Protocols for Watermark Verification

K. Gopalakrishnan
East Carolina University

Nasir Memon
Polytechnic University

Poorvi L. Vora
Hewlett-Packard

In current digital watermarking schemes used to deter piracy of multimedia content, the owner typically reveals the watermark in the process of establishing piracy. Once revealed, a watermark can be removed. We eliminate this limitation by using cryptographic protocols to demonstrate the presence of a watermark without revealing it.

Consider an application where multimedia content is electronically distributed over a network. To discourage unauthorized duplication and distribution, the content owner can embed a unique watermark (or a fingerprint) in each distributed copy of the data. If the owner later finds unauthorized copies of the data, then the origin of the copy and the identity of the erring buyer could be determined by retrieving the unique watermark corresponding to each buyer. These schemes are sometimes called copy deterrence watermarking schemes or digital fingerprinting schemes. We'll focus our attention on digital image watermarking, although the same problems exist for other multimedia data, such as video or audio.

A *watermark* is a signal added to the digital image that can later be extracted or detected to make an assertion about the image.[1] Two types of watermarks exist: visible and invisible. Visible watermarks typically contain conspicuously visible messages or company logos indicating the ownership of the image. Invisible watermarks, on the other hand, are unobtrusive modifications to the image and the invisibly watermarked image visually appears similar to the original. Users can determine the existence of an invisible watermark only by using an appropriate watermark extraction or detection algorithm. Companies generally prefer invisible watermarks as their unobtrusiveness makes them more desirable.

We can also classify watermarking techniques as *fragile* and *robust*. Any image processing procedure will corrupt a fragile watermark, whereas a robust watermark can resist common image manipulation procedures (such as rotation, reflection, scaling, cropping, smoothing, contrast or brightness adjustment, or lossy compression). Clearly, a watermark used for the purpose of copy deterrence must be robust.

Yet another classification of watermarking techniques is into oblivious and nonoblivious schemes. A nonoblivious scheme requires an original or reference image in the watermark detection procedure. On the other hand, an oblivious scheme doesn't require the use of an original or reference image. Thus, oblivious schemes are attractive for many applications.

In copy deterrence watermarking schemes, the watermarks used are generally invisible, robust, and oblivious. Recall that we deter copying by inserting a unique watermark into each copy of the image sold (which we can use to trace unauthorized copies to the erring buyer). In such a scenario, to indict the erring buyer, the seller has to demonstrate the presence of the unique watermark on an unauthorized copy of the image and provide evidence that binds the specific watermark to the buyer. To establish that the watermark was bound to the buyer, the seller must obtain a certificate at the time of sale which, say, is in the form of the encryption of the watermark with the seller's public key, details of the terms of the sale, and the identity of the buyer, all time stamped and signed by a trusted authority. To establish that a watermark exists in the unauthorized copy, we generally assume in the literature that the seller reveals the embedded watermark to the buyer or trusted third party. Once a company reveals the watermark, the buyer or trusted third party could subsequently remove it and resell multiple copies of the image with complete impunity. While this limitation appears inherent, we can actually eliminate it by using appropriate tools from cryptography. Our approach demonstrates the presence of a watermark in an image without revealing the watermark to the other party. This prevents the adversary from subsequently removing the watermark.

Proposed protocol for watermark verification

The watermark verification protocol we propose works with linear and additive watermarking

1070-986X/01/$10.00 © 2001 IEEE

techniques where we detect watermarks by correlating them. However, for ease of exposition, we present it in terms of the spread-spectrum watermarking technique proposed by Cox et al.,[2] which is remarkably robust against malicious attacks aimed at its removal. Before we present our protocol, we first briefly review this technique.

Cox et al.[2] embed a set of independent real numbers $W = \{w_1, w_2, \ldots w_n\}$ drawn from a zero mean, variance 1, Gaussian distribution into the n largest discrete cosine transform (DCT) AC coefficients of an image. Results reported using the largest 1,000 AC coefficients show that the technique is remarkably robust against various image-processing operations, including rescanning after printing.

Specifically, they take the 2D DCT of an image X and insert the watermark W into the largest n AC coefficients $\{x_1, x_2, \ldots, x_n\}$ by a suitable insertion formula to yield modified coefficients $\{x'_1, x'_2, \ldots, x'_n\}$. For example, the insertion formula used could be

$$x'_i = x_i (1 + \alpha w_i)$$

where α is a small constant. Cox et al.[2] then take an inverse 2D DCT, yielding the watermarked image X'. To determine if a given image Y contains the watermark W, the decoder first takes the 2D DCT of the image and extracts the largest N DCT coefficients $Y = \{y_1, y_2, \ldots y_n\}$. They take the confidence measure on the presence of the watermark W in Y to be the correlation between W and Y. Note that this version of their technique is invisible, robust, and oblivious.

