Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition Letters 25 (2004) 967–973

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Invertibility attack against watermarking based on forged algorithm and a countermeasure

Xinpeng Zhang *, Shuozhong Wang

School of Communication and Information Engineering, Shanghai University, 149 Yanchang Road, Shanghai 200072, China

Received 13 August 2003; received in revised form 15 October 2003
Available online 19 March 2004

Abstract

It is shown in this paper that, even with a non-invertible watermarking algorithm or an asymmetric watermarking protocol, it is still possible to effect an invertibility attack, which relies on a forged watermarking algorithm, a counterfeit mark, and a fake key. Two examples are given to show the vulnerability of the unfortified non-invertible algorithm/asymmetric protocol. As a solution, a secure watermarking protocol is proposed, which establishes correlation between the watermarking algorithm and the embedded mark.

Keywords: Digital watermarking; Protocol; Invertibility attack

1. Introduction

As an effective means for the protection of intellectual property rights, watermarks can be embedded imperceptibly into multimedia data, typically containing information about the owner, origin, status, and destination of the product (Petitcolas et al., 1999; Hartung and Kutter, 1999). In the meantime, hostile attacks try to invalidate watermarking systems for illegal profits. The attacks can generally be classified into four categories: removal attack, geometric distortion attack, cryptographic attack, and protocol attack (Voloshynovskiy et al., 2001).

Protocol attacks do not aim at destroying the embedded information or disabling the synchronization. The goal is to demolish the very concept of the watermarking application itself. For example, copy attack (Kutter et al., 2000) is based on estimation of the embedded watermark in the spatial domain through a filtering process without knowing the watermarking algorithm and the key. The estimate is then used and inserted into the target data. In this way, the watermark is copied from a watermarked product into another.

Craver et al. (1997, 1998) devised another protocol attack, invertibility attack. It creates counterfeit watermarks in the same digital product to cause confusion therefore prevent rightful assertion of ownership. In this regard, it has been suggested

* Corresponding author. Tel.: +86-2156331435; fax: +86-2156331964.

E-mail addresses: zhangxinpeng@263.net (X. Zhang), shuowang@yc.shu.edu.cn (S. Wang).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.02.007

(Swanson et al., 1998; Hwang et al., 1999) that non-invertible watermarking algorithms be used as they cannot be abused as tools in invertibility attacks. An alternative countermeasure is to design a special watermarking protocol in conjunction with an invertible algorithm to defeat the attack (Katzenbeisser and Veith, 2002).

In this paper, however, we show that loopholes still exist even though a non-invertible algorithm or protocol is used. Having obtained any particular watermarked multimedia data, an attacker can always tailor-make a non-invertible inserter and a corresponding detector, as well as a fake original and a workable key. Since the watermarking algorithm constructed as such conforms to the non-invertibility requirements, the attacker can claim ownership of the data. A novel watermarking protocol is then introduced to provide a remedy.

2. Invertibility attack and the forged-algorithm attack

An invertibility attack is described as follows (Craver et al., 1997, 1998). Suppose that the copyright owner A embeds his watermark $\boldsymbol{W}_{\mathrm{A}}$ into the host medium $\boldsymbol{I}$ with an inserter $\boldsymbol{E}_{\mathrm{A}}$. A watermarked medium $\boldsymbol{I}_{\mathrm{A}}$ is thus produced that is available to both the legitimate user and a hacker named B. Assume that B possesses a watermarking tool consisting of an inserter $\boldsymbol{E}_{\mathrm{B}}$ and a detector $\boldsymbol{D}_{\mathrm{B}}$. If B is able to construct a watermark $\boldsymbol{W}_{\mathrm{B}}$ as well as a fake data $\boldsymbol{I}'$ from $\boldsymbol{I}_{\mathrm{A}}$ such that (1) $\boldsymbol{E}_{\mathrm{B}}(\boldsymbol{I}', \boldsymbol{W}_{\mathrm{B}}) = \boldsymbol{I}_{\mathrm{A}}$, (2) $\boldsymbol{D}_{\mathrm{B}}(\boldsymbol{I}_{\mathrm{A}}, \boldsymbol{W}_{\mathrm{B}}) = 1$, (here 1 means "watermarked", and 0 means "not watermarked"), and (3) $\boldsymbol{I}'$ is similar to $\boldsymbol{I}_{\mathrm{A}}$, then the watermarking system ($\boldsymbol{E}_{\mathrm{B}}$, $\boldsymbol{D}_{\mathrm{B}}$) is said to be invertible. According to the above operation, both the authentic watermark $\boldsymbol{W}_{\mathrm{A}}$ and the counterfeit mark $\boldsymbol{W}_{\mathrm{B}}$ can be detected from $\boldsymbol{I}_{\mathrm{A}}$ using $\boldsymbol{D}_{\mathrm{A}}$ and $\boldsymbol{D}_{\mathrm{B}}$, respectively. Therefore, the rightful owner of $\boldsymbol{I}_{\mathrm{A}}$ cannot be identified based on the extracted watermark.

