IS ASYMMETRIC WATERMARKING NECESSARY OR SUFFICIENT?

Matt L. Miller

NEC Research Institute, Princeton, NJ

ABSTRACT

In public watermarking applications, the public must have the ability to detect watermarks, but must not have the ability to remove them. It has been proposed that such systems might be made secure by using asymmetric key watermarking*, in which the embedder and detector use different keys.*

This paper asks whether asymmetric-key watermarking is sufficient and necessary for secure public watermarking applications. The answer found, both to the question of sufficiency and the question of necessity, is basically no. Asymmetric-key watermarking, by itself, is demonstrably insufficient, because there exist asymmetric-key watermarking systems that do not provide the required security. Asymmetric-key watermarking might not be necessary, because it is possible to imagine irreversible *watermarking systems, in which even complete knowledge of the embedder would not help an adversary remove watermarks.*

The paper concludes by suggesting that research in secure public watermarking should focus on the design of the detector. The embedding algorithm, and whether it employs any information that must be kept secret, is of secondary concern.

1. INTRODUCTION

In several applications of watermarking, the public must have the ability to detect watermarks, but must not have the ability to remove them. For example, in copy-prevention applications, such as proposed for DVD video [1] or digital music [10], recording devices contain watermark detectors and refuse to record watermarked content. Such applications are referred to as *public* watermarking applications, as opposed to *private* applications in which the general public is *not* allowed to detect watermarks.

Unfortunately, with any known watermarking system, access to a watermark detector significantly simplifies the task of removing watermarks. The simplest systems detect watermarks by correlating against a fixed reference pattern. Anyone who can examine a detector and obtain this pattern can remove the watermark by subtracting it from a Work. More complex methods of watermark detection, such as normalized correlation or phase-only correlation, can also be thwarted by simple attacks based on knowledge of the reference pattern.

It has been proposed that watermarks with public detectors might be made secure by using *asymmetric key watermarking* [11, 5, 6, 9]. This idea is analogous to asymmetric key cryptography, in which encryption and decryption employ different keys. Knowledge of one key does not imply knowledge of the other, so someone who has, say, the *de*cryption key cannot *en*crypt messages. In watermarking, the embedder and detector would use different keys, and knowledge of the detection key would not imply knowledge of the embedding key. Thus, someone who has access to a detector would not know what pattern has been added to the Work to watermark it, and, it is hoped, would be unable to remove the mark.

The present paper takes a critical look at whether asymmetric watermarking is truly the key to making public watermarks secure. It asks, first, whether key-asymmetry is sufficient to provide some security against watermark removal, and, second, whether it is necessary.

Section 2 addresses the question of sufficiency. Although some systems have been proposed in which embedding keys cannot be deduced from detection keys, many of these systems are susceptible to attacks based only on information available in the detector. Thus, secure key asymmetry, by itself, is demonstrably insufficient to ensure security in a public watermarking application.

Section 3 then turns to the question of necessity. Although insufficient by itself, key asymmetry is often assumed to be necessary for public watermarking applications. However, this assumption might not be correct. The section describes some ways in which a watermarking system might be secure against watermark removal when the adversary knows all details of the detector *and the embedder*. Such systems can be termed *irreversible watermarks*, since the embedding process cannot be reversed even with complete knowledge of the embedding key.

Finally, Section 4 concludes by suggesting that research in secure public watermarking should focus on the design of the detector. The embedding algorithm, and whether it employs any information that must be kept secret, is of secondary concern.

2. IS KEY ASYMMETRY SUFFICIENT TO ENSURE SECURITY?

To begin the discussion of sufficiency, let us first be more specific about the type of security we hope to obtain with asymmetric-key watermarking. There are many attacks that do not involve knowledge of *any* keys, and thus are unaffected by key asymmetry. For example, most image watermarking systems succumb to one or more of the attacks implemented in the Stirmark program [8], which does not employ any knowledge about the watermarking algorithm at all. To avoid including these attacks in this discussion, we should ask a more specific question: if we have a secure asymmetric-key watermarking system, can we conclude that knowledge of the detector does not allow for additional attacks, beyond those that can be performed without this knowledge?

This question can be answered by examining some of the asymmetric key watermarking systems that have been proposed. Below, some effective attacks against two such proposals are described. These attacks exploit the detection key, but do not require the embedding key. Thus, we can conclude that key asymmetry is not sufficient for secure public watermarking.

