Watermarking Protocol Compatible with Secret Algorithms for Resisting Invertibility Attack

Xinpeng Zhang and Shuozhong Wang

School of Communication and Information Engineering, Shanghai University
Shanghai 200072, P.R. China
{xzhang,shuowang}@staff.shu.edu.cn

Abstract. Invertibility attack is a hostile measure to breach watermarking systems. In this paper, a novel watermarking protocol using a one-way hash function and a check of random watermarks is proposed in order to combat invertibility attacks. The described technique can be used in conjunction with any watermarking algorithm, no matter it is kept secret or made public, without resorting to a third party jury as required by some previous approaches. By introducing a set of reference sequences, segmentation of the digital information and iterative computation of watermarks, the protocol is further enhanced so that it can resist more sophisticated types of attack based on forging an illegitimate detector.

1 Introduction

Since mid-1990s digital watermarking as an effective means for intellectual property rights protection has attracted a great deal of research interests. Watermark is a digital code embedded imperceptibly and robustly into a multimedia signal, typically containing information about the owner, origin, status, and/or destination of the host material [1, 2]. With the development of watermarking techniques, various hostile attacks have emerged, such as geometric distortion attack, implementation attack, and protocol attack, which attempt to destroy watermarking systems in order to make illegal profits [3].

Craver et al. [4] proposed another smart protocol attack, invertibility attack. It does not aim at eliminating the embedded information or destroying synchronization, but demolish the very concept of the watermarking application itself. In invertibility attack, an attacker can create a counterfeit watermark in a digital product containing another legal mark to cause confusion therefore prevent rightful assertion of ownership when the attacker's watermarking algorithm is invertible. To combat this attack, non-invertible watermarking algorithms [5, 6] may be used since they cannot be abused as tools in invertibility attacks. Using these techniques, the structure of watermarking algorithms must be made public by submitting to an authoritative organization to check its non-invertibility. An alternative countermeasure is to design a special non-invertible watermarking protocol in conjunction with a traditional algorithm to defeat this attack even if the watermarking algorithm is invertible [7, 8]. However, a loophole still exists when a non-invertible algorithm or protocol is used. Having obtained any

R. Khosla et al. (Eds.): KES 2005, LNAI 3682, pp. 1134–1144, 2005.

particular watermarked multimedia data, an attacker can always tailor-make a counterfeit inserter and a corresponding detector, as well as a fake original and a workable key. In order to defeat such an attack, a secure watermarking protocol has been proposed in which correlation between the detecting algorithm and the embedded mark is imperative [9].

In the above-mentioned secure frameworks for resisting invertibility attack, a watermark hider must reveal his embedding/detecting scheme to an authorized jury to confirm its eligibility, i.e., the watermarking algorithm should be non-invertible or based on some normal mechanism, for example, QIM or spread spectrum method. As such, security of the embedded watermark will rely solely on the confidentiality of the key as stipulated by the Kerckhoff's principle universally observed in cryptography. In watermarking applications, however, this authorization procedure is rather cumbersome therefore lacks practicality. Without a general consensus within the watermarking community, a watermark hider may not wish to reveal his embedding/detecting algorithm to any third party, whether being an independent jury or the general public. The Kerckhoff's principle requires that an embedded watermark should be secure or cannot be removed even if the watermarking scheme is not a secret. It does not mean, however, that a watermarking algorithm must be revealed. In other words, watermarking is not necessary bound to the requirement of publicizing the marking mechanism. So, a watermarking technique capable of resisting invertibility attacks that is compatible with secret algorithms is certainly desirable.

However, a problem may occur in case the marking algorithm is kept secret. An attacker can forge an arbitrary mark-detector to declare presence of his fake watermark in a digital product since no one can open the black box to disclose the illegitimate mechanism. To resolve this problem, a secure watermarking protocol is proposed in this paper to prevent attackers from performing an invertibility attack by fabricating a fake original, a counterfeit key and a constructed secret algorithm. In this protocol, correlation between the detecting algorithm and the embedded mark is also necessary as in [9]. In addition to this common feature, the present framework also checks the detected results of random watermarks, and uses a set of reference sequences, segmentation of the digital product and iterative computation of marks. In this way, arbitrary forge of secret mark-detecting mechanisms by the attackers can be prevented.