In addition, a quasi-invertible watermarking tool can also be designed. With this tool the attacker can find $\boldsymbol{W}_{\mathrm{B}}$, $\boldsymbol{I}'$, and $\boldsymbol{I}_{\mathrm{B}}$ so that (1) $\boldsymbol{E}_{\mathrm{B}}(\boldsymbol{I}', \boldsymbol{W}_{\mathrm{B}}) = \boldsymbol{I}_{\mathrm{B}}$, (2) $\boldsymbol{D}_{\mathrm{B}}(\boldsymbol{I}_{\mathrm{B}}, \boldsymbol{W}_{\mathrm{B}}) = 1$, $\boldsymbol{D}_{\mathrm{B}}(\boldsymbol{I}_{\mathrm{A}}, \boldsymbol{W}_{\mathrm{B}}) = 1$, and (3) both $\boldsymbol{I}'$ and $\boldsymbol{I}_{\mathrm{B}}$ are similar to $\boldsymbol{I}_{\mathrm{A}}$. 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".

To defeat invertibility attacks, Craver et al. suggested that non-invertible watermarking techniques be used for copyright protection. In other words, all legitimate watermarking algorithms must be designed as non-invertible. If a non-invertible watermarking system ($\boldsymbol{E}_{\mathrm{B}}$, $\boldsymbol{D}_{\mathrm{B}}$) and a marked copy $\boldsymbol{I}_{\mathrm{A}}$ are given, the attacker cannot find a fake watermark $\boldsymbol{W}_{\mathrm{B}}$ and a fake original copy $\boldsymbol{I}'$ satisfying all the three conditions listed in either of the previous two paragraphs.

In fact, however, after obtaining the marked copy $\boldsymbol{I}_{\mathrm{A}}$, an attacker can always construct a non-invertible watermarking system ($\boldsymbol{E}_{\mathrm{B}}$, $\boldsymbol{D}_{\mathrm{B}}$) as well as $\boldsymbol{W}_{\mathrm{B}}$ and $\boldsymbol{I}'$ satisfying the above-mentioned conditions. In this way, the attacker can publish the mechanism of his illicit watermarking system ($\boldsymbol{E}_{\mathrm{B}}$, $\boldsymbol{D}_{\mathrm{B}}$) to show its non-invertibility, therefore declare the legitimacy of the fake mark $\boldsymbol{W}_{\mathrm{B}}$ in $\boldsymbol{I}_{\mathrm{A}}$. Note that the illicit watermarking system ($\boldsymbol{E}_{\mathrm{B}}$, $\boldsymbol{D}_{\mathrm{B}}$) is tailor-made for the particular object $\boldsymbol{I}_{\mathrm{A}}$ and cannot be used to perform attack against any other media because of its non-invertibility. Nonetheless the copyright of $\boldsymbol{I}_{\mathrm{A}}$ is breached, and a second forged system can be found against another watermarked object. Such an object-specific attack is termed forged-algorithm attack in this paper.

Katzenbeisser and Veith proposed an alternative watermarking system or protocol for resisting invertibility attack. The method can be used in combination with any watermarking algorithm regardless of being non-invertible or not. It will be shown in the next section, however, that Katzenbeisser's method is also vulnerable to the forged-algorithm attack.

3. Scenarios of possible attack against existing non-invertible schemes

3.1. Attack against non-invertible algorithms

In this subsection, we show how a forged algorithm can be obtained to perform a successful attack against a non-invertible watermarking system.

