On the Possibility of Non-invertible Watermarking Schemes

Qiming Li and Ee-Chien Chang

[1] Temasek Laboratories
National University of Singapore
tslliqm@nus.edu.sg
[2] Department of Computer Science
National University of Singapore
changec@comp.nus.edu.sg

Abstract. Recently, there are active discussions on the possibility of non-invertible watermarking scheme. A non-invertible scheme prevents an attacker from deriving a valid watermark from a cover work. Recent results suggest that it is difficult to design a provably secure non-invertible scheme. In contrast, in this paper, we show that it is possible. We give a scheme based on a cryptographically secure pseudo-random number generator (CSPRNG) and show that it is secure with respect to well-accepted notion of security. We employ the spread spectrum method as the underlying watermarking scheme to embed the watermark. The parameters chosen for the underlying scheme give reasonable robustness, false alarm and distortion. We prove the security by showing that, if there is a successful attacker, then there exists a probabilistic polynomial-time algorithm that can distinguish the uniform distribution from sequences generated by the CSPRNG, and thus contradicts the assumption that the CSPRNG is secure. Furthermore, in our scheme the watermark is statistically independent from the original work, which shows that it is not necessary to enforce a relationship between them to achieve non-invertibility.

1 Introduction

There are many discussions on the uses of watermarking schemes in resolving ownership disputes. An interesting and well-known scenario is the *inversion attacks* studied by Craver et al. [7]. Under this scenario, Alice has the original image I and a secret watermark $W_{\mathtt{A}}$. She releases the watermarked image $\widetilde{I} = I + W_{\mathtt{A}}$ into the public domain. Given $\widetilde{I}$ and not knowing $W_{\mathtt{A}}$, Bob (who is an attacker) wants to find a watermark $W_{\mathtt{B}}$ that is present in both $\widetilde{I}$ and I. If such a watermark $W_{\mathtt{B}}$ is found, Bob can create confusion of the ownership by claiming that: (1) $\widetilde{I}$ is watermarked by his watermark $W_{\mathtt{B}}$, and (2) the image $\widetilde{I} - W_{\mathtt{B}}$ is the original. If Bob can successfully and efficiently find such $W_{\mathtt{B}}$, we say that the scheme is *invertible*.

Craver et al. [7] give an attacker when the underlying watermarking scheme is the well-known spread spectrum method. To overcome such attackers, they

J. Fridrich (Ed.): IH 2004, LNCS 3200, pp. 13–24, 2004.

propose a protocol that employs a secure hash, and claim that it is non-invertible. Qiao et al. [8, 9] also give watermarking schemes for video and audio which are claimed to be non-invertible. Subsequently, there are a number of works [10, 1, 2] exploiting weaknesses of known non-invertible schemes. Ramkumar et al. [10] give an attack for the scheme by Craver et al. [7], and they also give an improved scheme. On the other hand, [1, 2] give a formal definition of ambiguity attacks and mention that most proposed non-invertible schemes either do not come with a satisfactory proof of security, or the proofs are flawed. They also point out that if the false alarm of the underlying watermarking scheme is high (for e.g. 2^{-10}), then successful ambiguity attacks are possible. However, there is no mention of cases when the false alarm is low. Thus, it is interesting to know whether non-invertibility can be achieved when false alarm is low. Due to the difficulty of obtaining a non-invertible scheme, [2] propose to use a *trusted third party* (TTP) to issue valid watermarks. Although using a TTP is provably secure, there is still a question of whether it can withstand attackers that probe the system. The development of the studies of non-invertibility seems to lead to the conclusion that a stand-alone (in the sense that there is no TTP) non-invertible scheme does not exist. In this paper, in contrast, we argue that with low false alarm, it is possible to have a non-invertible scheme. We support our argument by giving a provably secure protocol that employs a *cryptographically secure pseudo-random number generator* (CSPRNG). The main idea is to show that if the scheme is invertible, then the CSPRNG is not secure, and thus lead to a contradiction.

Our protocol requires a computationally secure one-way function, whose existence is a major open problem in computer science. Nevertheless, it is well accepted that such functions exist. In practice, many cryptographic protocols rely on this unproven assumption.