2 Previous Works on Invertibility Attack

2.1 Invertibility Attack

Suppose that the copyright owner A embeds his watermark $\mathbf{W}_{\mathrm{A}}$ into the host medium $\mathbf{I}$ with an inserter $\mathbf{E}_{\mathrm{A}}$ and a key $\mathbf{K}_{\mathrm{A}}$. A watermarked medium $\mathbf{I}_{\mathrm{A}}$ is thus produced that is available to both the legitimate user and an attacker named B. Assume that B possesses a watermarking tool consisting of an inserter $\mathbf{E}_{\mathrm{B}}$ and a detector $\mathbf{D}_{\mathrm{B}}$. If B is able to construct a watermark $\mathbf{W}_{\mathrm{B}}$ as well as a counterfeit key $\mathbf{K}_{\mathrm{B}}$ and a fake data $\mathbf{I}'$ from $\mathbf{I}_{\mathrm{A}}$ such that

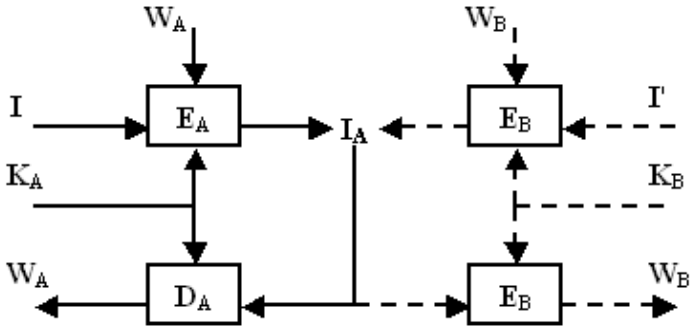

Fig. 1. Sketch of invertibility attack.

1. $\mathbf{E}_{\mathrm{B}}(\mathbf{I}', \mathbf{W}_{\mathrm{B}}, \mathbf{K}_{\mathrm{B}}) = \mathbf{I}_{\mathrm{A}}$,
2. $\mathbf{D}_{\mathrm{B}}(\mathbf{I}_{\mathrm{A}}, \mathbf{W}_{\mathrm{B}}, \mathbf{K}_{\mathrm{B}}) = 1$, (here 1 means "watermarked", and 0 "not watermarked") or $\mathbf{D}_{\mathrm{B}}(\mathbf{I}_{\mathrm{A}}, \mathbf{K}_{\mathrm{B}}) = \mathbf{W}_{\mathrm{B}}$, and
3. $\mathbf{I}'$ is similar to $\mathbf{I}_{\mathrm{A}}$,

the watermarking system ($\mathbf{E}_{\mathrm{B}}$, $\mathbf{D}_{\mathrm{B}}$) is said to be invertible. In the above operation, both the authentic watermark $\mathbf{W}_{\mathrm{A}}$ and the counterfeit mark $\mathbf{W}_{\mathrm{B}}$ can be detected from $\mathbf{I}_{\mathrm{A}}$ using $\mathbf{D}_{\mathrm{A}}$ and $\mathbf{D}_{\mathrm{B}}$, respectively. Therefore, the rightful owner of $\mathbf{I}_{\mathrm{A}}$ cannot be identified based on the extracted watermark. Figure 1 is a sketch of invertibility attack, in which the two sides of $\mathbf{I}_{\mathrm{A}}$ are symmetric, the left half being a legal watermarking procedure and the right half an illegal procedure counterfeited by the attacker.

In addition, a similar term "quasi-invertibility" can also be defined [4]. Quasi-invertibility is less stringent than full-invertibility therefore more harmful to legitimate watermarking system users. In the following discussion, quasi-invertibility and full-invertibility are not distinguished and will simply be referred to as "invertibility".

2.2 Non-invertible Algorithm and Protocol

To defeat invertibility attack, Craver et al. suggested that non-invertible watermarking techniques be used. In their approach, all legitimate watermarking schemes must be designed to be noninvertible. A one-way hash function or public-key cryptology is employed for this purpose. For example, let the watermark signal be a hash output of the host data [5], $\mathbf{W} = h(\mathbf{I})$. The host data are needed in detection. This way, the attacker can no longer forge a counterfeit watermark and counterfeit host data. In another method [6], the original medium is not needed in detection. The watermark is a hash of pseudorandom data selected by the digital product owner. As a key, the pseudorandom data is used to compute the watermark in detection. Since it is almost impossible to derive a counterfeit watermark $\mathbf{W}_{\mathrm{B}}$ and counterfeit host data $\mathbf{I}'$ from the watermarked data $\mathbf{I}_{\mathrm{A}}$, invertibility attack is virtually impossible. In this case, the mechanism of watermarking algorithm should be made public or submitted to an authorized organization to prove its non-invertibility.

