CERIAS Tech Report 2003-09

WATERMARKING WITH QUADRATIC RESIDUES

by Mikhail J. Atallah and Samuel S. Wagstaff, Jr.

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907

Watermarking with quadratic residues

Mikhail J. Atallah*, Samuel S. Wagstaff, Jr.†

Computer Operations, Audit, and Security Technology Laboratory
and Department of Computer Sciences, Purdue University
West Lafayette, IN 47907-1398 USA

ABSTRACT

We describe novel methods of watermarking data using quadratic residues and random numbers. Our methods are fast, generic and improve the security of the watermark in most known watermarking techniques.

Keywords: Watermarking, steganography, quadratic residues.

1. INTRODUCTION

Watermarking is a technique for adding a message, perhaps secret, to data to identify the owner or originator, or to detect tampering with the data. Methods for watermarking image data are well known. See, for example, Tirkel et al.,[1] Swanson et al.,[2] Wolfgang and Delp,[3] and Cox et al.[4]

We propose a scheme that can improve the security of the watermark in most known watermarking techniques. Further, our methods are faster and more robust than most known watermarking schemes. Here is the general idea of our scheme: Let $\mathcal{W}$ be any watermarking method that encodes a 0 or 1, a bit of the watermark message, in a data item x according to some rules. We make no assumptions about which portions of x can be modified by $\mathcal{W}$, and no assumptions about the nature of x. Therefore, x could be a physical measurement, or color or brightness data in a image, or an entry in the Fourier or Discrete Cosine Transform of an image. The portion of x which $\mathcal{W}$ could modify need not be its least significant bit. (Indeed, the recent frequency domain methods actually deliberately store the watermark by substantially modifying some high energy components of the spectrum of the image. Our scheme improves these methods, too.)

The basic idea is illustrated first, in Section 3, for the simple case of watermarking numerical data by encrypting the message into the low order bits of floating point numbers. These numbers, interpreted as integers, are modified slightly to force them to be quadratic residues or nonresidues modulo a secret prime, according to the parity of the next bit of the watermark message. Because of the nearly uniform distribution of quadratic residues and nonresidues, usually little change in the number is needed. By adding a simple random number generator and changing the way the watermark bit is encoded, we get an algorithm which damages at most one bit of each number.

In Section 4 we explain how the methods can be applied to improve any watermarking technique in which the watermark is inserted into a list of numbers one bit per number. Part of the number in which the next watermark bit is to be stored is combined with the next random number. If this combination is a quadratic residue, then the number is changed one way according to the watermark bit. But if the combination is a quadratic nonresidue, the number is changed the other way. The secret prime and random numbers make it harder to cryptanalyze the watermark. Even if the cryptanalyst knew where and how the watermark was placed in the data, he could not read it without knowing the secret prime. However, he could still erase it. This is why many recent image watermarking algorithms[5] keep secret which entries of the Discrete Cosine Transform of the image contain the watermark. This secret makes it impossible to remove the watermark without doing the removal operation on all of the transform's entries and thus making the data useless in the process. This is because the change needed to watermark one high-energy entry in the transform is substantial enough that, although of no serious consequence to image quality if done on a small number of such entries, it would make the image useless if done to all of the significant entries of its transform.

With our techniques the original data (as well as the watermarked data) are needed to read the watermark if and only if they would be needed to read it by the method $\mathcal{W}$ that we are improving. Our methods are robust with respect to truncation of the data in that one may read every bit of the watermark which survives truncation.

*Email: mja@cs.purdue.edu
†Email: ssw@cs.purdue.edu

2. PRELIMINARIES

This section introduces notation and terminology, and reviews basic facts that will be used in later sections.