Without loss of generality, assume that the marked copy $\boldsymbol{I}_{\mathrm{A}}$ is a still image and the number of pixel is N. Randomly generate a binary sequence $\boldsymbol{R}_{\mathrm{B}}$ of length N which consists of +1 and −1, and then subtract $\boldsymbol{R}_{\mathrm{B}}$ from $\boldsymbol{I}_{\mathrm{A}}$ to produce a fake original $\boldsymbol{I}'$:

$$\boldsymbol{I}' = \boldsymbol{I}_{\mathrm{A}} - \boldsymbol{R}_{\mathrm{B}} \tag{1}$$

where "−" is a subtraction operation between two vectors.

Define a one-way hash function h, which generates a binary sequence of length N. Arbitrarily select a key k_{B} and compute

$$\boldsymbol{W}_{\mathrm{B}} = h(\boldsymbol{I}', k_{\mathrm{B}}) \tag{2}$$

In both $\boldsymbol{R}_{\mathrm{B}}$ and $\boldsymbol{W}_{\mathrm{B}}$, about $N/2$ elements are +1, and the rest are −1. So, one can find a permutation Π for re-organizing elements of $\boldsymbol{W}_{\mathrm{B}}$ to make the permuted result very similar to $\boldsymbol{R}_{\mathrm{B}}$. For example, assuming that $\boldsymbol{W}_{\mathrm{B}} = [+1,+1,-1,-1,+1,-1,+1,-1,-1,+1,+1,-1,-1,-1,+1,+1,\ldots]$, and $\boldsymbol{R}_{\mathrm{B}} = [-1,+1,+1,-1,+1,-1,-1,+1,-1,-1,+1,+1,+1,-1,-1,+1,\ldots]$, the attacker may define a permutation Π: $[1\rightarrow 3, 2\rightarrow 2, 3\rightarrow 4, 4\rightarrow 1, 5\rightarrow 8, 6\rightarrow 7, 7\rightarrow 5, 8\rightarrow 6, 9\rightarrow 10, 10\rightarrow 11, 11\rightarrow 12, 12\rightarrow 9, 13\rightarrow 15, 14\rightarrow 14, 15\rightarrow 16, 16\rightarrow 12, \ldots]$ to make $\Pi(\boldsymbol{W}_{\mathrm{B}}) \approx \boldsymbol{R}_{\mathrm{B}}$, where $\alpha \rightarrow \beta$ means to place the αth element on the βth position. Thus an illegal non-invertible watermarking system can be counterfeited using Π and h as follows.

The embedding procedure $\boldsymbol{E}_{\mathrm{B}}$:

1. Assuming that the original image is $\boldsymbol{I}$, a secret key k can be selected to compute a watermark by using the one-way hash function h:

$$\boldsymbol{W} = h(\boldsymbol{I}, k) \tag{3}$$

2. Permute $\boldsymbol{W}$ with Π to yield $\boldsymbol{W}_1$, and add it into the original $\boldsymbol{I}$ to produce a marked image.

Detecting procedure $\boldsymbol{D}_{\mathrm{B}}$:

1. If anyone announces that the rightful owner is himself, he must provide his original image and the key to compute the permuted mark $\boldsymbol{W}_1$. The difference between marked and original images, $\boldsymbol{R}$, can also bef obtained.
2. Computing the correlation coefficient ρ between $\boldsymbol{W}_1$ and $\boldsymbol{R}$:

$$\rho = \frac{1}{N}\sum_{i=1}^{N}[\boldsymbol{R}(i)\cdot \boldsymbol{W}_1(i)] \tag{4}$$

When ρ is greater than 1/2 by a specified margin, a "watermarked" decision is made, otherwise "not watermarked".

In this way, the attacker can provide his fake original $\boldsymbol{I}'$ and a fake key k_{B} to claim copyright. Because $\boldsymbol{W}_1$, the permuted version of $\boldsymbol{W}_{\mathrm{B}}$, is very similar to $\boldsymbol{R}_{\mathrm{B}}$, the value of ρ must be close to 1.

