IMAGE WATERMARKING WITH BETTER RESILIENCE

Ramarathnam Venkatesan and Mariusz H. Jakubowski

Microsoft Research, One Microsoft Way, Redmond, WA 98073 (USA)
Email: {venkie,mariuszj}@microsoft.com

ABSTRACT

We present schemes for hardening image watermarks against adversarial jamming. Our techniques improve upon standard correlation- and spread spectrum-based methods, and withstand various image distortions and attacks intended to render watermarks unreadable. The watermarking schemes we propose explicitly locate and amplify watermark data remaining after attacks. Key ideas include embedding of watermarks in specially chosen domains, application of image enhancement, computation of responses over watermark subsets, and use of redundancy and search. For watermark detection, our schemes do not require the original image or any information about it.

1. INTRODUCTION

During the past several years, a variety of image watermarking schemes have been introduced in literature and applied in practice [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. Such schemes hide a small amount of data, typically one to 1K bits, in images for purposes such as copyright protection and image identification. Unfortunately, virtually all published watermarking schemes fail to withstand attacks aimed at rendering embedded data unreadable, and most watermarks can be easily defeated even by simple image manipulations. This reduces the practical usefulness of such watermarks, and leads to the question of whether more robust schemes can be developed.

We describe new watermarking algorithms designed to resist a wide variety of attacks [11, 12, 13]. These include both common image-processing distortions, such as lossy compression, rescaling, and cropping; and malicious attacks, including watermark-distorting software such as StirMark and unZign [14, 15]. For embedding and detecting one or more bits in images, we use a standard correlation-based approach and spread-spectrum methods [16, 17, 18], and add techniques to harden watermarks against attacks. In particular, we compute watermark responses over subsets and apply image-enhancement procedures that locate and amplify watermark data after distortions. Our detection procedures do not use any information about original non-watermarked images. Implementations and extensive tests show that for detecting watermarks attacked in various ways, these methods are substantially more effective than past approaches.

2. A BASIC ALGORITHM

We first review a typical correlation-based algorithm for watermarking, and then describe our algorithm as a variation on this. Our methods anticipate potential new attacks as well. The algorithms have many steps performed in a randomized fashion, and use a seed s for a cryptographically strong pseudorandom generator $G(s)$ [19] as the watermark secret.

Let $M = m_1, ..., m_n$ denote an input image. The values m_i typically represent the coefficients of some image transform, such as a DCT or wavelet transform, but could denote any data comprising the image. To insert a single watermark bit b into M, we do the following:

1. Using b as a part of seed selection for the generator G, spread b into a pseudorandom vector $Y = y_1, ..., y_n$, where $y_i = +d$ or $-d$ (or chosen from a symmetric distribution) and d is a real constant. The values y_i could also be chosen pseudorandomly from ranges such as $[d/2, 3d/2]$ and $[-3d/2, -d/2]$, which give better results.

2. Compute the sum $W = M + Y = w_1, ..., w_n$, where $w_i = m_i + y_i$.

The signal W is the resulting watermarked image. To detect a bit in W, we generate Y as above, and compute a correlation c, or normalized dot product, between Y and W:

$$c = (y_1 w_1 + ... + y_n w_n)/(nd^2).$$

This value is normally close to 1 if the watermark bit is present, and close to 0 otherwise. To distinguish between "0" and "1" bits, we can use two different pseudorandom sequences to compute the correlation c twice. Alternately, we can use the sequences Y and $-Y$ for embedding 0 and 1, respectively; then we compute c once and judge which bit is present based on whether c is close to 1 or to -1. To embed

more than one bit, we can embed different bits into different image regions. Also, we can use 2^k different pseudorandom sequences to embed strings of k bits; we then compute c for each sequence to determine which of the 2^k possible bit strings was embedded. In practice, k is likely to be small (less than 10).

3. OUR ENHANCEMENTS

In our tests, the above algorithm works well when no malicious attacks are applied to watermarked images. Embedded data are easily detectable in reasonably sized images (100 by 100 and above) even when images are degraded by lossy compression, such as JPEG at 25 percent quality. However, small amounts of cropping and scaling suffice to foil the simple detection process outlined above. Additionally, programs such as StirMark and unZign, which apply slight geometric and noise distortions, often degrade watermarks beyond detection even when images are not cropped or scaled. To counter these anti-watermark processes, we use several enhancements of the basic algorithm. Not all of these techniques were equally effective on each image in our experiments, which we describe later. The first two methods below aim against both current and potential new attacks, while our image-enhancement procedure can be seen as putting an attacked image in canonical form before testing.