We shall illustrate the basic idea in the simple situation where we are given a list of floating point numbers $d_1, d_2, \ldots$ and the (insecure) watermarking method $\mathcal{W}$ which consists of storing the watermark message in the least significant bit of each d_i. This could occur when each d_i comes from, for example, physical measurements made with a certain level of imprecision, such that the imprecision introduced by $\mathcal{W}$ is within the range of measurement errors, or it could be that the d_i are the secret entries of the transform of an image that we mentioned in the previous section. The security of any such scheme can be improved as described in Section 3.

Let the watermark message be the bit string $w_1, w_2, \ldots$, so that $w_i \in \{0, 1\}$ for every i. (There must be at least as many floating point numbers as bits in the watermark. In most applications there will be many more floating point numbers than watermark bits and the watermark is then repeated over and over again.)

Definitions. Let p be an odd prime number. An integer n is a *quadratic residue* modulo p if p does not divide n and there exists an integer x for which $x^2 \equiv n \bmod p$. An integer n is a *quadratic nonresidue* modulo p if p does not divide n and there does not exist an integer x for which $x^2 \equiv n \bmod p$. The *Legendre symbol* (n/p) is $+1$ if n is a quadratic residue modulo p and -1 if n is a quadratic nonresidue modulo p. Whether n is a quadratic residue or nonresidue modulo p is called the *quadratic character* of n modulo p.

Using Gauss's quadratic reciprocity law, it is easy to determine the quadratic character of n modulo p in time $\mathrm{O}((\log n)(\log p))$. See Theorem 5.9.3 of Bach and Shallit.[6] The problem is just about as easy as computing the greatest common divisor of two numbers as large as n and p by the Euclidean algorithm.

Quadratic residues have many applications in Computer Science. For example, they have been used by Maurer[7] and Radke[8] in constructing efficient hashing functions.

We use $x \vee y$, $x \wedge y$ and $x \oplus y$ to mean the bitwise logical `or`, `and` and `exclusive or`, respectively, of the binary numbers x and y.

3. EXAMPLES ILLUSTRATING THE IDEA

This section illustrates, by example, the algorithm. The next section will state abstractly the general version of the approach.

Since the number 0 often has special meaning, floating point zeros should not participate in the watermarking. We assume they are just copied from the old list to the new list unchanged and that they are ignored when the watermark is read. From now on we assume that our given list of floating point numbers contains no zeros. In all floating point representations of which we are aware the number zero is represented by a word with all zero bits.

Let p be a secret prime at least 20 decimal digits long.

3.1. A preliminary version

We begin with a simple, preliminary (and flawed) version of the algorithm for embedding a watermark message in the list of non-zero floating point numbers $d_1, d_2, \ldots$. Encode the i-th bit w_i of watermark by changing d_i to a nearby number d'_i with specified quadratic character when considered to be an integer.

Let n_i be the integer whose bit representation is the same as that of the floating point number d_i. Then n_i will not be 0 because d_i is not zero. The low order bits of n_i will be the least significant bits of d_i, at least on the machines we checked. Assume that $|n_i| < p$. (If the floating point numbers have 32 or 64 bits, then a prime p of 20 decimal digits will be large enough to meet this requirement.)

We will find an integer n'_i near n_i which is a quadratic residue or nonresidue modulo p according as w_i is 0 or 1. Specifically, try the integers n_i, $n_i + 1$, $n_i - 1$, $n_i + 2$, $n_i - 2$, etc., until some $n'_i = n_i + j$ is found for which the Legendre symbol $(n'_i/p) = 1 - 2w_i$.

Finally, let d'_i be the floating point number whose bit representation is the same as that of the integer n'_i.

One can read the watermark in the sequence $d'_1, d'_2, \ldots$ using p by finding n'_i from d'_i and computing $w_i = (1 - (n'_i/p))/2$.

