Robust Bit Extraction from Images

Jiri Fridrich
Center for Intelligent Systems, SUNY Binghamton, Binghamton, NY 13902-6000
Mission Research Corporation, 1720 Randolph Rd SE, Albuquerque, NM 87501
fridrich@binghamton.edu

Abstract

In this paper we describe an algorithm for robust extraction of bits from image blocks and a method for synthesizing a Gaussian pseudo-random sequence from those bits. The bits are extracted by thresholding projections onto random smooth patterns generated from a user-specified key. The extracted bits are further utilized to synthesize a Gaussian pseudo-random sequence that changes continuously with the image block yet depends sensitively on the secret key. The proposed technique is quite general and can be combined with majority of oblivious watermarking schemes that generate watermarks from pseudo-random sequences. We anticipate that this algorithm will find applications in many oblivious watermarking schemes including secure data embedding into videos and watermarking images for tamper detection.

1. Introduction

Digital watermark is a perceptually invisible pattern robustly embedded in a digital image. The watermark can carry useful information about the owner of the image or the recipient, or some additional information accompanying the image (image caption). Depending on the application, we may require that only a selected group of people sharing a secret password should have access to the watermark. A good example of such a scheme would be watermarking for copyright protection, fingerprinting images or videos, and watermarking for tamper detection. In this case, the watermarking algorithm is an instant of a private-key symmetric encryption scheme. In other applications, the watermark should be accessible to everybody using a public key. An example of such an application is adding subtitles or audio-tracks to videos, or supplying captions to images.

Watermarking schemes can be further divided into two groups depending on whether or not the original image is required for watermark extraction. In non-oblivious watermarking, the original image is needed for watermark extraction. Although this makes non-oblivious techniques more robust to all kinds of attack, the necessity of having the original image is clearly a disadvantage that severely limits the applicability of non-oblivious techniques. In oblivious techniques, the watermark can be extracted from the watermarked / attacked image without access to the original image. In some watermarking techniques, one must have access at least to a hash of the image (or a hash of the whole video) in order to recreate the watermark sequence at the receiving end in order to be able to correlate the watermark with the watermark extracted from the image itself [1]. Such techniques are not truly oblivious because the hash needs to be exchanged prior to watermark detection.

Secure oblivious watermarking of videos for fingerprinting or data embedding requires watermarks that depend on each frame. Indeed, one watermark pattern inserted into each frame would lead to a very weak watermarking scheme. However, the requirement of the technique to be oblivious means that either the watermark depends on the frame index or it is determined by the frame itself. Obviously, the latter case leads to more versatile schemes. A reliable method for generating a good approximation of the watermark from the image itself (even after watermarking and attacks) will clearly lead to more useful and elegant oblivious watermarking schemes.

Tewfik et al. [1] describe a watermarking technique in which a user-defined noise-like signature is modulated with a perceptual mask calculated from small blocks using frequency masking. The same signature is used for all video-frames. The watermark pattern in this application is frame dependent and does not depend on the frame index. However, the frame dependency is not too strong because the perceptual mask can be calculated from each frame, which makes the technique equivalent to watermarking with a fixed watermark pattern.

Image watermarking for tamper detection leads to a similar situation as watermarking videos. Each digital image with a digital camera or digital video-camera would be watermarked on the fly so that later on we can prove image integrity or indicate blocks in the image that have been tampered with. For a review of watermarking techniques for tamper detection, see [2, 3]. In this

particular application, the original image will never be available because only the watermarked image is stored on the camera's storage medium. Again, using one pattern that does not depend on the image would be insecure.

What is needed in both applications is a watermark W that depends sensitively on a secret key K and continuously on the frame I. More formally, $W(K, I)$ is uncorrelated with $W(K, I')$ whenever I and I' are dissimilar, $W(K, I)$ is strongly correlated with $W(K, I')$ whenever I and I' are similar (I' is the frame I after an attack or watermarking), and $W(K, I)$ is uncorrelated with $W(K', I)$ for $K \neq K'$. Linnartz and Cox [4, 5] proposed similar requirements for watermarking DVDs. This paper presents a solution to this problem.

The method presented here is an extension and improvement of the robust bit-extraction method proposed in [3]. In Section 2, we describe a method for extracting N bits from image blocks so that all similarly looking blocks produce almost the same N bits. The extraction is based on thresholding projections onto N key-dependent random smooth patterns. The bit error rate for the bits extracted from images attacked using common image processing operations is also given. In Section 3 we explain how to generate a Gaussian pseudo-random sequence from the extracted bits so that a gradual decrease in correlation with the original sequence (obtained if all N bits are correctly extracted) is achieved with increasing bit error rate in the extracted bits. Section 4 contains conclusion and directions for future research.