An alternative method for resisting invertibility attack is to design some special watermarking protocol. The protocol presented in [7] can be used in conjunction with traditional algorithms based on addition of a pseudo-random sequence. Another protocol [8] is applicable to any watermarking algorithm regardless of being non-invertible or not, with which a watermark hider must embed, in addition to the watermark itself, the digital signatures of the original product and the mark into the digital content. Given a claimed watermarked copy $\mathbf{I}_{\mathrm{A}}$, the signatures should be verified after all the embedded data are extracted. Note that the asymmetry is derived from the watermarking protocol, which uses a signature mechanism, but not from the embedding/detecting algorithms. If the attacker's watermarking algorithm ($\mathbf{E}_{\mathrm{B}}$, $\mathbf{D}_{\mathrm{B}}$) is given, it will be very difficult to find a fake original $\mathbf{I}'$, a fake mark $\mathbf{W}_{\mathrm{B}}$ and a fake key $\mathbf{K}_{\mathrm{B}}$ to satisfy every aspect of this protocol, though the watermarking algorithm may be invertible. In this case, the watermarking mechanism must also be revealed to show it is based on a normal technique, such as QIM or spread spectrum method. Otherwise, an attacker can arbitrarily create a detected result that serves his purpose since the marking tool is a black box.

2.3 Invertibility Attack Based on Forged Algorithm and a Countermeasure

Although a non-invertible protocol or a rule requiring that all legitimate watermarking tools must be non-invertible is used, a pirate, having obtained a single watermarked product, can tailor-make an inserter $\mathbf{E}_{\mathrm{B}}$, a corresponding detector $\mathbf{D}_{\mathrm{B}}$, as well as a fake original, a counterfeit mark and a workable key to perform an invertibility attack [9]. When the invertibility of marking algorithm is required, the attacker can publish the mechanism of his forged watermarking algorithm ($\mathbf{E}_{\mathrm{B}}$, $\mathbf{D}_{\mathrm{B}}$) to show its non-invertibility, therefore declare the legitimacy of the fake mark $\mathbf{W}_{\mathrm{B}}$ in $\mathbf{I}_{\mathrm{A}}$. When a non-invertible protocol is used, the attacker can also obey the protocol and publish ($\mathbf{E}_{\mathrm{B}}$, $\mathbf{D}_{\mathrm{B}}$) like a normal algorithm. So, this invertibility attack based on forged algorithm is more powerful than the basic invertibility attack.

In order to combat this attack, a secure protocol has been proposed in which the embedded mark signal is derived from the detecting software $\mathbf{D}$ and an embedding key using a one-way hash function [9]. By establishing correlation between the embedded signal and the watermarking algorithm, an attacker cannot obtain a suitable mark and a workable key when he constructs a counterfeit watermarking algorithm. Thus the forged-algorithm attack can no longer succeed. Similarly, the mechanism of marking algorithm should also be publicized to show its embedding technique as normal.

3 A Basic Protocol Compatible with Secret Algorithms

In the previous anti-invertibility-attack strategies, the requirement is too stringent for most watermarking applications since all watermarking algorithms must

be checked by an authorized organization to prove its non-invertibility or normality. In practice, watermarking system users are usually reluctant to reveal the marking procedure to any outsider. Furthermore, it is difficult to define "normality" of a watermarking algorithm. It is also possible that an attacker can design an invertible or abnormal algorithm that may be mistakenly judged as non-invertible or normal to cheat the third-party jury.

Thus, a more convenient and user-friendly watermarking framework capable of resisting invertibility attacks without the need of revealing the embedding and detecting algorithms is desired. With this system, although revelation of the marking algorithm is not compulsory, the watermark hider, to conform to the Kerckhoff's principle, should still ensure that an embedded mark cannot be removed even if the watermarking mechanism is known to attackers.

Without an authorized organization, an attacker, B, can arbitrarily construct not only a counterfeit mark $\mathbf{W}_\mathrm{B}$ and a fake key $\mathbf{K}_\mathrm{B}$, but also his own watermarking algorithm ($\mathbf{E}_\mathrm{B}$, $\mathbf{D}_\mathrm{B}$), in which the detector $\mathbf{D}_\mathrm{B}$ can be considered a black box. Specifically, the forged detector $\mathbf{D}_\mathrm{B}$ may have the same input-output relation as an authentic detector but with an entirely different internal structure. Assuming that inputs to the detector $\mathbf{D}_\mathrm{B}$ are a received cover $\mathbf{I}_\mathrm{I}$, a key $\mathbf{K}_\mathrm{I}$ and a watermark $\mathbf{W}_\mathrm{I}$, and output from $\mathbf{D}_\mathrm{B}$ is "1" or "0", the attacker can forge an detecting algorithm as follows to confuse the copyright of a digital media $\mathbf{I}_\mathrm{A}$:

If $\mathbf{I}_\mathrm{I}$ is similar to $\mathbf{I}_\mathrm{A}$, $\mathbf{W}_\mathrm{I} = \mathbf{W}_\mathrm{B}$ and $\mathbf{K}_\mathrm{I} = \mathbf{K}_\mathrm{B}$
  $\mathbf{D}_\mathrm{B}(\mathbf{I}_\mathrm{I}, \mathbf{K}_\mathrm{I}, \mathbf{W}_\mathrm{I}) = 1$
Else
  $\mathbf{D}_\mathrm{B}(\mathbf{I}_\mathrm{I}, \mathbf{K}_\mathrm{I}, \mathbf{W}_\mathrm{I}) = 0$

Although the output result is directly derived from the inputs, no one can find the illegal mechanism.