Fig. 1 sketches the forged embedding algorithm. It should be noted that the illegal watermarking system ($\boldsymbol{E}_{\mathrm{B}}$, $\boldsymbol{D}_{\mathrm{B}}$) is not invertible because a one-way hash function h is involved. When a watermarking system, including the one-way hash function h and the permutation Π, is given, it is virtually impossible to fabricate a fake original and a usable key. In the above procedure, the attacker not only "creates" the original image and the key, but also a permutation Π. In other words, he has successfully forged a workable watermarking algorithm. Although the forged watermarking system ($\boldsymbol{E}_{\mathrm{B}}$, $\boldsymbol{D}_{\mathrm{B}}$) cannot be used to attack any other watermarked image, the copyright of $\boldsymbol{I}_{\mathrm{A}}$ is breached. To

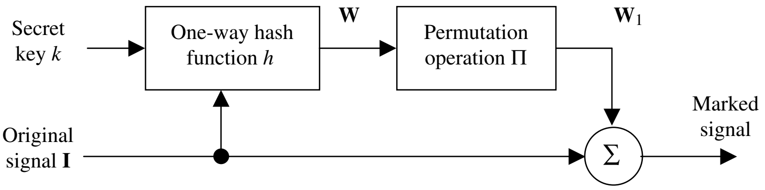

Fig. 1. Sketch of forged embedding algorithm.

attack another medium, the attacker must find a new permutation and construct another watermarking algorithm.

3.2. Forged-algorithm attack against the asymmetric watermarking protocol

In (Katzenbeisser and Veith, 2002), Katzenbeisser and Veith constructed an asymmetric watermarking system using a cryptographic signature on top of a traditional watermarking algorithm $\boldsymbol{E}$, as sketched in Fig. 2. The purpose was to defeat the invertibility attack even if $\boldsymbol{E}$ is invertible, since the asymmetry is not from $\boldsymbol{E}$, but from the non-invertible signature mechanism that includes a public key P and a private key S. The copyright owner A uses the original object $\boldsymbol{I}$, and a watermark $\boldsymbol{W}_{\mathrm{A}}$, to derive $\boldsymbol{W}'_{\mathrm{A}} = \boldsymbol{W}_{\mathrm{A}} \| S_{\mathrm{S}}(\boldsymbol{I}, P) \| S_{\mathrm{S}}(\boldsymbol{W}_{\mathrm{A}}, P)$, where the operator $\|$ concatenates two strings and S_{S} is a signature generator. $\boldsymbol{W}'_{\mathrm{A}}$ is then embedded into $\boldsymbol{I}$ using an embedding algorithm $\boldsymbol{I}_{\mathrm{A}} = \boldsymbol{E}(\boldsymbol{I}, \boldsymbol{W}'_{\mathrm{A}}, k_{\mathrm{A}})$, where k_{A} is a key. In detection, given a claimed watermarked copy $\boldsymbol{I}_{\mathrm{A}}$, the corresponding original $\boldsymbol{I}$, a watermark $\boldsymbol{W}$, and a key k, $\boldsymbol{W}'_{\mathrm{A}}$ is computed and split into three concatenated parts: $\boldsymbol{W}' = \boldsymbol{W}'_1 \| \boldsymbol{W}'_2 \| \boldsymbol{W}'_3$. Check whether $\boldsymbol{D}(\boldsymbol{I}_{\mathrm{A}}, \boldsymbol{I}, \boldsymbol{W}', k) = \mathrm{TRUE}$, $V_{\mathrm{S}}\ (\boldsymbol{I}, \boldsymbol{W}'_2, P) = \mathrm{TRUE}$, and $V_{\mathrm{S}}(\boldsymbol{W}'_1, \boldsymbol{W}'_3, P) =$ TRUE, where $\boldsymbol{D}$ is the detecting algorithm corresponding to $\boldsymbol{E}$, and V_{S} the signature verification process. Only if all the three are true, the watermark is accepted as genuine. If the attack's watermarking algorithm ($\boldsymbol{E}_{\mathrm{B}}$, $\boldsymbol{D}_{\mathrm{B}}$) is given, it is virtually impossible to find a fake original $\boldsymbol{I}'$, a fake mark $\boldsymbol{W}_{\mathrm{B}}$ and a fake key k_{B} satisfying all the three tests, even though the watermarking algorithm may be invertible.