Actually, we show that our protocol is secure against *ambiguity attacks*, of which inversion attacks are a special case. Given a work $\widetilde{I}$, a successful ambiguity attack outputs a watermark W that is embedded in $\widetilde{I}$, and a key K that is used to generate W. In a weaker form, the attack is also required to output an original I. In our discussion, we do not require the attacker to do so.

There are two components in our scheme. The first component addresses the issue of robustness, false alarm and distortion. This component is often called the *underlying* watermarking scheme. Due to the theoretical nature of this problem, we adopt the usual assumption that the hosts and noise are Gaussian, and distortion is measured by Euclidean 2-norm. In our protocol, we employ the well-known spread spectrum method as the underlying scheme.

The second component consists of key-management and watermark generation. In our setting, Alice (the owner) has a secret key $K_{\mathtt{A}}$, and she generates a watermark $W_{\mathtt{A}}$ using a CSPRNG with $K_{\mathtt{A}}$ as the seed. Next, she watermarks the original I using $W_{\mathtt{A}}$. To prove the ownership, Alice needs to reveal (or show that she knows) $K_{\mathtt{A}}$ and $W_{\mathtt{A}}$. Interestingly, our scheme does not use the original I to derive the key $K_{\mathtt{A}}$, nor the watermark $W_{\mathtt{A}}$. Hence the watermark is statistically independent from the original. This is in contrast to the method given by Craver

et al. [7], where Alice computes the hash of the original I, and uses the hash value $h(I)$ to generate the watermark $W_{\mathtt{A}}$. Hence, to achieve non-invertibility, it is not necessary to enforce a relationship between the watermark and the original work.

We give our main idea of our protocol in Section 2. We further give precise notations and describe the models that we use in Section 3. The details of the non-invertible scheme will be given in Section 4, followed by a proof of security in Section 5. Finally we give some remarks (Section 6) and conclude our paper (Section 7).

2 Main Idea

In our scheme, a watermark W is a sequence of -1 and 1 of length n, i.e. $W \in \{-1, 1\}^n$. We call W a *valid watermark* if it is generated by a CSPRNG using some m-bit seed, where $m < n$. Thus, the number of valid watermarks is not more than 2^m, and not all sequences in $\{-1, 1\}^n$ are valid watermarks.

Suppose we have a probabilistic polynomial-time algorithm B such that given any work $\widetilde{I}$ that is embedded using some valid watermark W, B can successfully find a valid watermark $\widehat{W}$ embedded in $\widetilde{I}$ with probability that is not negligible[3].

Now, we want to use B to construct a polynomial statistical test $\mathcal{T}$ that distinguishes a truly random sequence from a sequence generated by the CSPRNG, thus lead to a contradiction.

Given a sequence W, $\mathcal{T}$ carried out the following steps:

1. Embed W in I to get $\widetilde{I}$, where I is a randomly chosen work.
2. Ask B for a valid watermark $\widehat{W}$ embedded in $\widetilde{I}$.
3. Declare that W is from the random source if B fails to find such a watermark, and declare that W is generated by the CSPRNG otherwise.

By carefully choosing parameters for the underlying watermarking scheme, the probability that a valid watermark exists in a randomly chosen $\widetilde{I}$ can be exponentially small.

Hence, if W is generated by the truly random source, then it is very unlikely that a valid watermark exists in $\widetilde{I}$, and thus most of the time, B fails and the decision by $\mathcal{T}$ is correct. On the other hand, if W is indeed generated from the CSPRNG, the chances that a valid $\widehat{W}$ can be found is not negligible since B is a successful attacker. So, with probability that is not negligible, the decision made by $\mathcal{T}$ is correct.

Combining the above 2 cases leads to the conclusion that $\mathcal{T}$ can distinguish the two distributions. This contradicts with the assumption that the pseudo random number generator is secure. Therefore, no such B exists, and the scheme is non-invertible as a consequence.

[3] W and $\widehat{W}$ can be different.

3 Notations and Models

3.1 Overall Setting

A *work* is a vector $I = (x_1, x_2, \ldots, x_n)$ where each x_i is a real number. A *watermark* W is a sequence in $\{-1, 1\}^n$. A *key* K is a sequence of m binary bits. A watermark generator $f : \{0, 1\}^m \to \{-1, 1\}^n$ maps a key to a watermark. We say that a watermark W is *valid* if and only if w is in the range of f, i.e., it is generated from some key K by f.