To prevent anyone from carrying out such an attack, we propose a novel watermarking protocol, which specifies the way of watermark generation, but not the internal structure of watermarking algorithm. Here we view any mark-detecting software as a binary sequence, although its actual structure is unknown. The proposed watermarking protocol employs a one-way hash function h and a concept of *key space*. The key space includes a large number of candidate keys, say, 2^{128} binary sequences, each having a length of 128. After selecting a key from the key space according to the protocol, the rightful owner, A, produces a mark signal by using h, his key, and a binary sequence corresponding to his detector, and embeds them into the cover data. On the detection side, the mark signals derived from both the key provided by A and other keys randomly selected from the key space, are used as the detector inputs for the judgment as to whether or not A is a rightful owner.

The key of this protocol is to establish correlation between the watermarking algorithm, the embedding key, and the mark signal, and to check the detection results of marks derived both from a given key and from randomly selected keys, so that any attacker cannot find a workable key, a counterfeit mark and a constructed secret algorithm simultaneously. Details of the proposed protocol are described as follows:

1. The protocol employs a one-way hash function h that is made public.
2. Embedding procedure: The detecting software $\mathbf{D}$ can be considered and treated as a binary sequence $\mathbf{S_D}$. The embedded mark signal $\mathbf{W}$ must be determined by $\mathbf{S_D}$ and $\mathbf{K}$, that is, $\mathbf{W} = h(\mathbf{S_D}, \mathbf{K})$, where the key $\mathbf{K}$ is selected by the copyright owner from the key space. Thus, the watermarked medium $\mathbf{I}_\mathrm{A}$ is generated: $\mathbf{I}_\mathrm{A} = \mathbf{E}_\mathrm{A}\ (\mathbf{I}, \mathbf{K}_\mathrm{A}, \mathbf{W}_\mathrm{A}) = \mathbf{E}_\mathrm{A}\ [\mathbf{I}, \mathbf{K}_\mathrm{A}, h(\mathbf{S}_{\mathbf{D}_\mathrm{A}}, \mathbf{K}_\mathrm{A})]$.
3. Detecting procedure: Suppose a consumer, C, has obtained the digital product $\mathbf{I}_\mathrm{A}$, and wants to know who is the rightful owner of $\mathbf{I}_\mathrm{A}$. If someone claims that he is the rightful owner, he must provide his detector $\mathbf{D}$, but no its mechanism, and his key $\mathbf{K}$ for the one-way hash function h. After obtaining $\mathbf{W}$ using h, he must also provide his key $\mathbf{K}$ and the signal $\mathbf{W}$ for his detector $\mathbf{D}$ to show that $\mathbf{D}(\mathbf{I}_\mathrm{A}, \mathbf{K}, \mathbf{W}) = 1$. The procedure should be under C's supervision. The claimer may keep his key secret but must prove that the keys entered into the hash function and the detector $\mathbf{D}$ are identical. For example, he may store his key in an IC card and insert the card into a special device to provide the content of his key. Furthermore, he must randomly select many, say, 100, other keys $\mathbf{K}'$ in the key space to compute the derived signals $\mathbf{W}'$ using h, and checks if $\mathbf{D}[\mathbf{I}_\mathrm{A}, \mathbf{K}, h(\mathbf{S_D}, \mathbf{K}')] = 0$. The claimer must also prove that the keys $\mathbf{K}$ entered into the detector $\mathbf{D}$ now are the same as the previous one. If the above procedure is successfully completed, the claim of copyright is accepted.

Here, the purpose of checking whether or not $\mathbf{D}[\mathbf{I}_\mathrm{A}, \mathbf{K}, h(\mathbf{S_D}, \mathbf{K}')]$ equals zero is to prevent anyone from forging a detector in which the output is always 1 when using his key. The purpose of proving identity of the two entered keys is to verify the prescribed correlation between the key, the algorithm, and the mark signal.

Using this protocol, the rightful owner A can provide his legitimate key $\mathbf{K}_\mathrm{A}$ and the detector software $\mathbf{D}_\mathrm{A}$ to satisfy the above requirements. On the other hand, even if the attacker B can construct a watermarking algorithm $(\mathbf{E}_\mathrm{B}, \mathbf{D}_\mathrm{B})$ and a forged mark $\mathbf{W}_\mathrm{B}$ with $\mathbf{D}_\mathrm{B}(\mathbf{I}_\mathrm{A}, \mathbf{K}_\mathrm{B}, \mathbf{W}_\mathrm{B}) = 1$, he cannot satisfy $\mathbf{W}_\mathrm{B} = h(\mathbf{S}_{\mathbf{D}_\mathrm{B}}, \mathbf{K}_\mathrm{B})$ so that fails to pass himself off as a copyright owner.