However, the system in Fig. 2 is also susceptible to the forged-algorithm attack, which fabricates an illegal watermarking algorithm ($\boldsymbol{E}_{\mathrm{B}}$, $\boldsymbol{D}_{\mathrm{B}}$) together with a fake original, a derived watermark and a key. This is shown in the following where ($\boldsymbol{E}_{\mathrm{B}}$, $\boldsymbol{D}_{\mathrm{B}}$) is based on a dither modulation technique (Chen and Wornell, 1999).

Assuming that the length of $\boldsymbol{W}'$ is always M, convert $\boldsymbol{I}_{\mathrm{A}}$ into $\boldsymbol{Y}_{\mathrm{A}}$ using an orthogonal transform such as DCT or DFT, and select a quantization step Δ. The purpose of orthogonal transform is to spread the mark information $\boldsymbol{W}'$ over the entire space of host object. Then, select M transform coefficients in $\boldsymbol{Y}_{\mathrm{A}}$, denoted $y_1, y_2, \dots, y_M$. The dither coefficients are derived:

$$y'_k = y_k + e_k \quad k = 1, 2, \dots, M \tag{5}$$

where e_k is randomly and independently picked from $[-\Delta/2, \Delta/2]$. The coefficients $y_1, y_2, \dots,$ and y_M are replaced with $y'_1, y'_2, \dots,$ and y'_M and inversely transformed to produce an object $\boldsymbol{I}'$ as a fake original copy. The fake key k_{B} is made up of Δ and the way of selecting M coefficients from $\boldsymbol{Y}_{\mathrm{A}}$.

Select a fake $\boldsymbol{W}_{\mathrm{B}}$ to compute $\boldsymbol{W}'_{\mathrm{B}}$ according to the watermarking protocol. A vector of length M is obtained

$$d_k = \mathrm{round}\left(\frac{y_k}{\Delta}\right) \cdot \Delta - y_k \quad k = 1, 2, \dots, M \tag{6}$$

where round $(\cdot)$ takes the nearest integer of its argument. The values of d_k also lie within $[-\Delta/2, \Delta/2]$. A new dither vector $[D_1, D_2, \dots, D_M]^{\mathrm{T}}$ is used as the parameter of watermarking algorithm, where

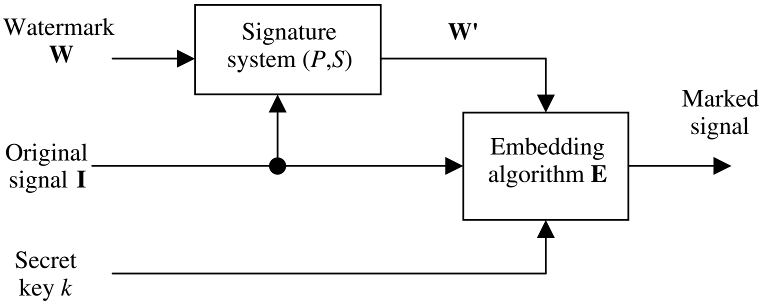

Fig. 2. Sketch of asymmetric watermarking protocol.

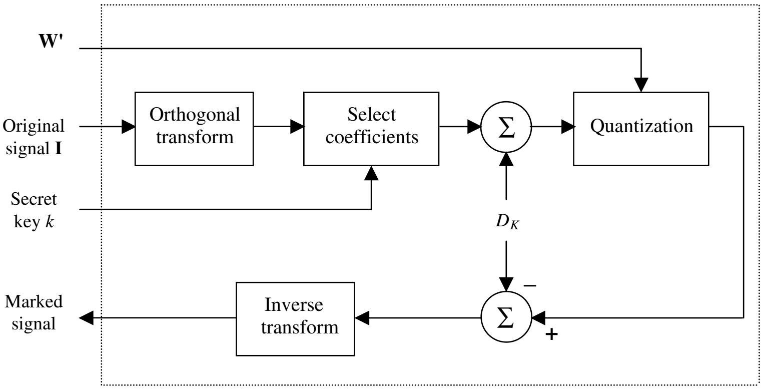

Fig. 3. Forged embedding algorithm $\boldsymbol{E}_{\mathrm{B}}$ in the asymmetric watermarking protocol.

$$D_k = \begin{cases} d_k, & \text{if} \quad W'_{\mathrm{B},k} = 0 \\ d_k - \Delta/2, & \text{if} \quad W'_{\mathrm{B},k} = 1, d_k \geqslant 0 \\ \Delta/2 + d_k, & \text{if} \quad W'_{\mathrm{B},k} = 1, d_k < 0 \end{cases}$$
$$k = 1, 2, \ldots, M \tag{7}$$

Thus, the attacker is able to construct its illegal watermarking algorithm as described in the following. Note that the original is not needed in detection.