Because of the way floating point numbers are represented, if the difference $|n_i - n'_i|$ is small, then the percentage difference between d_i and d'_i will be small, too. Of the $p-1$ numbers between 1 and $p-1$ (or between -1 and $-p+1$), inclusive, exactly $(p-1)/2$ are quadratic residues and $(p-1)/2$ are quadratic nonresidues. One would expect that about 50% of the time n_i will already have the desired quadratic character, so that $n'_i = n_i$. In about 50% of the remaining cases, $n_i + 1$ will have the desired quadratic character. In about 50% of the remaining cases, $n'_i = n_i - 1$, etc. Experiments show that the difference $|n_i - n'_i|$ is close to the expected exponential distribution having $|n_i - n'_i| \leq k$ with probability $1 - 2^{-2k-1}$.

The worst case upper bound on $|n_i - n'_i|$ has been studied for a long time. The best that number theorists can prove is probably far from the truth. Let $a > 1/4$. Burgess[9] proved that every interval of length $O(p^a)$ must contain at least one quadratic residue and at least one quadratic nonresidue modulo p. See Norton[10] for the latest work on this subject. When starting at the beginning ($n_i = 1$) and seeking a quadratic nonresidue, one can do better because there is a way to take advantage of numbers having only small prime factors: Ankeny,[11] assuming the (unproved but widely believed) Extended Riemann Hypothesis, proved that the least positive quadratic nonresidue modulo p must be $O((\log p)^2)$.

The above method does not require storing the original data in order to retrieve the watermark, but it suffers from three drawbacks. The most serious drawback is that if $d_1 = d_2 = \cdots$, then it is possible to read the watermark even without knowing p because the watermarked version of such a sequence of equal d_i's is a sequence S containing only two distinct symbols (call them a and b), and the watermark message is one of only two possibilities (either interpreting every a in S as a 0 and every b as a 1, or vice-versa). Another drawback is the case of distinct but very small n_i's, because we would then be dealing with the quadratic character of very small numbers, and there would be many dependencies among these Legendre symbols. Finally, although we pointed out that it is quite unlikely that $|n_i - n'_i|$ is large, it is better to guarantee that it does not exceed 1. The modifications outlined in the next section take care of all three drawbacks.

3.2. A better version

We have seen that the preliminary algorithm, just presented, changes only a few low order bits of d_i. Half of the time it changes nothing and in only about one case in four does it change more than one bit of d_i.

We now describe how to write the same watermark by changing at most one bit of each d_i. Half of the time d_i will be unchanged and the other half of the time the only change to d_i will be that its least significant bit will be complemented. The one-bit algorithm consist of simply using the quadratic character of the integer represented by all of the bits of n_i *except* the rightmost bit of n_i, and modifying that rightmost bit according as the quadratic character just computed and the corresponding watermark message bit w_i.

In more detail, let m_i be n_i with its low order bit set to 0. That is, $m_i = n_i - (n_i \wedge 1) = n_i \wedge \ldots 1110_2$. Let b_i be the one-bit number $b_i = [(1 + (m_i/p))/2] \oplus w_i$. Finally, $n'_i = m_i + b_i$. We can read the watermark knowing only p and the n'_i's because m_i is just n'_i with its low order bit set to 0 and because $w_i = [(1 + (m_i/p))/2] \oplus b_i$.

So far we have solved the problem of damaging not more than one bit. The drawbacks for the cases of equal n_i's mentioned earlier, and the one of very small n_i's with dependent Legendre symbols, are both solved by introducing random numbers. Let the watermark be k bits long. Define w_i for $i > k$ by making the bits repeat: $w_{k+i} = w_i$ for $i > 0$. Use a pseudo-random number generator with p as seed to generate k random numbers $r_1, r_2, \ldots, r_k$. Define r_i for $i > k$ by making the numbers repeat: $r_{k+i} = r_i$ for $i > 0$. Then we use $(m_i + r_i)$ in place of m_i in the definition of b_i above. That is, $b_i = [(1 + ((m_i + r_i)/p))/2] \oplus w_i$ in the version with random numbers. This takes care of all three drawbacks but it introduces a synchronization problem when trying to retrieve the watermark from truncated data: Since we do not know where the truncation occurred, we must try up to k possibilities (one for each possible starting position modulo k of the truncation).