The *underlying* watermarking scheme consists of an embedder and a detector. Given an original work I and a watermark W, the embedder computes a watermarked work $\widetilde{I}$. Given a work $\widetilde{I}$ and a watermark W, the detector declares whether W is embedded in $\widetilde{I}$, or not.

Before watermarking an original work I, Alice chooses a secret key $K_{\mathtt{A}}$ and generates a watermark $W_{\mathtt{A}} = f(K_{\mathtt{A}})$. Alice then embeds $W_{\mathtt{A}}$ into I. To resolve disputes of ownership, Alice has to reveal both the secret key $K_{\mathtt{A}}$ and the watermark $W_{\mathtt{A}}$. (In zero-knowledge watermarking setting [3, 6], Alice only has to prove that she knows $K_{\mathtt{A}}$ and $W_{\mathtt{A}}$).

In a successful ambiguity attack, given $\widetilde{I}$, Bob (the attacker) manages to find a pair $K_{\mathtt{B}}$ and $W_{\mathtt{B}}$ such that $f(K_{\mathtt{B}}) = W_{\mathtt{B}}$ and $W_{\mathtt{B}}$ is already embedded in $\widetilde{I}$. A formal description of ambiguity attacks will be presented in Section 3.3.

It is unreasonable to require a successful attacker to be always able to find the pair $K_{\mathtt{B}}$ and $W_{\mathtt{B}}$ for every work $\widetilde{I}$. Thus, we consider an attacker *successful* as long as the probability that he succeeds, on a randomly chosen $\widetilde{I}$, is non-negligible (greater than $1/p(n)$ for some positive polynomial $p(\cdot)$). Note that the probability distribution to be used in the definition of a successful attacker is important in the formulation. In Section 3.3 we will give more details on this.

We measure computational efficiency with respect to n, the number of coefficients in a work. Thus, an algorithm that runs in polynomial time with respect to n is considered efficient.

3.2 Statistical Models of Works and Watermarked Works

In this section, we give the statistical models of works. Recall that a work I is expressed as $I = (x_1, x_2, \ldots, x_n)$, where each x_i is a real number. We assume that I is Gaussian. That is, the x_i's are statistically independent and follow zero-mean normal distribution. Thus, to generate a random I, each x_i is to be independently drawn from the normal distribution $\mathcal{N}(0, 1)$. Note that the expected energy $E(\|I\|^2)$ is n.

Although the distribution of the original works is Gaussian, the distribution of the watermarked works is not necessarily Gaussian. Consider the process where an $\widetilde{I}_r$ is obtained by embedding a randomly chosen W_r from $\{-1, 1\}^n$ into a randomly chosen original work I. If the embedder simply adds the watermark to the original work, then the distribution of such watermarked work $\widetilde{I}_r$ is the convolution of the distribution of the watermarks and that of the original works,

which is not necessarily Gaussian. Let us denote the distribution of $\widetilde{I}_r$ as $\mathbf{X}_\mathbf{r}$ and call it the distribution of *randomly watermarked works*.

Now, consider the process where a **valid** watermark W_v is uniformly chosen (by uniformly choosing the key for the watermark generator), and then the watermarked work $\widetilde{I}_v$ is obtained by embedding W_v into a randomly chosen original work I. Let us denote the distribution of such $\widetilde{I}_v$ as $\mathbf{X}_\mathbf{v}$, and call it the distribution of *valid watermarked works*.

For clarity in notation, we use the symbol I to denote an original work, and add the tilde $\widetilde{I}$ to denote a work drawn from either $\mathbf{X}_\mathbf{r}$ or $\mathbf{X}_\mathbf{v}$ [4].

3.3 Formulation of Ambiguity Attacks

We follow the formulation of ambiguity attacks given in [2] with slight but important modification.