The embedding procedure $\boldsymbol{E}_{\mathrm{B}}$ (see Fig. 3):

1. Convert the original data using an orthogonal transform and select M coefficients $y'_1, y'_2, \ldots, y'_M$ according to the key.
2. Embed $\boldsymbol{W}' = \boldsymbol{W} \| S_{\mathrm{S}}(\boldsymbol{I}, P) \| S_{\mathrm{S}}(\boldsymbol{W}, P)$ using the dither modulation method

$$y_k = \begin{cases} Q_0(y'_k + D_k) - D_k, & \text{if} \quad W'_k = 0 \\ Q_1(y'_k + D_k) - D_k, & \text{if} \quad W'_k = 1 \end{cases}$$
$$k = 1, 2, \ldots, M \tag{8}$$

where

$$Q_0(x) = \mathrm{round}\left(\frac{x}{\Delta}\right) \cdot \Delta \tag{9}$$

$$Q_1(x) = \mathrm{round}\left(\frac{x + \Delta/2}{\Delta}\right) \cdot \Delta - \frac{\Delta}{2} \tag{10}$$

3. Perform the inverse transform to produce a marked copy.

The detecting procedure $\boldsymbol{D}_{\mathrm{B}}$:

1. Transform the marked copy and pick M coefficients $y_1, y_2, \ldots, y_M$ according to the key.
2. Compute elements of the vector $\boldsymbol{V}$:

$$V_k = 1 - \frac{1}{2}\left[(-1)^{\mathrm{round}\left(\frac{y_k + D_k}{\Delta/2}\right)} + 1\right]$$
$$k = 1, 2, \ldots, M \tag{11}$$

3. Compare $\boldsymbol{V}$ with $\boldsymbol{W}'$. If they are identical, the output is TRUE, otherwise FALSE.

When the attacker publishes his fake watermarking algorithm and announces his mark $\boldsymbol{W}_{\mathrm{B}}$ and the key k_{B}, $\boldsymbol{V}$ and $\boldsymbol{W}'_{\mathrm{B}}$ should be identical according to (7) and (11). Eq. (5) indicates that e_k is the difference between the fake "original" y'_k and the marked coefficient y_k, which obeys a uniform distribution within $[-\Delta/2, \Delta/2]$, similar to the distribution of quantization error due to embedding a legal watermark using $\boldsymbol{E}_{\mathrm{B}}$. This way, the asymmetric watermarking system is broken under the forged-algorithm attack. As in the previous example, the attack is aimed at a specific object. Another fake watermarking algorithm must be constructed in an attack against a new object.

4. A secure watermarking protocol capable of resisting forged-algorithm attacks

As mentioned in Section 3, an attacker can arbitrarily construct a non-invertible watermarking

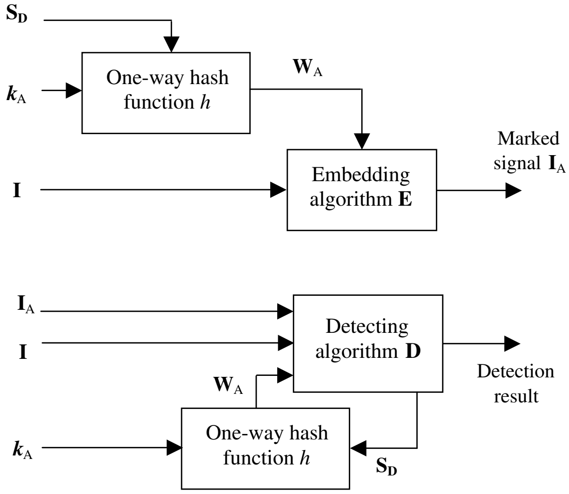

Fig. 4. The proposed secure watermarking protocol.

algorithm, or an invertible algorithm under an asymmetric protocol, to find a fake watermark in any media to falsely claim his "ownership". In order to defeat this kind of attack, correlation between the embedded signal and the watermarking algorithm should be established.