If the above is applied to a long stretch of constant data, then the resulting periodicity (of period k) in the watermarked version of that stretch reveals the length k of the watermark. This can be avoided by letting the pseudo-random number generator have a period k' different from k. Then the length of the block of constant data would have to exceed $\gcd(k, k')$ before any repetition would result. It would take $O(\gcd(k, k'))$ steps to recover from truncation. This device would hide k without impairing our ability to recover the watermark from truncated data.

Another way to hide the length k of the watermark is to pad the length by choosing some $k' > k$ and random bits $w_{i+1}, \ldots, w_{k'}$. Then store the bits $w_1, \ldots, w_{k'}$ as the watermark.

3.3. Additional practical considerations

We mentioned earlier that the watermark should be repeated many times throughout the data. This is so that it may be detected if just part of the data is pirated.

If the problem domain from which the data is drawn is such that it makes sense to sample the data and pirate only every i-th number, say, then each bit of the watermark should be repeated i times, so that it still appears in the sample data.

If one is worried about a disgruntled employee tampering with unwatermarked data by changing a few values of d_i, one could watermark the data by making each d_i' be a quadratic residue modulo a secret prime p. If tampering is suspected, one can determine which data values were changed by looking for quadratic nonresidues modulo p. About half of the changes made by someone who did not know p would produce quadratic nonresidues.

Why is our scheme better than simply enciphering the watermark before inserting it? First, our scheme is faster than most encryption algorithms because there is a very swift algorithm (Bach and Shallit[6] Theorem 5.9.3) for determining quadratic character. Another advantage of our method is that it is more robust than a block cipher because our method can recover all bits of the watermark not truncated, while, with a block cipher, one could recover only those bits from whole blocks unaffected by truncation.

3.4. Cryptanalysis

If a pirate knows or suspects the data has been watermarked, he can destroy the watermark by setting all the low order bits to 0. This vulnerability is due to the simple example we used and is not inherent to our method. Later in the paper we describe how our method can be used in conjunction with a large class of watermarking schemes, some of which do not suffer from such a vulnerability.

Suppose one knows that a watermark is present, but does not know p. Can one read the watermark? At first glance, this may seem to be possible depending on the data. For example, if the data consists of numbers rounded to three significant decimal digits (or ten significant bits), one could infer the differences between the original and watermarked data and recover the original data. But one could not determine the bits of the watermark without knowing p because p defines the quadratic character of the numbers n_i and the pseudo-random numbers mask any constant data.

What about a known plaintext attack in which the original data, the watermarked data and the watermark text are all known and no random numbers are used, and p is sought? The given data determine the value of the Legendre symbol (n_i/p) for every i. By the Quadratic Reciprocity Law (see Theorem 5.8.1 of Bach and Shallit[6]), the value of (n_i/p) restricts p to lie in only one-half of the possible arithmetic progressions modulo $4n_i$. Thus, $\mathrm{O}(\log_2 p)$ known values of (n_i/p) should provide enough information to determine p. However, the problem of manipulating the huge arithmetic progressions makes this information useless. The only known solution to the problem of finding *pseudosquares*, positive integers which are not exact squares but which are quadratic residues modulo each of the first k primes, is trial and error. Special hardware devices (sieves) have been built[12] to search for pseudosquares by trying numbers one after the other. Mathematicians[12] have searched up to about 10^{20} for such numbers using years of sieve time. Similar, but more complicated, hardware devices could be built to solve the known plaintext attack on watermarking, but they would take years to find a 20-digit prime p. Of course, the use of random numbers makes the cryptanalysis even harder.