Let B be a probabilistic polynomial-time algorithm. Given some watermarked work $\widetilde{I}$, we say that B successfully attacks $\widetilde{I}$ if it outputs a pair (W, K) s.t. $\widetilde{I}$ contains the watermark W and $W = f(K)$, or outputs a symbol $\perp$ to correctly declare that such pair does not exist. Let us write $B(\widetilde{I}) = \texttt{PASS}$ when the attack is successful. We denote $\Pr[B(\widetilde{I}) = \texttt{PASS}]$ to be the probability that B successfully attacks a particular $\widetilde{I}$. The probability distribution is taken over the coin tosses made by B. Note that for $\widetilde{I}$ there does not exist such a pair (W, K), B has to output $\perp$ and hence is always successful.

We further denote $\widetilde{\mathcal{I}_n}$ to be a work that consists of n coefficients, and that is randomly drawn from the distribution of valid watermarked works $\mathbf{X}_\mathbf{v}$. Let $\Pr[B(\widetilde{\mathcal{I}_n}) = \texttt{PASS}]$ to be the probability that an attack by B is successful. In this case, the probability distribution is taken over the coin tosses made by B, as well as the choices of watermarked $\widetilde{\mathcal{I}_n}$. Then we have the

Definition 1 *Let B be a probabilistic polynomial-time algorithm. We say that B is a successful attacker if, there exists a positive polynomial $p(\cdot)$, s.t. for all positive integer n_0, there exists an integer $n > n_0$, and*

$$\Pr[B(\widetilde{\mathcal{I}_n}) = \texttt{PASS}] > 1/p(n).$$

In other words, B is a successful attacker if B successfully output a watermark-key pair with probability that is not negligible.

Note that our definition is a slight modification from [2]. The definition in [2] does not take into account cases where there is no valid watermark in a work. Moreover, the distribution of the watermarked work $\widetilde{I}$ is taken over the random choices of the original works. In our formulation, the watermarked work is drawn from $\mathbf{X}_\mathbf{v}$, and we differentiate the case where there are some valid watermarks in the given work from the case where there is not any.

[4] Clearly these two distributions $\mathbf{X}_\mathbf{r}$ and $\mathbf{X}_\mathbf{v}$ are different. However, by an argument similar to that in Section 5, it is not difficult to show that these two distributions are computationally indistinguishable.

This modification is important. We cannot simply say that an attacker is successful if $\Pr[B(\widetilde{\mathcal{I}_n}) = \texttt{PASS}]$ is high. This is because we observe that, it is possible to design a watermarking scheme such that for a randomly chosen work $\widetilde{I}$, the probability that it does not contain a valid watermark is very high. In that case, a trivial algorithm that always declares "can not find a valid watermark" is correct with high probability, and thus by definition is a successful attacker. Due to this consideration, we decide to consider $\mathbf{X}_\mathrm{v}$ in the definition, and separate the two cases where valid watermarks do or do not exist.

3.4 Cryptographically Secure Pseudo-random Number Generator

Loosely speaking, a *pseudo-random number generator* (PRNG) takes a seed of a certain length as input and outputs a string, which is of a longer length than that of the seed.

A *cryptographically secure pseudo-random number generator* (CSPRNG) is a PRNG whose output string cannot be computationally distinguished from a truly random distribution. Formal definition of the security of CSPRNG is done in terms of polynomial statistical tests [11]. We follow a simplified definition of statistical tests used in [4].

Let $\{0,1\}^n$ be the set of binary strings of length n, and $\{0,1\}^*$ denotes the set of all binary strings of all lengths. Formally, we have the following definitions.

Definition 2 *A PRNG g is a deterministic polynomial-time algorithm $g : \{0,1\}^m \to \{0,1\}^{q(m)}$, for some positive integer m and positive polynomial $q(m)$.*

Definition 3 *A probabilistic polynomial-time statistical test $\mathcal{T}$ is a probabilistic polynomial-time algorithm that assigns to every input string in $\{0,1\}^*$ a real number in the interval $[0,1]$.*

In other words, $\mathcal{T}$ can be considered as a function $\mathcal{T} : \{0,1\}^* \to [0,1]$, which terminates in polynomial time, and whose output depends also on the coin tosses during execution. Let r_n be the expected output of $\mathcal{T}$ over all truly random n-bit strings drawn uniformly from $\{0,1\}^n$, and all coin tosses made by $\mathcal{T}$. We have