As the structure of the watermarking algorithm must be published according to Kerckhoff's principle, and the detecting software $\boldsymbol{D}$ can be considered and treated as a binary sequence $\boldsymbol{S_D}$, we propose a secure watermarking protocol as follows (see Fig. 4):

1. The protocol employs a one-way hash function h and publicizes it.
2. The copyright owner A selects his key k_A and compute $\boldsymbol{W}_A = h(\boldsymbol{S_D}, k_A)$ to be embedded into the original copy, $\boldsymbol{I}_A = \boldsymbol{E}(\boldsymbol{I}, \boldsymbol{W}_A) = \boldsymbol{E}[\boldsymbol{I}, h(\boldsymbol{S_D}, k_A)]$.
3. If one claims rightful ownership of a multimedia object, he must provide his original copy $\boldsymbol{I}$, the key k and the detecting software $\boldsymbol{D}$. A "watermarked" decision is made when $\boldsymbol{D}[\boldsymbol{I}_A, \boldsymbol{I}, h(\boldsymbol{S_D}, k_A)] = \text{TRUE}$, otherwise "not watermarked".

In the proposed protocol, the embedded data is a code derived from the detecting software $\boldsymbol{D}$ and a secret key, but not the detecting software itself. When an attacker constructs his counterfeit watermarking algorithm, he cannot obtain a suitable mark $\boldsymbol{W}$ resulting from h, therefore is unable to make $\boldsymbol{D}[\boldsymbol{I}_A, \boldsymbol{I}, h(\boldsymbol{S_D}, k)] = \text{TRUE}$. Thus the forged-algorithm attack can no longer succeed.

5. Conclusion

In this paper, we first describe a type of invertibility attack, which can be performed when a non-invertible watermarking algorithm or an asymmetric protocol is used. If a watermarking algorithm is properly constructed, any watermark signal needed for ownership authentication can be derived from a marked copy so that the copyright is confused. In order to defeat this attack, a secure watermarking protocol is proposed, in which correlation between the detecting algorithm and the embedded mark is imperative. Because the attacker cannot generate the algorithm and the required watermark simultaneously, the forged-algorithm attack is defeated.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (60372090).

References

Chen, B., Wornell, G.W., Dither modulation: a new approach to digital watermarking and information embedding. In: Security and Watermarking of Multimedia Contents. Proc. SPIE, vol. 3657, San Jose, CA, January 1999, pp. 342–353.

Craver, S., Menon, N., Yeo, B.L., Yeung, M., 1997. On the invertibility of invisible watermarking techniques. Proc. IEEE Internat. Conf. on Images Process., Chicago, Illinois, USA, October 1997, pp. 540–543.

Craver, S., Menon, N., Yeo, B.L., Yeung, M., 1998. Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks, and implications. IEEE J. Selected Areas Comm. 16 (4), 573–586.

Hartung, F., Kutter, M., 1999. Multimedia watermarking techniques. Proc. IEEE 87, 1079–1107.

Hwang, M.S., Chang, C.C., Hwang, K.F., 1999. A watermarking technique based on one-way hash functions. IEEE Trans. Consumer Electron. 45, 286–294.

Katzenbeisser, S., Veith, H., 2002. Securing symmetric watermarking schemes against protocol attacks. In: Security and

Watermarking of Multimedia Contents IV, Proc. SPIE, vol. 4675, San Jose, CA, January 2002, pp. 260–268.

Kutter, M., Voloshynovskiy, S., Herrigel, A., 2000. The watermark copy attack, in security and watermarking of multimedia contents II. Proc. SPIE 3971, 371–379.

Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G., 1999. Information hiding—A survey. Proc. IEEE 87, 1062–1078.

Swanson, M.D., Zhu, B., Tewfik, A.H., 1998. Multiresolution scene-based video watermarking using perceptual models. IEEE J. Selected Areas Comm. 16, 540–550.

Voloshynovskiy, S., Pereira, S., Pun, T., Eggers, J., Su, J., 2001. Attacks on digital watermarking: classification, estimation-based attacks, and benchmarks. IEEE Comm. Mag. (August), 118–126.