1. **Embedding in a specially chosen domain**: As has been done in the literature [1], we insert watermark data into the DCT or wavelet transform of an entire image, and we choose a *random subset* of coefficients with the highest power in a transform region that omits both the lowest and highest frequencies. This region comprises about 10% of the entire DCT, and was experimentally chosen by tests run on a variety of images. The coefficients we choose are "important" for the image, and thus are likely to retain our embedded watermark data despite visually unimportant distortions. The random subsets help deflect averaging attacks that collect many distinct images watermarked with the same secret and use averaging to read out (and possibly reduce) components of the watermark. We note that the random subset of coefficients chosen for watermarking the original image may differ partially from the subset chosen during detection, after an image is distorted. However, this does not cause major problems.

2. **Subset computations**: We compute individual correlations over *pseudorandom subsets* of the watermark data to generate many different watermark responses $c_1, ..., c_p$. The watermark subsets can overlap, but the correlations have some formal independence properties, allowing us to use standard statistical methods for detection when the overall correlation is low. In a sense, we can "zoom in" on the watermark sections that are strongest after attacks. We may ignore subsets with the lowest correlations and emphasize remaining subsets, or apply a more involved detection process. As an example where subsets help, note that an entire correlation-based watermark can theoretically be removed by changing one or a few watermarked coefficients by appropriate amounts; however, subset correlations will ignore such coefficients, and consequently allow us to detect the watermark data that remain in other coefficients. Thus, an effective attack must try to reduce or desynchronize the watermark data across a noticeable fraction of the image DCT or wavelet coefficients.

3. **Image enhancement**: This step is not needed while embedding watermarks, but only during detection. To amplify a watermark embedded in high-power, low- to middle-frequency DCT coefficients, we apply histogram equalization to an image before we attempt watermark detection. Typically, such image enhancement increases watermark response by $25 - 200$ percent, and occasionally much more, depending on the specific image. For non-watermarked images, watermark responses are similar with and without histogram equalization. When a program such as StirMark or unZign degrades the watermark in an image, the enhancement procedure often makes the difference between failed and successful detection.

4. **Grid snapping**: To counter moderate amounts of resizing and cropping, we rescale images before watermarking, either to a standard size or to some quantized dimensions (e.g., rounded to the nearest 20 pixels), and then restore original size. For detection, we similarly rescale images; with quantized dimensions, we try several different rescalings. The parameters of this search should be chosen to balance time complexity and desired likelihood of watermark detection. In domains such as the DCT, minor cropping leaves enough watermark data intact for detection, and resizing causes slight watermark degradation, as shown by our experiments. Resizing to quantized dimensions tends to give better results than rescaling to a standard size, because of coalescing and averaging phenomena that affect DCT coefficients in significantly resized images.

5. **Multi-region embedding**: To decrease the probability of false positives, and to increase the probability of detection, we can embed separate watermarks into two or more possibly overlapping regions of the

image. During detection, we use the responses for all regions simultaneously. For non-watermarked images, accidentally high watermark values can cause false positives, but the probability is significantly decreased when multiple watermark responses for different regions are used. When watermark response is weak in one or more regions of an attacked watermarked image, responses for other regions can help in determining watermark presence.

4. SECURITY ISSUES

The secret key used in a watermark may be be vulnerable to attacks other than destroying or hiding (desynchronizing) the watermark. Watermarking many images with the same secret, or with one of a handful of different secrets, is necessary for any reasonable use of a watermarking system. However, if one uses the same secret for many images, each image may reveal some small portion of the pseudorandom sequence used. Combined over several images, such leaks may compromise the secret. To counter this, we can derive an image-dependent, secret, random binary string $K_I = Hash(K, I)$ from a master secret K; such an *image hash* (e.g., 128 bits) should be the same after an image is watermarked and/or attacked, while the hash values of visually distinct images should be uncorrelated. For such a proposal, see [20, 21]. Image hashes are also useful for searching image databases, because they reduce the problem of inexact image comparison to exact binary-string comparison. This is a direct analogue of authentication in cryptography [19].

Some of the randomizing features of our algorithms seek to minimize the assumptions on how input images are generated. We believe this is important for watermarking techniques to work well across a range of images with varying characteristics, including images traditionally difficult to watermark robustly. A combinatorial approach to formulating and analyzing the problem at hand is in progress and will appear elsewhere.

5. RESULTS

Figure 1 shows watermark responses for 100 images, each watermarked and then distorted by medium JPEG compression and the basic StirMark transformation. One correct watermark key results in high responses, further improved by image enhancement, and 19 incorrect keys generate low responses. While results vary among images, in general our techniques are effective at enabling watermark detection despite distortions, although a process more involved than threshold decision may be required.