Definition 4 *A PRNG g passes test $\mathcal{T}$ if, for every positive integer t, and every positive polynomial $q(m)$, there exists a positive integer m_0, such that for all integers $m > m_0$, the expected output of $\mathcal{T}$, given a $q(m)$-bit string generated by g, lies in the interval $(r_{q(m)} - m^{-t}, r_{q(m)} + m^{-t})$, assuming the seed of g is uniformly distributed over $\{0,1\}^m$.*

If a PRNG g does not pass a test $\mathcal{T}$, we say that $\mathcal{T}$ has an *advantage* in distinguishing g from a truly random source. Then we can define CSPRNG as

Definition 5 *A CSPRNG is a PRNG g that passes every probabilistic polynomial-time statistical test $\mathcal{T}$.*

In other words, no test $\mathcal{T}$ can have an advantage in distinguishing a CSPRNG g from a truly random source.

In this paper, we employ the CSPRNG due to Blum et al. [4]. A Blum number N is an integer that is the product of two primes, each congruent to 3 (mod 4). Let QR_N be the set of all quadratic residues in $\mathbb{Z}_N^*$. That is, $x \in QR_N$ if and only if there exists an $x_0 \in \mathbb{Z}_N^*$ such that $x_0^2 \equiv x \mod N$. Let $s \in QR_N$ be a seed to the Blum CSPRNG, the i-th bit b_i in the output string is computed as

$$b_i = (s^{2^i} \mod N) \mod 2. \tag{1}$$

In other words, we compute the output string by squaring the current number (starting from the seed) to get the next number, and take the least significant bit as the output.

Following the above notations, we have the

Definition 6 *A Blum PRNG is a function $g : QR_N \to \{0,1\}^{q(m)}$ defined as $g(s) = b_0, b_1, \cdots, b_{q(m)-1}$, where $b_i = (s^{2^i} \mod N) \mod 2$, N is a Blum number of length m, and $q(m)$ is a positive polynomial of m.*

It is proved in [4] that, under the well accepted assumption that integer factorization is hard, this PRNG is secure. That is, it passes every polynomial statistical test $\mathcal{T}$. We shall refer to it as the Blum CSPRNG.

4 A Non-invertible Scheme

Now, we describe the proposed secure protocol. The parameters for the protocol are three constants T, k and m.

In the proof of security, the parameters should be expressed in terms of n. We will choose

$$k = 1/100, \qquad T = nk/2 = n/200, \qquad m = \sqrt{n}. \tag{2}$$

4.1 Underlying Watermarking Scheme

The underlying watermarking scheme is essentially the spread spectrum method. For completeness and clarity, we describe the embedding and detection processes.

Embedding: Given an original I and a watermark W, the watermarked $\widetilde{I}$ is

$$\widetilde{I} = I + kW,$$

where k is a predefined parameter.

Detection: Given a work $\hat{I}$ and a watermark W, declare that $\hat{I}$ is watermarked if and only if

$$\hat{I} \cdot W \geq T,$$

where $\cdot$ is the vector inner product and T is a predefined parameter.

For simplicity, we omit normalization in the embedding. Thus, the energy $\|\widetilde{I}\|^2$ of a watermarked work is expected to be higher than the original work. Our proof can be modified (but tedious) when normalization is to be included.

4.2 False Alarm, Robustness, and Distortion (Parameters T and k)

The performance of a watermarking scheme is measured by its false alarm, robustness and distortion. Detailed analysis can be found in [5]. Here, we are more concerned with the false alarm.

The false alarm F is the probability that a randomly chosen $\widetilde{I}$ is declared to be watermarked by a random valid watermark W. That is

$$F = \Pr[\widetilde{I} \cdot W > T] \tag{3}$$

where $\widetilde{I}$ is drawn from the distribution of randomly watermarked works $\mathbf{X_r}$, and W is uniformly chosen from $\mathcal{W}$ the set of valid watermarks.

The false alarm F is small. To see that, consider any given $W \in \mathcal{W}$ and $\widetilde{I}$ randomly chosen from distribution $\mathbf{X_r}$, it is not difficult to show that the distribution $(\widetilde{I} \cdot W)$ is a zero-mean normal distribution with standard derivation δ where δ can be analytically derived. If $T = C_0\delta$ where $C_0 > 0$ is some positive constant, then the probability that a random $\widetilde{I}$ satisfies $(\widetilde{I} \cdot W > T)$ is less than $\exp(-C_0^2/2)$. Using the parameters in (2), $\delta < 2\sqrt{n}$. Since $T = n/200$, it is many times larger than the standard derivation δ.