Under our scenario of copy deterrence watermarking schemes using the spread-spectrum technique, the seller or distributor inserts a unique watermark that's distinct for each buyer into the image before distributing it to the buyer. The sellers also encrypt this watermark W using their public key of the well-known RSA public-key cryptosystem and obtains a time-stamped digital certificate binding $E(W)$ to the specific buyer. Let's say that later the seller encounters an image Y and contends that it's a pirated copy originating from a specific buyer. To establish this, the seller must prove that the answer to the following watermarking decision problem is a resounding yes:

- *Problem instance*. The digital image Y in dispute, seller's public key and $E(W)$, the encryption (using seller's public key) of a spread-spectrum watermark.
- *Question*. Is the watermark W present in the digital image Y?

Note that the seller can solve the watermarking decision problem by disclosing the watermark W and the digital certificate that binds $E(W)$ to the buyer. The verifier can check the certificate, that $E(W)$ is indeed the encryption of W, and that W is present in Y by using the watermark detection procedure of a spread-spectrum technique in the standard manner. But then the verifier knows the watermark W, can remove it from the image Y, and can resell multiple copies of it with complete impunity. So the seller has lost the power of demonstrating that a disputed copy is a pirated copy the moment he discloses the unique watermark. However, there's no reason why the seller should prove that the answer to the watermarking decision problem is yes in the above manner. It's possible to prove that the answer is yes without revealing the watermark by using tools from cryptography.

Specifically, the seller can use the following protocol to prove that the answer to the watermarking decision problem is yes without revealing the watermark itself. Here's the protocol:

Input. The digital image Y in dispute, seller's public key and $E(W)$, the encryption (using seller's public key) of a spread-spectrum watermark.

1. Repeat the following steps k times.
2. The seller chooses a random number r and uses it to generate a sequence ε in a one-way manner. The seller then adds ε to Y to get an image $Y' = Y + \varepsilon$. The seller encrypts Y' and sends $E(Y')$ to the verifier.
3. The verifier chooses a random integer $j = 1$ or 2 and sends it to the seller.
4. If $j = 1$, the seller reveals Y' and r. The verifier encrypts Y' and checks that it's the same as $E(Y')$ that the seller previously sent. The verifier generates ε from r, adds it to Y, and checkes that it's the same as Y'. If $j = 2$, the seller demonstrates that Y' and W correlate.
5. The verifier accepts the seller's proof if the computation of step 4 is verified in each of the k rounds.

Although most of the protocol is self-explanatory,

we should clarify step 4. When $j = 1$, the seller reveals Y' and r. The two checks performed by the verifier convinces him or her that $E(Y')$ sent by the seller previously was created by the seller as dictated by the protocol and not in an arbitrary manner. In particular, this step ensures that the sequence ε added to Y is random and doesn't correlate with W by design.

To understand step 4, in cases where $j = 2$, consider a sequence $a = (a_1, a_2, \ldots a_n)$ and another sequence $b = (b_1, b_2, \ldots b_n)$. Essentially the value of the inner product $a_1b_1 + a_2b_2 + \ldots + a_nb_n$ determines whether these two sequences correlate. If $E(a)$ and $E(b)$ are available to the verifier, then the seller could disclose the sequence $(a_1b_1, a_2b_2, \ldots, a_nb_n)$ to the verifier. The verifier can simply add the elements of this sequence and thus determine whether the sequences a and b correlate. The verifier can be confident that the sequence given by the seller isn't arbitrary by checking that

$$E(a_ib_i) = E(a_i)\,E(b_i)$$

for $i = 1, 2, \ldots n$. This checking is possible as the verifier is in possession of both $E(a)$ and $E(b)$, has been given the plaintext values of a_ib_i by the seller, and the RSA cryptosystem has the multiplicative homomorphic property.

If $j = 2$, the seller discloses the sequence $(y'_1w_1, y'_2w_2, \ldots y'_nw_n)$ to the verifier. The verifier can then check the given sequence's legitimacy—using the seller's public key—since the verifier is already in possession of both $E(Y')$ and $E(W)$. The verifier can then add up the elements of this sequence and use the result to check that Y' and W correlate.

Because the verifier knows that Y' is derived from Y by insertion of ε, Y' correlates with W, and the random sequence ε doesn't correlate with W, the verifier can conclude that Y must correlate with W. Therefore, the pirated copy must originate from the specific buyer.

Note that in each protocol round, the seller only proves one of two statements, namely that Y' correlates with W or that Y' is derived from Y by insertion of ε. However, since the seller doesn't know which one of these two statements he or she will be asked to prove before committing to $E(Y')$, he or she can't choose ε by malicious design.