2.1. Attacks on eigenvector watermarking

In [11], van Schyndel, Tirkel, and Svalbe proposed watermarking by embedding Legendre sequences. This scheme was later generalized by Eggers, Su, and Girod [5] to embed eigenvectors of any linear transform. In such an *eigenvector watermarking* system, the detection key is a transform matrix, $\mathbf{G}$, and the embedding key is a real-valued eigenvector, $\mathbf{w}$, of $\mathbf{G}$, with positive corresponding eigenvalue. The embedder adds a scaled version of $\mathbf{w}$ to the cover Work, $\mathbf{c_w} = \mathbf{c_o} + \alpha\mathbf{w}$. The detector computes the correlation between a received Work, $\mathbf{c}$, and the transform of that Work, $\mathbf{Gc}$. If any eigenvector of $\mathbf{G}$ (with positive eigenvalue) has been embedded in $\mathbf{c}$, then this correlation is high, and the detector can conclude that the watermark is present, without revealing $\mathbf{w}$.

An attack on this system, due to Furon, is reported in [4]. The attack involves knowledge of $\mathbf{G}$, but does not involve knowledge of the embedding secret. First, project a watermarked Work into a subspace defined by some eigenvectors of $\mathbf{G}$. Next, scale this vector, project it back into media space, and subtract it from the Work. Regardless of which eigenvector (or combination of eigenvectors) was used in the embedder, this reduces the correlation between the Work and its transform, and can render the watermark undetectable.

A more powerful attack can often be implemented by analyzing the shape of the detection region. Consider the case in which all the eigenvalues of $\mathbf{G}$ are real and positive, and the eigenvectors are orthogonal. An attack could proceed as follows: first, rotate media space into the coordinate system defined by $\mathbf{G}$'s eigenvectors. In this coordinate system, all the points that will be detected as containing the watermark lie outside an ellipsoid alligned with the axes. To remove a watermark from a given Work, we need only find a nearby point inside this ellipsoid, and project it back into media space. Finding a nearby unwatermarked point can be accomplished by means of an efficient search. The resulting attack is illustrated in Figure 1.

2.2. Attacks on Furon and Duhamel's method

A different asymmetric-key watermarking system was first proposed by Furon and Duhamel in [6], and later improved upon in [7]. In their system, the embedding key is a white-noise pattern and a filter. The detection key is the power spectrum of the filter. The detection key, then, is the result of a *one-way signal processing operation*, in the sense that the white-noise pattern and the filter cannot be deduced from their power spectrum.

The embedder in Furon and Duhamel's system applies the filter to the white-noise pattern, and adds it to a vector extracted from the cover Work. An inverse extraction process is then applied to obtain the watermarked Work. The detector applies the extraction process to a received Work, computes its power spectrum, and applies a statistical test to determine whether it is closer to a flat spectrum (indicating that the watermark is not present), or to the spectrum of the filtered noise (indicating that the watermark is present). The detector can thus detect watermarks using only the spectrum of the filter, and does not reveal enough information for a hacker to determine either the filter itself or the original white-noise pattern.

The early versions of Furon and Duhamel's approach are susceptible to a *spectral whitening* attack that does not require information about the embedder [4]. First, extract a vector from the watermarked Work. Next, take its Fourier transform, and scale the coefficients so that the spectrum is nearly flat. Finally, apply the inverse extraction function to obtain a Work in which the watermark is undetectable.

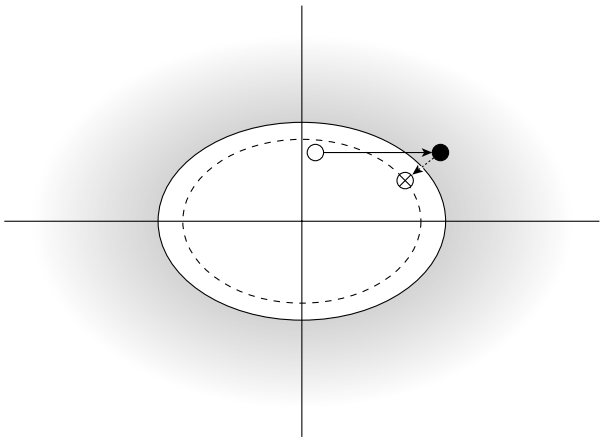

Fig. 1. *Attack on an eigenvector watermarking system. The two axes of this figure correspond to eigenvectors of the detection matrix,* $\mathbf{G}$. *The gray area outside the ellipse is the detection region defined by* $\mathbf{G}$. *The white dot is an unwatermarked Work. Watermarks are embedded by adding an eigenvector. This is shown with the solid arrow and black dot. The attack, shown with the dotted arrow and 'X' dot, moves the watermarked Work to the closest point on an ellipse (dashed line) within the unwatermarked side of the detection boundary. Note that the attacked Work is closer to the original than is the watermarked Work.*

