provably secure zero-knowledge watermark detection protocols. These protocols are also the first which allow non-blind zero-knowledge detection of watermarks when embedded by the well known watermarking scheme from Cox et al. They can greatly improve the security of all applications in which the presence of a watermark needs to be proven to any untrusted party.

Further, we showed how zero-knowledge detection protocols can be used to construct efficient direct proofs of ownership without requiring a trusted third party to participate in the ownership proofs. This leads to a significant improvement of ownership proofs in terms of scalability and practicality.

[11] Note that an attacker has no knowledge about *WM*, even if he participated in a run of the *PROVE* protocol with $\mathcal{H}$. Thus he can't remove the watermark without severely damaging the image.

Acknowledgment

Our special thanks go to Michael Steiner for fruitful discussions. We also thank Birgit Pfitzmann for helpful comments.

References

1. André Adelsbach, Birgit Pfitzmann, Ahmad-Reza Sadeghi: Proving Ownership of Digital Content; Information Hiding: Third International Workshop, LNCS 1768, Springer-Verlag, Berlin, 2000, pp. 117-133 275, 282, 283, 284, 285
2. Mihir Bellare, Oded Goldreich: On Defining Proofs of Knowledge; Crypto '92, LNCS 740, Springer-Verlag, Berlin 1993, pp. 390-420 277
3. Mihir Bellare, Phillip Rogaway: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols; 1st ACM Conference on Computer and Communications Security, ACM Press, New York, 1993, pp. 62-73 276
4. Dan Boneh, James Shaw: Collusion-Secure Fingerprinting for Digital Data; Crypto '95, LNCS 963, Springer-Verlag, Berlin 1995, pp. 452-465 273
5. Fabrice Boudot: Efficient Proofs that a Committed Number Lies in an Interval; Eurocrypt '00, LNCS 1807, Springer-Verlag, Berlin 2000, pp. 431-444 276, 281, 282
6. Jan Camenisch, Markus Michels: Proving in Zero-Knowledge that a Number is the Product of Two Safe Primes; Eurocrypt '99, LNCS 1592, Springer-Verlag, Berlin, 1999, pp. 107-122 276, 280, 282
7. Scott Craver: Zero Knowledge Watermark Detection; Information Hiding: Third International Workshop, LNCS 1768, Springer-Verlag, Berlin, 2000, pp. 101-116 274
8. Ingemar J. Cox, Joe Kilian, Tom Leighton, Talal Shamoon: A Secure, Robust Watermark for Multimedia; Information Hiding, LNCS 1174, Springer-Verlag, Berlin, 1996, pp. 185-206 273, 274, 275
9. Scott Craver, Nasir Memon, Boon-Lock Yeo, Minerva M. Yeung: Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications; IEEE Journal on Selected Areas in Communications, Vol. 16, No. 4, Mai 1998, pp. 573-586 273, 274, 275, 277
10. Ingemar J. Cox, Jean-Paul M. G. Linnartz: Some General Methods for Tampering with Watermarks, IEEE Journal on Selected Areas in Communications, Vol. 16, No. 4, May 1998, pp. 587-593 274, 282
11. J. J. Eggers, J. K. Su, B. Girod: Asymmetric Watermarking Schemes; Sicherheit in Mediendaten, Berlin, Germany, Springer Reihe: Informatik Aktuell, September 2000 274
12. J. J. Eggers, J. K. Su, B. Girod: Public Key Watermarking By Eigenvectors of Linear Transforms; European Signal Processing Conference, Tampere, Finland, September 2000 274
13. Amos Fiat, Adi Shamir: How to Prove Yourself: Practical Solutions to Identification and Signature Problems; Crypto '86, LNCS 263, Springer-Verlag, Berlin 1987, pp. 186-194 276
14. Teddy Furon, Pierre Duhamel: An Asymmetric Public Detection Watermarking Technique; Information Hiding: Third International Workshop, LNCS 1768, Springer-Verlag, Berlin, 2000, pp. 88-100 274

15. Eiichiro Fujisaki, Tatsuaki Okamoto: A practical and provably secure scheme for publicly verifiable secret sharing and its applications; Eurocrypt '98, LNCS 1403, Springer-Verlag, Berlin 1998, pp. 32-46 276
16. Shafi Goldwasser, Silvio Micali, Charles Rackoff: The Knowledge Complexity of Interactive Proof Systems; SIAM Journal on Computing 18/1 (1989), pp. 186-207 274
17. Oded Goldreich, Jair Oren: Definitions and Properties of Zero-Knowledge Proof Systems; Journal of Cryptology, 1994, 7(1), pp. 1-32 277
18. K. Gopalakrishnan, Nasi Memon, Poorvi Vora: Protocols for Watermark Verification; Multimedia and Security, Workshop at ACM Multimedia 1999, pp. 91-94 274
19. Frank Hartung, Martin Kutter: Multimedia Watermarking Techniques; Proceedings of the IEEE, Vol. 87, No. 7, July 1999, pp. 1079-1107 273, 275, 277
20. Alexander Herrigel, Joseph Ó Ruanaidh, Holger Petersen, Shelby Pereira, Thierry Pun: Secure Copyright Protection Techniques for Digital Images; Information Hiding, LNCS 1525, Springer-Verlag, Berlin, 1998, pp. 169-190 273, 275, 277
21. Jean-Paul M. G. Linnartz, Marten van Dijk: Analysis of the Sensitivity Attack against Electronic Watermarks in Images; Information Hiding: Second International Workshop; LNCS 1525, Springer-Verlag, Berlin 1998, pp. 258-272 274
22. Birgit Pfitzmann, Matthias Schunter: Asymmetric Fingerprinting (Extended Abstract); Eurocrypt '96, LNCS 1070, Springer-Verlag, Berlin 1996, pp. 84-95 273
23. Lintian Qiao, Klara Nahrstedt: Watermarking Methods for MPEG Encoded Video: Towards Resolving Rightful Ownership; International Conference on Multimedia Computing and Systems, Austin, Texas, USA, 1998, pp. 276-285 273, 275, 277
24. Mitchell D. Swanson, Mei Kobayashi, Ahmed H. Tewfik: Multimedia Data-Embedding and Watermarking Technologies; Proceedings of the IEEE, Vol. 86, No. 6, June 1998, pp. 1064-1087 273