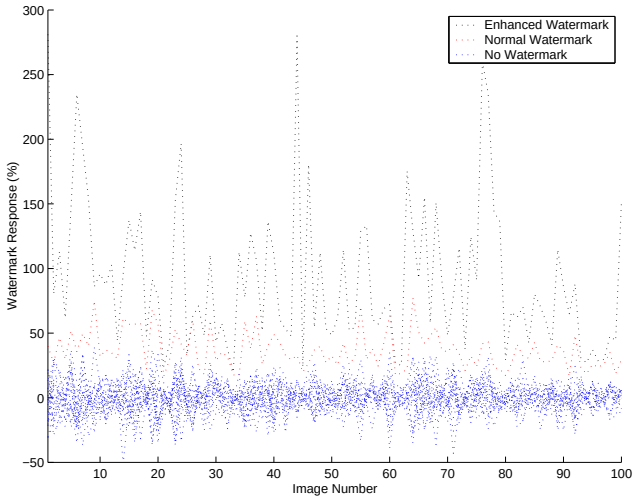

Fig. 1. Watermark responses for 100 images after JPEG and StirMark distortion. The horizontal and vertical axes denote image number (1-100) and normalized watermark response, respectively. The bottom curves show responses on incorrect keys, and the top two curves show normal and enhanced responses on the watermark keys.

6. CONCLUSION

We presented a variety of techniques for enhancing the robustness of image watermarks. Our techniques often make the difference between successful and failed detection, particularly when programs such as StirMark and unZign distort images to degrade watermarks. In this paper we did not address the problem of Web "presentation" attacks, such as the mosaic attack [14]; this is an area that requires a separate approach, because such attacks split up images, or combine several images into one. However, as shown by extensive experiments, our methods help significantly in recovering watermarks from distorted images provided separately and in their entirety.

Acknowledgements: We would like to thank Yacov Yacobi, Rico Malvar, William Koon, and Dan Boneh for many useful discussions. We thank Jim Kajiya for directing our attention to image-enhancement methods.

7. REFERENCES

[1] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "A secure, robust watermark for multimedia," in *Workshop on Information Hiding*, Univ. of Cambridge, England, May 1996.

[2] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Robust data hiding for images," in *IEEE Digital Signal Processing Workshop*, Loen, Norway, Sept. 1996.

[3] J. R. Smith and B. O. Comiskey, "Modulation and information hiding in images," in *Workshop on Information Hiding*, Univ. of Cambridge, England, May 1996.

[4] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proc. IEEE*, vol. 86, June 1998.

[5] C. Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, May 1998.

[6] J. R. Hernandez, F. Perez-Gonzalez, J. M. Rodriguez, and G. Nieto, "Performance analysis of a 2-D-multipulse amplitude modulation scheme for data hiding and watermarking of still images," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, May 1998.

[7] J. J. K. O'Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Processing*, vol. 66, no. 3, May 1998.

[8] B. Chen and G. W. Wornell, "Provably robust digital watermarking," in *Proc. SPIE: Multimedia Systems and Applications II*, Boston, MA (USA), Sept. 1999, vol. 3845.

[9] B. Chen and G. W. Wornell, "Achievable performance of digital watermarking systems," in *IEEE Int. Conf. Multimedia Computing and Systems*, Florence, Italy, June 1999.

[10] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and videos," *Proc. IEEE*, vol. 87, no. 7, July 1999.

[11] I. J. Cox and J.-P. M. G. Linnartz, "Some general methods for tampering with watermarks," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, May 1998.

[12] M. Maes, "Twin peaks: The histogram attack on fixed depth image watermarks," in *Second Workshop on Information Hiding*, Portland, OR (USA), Apr. 1998.

[13] G. C. Langelaar, R. L. Lagendijk, and J. Biemond, "Removing spatial spread spectrum watermarks by nonlinear filtering," in *9th European Signal Processing Conference*, Rhodes, Greece, Sept. 1998.

[14] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *Second Workshop on Information Hiding*, Portland, OR (USA), Apr. 1998.

[15] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding – a survey," *Proc. IEEE*, vol. 87, no. 7, July 1999.

[16] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of spread spectrum communications—A tutorial," *IEEE Trans. on Communications*, vol. 30, May 1982.

[17] R. C. Dixon, *Spread Spectrum Systems with Commercial Applications*, Wiley, New York, 1994.

[18] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[19] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[20] R. Venkatesan and M. H. Jakubowski, "Image hashing," in *DIMACS Conf. on Intellectual Property Protection*, Piscataway, NJ (USA), Apr. 2000.

[21] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *current proceedings*, 2000.