However, since the internal structure of the detecting software $\mathbf{D}$ can be arbitrary, that is, $\mathbf{D}$ is treated as a black box, the attacker B can define an arbitrary mapping between the input and output at his disposal using any possible method. This leaves some flaws for more sophisticated attacks. Therefore the protocol needs to be further enhanced as described in the following sections.

4 Scenarios of Possible Attacks

Although performing invertibility attacks by constructing counterfeit watermark and watermarking algorithm is difficult with the above-described protocol, more sophisticated attacks are still possible. Two different kinds of possible attacks are described in the following.

4.1 First Type of Possible Attack: Hash-Based Detector Attack

The first type of attack employs the identical one-way hash function h in the forged detector $\mathbf{D}_\mathrm{B}$ so that correlation between a fake key, a counterfeit mark, and a workable secret algorithm can be established. Suppose that inputs to $\mathbf{D}_\mathrm{B}$ are a received cover $\mathbf{I}_\mathrm{I}$, a key $\mathbf{K}_\mathrm{I}$, and a watermark signal $\mathbf{W}_\mathrm{I}$, and the output is 1 or 0. An attacker B selects a key $\mathbf{K}_\mathrm{B}$ from the key space as his counterfeit key and constructs the inside mechanisms of his forged detector $\mathbf{D}_\mathrm{B}$ as follows.

If $\mathbf{I}_\mathrm{I}$ is similar to $\mathbf{I}_\mathrm{A}$, and $\mathbf{W}_\mathrm{I} = h(\mathbf{S}_{\mathbf{D}_\mathrm{B}}, \mathbf{K}_\mathrm{I})$

$\mathbf{D}_\mathrm{B}(\mathbf{I}_\mathrm{I}, \mathbf{K}_\mathrm{I}, \mathbf{W}_\mathrm{I}) = 1$

Else

$\mathbf{D}_\mathrm{B}(\mathbf{I}_\mathrm{I}, \mathbf{K}_\mathrm{I}, \mathbf{W}_\mathrm{I}) = 0$

Here, the detector treats itself as a binary sequence $\mathbf{S}_{\mathbf{D}_\mathrm{B}}$ and computes the hash of $\mathbf{S}_{\mathbf{D}_\mathrm{B}}$ and $\mathbf{K}_\mathrm{B}$. So, the attacker can claim that he satisfies the protocol as described in the previous section. After providing his counterfeit key $\mathbf{K}_\mathrm{B}$ and the forged detector $\mathbf{D}_\mathrm{B}$ for the one-way hash function h to yield $\mathbf{W}_\mathrm{B}$, he input $\mathbf{K}_\mathrm{B}$ and $\mathbf{W}_\mathrm{B}$ into the forged detector $\mathbf{D}_\mathrm{B}$, output of the detector will be 1, while other mark signals derived from the keys randomly selected in the key space will produce 0. Thus the basic protocol is defeated.

4.2 Second Type of Possible Attack: Forged Detector Attack Based on Watermark Subset

With the protocol proposed in Section 3, for the given legitimate key $\mathbf{K}_\mathrm{A}$, $\mathbf{D}_\mathrm{A}[\mathbf{I}_\mathrm{A}, \mathbf{K}_\mathrm{A}, h(\mathbf{S}_{\mathbf{D}_\mathrm{A}}, \mathbf{K}_\mathrm{A})]$ must be 1, and for any other randomly selected key $\mathbf{K}'$, $\mathbf{D}_\mathrm{A}[\mathbf{I}_\mathrm{A}, \mathbf{K}_\mathrm{A}, h(\mathbf{S}_{\mathbf{D}_\mathrm{A}}, \mathbf{K}')]$ must be 0. Assume that $\mathcal{W}$ is the entire set of $\mathbf{W}$. The attacker can arbitrarily define a set $\mathcal{W}_1$ that is a small subset of $\mathcal{W}$, and construct the internal structure of a fake detector $\mathbf{D}_\mathrm{B}$ as follows.