More sophisticated attacks, which may prove effective against recent versions of the algorithm, would search for the nearest power spectrum that causes the detection test to fail. The spectrum found might or might not be a flattened version of the spectrum obtain from the watermarked Work.

2.3. Generalization: closest-point attacks

The attacks described above can be thought of as examples of a very broad class of *closest-point* attacks. In such attacks, the adversary finds the unwatermarked Work that is closest (or nearly closest) to a given watermarked Work. With knowledge of the detector alone, this is usually possible to do either analytically or by means of an efficient search.

In most cases, the unwatermarked Work closest to a given watermarked Work is close enough to the original that the adversary will be satisfied with its fidelity. For example, in the attack shown in Figure 1, the attacked Work is actually *closer* to the original than is the watermarked Work. Ensuring that the attacked Work will be farther from the original – far enough that the adversary is not satisfied with the result – would probably require a detection region with a more complicated shape.

2.4. Answer: asymmetric-key watermarking by itself is insufficient

Upon examining attacks against the proposed watermarking systems described above, it is clear that asymmetric-key watermarking alone is insufficient for secure public watermarking. By analyzing detectors, adversaries can devise methods of finding the closest unwatermarked Works to given watermarked Works, thus successfully removing the watermarks. These attacks do not require any knowledge of the embedding algorithm used, so keeping

embedder keys secret does not prevent them.

These closest-point attacks can only be prevented by using sufficiently sophisticated detectors, irrespective of the methods used for embedding. The detection region must have at least one of two properties:

1. The closest unwatermarked point to a given watermarked Work should have a high probability of being unacceptably far from the original,
   or
2. given a point inside the detection region, it must be computationally infeasible to find a nearby point outside the detection region.

Neither of these properties is ensured by key asymmetry.

3. IS KEY ASYMMETRY NECESSARY FOR SECURITY?

The next question is whether key asymmetry, while insufficient by itself, is *necessary* for secure public watermarking applications. That is, must adversaries be prevented from obtaining the embedding keys?

If we could find a symmetric-key watermarking system that were secure for public watermarking, then this would prove that key asymmetry is not necessary. Such a system could be termed an *irreversible watermark*, since the embedding process cannot be reversed, even by someone with full access to the keys. Unfortunately, as far as this author knows, no such system exists[1]. Thus, we must satisfy ourselves with speculations about how such a system might work. If it appears plausible that these speculations might some day be realized, then it is plausible that secure public watermarking does not require key asymmetry.

This section begins by examining one reason that key asymmetry might be necessary to embed for detectors that are secure against closest-point attacks. It then goes on to speculate about ways to embed for these detectors without using any secret keys.

3.1. An approach in which key asymmetry is necessary

Suppose we have a detector that makes it virtually impossible to apply a closest-point attack. That is, given a point inside the detection region (a watermarked Work), it is computationally infeasible to find a nearby point outside the detection region (a successfully attacked Work). This detector, then, satisfies the second of the two alternative requirements listed at the end of the preceding section.

The task of embedding a watermark is essentially the reverse of performing a closest-point attack. Given a point *outside* the detection region (an original, unwatermarked Work) the embedder must find a nearby point *inside* the detection region. Thus, if the closest-point attack is virtually impossible for our detector, it seems likely that watermark embedding should be virtually impossible as well.

One way to get around this might be to provide the embedder with some additional information that makes embedding feasible. If this information also makes closest-point attacks feasible, then it must be kept secret, and secure key asymmetry is required.

[1] It was suggested, in [2], that the technique known as *dither index modulation* might be secure for public watermarking, even though it uses symmetric keys. The argument, essentially, was that a closest-point attack is very likely to produce a Work farther from the original. Unfortunately, it was reported in [4] that the decrease in fidelity resulting from such attacks is quite small in practice.