Because ε is selected at random by the seller, it's reasonable to assume it's orthogonal to W and that its inner product or correlation with W is small. Hence, it's possible to use the correlation of Y' and W to estimate the correlation of Y and W. The difference between the two correlations is $e = \sum_i \varepsilon_i W_i$, where the sum is over the DCT coefficients of the image. The expected value of e is zero for zero-mean W. The variance of e—large values that decrease the robustness of the watermarking procedure—increases with an increase in the variance of ε.

Larger values of ε will provide more protection to the value of the watermark because the seller reveals y'_i. They will, however, also decrease the scheme's robustness. Hence, watermarking scheme's robustness, and the number of distinct watermarks that can be embedded in an image, will be traded off with the degree to which the seller prevents knowledge about the watermark being revealed during the detection procedure.

Zero knowledge proofs and digital watermarks

The protocol presented in the previous section closely follows a well-known tool in cryptography called zero knowledge proofs. A nonmathematical introduction to zero knowledge proofs is provided in Quisquater et al.[3] Zero knowledge proof systems is an active area in cryptography and a formal and detailed introduction to it can be found in Stinson's[4] and Menezes et al.'s[5] texts. Informally, a zero knowledge proof system lets one person, Peggy, convince another person, Vic, of some fact without revealing any information about the proof. At the end of the protocol, Vic is completely convinced of the same fact, but doesn't gain any additional knowledge whatsoever.

In Kinoshita's work,[6] he attempted to use the zero knowledge interactive proofs to assert ownership rights on an image. He used the zero knowledge interactive proof for the graph isomorphism problem presented in Goldreich et al.[7]

Kinoshita's scheme works in the image's spatial domain. Essentially, he generates a graph with n nodes, called the region graph G_r, from the most significant bits of the image's pixels in a fixed manner. He then applies a permutation σ on n points to the region graph G_r to obtain an isomorphic graph called the concealed graph G_c. He then conceals the G_c in the least significant bits of the pixels. To assert ownership rights over the image, Kinoshita suggests that the owner could extract the region graph from the most significant bits and the concealed graph from the least significant bits; then the owner could demonstrate that these two graphs are isomorphic to each other without revealing the permutation σ using the zero knowledge interactive proof.

While the zero knowledge interactive proof for the graph isomorphism problem presented in Goldreich et al.[7] is perfect, the way Kinoshita uses it here is fundamentally flawed. The first problem is that this watermarking scheme isn't robust. The concealed graph is encoded into the least significant bits of the pixels. Adversaries can always modify the least significant bits, thus preventing the real owner from proving his ownership of the image. More importantly, the adversary can construct the region graph G_r from the most significant bits of the pixels in exactly the same manner as the owner. The adversary can then apply a permutation p, known only to himself or herself, to the region graph G_r and obtain an isomorphic graph G'_c. He or she can then embed G'_c into the least significant bits of the image and can claim that the image actually belongs to himself or herself. Moreover, the adversary can prove this by using the same zero knowledge interactive proof for graph isomorphism.

The previous example shows that one must be careful in applying the subtle concept of zero knowledge interactive proofs to practical problems. More recently, Craver[8] presents two attempts at developing protocols for zero knowledge watermark detection. Craver bases the first one on the Pitas scheme,[9] which works in the spatial domain. Without going into details, this protocol relies on some scramblings (permutations) of images, where the scrambling itself must be kept secret, even though the verifier knows the scrambled image (and the original image). As uncommon intensity values in the original image are mapped to uncommon values in the scrambled image, this leaks partial information about the scrambling, so it isn't a true zero knowledge protocol. The same problem also exists in the second attempt Craver[8] describes.

The question that arises is whether the protocol we present is a perfect zero knowledge protocol. We don't know whether this is true because we're unable to prove that it leaks no knowledge; but the protocols Craver presents[8] do leak partial knowledge.

It isn't too difficult to see that the watermarking decision problem belongs to the complexity class Nondeterministic Polynomial (NP) time. If the watermark W is present in the image Y, we can nondeterministically guess the watermark W, check that W is present in Y, and that the encryption of W is indeed $E(W)$ in polynomial time. Hence, a computational zero knowledge proof for the watermarking decision problem exists, as all problems in NP have a computational zero knowledge proof (see Goldreich et al.[7]). Note that computational zero knowledge proofs are weaker forms of perfect zero knowledge proofs. In the case of perfect zero knowledge proofs, the verifier shouldn't gain any knowledge even if the verifier has access to unbounded computational resources. On the other hand, in the case of computational zero knowledge proofs, a verifier with bounded computational resources (for example, within polynomial time) shouldn't gain any knowledge by participating in the protocol. There's theoretical evidence that perfect zero knowledge proofs don't exist for NP-complete problems. However, we don't know whether the watermarking decision problem is NP-complete. If it is, it's probably futile trying to develop a perfect zero knowledge proof for the problem.