Damgård[13] considers the following problem: Given the sequence of Legendre symbols (a/p), $(a+1/p)$, …, (b/p) (with p prime and $a < b$), find $(b+1/p)$. (Assume that $b+1$ is not the square of an integer.) While no one knows how to find this Legendre symbol without first finding p, no one has shown that it cannot be done either. He argues that the sequence of Legendre symbols (a/p), $(a+1/p)$, …, (p prime) is random (unpredictable), and shows that a sequence of Jacobi symbols (p composite) is even more random. If one believes his heuristic argument, then our watermarking scheme would become even more secure if one replaced the prime p by the product of several primes. Peralta[14] presents further evidence of the randomness of Legendre symbols by showing that their distribution is like that of independent fair coins.

4. A GENERIC DESCRIPTION

The generic description of the scheme is as follows.

Let $\mathcal{W}$ be any watermarking scheme that stores a 0 or a 1 in each of data items $x_1, x_2, \ldots, x_n$. Let $w_1, w_2, \ldots, w_k$ be the bits of the watermark message; if $k < n$ then the watermark is written repeatedly (roughly n/k times). To hide the length k of the watermark, choose some $k' > k$ (as mentioned at the end of Subsection 3.2) and choose random bits $w_{i+1}, \ldots, w_{k'}$. Define w_i for $i > k'$ by making the bits repeat: $w_{k'+i} = w_i$ for $i > 0$. Let y_i be the number representing the portions of x_i that cannot be changed by $\mathcal{W}$, and let z_i be the number representing the remaining portions of x_i (we do not even assume that the bits of y_i are contiguous in x_i).

Let $r_1, ..., r_{k'}$ be a sequence of pseudo-random numbers generated using p, a secret prime, as seed. Define r_i for $i > k'$ by making the numbers repeat: $r_{k'+i} = r_i$ for $i > 0$. An x_i is processed by our scheme as follows. We check whether $y_i + r_i$ is a quadratic residue mod p. If it is, then we modify z_i so that it stores w_i according to $\mathcal{W}$'s rules. Otherwise we modify z_i so that it stores the complement of w_i. Suppose $\mathcal{W}$ changes z_i to z_i' if $w_i = 0$ and to z_i'' if $w_i = 1$. Our scheme modifies $\mathcal{W}$ to $\mathcal{W}'$ which changes z_i to z_i' if $w_i = 0$ and $y_i + r_i$ is a quadratic residue modulo p or if $w_i = 1$ and $y_i + r_i$ is a quadratic nonresidue modulo p. $\mathcal{W}'$ changes z_i to z_i'' if $w_i = 1$ and $y_i + r_i$ is a quadratic residue modulo p or if $w_i = 0$ and $y_i + r_i$ is a quadratic nonresidue modulo p. In other words, $\mathcal{W}'$ stores w_i in the same way that $\mathcal{W}$ would store $[(1 + ((y_i + r_i)/p))/2] \oplus w_i$.

4.1. A frequency domain example

Suppose $x_1, \ldots, x_n$ are the n frequencies in the spectrum of an image which are to be used to store a watermark. See Cox and Miller[5] for this kind of watermark. Suppose $k' = n$ because the watermark is written just once. Suppose $\mathcal{W}$ decreases x_i by 10% if $w_i = 1$ and leaves x_i unchanged if $w_i = 0$. This scheme requires comparison of the original and watermarked images to read the watermark.

In our modification $\mathcal{W}'$ of $\mathcal{W}$ we would decrease z_i by 10% if $w_i = 1$ and $y_i + r_i$ is a quadratic residue modulo p or if $w_i = 0$ and $y_i + r_i$ is a quadratic nonresidue modulo p. $\mathcal{W}'$ would leave z_i unchanged if $w_i = 0$ and $y_i + r_i$ is a quadratic residue modulo p or if $w_i = 1$ and $y_i + r_i$ is a quadratic nonresidue modulo p. The watermark can be read by comparing x_i from the original image with z_i from the watermarked image, using p to generate the r_i and to evaluate the Legendre symbols $((y_i + r_i)/p)$.