If $\mathbf{I}_\mathrm{I}$ is similar to $\mathbf{I}_\mathrm{A}$, and $\mathbf{W}_\mathrm{I} \in \mathcal{W}_1$

$\mathbf{D}_\mathrm{B}(\mathbf{I}_\mathrm{I}, \mathbf{K}_\mathrm{I}, \mathbf{W}_\mathrm{I}) = 1$

Else

$\mathbf{D}_\mathrm{B}(\mathbf{I}_\mathrm{I}, \mathbf{K}_\mathrm{I}, \mathbf{W}_\mathrm{I}) = 0$

A counterfeit key $\mathbf{K}_\mathrm{B}$ can be found using a method of exhaustive enumeration so that $h(\mathbf{S}_\mathbf{D}, \mathbf{K}_\mathrm{B})$ belongs to $\mathcal{W}_1$. In this way, the attacker can cheat the customer by providing a fake key $\mathbf{K}_\mathrm{B}$.

Cost of this type of attack is estimated as follows. Assume that the ratio between the sizes of $\mathcal{W}_1$ and $\mathcal{W}$ is p, the average number of attempts for finding the first useful counterfeit key $\mathbf{K}_\mathrm{B}$ is $1/p$ because

$$\sum_{j=1}^{\infty} p\,(1-p)^{(j-1)}\, j = \frac{1}{p} \tag{1}$$

On the detection side, a consumer randomly selects other N keys to check for $\mathbf{D}_\mathrm{B}[\mathbf{I}_\mathrm{A}, \mathbf{K}_\mathrm{B}, h(\mathbf{S}_{\mathbf{D}_\mathrm{B}}, \mathbf{K}')] = 0$. The probability of satisfying the condition, Q, can be obtained:

$$Q = (1-p)^N \tag{2}$$

This indicates that the attacker has constructed a workable detector and found a fake key, leading to a successful attack.

When p is very close to zero,

$$p \approx \frac{1-Q}{N} \tag{3}$$

For example, if p=0.0001, a fake key can be found after, in average, 10000 attempts. In this case, if the consumer makes N=100 checks, he will accept the illegal claim of the attacker with a probability Q=99%. Since a large N is unrealistic, the cost of attack is not a problem for the attacker.

5 Protocol Enhancement Against Sophisticated Attacks

Since the two types of attack as described in the previous section form a threat to the basic protocol proposed in Section 3, we introduce further enhancement in order to provide additional resistance against possible attacks by using a forged detector.

5.1 Enhancement Against Hash-Based Attack

The core of the hash-based attack is that the fake detector can be treated as a binary sequence by itself, and the one-way hash function h can be used to compute a watermark signal in a forged detector. In order to prevent this, a set of reference sequences with different lengths is introduced into the protocol. The reference sequences are made of random numbers containing little redundant information, and therefore are hard to be compressed. And these reference sequences are made available to the public. In the enhanced protocol, the embedded watermark should be the hash of the sequence $\mathbf{S_D}$, the key, and in addition, the reference. In other words, $\mathbf{W} = h(\mathbf{S_D}, \mathbf{K}, \mathbf{R})$. The reference sequence $\mathbf{R}$ is the shortest among all reference sequences that are longer than the binary sequence $\mathbf{S_D}$ corresponding to $\mathbf{D}$.

When someone claims his copyright, he must obtain $\mathbf{W}$ by providing the key $\mathbf{K}$, the detector $\mathbf{D}$, and the reference $\mathbf{R}$ to h, then enter $\mathbf{W}$, together with $\mathbf{K}$ and $\mathbf{I_A}$, to the detector $\mathbf{D}$ in an environment isolated from the reference, as shown in Figure 2. The consumer should check an output "1" from $\mathbf{D}$. With other mark signal derived from any arbitrarily selected key, however, the result is always "0". It should be noted that the reference $\mathbf{R}$ cannot be included in the detector $\mathbf{D}$, since it is longer than $\mathbf{S_D}$. Therefore, a hash $h(\mathbf{S_D}, \mathbf{K}, \mathbf{R})$ cannot be computed in a forged detector when the detector is isolated from $\mathbf{R}$. Thus, the first type of the detector attack is defeated.

5.2 Enhancement Against Watermark Subset Based Attack

To deal with the above-described second type of attack, an additional mechanism is introduced to make the cost of the attack too high to be feasible. The host

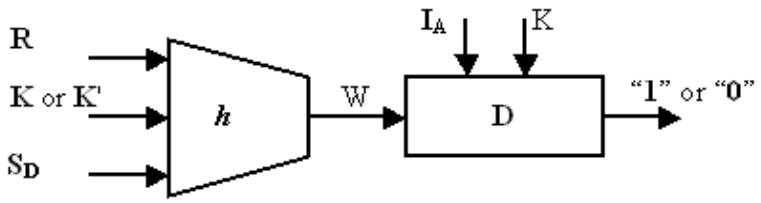

Fig. 2. Structure of the proposed watermarking protocol in which a reference **R** is added to prevent construction of a fake detector with the one-way hash.