For each $W_i \in \mathcal{W}$, where $1 \leq i \leq |\mathcal{W}|$, let F_i be the probability that $\widetilde{I} \cdot W_i > T$ for random $\widetilde{I}$ from $\mathbf{X_r}$. By the argument above, F_i is exponentially small with respect to n. More precisely, given the parameters in (2) and random $\widetilde{I}$ from $\mathbf{X_r}$,

$$F_i = \Pr[\widetilde{I} \cdot W_i > T] = \exp(-d_i n) \tag{4}$$

for some positive constant d_i. Therefore,

$$F = \sum_{i=1}^{|\mathcal{W}|} F_i \Pr[W = W_i] \leq \exp(-C_1 n) \tag{5}$$

where C_1 is the maximum d_i in (4), which is a positive constant.

By choosing $k = 1/100$, the distortion introduced during embedding is 1% of the original work. We could also choose k to be a slow decreasing function, for e.g. $k = 1/\sqrt{\log n}$, so that the ratio of the distortion over the energy of the work tends to 0 as n increases. Our proof still holds for this set of parameters.

Similarly, the scheme is very robust. Since the expected inner product of a watermarked image and the watermark is $E[(I + kW) \cdot W] = kn$, a noise of large energy is required to pull the inner product below the threshold $T = kn/2$. In this case, for noise with energy n (i.e. same as the original image), the watermark can still be detected in the corrupted work with high probability.

4.3 Watermark Generation (Parameter m)

A watermark is generated using a CSPRNG $f : \{0,1\}^m \rightarrow \{-1,1\}^n$ where $m \leq n$. Thus, it takes a small seed of m bits and produces a watermark. Note that this CSPRNG can be easily translated from the Blum CSPRNG by mapping

the output 0 to -1, and 1 unchanged. Let $\mathcal{W}$ to be the range of the function f, and it is actually the set of valid watermarks. Clearly, $|\mathcal{W}| \leq 2^m$.

Intuitively, for better security, we should have large m so that given a valid watermark, it is computationally difficult for an attacker to find the key K, such that $f(K) = W$. However, in some applications and our proof, we need the number of valid watermark to be small, so that it is computationally difficult for an attacker to find a valid watermark. On the other hand, if m is too small, an attacker can look for a suitable valid watermark using brute-force search.

In our construction, we choose $m = \sqrt{n}$, thus $|\mathcal{W}| = 2^{\sqrt{n}}$. As a result, it is computationally infeasible to do a brute-force search in the set of valid watermarks. At the same time, consider a randomly watermarked work $\widetilde{\mathcal{I}_n}$ drawn from distribution $\mathbf{X_r}$, which is of length n. With the parameters as in (2), the probability that $\widetilde{\mathcal{I}_n}$ contains any valid watermark $W \in \mathcal{W}$ is very small. Let us denote this probability $V(n)$ as a function of n, that is,

$$V(n) = \Pr[\exists W \in \mathcal{W}, \quad \widetilde{\mathcal{I}_n} \cdot W > T] \tag{6}$$

where $\widetilde{\mathcal{I}_n}$ is drawn from $\mathbf{X_r}$. Recall from Section 4.2 that the probability F_i that a randomly watermarked work can be declared as watermarked by a given valid watermark $W_i \in \mathcal{W}$ is exponentially small with respect to n. In particular, $F_i \leq \exp(-C_1 n)$ for some positive constant C_1 and for all $1 \leq i \leq |\mathcal{W}|$. Therefore,

$$\begin{aligned} V(n) &= 1 - \prod_{i=1}^{|\mathcal{W}|}(1 - F_i) \leq 1 - (1 - \exp(-C_1 n))^{2^m} \\ &< 2^m \exp(-C_1 n) < \exp(-C_1 n + \sqrt{n}) \end{aligned} \tag{7}$$

where C_1 is some positive constant. Note that $V(n)$ is a negligible function of n.