4.2. A text example

Suppose $\mathcal{W}$ places a watermark in a text document by shifting a word to the right by a small amount if $w_i = 1$ and shifting it to the left if $w_i = 0$. See Low et al.[15] and Chapman and Davida[16] for this kind of watermark.

In our modification $\mathcal{W}'$ of $\mathcal{W}$ we would shift a word to the right if $w_i = 1$ and $y_i + r_i$ is a quadratic residue modulo p or if $w_i = 0$ and $y_i + r_i$ is a quadratic nonresidue modulo p. $\mathcal{W}'$ would shift the word to the left if $w_i = 0$ and $y_i + r_i$ is a quadratic residue modulo p or if $w_i = 1$ and $y_i + r_i$ is a quadratic nonresidue modulo p. The watermark can be read by comparing x_i from the original image with z_i from the watermarked image, using p to generate the r_i and to evaluate the Legendre symbols $((y_i + r_i)/p)$.

5. ACKNOWLEDGEMENTS

Portions of this work were supported by sponsors of the COAST Laboratory. We wish to thank Laura C. Feinstein for implementing the preliminary version in Section 3.1 and Tom Daniels for watermarking some images with another algorithm.

REFERENCES

1. A. Z. Tirkel, G. A. Rankin, R. M. V. Schyndel, W. J. Ho, N. R. A. Mee, and C. F. Osborne, "Electronic watermark," in *Digital Image Computing, Technology and Applications*, pp. 666–673, DICTA '93, Macquarie University, Sidney, 1993.
2. M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent robust image watermarking," in *IEEE International Conference on Image Processing*, vol. III, pp. 211–214, 1996.
3. R. B. Wolfgang and E. J. Delp, "A watermark for digital images," in *IEEE International Conference on Images Processing Lausanne, Switzerland*, pp. 219–222, September 1996.
4. I. J. Cox, J. Killian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing* **6**, pp. 1673–1687, December 1997.

5. I. J. Cox and M. L. Miller, "A review of watermarking and the importance of perceptual modeling," in *Human Vision and Electronic Imaging II*, pp. 92–99, SPIE 3016, San Jose, CA, USA, February 1997.
6. E. Bach and J. Shallit, *Algorithmic Number Theory, Volume I: Efficient Algorithms*, The MIT Press, Cambridge, Massachusetts, 1996.
7. W. D. Maurer, "An improved hash code for scatter storage," *Communications of the ACM* **11**(1), pp. 35–38, 1968.
8. C. E. Radke, "The use of quadratic residue research," *Communications of the ACM* **13**(2), pp. 103–105, 1970.
9. D. A. Burgess, "A note on the distribution of residues and non-residues," *Journal of the London Math. Society* **38**, pp. 253–256, 1963.
10. K. K. Norton, "A character-sum estimate and applications," *Acta Arithmetica* **85**(1), pp. 51–78, 1998.
11. N. C. Ankeny, "The least quadratic non residue," *Annals of Math.* **55**, pp. 65–72, 1952.
12. R. F. Lukes, C. D. Patterson, and H. C. Williams, "Some results on pseudosquares," *Mathematics of Computation* **65**, pp. 361–372, 1996.
13. I. B. Damgård, "On the randomness of Legendre and Jacobi sequences," in *Advances in Cryptology—CRYPTO '88*, S. Goldwasser, ed., no. 403 in Lecture Notes in Computer Science, pp. 163–172, Springer-Verlag, (Berlin, Heielberg, New York), 1990.
14. R. Peralta, "On the distribution of quadratic residues and nonresidues modulo a prime number," *Mathematics of Computation* **58**(197), pp. 433–440, 1992.
15. S. H. Low, N. F. Maxemchuk, J. T. Brassil, and L. O'Gorman, "Document marking and identification using both line and word shifting," in *Infocom 95*, April 1995.
16. M. Chapman and G. Davida, "Hiding the hidden: A software system for concealing ciphertext in innocuous text," in *ICICS 97*, pp. 335–345, 1996.