Concluding remarks

We developed a novel way of demonstrating the presence of a watermark in an image without revealing the watermark that could lead to the possibility of adversaries removing the watermark and reselling multiple copies of the image with impunity.

For the sake of brevity, we focused on the problem of demonstrating the presence of a watermark in an image without revealing it. Memon and Wong[10] discuss some other aspects of copy deterrence watermarking schemes—such as preventing the ability of a malicious seller to frame the buyer. Indeed, the protocol presented here could be coupled with the buyer–seller protocol presented in Memon and Wong[10] to form a more comprehensive solution to the problem of copy deterrence.

In addition to copy deterrence applications, the fundamental problem we point to also applies to watermarking for ownership assertion. Current techniques assume that the watermark must be revealed to assert ownership. A similar protocol could potentially address this problem. **MM**

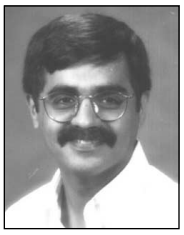
Acknowlegments

K. Gopalakrishnan was supported in part by the ECU Faculty Senate through a Research/Creative Activity Grant. Nasir Memon was supported in part by US NSF Grant NCR-9996145 and the Air Force Office of Scientific Research (AFOSR) F49620-01-1-0243. An initial version of this paper was presented in the 1999 ACM Multimedia Security Workshop, Orlando, Florida.

References

1. N. Memon and P.-W. Wong, "Protecting Digital Media Content: Watermarks for Copyrighting and Authentication," *Comm. ACM*, July 1998, pp. 35-42.
2. I.J. Cox et al., "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. Image Processing*, vol. 6, no. 12, Dec. 1997, pp. 1673-1687.
3. J.J. Quisquater, L. Guillo, and T. Berson, "How to Explain Zero Knowledge Protocols to Your Children," *Advances in Cryptology* (CRYPTO 89), Lecture Notes in Computer Science, Springer Verlag, Berlin, vol. 435, 1989, pp. 628-631.
4. D.R. Stinson, *Cryptography—Theory and Practice*, CRC Press, Boca Raton, Fla., 1995.
5. A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, Fla., 1997.
6. H. Kinoshita, "An Image Digital Signature System with ZKIP for the Graph Isomorphism Problem," *Proc. IEEE Int'l Conf. Image Processing* (ICIP 96), IEEE Press, Piscataway, N.J., vol. 3, 1996, pp. 247-250.
7. O. Goldreich, S. Micali, and A. Wigderson, "Proofs That Yield Nothing but Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems," *J. ACM*, vol. 38, no. 1, 1991, pp. 691-729.
8. S. Craver, "Zero Knowledge Watermark Detection," *Information Hiding* (IH 99), Lecture Notes in Computer Science, Springer Verlag, Berlin, vol. 1768, 1999, pp. 101-116.
9. I. Pitas, "A Method for Signature Casting on Digital Images," *Proc. IEEE Int'l Conf. Image Processing* (ICIP 96), IEEE Press, Piscataway, N.J., vol. 3, 1996, pp. 215-218.
10. N. Memon and P.-W. Wong, "A Buyer–Seller Watermarking Protocol," *Proc. IEEE Second Workshop on Multimedia Signal Processing* (MMSP 98), IEEE Press, Piscataway, N.J., 1998.

K. Gopalakrishnan is currently working as an assistant professor of computer science at East Carolina University. He received his PhD in computer science from the University of Nebraska in 1994. His primary research interests are in the areas of cryptography and information security.



Nasir Memon is an associate professor in the computer science department at the Polytechnic University, New York. He received his BE in chemical engineering and MSc in mathematics from the Birla Institute of Technology, Pilani, India, in 1981 and 1982, respectively. He received his MS and PhD in computer science from the University of Nebraska in 1989 and 1992, respectively. His research interests include data compression, data encryption, image processing, multimedia data security, and multimedia communication and computing. He's currently an associate editor for *IEEE Transactions on Image Processing*.

Poorvi L. Vora works at Hewlett-Packard Laboratories in Corvallis, Oregon. She received a BTech degree in electrical engineering from the Indian Institute of Technology, Bombay, in 1982; an MS in electrical engineering from North Carolina State University in 1988; an MS in mathematics from Cornell University in 1990; and a PhD in computer engineering from North Carolina State University in 1993. Her current interests are in the applications of cryptography to problems in the fields of privacy protection and digital asset commerce.

Readers may contact Memon at the Computer Science Department, Polytechnic University, Brooklyn, NY 11201, email memon@poly.edu.