In this approach, the embedding and detection keys might constitute asymmetric *descriptions* of the same detection region [3]. Using one description – the embedding key – it is easy to find a point on the opposite side of the detection boundary that is near a given point. Using the other description – the detection key – it is easy to determine whether a point is inside or outside the detection region, but infeasible to find nearby points on the other side of the boundary.

3.2. The possibility of irreversible watermarking

The need for asymmetric descriptions of the detection region is based on the assumption that, if the detector is secure against closest-point attacks, then it is difficult to embed watermarks without additional, secret information. But there are at least two ways that this assumption might not be true. First, it might be possible to define a detector for which it *is* feasible to perform optimal embedding, without any additional information, but it is still infeasible to apply a closest-point attack. Second, it might be possible to define a system in which the embedder does need additional information, but this information does not enable adversaries to perform closest-point attacks, and thus need not be kept *secret*. Either of these approaches would result in an irreversible watermark.

To speculate about how the first type of irreversible watermark might be implemented, imagine we define a function, $f(\mathbf{c}, k)$, that yields a real number, z, for each Work, $\mathbf{c}$, and key, k, and has the following properties:

1. $f(\cdot, \cdot)$ is smooth over media space.
2. For a given value of z, and a given Work, $\mathbf{c}$, it is impossible to analytically find a nearby Work, $\mathbf{c_x}$, such that $z = f(\mathbf{c_x}, k)$. However, since the function is smooth, it *is* possible to use gradient ascent and descent to find nearby local maxima and minima.
3. For the vast majority of actual Works, $f(\mathbf{c}, k)$ is below some threshold, τ.
4. All local maxima of $f(\cdot, \cdot)$ are above τ.
5. Most local minima of $f(\cdot, \cdot)$ are also above τ, but some are below τ.

Figure 2 illustrates what such a function might look like when computed along a one-dimensional line through media space.

We build a detector that reports presence of the watermark in Work $\mathbf{c}$ if $f(\mathbf{c}, k) > \tau$. We can embed for this detector by using gradient ascent to find a local maximum that is near an original cover Work. However, if adversaries attempt to use gradient descent to find nearby points below the threshold, they are likely to get trapped in local minima that are above the threshold. Thus, even with complete knowledge of the embedding and detecting algorithms and key, it would not be feasible to remove the watermark.

To speculate about how the second type of irreversible watermark might be implemented, imagine that we describe a simple *embedding region* that provably lies entirely inside a given detection region. Imagine further that, even after examining this embedding region, it is not possible to apply a closest-point attack to our detection region. This is illustrated conceptually in Figure 3. Here, the embedder requires some additional information, namely the description of the embedding region, but knowing this information would not help an adversary remove the watermark. This idea is discussed in [3].

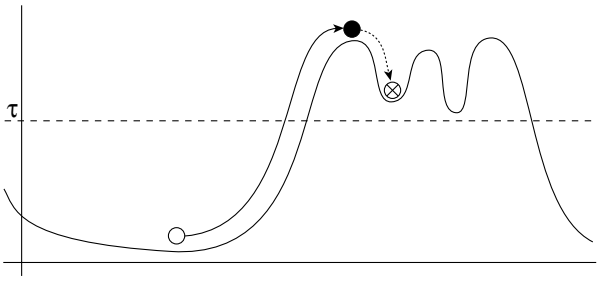

Fig. 2. *One possible method of implementing an irreversible watermark. The curve shows the value (vertical axis) of a hypothetical function, $f(\mathbf{c}, k)$, evaluated for Works (horizontal axis) that lie along some path through media space. The white dot indicates the value of the function for some unwatermarked Work. An embedder uses gradient ascent to find a nearby local maximum (black dot). Since all local maxima are above the detection threshold, τ, (dashed line), this corresponds to a watermarked Work. An adversary might try to use gradient descent to find a nearby point below the threshold, but there is a chance of going the wrong direction and finding a local minimum above the threshold instead ('X' dot). In this 2d figure, the chance of going the wrong way is 50/50, but in higher dimensions it can be made very likely, so the attack would be most likely to fail.*