5 Proof of Security

Now, we are ready to prove that the proposed protocol is secure. We assume that the function f is a CSPRNG. Suppose that there is a successful attacker B as defined in Definition 1, we want to extend it to a statistical test $\mathcal{T}$ that has an advantage in distinguishing sequences produced by f from that by a truly random source. Since f is a CSPRNG, this leads to a contradiction, and thus such a B is impossible.

Given an input $W \in \{-1, 1\}^n$, the following steps are carried out by $\mathcal{T}$:

1. Randomly pick an original work I.
2. Compute $\widetilde{I} = I + kW$. That is, embed W into I.
3. Pass $\widetilde{I}$ to B and obtain an output.
4. If the output of B is a pair $(\widehat{W}, \widehat{K})$, such that $\widehat{W} = f(\widehat{K})$, then $\mathcal{T}$ declares that W is generated by f by outputting a 0. Otherwise B outputs a $\perp$, then $\mathcal{T}$ declares that W comes from a random source by outputting a 1.

We want to calculate the expected output of $\mathcal{T}$ for the following 2 cases. If the difference of the expected outputs of these 2 cases is non-negligible, then by the definitions in Section 3.4, f is not a CSPRNG, thus leads to a contradiction.

Case 1: W is from a random source. Suppose W is from a random source, then the probability that there exists a valid watermark $\widehat{W} \in \mathcal{W}$ in $\widetilde{I}$ is exactly the probability $V(n)$ in (7), which is negligible with respect to n as we have shown in Section 4.3. Hence, we know that $\mathcal{T}$ will almost always output a 1 to correctly declare that it is from the random source, except in the unlikely event $\mathcal{E}$ where $\widetilde{I}$ happens to contain a valid watermark. Clearly $\mathcal{E}$ happens with negligible probability $V(n)$. We observe that, when $\mathcal{E}$ happens, $\mathcal{T}$ may output a 0 with a probability that is not negligible (since B is a successful attacker). We consider the obvious worst case (best case for the attacker) that, $\mathcal{T}$ always output 0 when $\mathcal{E}$ happens. In this case, the fraction of 0's output by $\mathcal{T}$ is $V(n)$, which is still negligible. Therefore, let $E_1(\mathcal{T})$ be the expected output of $\mathcal{T}$, we have

$$E_1(\mathcal{T}) > 1 - V(n). \tag{8}$$

Case 2: W is from the CSPRNG f. Suppose W is generated by f, then W is a valid watermark. Since B is a successful attacker, by definition B is able to find a valid watermark $\widehat{W}$ that is already embedded in $\widetilde{I}$ with a probability that is not negligible. More specifically, for any positive integer n_0,

$$\Pr[B(\widetilde{I}) = \texttt{PASS}] > 1/p(n)$$

for some positive polynomial $p(\cdot)$ and for some $n > n_0$. Hence, the probability that $\mathcal{T}$ decides that W is from the CSPRNG f is more than $1/p(n)$. Hence, let $E_2(\mathcal{T})$ be the expected output of $\mathcal{T}$ in this case, and we have

$$E_2(\mathcal{T}) < \left(1 - \frac{1}{p(n)}\right). \tag{9}$$

Consider the difference between (8) and (9). Since $V(n)$ is negligible but $1/p(n)$ is not, the difference cannot be negligible because the sum of two negligible functions is still negligible. Hence, the difference between $E_1(\mathcal{T})$ and $E_2(\mathcal{T})$ is not negligible. Thus $\mathcal{T}$ has an advantage in distinguishing the truly random source from the the output of f, therefore f by definition is not a CSPRNG, which is a contradiction. As a result, such a successful attacker B does not exist.

6 Remarks and Future Works

Choice of m. In our construction we require the parameter m to be small. However, it seems that even if it is large, say $m = n/2$, the protocol is still secure. Thus it would be interesting to find an alternative proof that handles large m.