medium $\mathbf{I}$ is first segmented into M sections, $\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_M$ in a way prescribed in the enhanced protocol. And the enhanced protocol also prescribes $\mathbf{W}_{\mathrm{A},1} = h(\mathbf{S}_{\mathbf{D}_\mathrm{A}}, \mathbf{K}_\mathrm{A})$, and $\mathbf{W}_{\mathrm{A},i} = h(\mathbf{S}_{\mathbf{D}_\mathrm{A}}, \mathbf{W}_{\mathrm{A},i-1})$ where i= 2, 3, …, M, so that all $\mathbf{W}_{\mathrm{A},i}$ form a chain. The marked sections are generated according to $\mathbf{I}_{\mathrm{A},i} = \mathbf{E}_\mathrm{A}(\mathbf{I}_i, \mathbf{K}_\mathrm{A}, \mathbf{W}_{\mathrm{A},i})$ where i = 1, 2, …, M. Finally, $\mathbf{I}_{\mathrm{A},i}$ are combined to yield the watermarked result $\mathbf{I}_\mathrm{A}$.

When someone claims his copyright, he should produce $\mathbf{W}_1 = h(\mathbf{S}_\mathbf{D}, \mathbf{K})$ and $\mathbf{W}_j = h(\mathbf{S}_\mathbf{D}, \mathbf{W}_{j-1})$ ($j > 1$), and show $\mathbf{D}(\mathbf{I}_{\mathrm{A},i}, \mathbf{K}, \mathbf{W}_i) = 1$ for each section of $\mathbf{I}_\mathrm{A}$ under supervision (i= 1, 2, …, M). With other $\mathbf{W}'_i$ derived from randomly selected keys, $\mathbf{D}(\mathbf{I}_{\mathrm{A},i}, \mathbf{K}, \mathbf{W}'_i)$ should be 0.

In this case the attacker B may still forge a detector $\mathbf{D}_\mathrm{B}$ to carry out the second type of fake-detector based attack as follows.

If $\mathbf{I}_\mathrm{I}$ is similar to $\mathbf{I}_{\mathrm{A},i}$, and $\mathbf{W}_\mathrm{I} \in \mathcal{W}_i$
  $\mathbf{D}_\mathrm{B}(\mathbf{I}_\mathrm{I}, \mathbf{K}_\mathrm{I}, \mathbf{W}_\mathrm{I}) = 1$
Else
  $\mathbf{D}_\mathrm{B}(\mathbf{I}_\mathrm{I}, \mathbf{K}_\mathrm{I}, \mathbf{W}_\mathrm{I}) = 0$

Here, $\mathcal{W}_i$ are subsets of $\mathcal{W}$. As previously explained, the attacker must find a suitable $\mathbf{K}_\mathrm{B}$ satisfying $\mathbf{W}_{\mathrm{B},1} = h(\mathbf{S}_{\mathbf{D}_\mathrm{B}}, \mathbf{K}_\mathrm{B}) \in \mathcal{W}_1$ and $\mathbf{W}_{\mathrm{B},i} = h(\mathbf{S}_{\mathbf{D}_\mathrm{B}}, \mathbf{W}_{\mathrm{B},i-1}) \in \mathcal{W}_i$ where i= 2, 3, …, M. Assume that all $\mathcal{W}_i$ have the same size, and the ratio between this size and that of $\mathcal{W}$ is p. Similar to Equation 1, the average number of attempts in finding the first useful counterfeit key $\mathbf{K}_\mathrm{B}$ is p^{-M}. If a consumer uses N randomly selected keys in the key space to derive the mark chains and check each $\mathbf{I}_{\mathrm{A},i}$, the probability of satisfying the conditions $\mathbf{D}(\mathbf{I}_{\mathrm{A},i}, \mathbf{K}_\mathrm{B}, \mathbf{W}'_i) = 0$ (i= 1, 2, …, M) will be:

$$Q = (1-p)^{M\,N} \tag{4}$$

Then, Q is the probability of being cheated by the attacker. The parameter p must be very close to 0 for a sufficiently large Q. Thus,

$$p \approx \frac{1-Q}{M\,N} \tag{5}$$

In this case, the cost of attack increases exponentially with M, therefore the attack scheme becomes infeasible. For example, when Q = 50%, N = 10, M = 10, the average number of attempts is greater than 10^{21}.

5.3 Enhanced Protocol

The integrated watermarking protocol against possible invertibility attacks is summarized as follows.