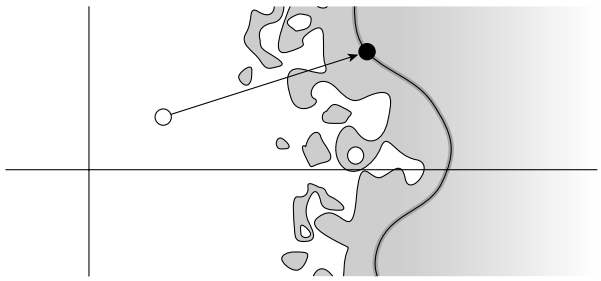

Fig. 3. *A second possible method of implementing an irreversible watermark. The gray area indicates a hypothetical detection region for which both closest-point attacks* and *optimal embedding are infeasible. The dark curve indicates an embedding region, specified to the embedder, that lies entirely within the detection region. It is easy for the embedder to move any unwatermarked Work (white dot) to a nearby point (black dot) in this embedding region.*

3.3. Answer: asymmetric-key watermarking might not be necessary

To assume that asymmetric keys are necessary for public-key watermarking amounts to assuming two things: a) if a detector does not give adversaries useful information for attacks, then the embedder must employ some information not found in the detector, and b) this additional embedder information *is* useful for adversaries constructing attacks. However, it is possible to at least imagine systems in which one or the other of these assumptions is incorrect. Until such imaginary systems are proven to be unrealizable, we must recognize that asymmetric-key watermarking might not be necessary.

4. CONCLUSIONS

This paper began by asking whether asymmetric-key watermarking is sufficient and necessary for secure public watermarking applications. The answer, both to the question of sufficiency and the question of necessity, is basically no. Asymmetric-key watermarking, by itself, is demonstrably insufficient, because there exist systems that prevent adversaries from determining embedding keys, but do not prevent them from removing watermarks. Asymmetric-key watermarking might not be necessary, because it is possible to imagine *irreversible* watermarking systems, in which even complete knowledge of the embedder would not help an adversary remove watermarks.

This is not to say that asymmetric-key watermarking would not be useful. Rather, it suggests that secure key-asymmetry might not be the most important property of a watermarking system. Our primary concern is whether adversaries can deduce effective attacks by studying a detector – not whether they can deduce how watermarks were embedded.

Perhaps the best approach to designing a system for secure public watermarking would be to first focus on defining a secure detector. Once such a detector is defined, we can turn to the question of how to embed watermarks. Whether asymmetric-key watermarking is important depends on whether the embedder needs additional information, and whether knowing that information would allow an adversary to remove the watermark.

5. REFERENCES

[1] J. A. Bloom, I. J. Cox, T. Kalker, J-P Linnartz, M. L. Miller, and B. Traw. Copy protection for DVD video. *Proc. IEEE*, 87(7):1267–1276, 1999.

[2] B. Chen and G. W. Wornell. Dither modulation: a new approach to digital watermarking and information embedding. In *Security and Watermarking of Multimedia Contents*, volume SPIE-3657, 1999.

[3] I. J. Cox, M. L. Miller, and J. A. Bloom. *Digital Watermarking*. Morgan Kaufmann, 2001.

[4] J. Eggers, J. Su, and B. Girod. Asymmetric watermarking schemes, 2000.

[5] Joachim J. Eggers, Jonathan K. Su, and Bernd Girod. Public key watermarking by eigenvectors of linear transforms. In *EUSIPCO*, Tampere, Finland, Sept. 2000.

[6] T. Furon and P. Duhamel. An asymmetric public detection watermarking technique. In *Proc. of the 3rd Int. Information Hiding Workshop*, pages 88–100, Dresden, Germany, Sept. 1999.

[7] T. Furon and P. Duhamel. Robustness of an asymmetric watermarking technique. *IEEE Int. Conference on Image Processing*, 3:21–24, 2000.

[8] Fabian A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Attacks on copyright marking systems. In *Workshop on Information Hiding, Portland, OR*, 15-17 April, 1998.

[9] J. Picard and A. Robert. Neural networks functions for public key watermarking. In *Proc. of the 4th Int. Information Hiding Workshop*, pages 142–156, Pittsburgh, PA, April 2001.

[10] SDMI Portable Device Specification – Part 1, Version 1.0. Technical Report pdwg99070802, Secure Digital Music Initiative, 1999.

[11] R. G. van Schyndel, A. Z. Tirkel, and I. D. Svalbe. Key independent watermark detection. In *Proc. International Conference on Multimedia Computing and Systems*, pages 580–585, Florence, Italy, 1999. IEEE.