Underlying watermarking scheme. For simplicity in the proof, we use a simple watermarking scheme, and "discretized" watermark $W \in \{-1, 1\}^n$. The draw back is that the performance of false alarm, robustness and distortion would be far from optimal. Recent results in communication theory offer schemes that can

achieve much higher performance. Thus, we can have much lower false alarm, with other requirement fixed. On the other hand, it is also not clear whether we can make these schemes secure against inversion attacks. This is because in these schemes, the watermark is usually derived from the original in an insecure manner. It is interesting to investigate this issue. Furthermore, our proof requires valid watermarks to be "sparsely populated" in $\{-1, 1\}^n$. On the other hand, schemes with high performance usually require the watermarks to be densely populated, so as to reduce the distortion. Therefore, it is interesting to know if our proof can be extended.

Proving ownership. As mentioned earlier, to prove the ownership of a work $\widetilde{I}$, Alice has to show that she knows a pair $(K_{\mathtt{A}}, W_{\mathtt{A}})$, such that $W_{\mathtt{A}}$ is correctly generated from $K_{\mathtt{A}}$ and is detectable in $\widetilde{I}$. However, directly revealing such a pair in the proof might leak out information that leads to successful attacks. One alternative is to use zero-knowledge interactive proofs to prove the relationship between $K_{\mathtt{A}}$ and $W_{\mathtt{A}}$ without revealing the actual values. We note that it is straight forward to apply known zero-knowledge interactive proofs efficiently in our scheme. This is an advantage of our construction over schemes that involves hash functions (such as [7]), which are difficult to prove using known zero-knowledge interactive proofs.

Generation of watermarks. In Craver et al. [7], Alice computes a secure hash of the original I, and uses the hash value $h(I)$ to generate the watermark $W_{\mathtt{A}}$, which is then embedded into I. It is commonly believed that we need to generate the watermark from the original in a one-way manner to achieve non-invertibility since the attacker would be forced to break the underlying one-way function.

Interestingly, our scheme does not use the original I to derive the key $K_{\mathtt{A}}$, nor the watermark $W_{\mathtt{A}}$. Hence the watermark is statistically independent from the original. Although we can view the hash value $h(I)$ as the secret key $K_{\mathtt{A}}$ in our setting, our results show that it is not necessary to enforce a relationship between the watermark and the original work.

7 Conclusions

Resistance to inversion attacks is an important requirement for a secure digital right management system. Many schemes have been proposed to improve security. On the other hand, there are also attacks proposed to break these schemes. In this paper, we give a provably secure protocol that is resistant to inversion (and ambiguity) attacks. We prove the security using well accepted techniques in cryptography. Specifically, we show that if an inversion attack is possible, then we can computationally distinguish a truly random sequence from a sequence generated from a cryptographically secure pseudo-random number generator. It is interesting to investigate how to bring our proposed protocol into practice.

References

[1] A. Adelsbach, S. Katzenbeisser, and A.-R. Sadeghi. On the insecurity of non-invertible watermarking schemes for dispute resolving. *International Workshop on Digital Watermarking (IWDW)*, pages 374–388, 2003.

[2] A. Adelsbach, S. Katzenbeisser, and H. Veith. Watermarking schemes provably secure against copy and ambiguity attacks. *DRM*, pages 111–119, 2003.

[3] A. Adelsbach and A. Sadeghi. Zero-knowledge watermark detection and proof of ownership. *4th Int. Workshop on Info. Hiding*, LNCS 2137:273–288, 2000.

[4] L. Blum, M. Blum, and M. Shub. A simple secure unpredictable pseudo-random number generator. *SIAM Journal on Computing*, 15:364–383, 1986.

[5] I.J. Cox, M.L. Miller, and J.A. Bloom. *Digital Watermarking*. Morgan Kaufmann, 2002.

[6] S. Craver. Zero knowledge watermark detection. *3rd Intl. Workshop on Information Hiding*, LNCS 1768:101–116, 2000.

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

[8] L. Qiao and K. Nahrstedt. Non-invertible watermarking methods for MPEG encoded audio. In *Proceedings of the SPIE 3675, Security and Watermarking of Multimedia Contents*, pages 194–202, 1998.

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

[10] M. Ramkumar and A. Akansu. Image watermarks and counterfeit attacks: Some problems and solutions. In *Symposium on Content Security and Data Hiding in Digital Media*, pages 102–112, 1999.

[11] A. Yao. Theory and application of trapdoor functions. *23rd IEEE Symposium on Foundation of Computer Science*, pages 80–91, 1982.