1. The protocol employs a one-way hash function h and a set including many reference sequences of different lengths that are made public. These reference sequences contain very little redundancy.
2. The legitimate owner of a digital product selects a key $\mathbf{K}_{\mathrm{A}}$ and a reference sequence $\mathbf{R}$ that possesses the smallest length among the existing sequences with a length greater than the binary array $\mathbf{S}_{\mathbf{D}_{\mathrm{A}}}$ corresponding to $\mathbf{D}_{\mathrm{A}}$. $\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_M$ are calculated according to $\mathbf{W}_1 = h(\mathbf{S}_{\mathbf{D}_{\mathrm{A}}}, \mathbf{K}_{\mathrm{A}}, \mathbf{R})$, $\mathbf{W}_i = h(\mathbf{S}_{\mathbf{D}_{\mathrm{A}}}, \mathbf{W}_{i-1})$ where $i = 2, 3, \ldots, M$.
3. The host medium $\mathbf{I}$ is segmented into M sections $\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_M$ in a prescribed way. Watermarked sections $\mathbf{I}_{\mathrm{A},i} = \mathbf{E}_{\mathrm{A}}(\mathbf{I}_i, \mathbf{K}_{\mathrm{A}}, \mathbf{W}_i)$ are generated accordingly, and then combined to produce $\mathbf{I}_{\mathrm{A}}$.
4. When someone claims that he is the rightful owner of $\mathbf{I}_{\mathrm{A}}$, he must compute $\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_M$ using his detector $\mathbf{D}$, his key $\mathbf{K}$, a reference sequence $\mathbf{R}$ and the function h. For each $\mathbf{I}_{\mathrm{A},i}$, the results $\mathbf{D}(\mathbf{I}_{\mathrm{A},i}, \mathbf{K}, \mathbf{W}_i) = 1$ and $\mathbf{D}(\mathbf{I}_{\mathrm{A},i}, \mathbf{K}, \mathbf{W}'_i) = 0$ should be checked where $\mathbf{W}'_i$ is derived from one of other N randomly selected keys. Here, the detecting software must be run in an environment isolated from the reference to ensure that it cannot make use of the reference sequences. The procedure of Step 4 should be under supervision and the copyright claimer may keep his key secret but must prove that the keys entered into the hash function and the detector $\mathbf{D}$ are identical.

6 Conclusion

A novel watermarking protocol capable of resisting invertibility attacks is introduced. Unlike the other methods subject to non-invertibility limitations, the proposed approach is applicable to any watermarking algorithm, whether the embedding scheme is kept secret or made public, and whether the algorithm is invertible or noninvertible. Using this protocol, a third party jury is not needed. The only additional requirements are that the embedded watermark signal should be generated in line with the protocol, and a series of detected results are used to decide the ownership of a digital product.

In the described protocol, the watermark signal is produced according to a key, a detector, as well as a set of reference sequences using a specified one-way hash function. Consequently, it is virtually impossible to generate an illegal watermark in a forged detector with a counterfeit key. In addition, by embedding several iteratively generated watermark signals into segmented medium data, the price for producing illegal detectors and keys is prohibitively high. In this way, invertibility attacks and possible attacks based on forged detector are defeated.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 60372090), Youth Project of Shanghai Municipality (No. 04AC93), and Key Project of Shanghai Municipality for Basic Research (No. 04JC14037).

References

1. Petitcolas, F.A.P., Anderson, R.J., Kuhn M.G.: Information Hiding–A Survey. Proc. IEEE. **87** (1999) 1062–1078
2. Hartung, F., Kutter, M.: Multimedia Watermarking Techniques. Proc. IEEE. **87** (1999) 1079–1107
3. Voloshynovskiy, S., Pereira, S., Pun, T., Eggers, J., Su, J.: Attacks on Digital Watermarking: Classification, Estimation-Based Attacks, and Benchmarks. IEEE Communications Magazine. (Aug. 2001) 118–126
4. Craver, S., Menon, N., Yeo, B. L., Yeung, M.: Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications. IEEE Journal on Selected Areas of Communications. **16** (1998) 573–586
5. Swanson, M. D., Zhu, B., Tewfik, A. H., Boney. L.: Robust Audio Watermarking Using Perceptual Masking. Signal Processing. **66** (1998) 337–355
6. Hwang, M., Chang, C., Hwang, K.: A Watermarking Technique Based on One-Way Hash Functions. IEEE trans. on Consumer Electronics. **45** (1999) 286–294
7. Li, Q., Chang, E.: On the Possibility of Non-Invertible Watermarking Schemes. 6th International Workshop on Information Hiding, Lecture Notes in Computer Science, **3200** (2004) 13–24
8. Katzenbeisser, S., Veith, H.: Securing Symmetric Watermarking Schemes against Protocol Attacks. Security and Watermarking of Multimedia Contents IV, Proc. SPIE. **4675** (2002) 260–268
9. Zhang, X., Wang, S.: Invertibility Attack against Watermarking Based on Forged Algorithm and a Countermeasure. Pattern Recognition Letters. **25** (2004) 967–973