2. Description of the bit extraction procedure

Our goal is to design a procedure for extracting bits from image blocks so that all similarly looking blocks, whether they are watermarked, unwatermarked or attacked, will produce almost the same bit sequence of length N. We base our method on the observation that if a low-frequency DCT coefficient of an image is small in absolute value, it cannot be made large without causing visible changes to the image. Similarly, if the absolute value of a low-frequency coefficient is large, we cannot change it to a small value without influencing the image significantly. To make the procedure key-dependent, we replace DCT modes with low-frequency DC-free (i.e., having zero mean) random smooth patterns generated from a secret key (with DCT coefficients equivalent to projections onto the patterns). For each image, we calculate a threshold Th so that on average 50% of projections have absolute value larger than Th and 50% are in absolute value less than Th. This will maximize the information content of the extracted N bits. Having extracted N bits from an image block, in the next section we explain how to synthesize a Gaussian pseudo-random sequence $\eta \in N(0,1)$ that depends sensitively on a secret key K and continuously on the image block B. Our design requirements are

1. $\eta(K, B)$ is uncorrelated with $\eta(K', B)$ for $K \neq K'$
2. $\eta(K, B)$ is uncorrelated with $\eta(K, B')$ whenever B and B' are dissimilar
3. $\eta(K, B)$ is highly correlated with $\eta(K, B')$ whenever B is similar to B'

In the requirements above, we used two subjective terms "similar" and "dissimilar". We will say that block B_1 is similar to B_2 if B_2 can be obtained from B_1 by applying some common image operation, such as contrast/brightness adjustment, gamma correction, histogram equalization and stretching, blurring, sharpening, edge enhancement, lossy compression, nonlinear filtering, noise adding, decreasing the color depth, printing and subsequent scanning, or watermarking.

Given an image I, we divide it into blocks of 64×64 pixels (for large images, larger block sizes could be used). Using a secret key K (a number uniquely associated with an author, movie distributor, or a digital camera) we generate N random matrices with entries uniformly distributed in the interval [0, 1]. Then, a low-pass filter is repeatedly applied to each random matrix to obtain N random smooth patterns. An example of four random patterns and their smoothened versions are shown in Figure 1. All patterns are then made DC-free by subtracting the mean from each pattern. Considering the block and the pattern as vectors, the image block B is projected on each pattern P_i, $1 \leq i \leq N$ and its absolute value is compared with a threshold Th to obtain N bits b_i:

$$\text{if } |B \cdot P_i| < Th \qquad b_i = 0$$
$$\text{if } |B \cdot P_i| \geq Th \qquad b_i = 1.$$

Since the patterns P_i have zero mean, the projections do not depend on the mean gray value of the block and only depend on the variations in the block itself. The distribution of the projections is image dependent and should be adjusted accordingly so that approximately half of the bits b_i are zeros and half are ones. This will guarantee the highest information content of the extracted N-tuple. This adaptive choice of the threshold becomes important for those image operations that significantly change the distribution of projections, such as contrast adjustment.

To test the robustness of the proposed bit extraction procedure, we subjected the test image "Lenna" with 256×256 pixels and 256 gray levels to various image processing operations available in typical commercial image manipulation software (we used PaintShop Pro 4.12). The test image "Lenna" had 16 blocks of 64×64 pixels. We extracted N=50 bits from each block for the

original image and the manipulated image and calculated the average number of correctly recovered bits over all 16 blocks. The results are shown in Table 1. To see the importance of the smoothing operation, we compared our technique with a similar technique in which random unsmoothened patterns were used (the rightmost column in Table 1). As can be seen, the performance for

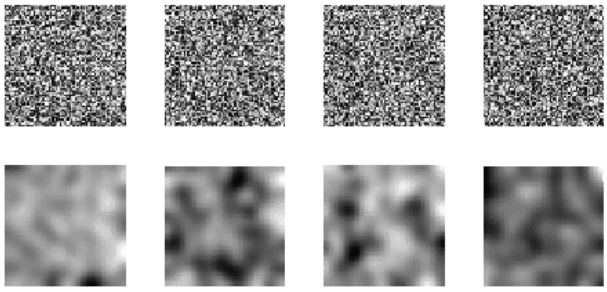

Figure 1 Examples of four random patterns and their smoothened version

Image operation	Random smooth patterns	Random patterns (non-smooth)
85, 50, 25, 15, 5% quality JPEG	49.69, 49.81, 49.63, 49.00, 47.69	47.88, 46.19, 44.63, 44.00, 41.82
Adding uniform noise in [−7, 7], [−22, 22], [−36, 36]	49.56, 48.81, 47.69	45.50, 37.13, 32.06
−50, −25,+25, +50% contrast	49.88, 49.94, 49.94, 48.75	49.50, 49.69, 49.94, 46.50
−30, −15,+15, +30% brightness	47.31, 49.94, 49.81, 47.44	45.56, 49.81, 49.81, 46.06
Posterizing to 2, 4, 8, 16 colors	39.56, 45.50, 48.25, 49.19	34.00, 40.00, 44.44, 47.56
Sharpen 1×, 2×	49.81, 49.56	46.56, 41.56
Histogram equalization	48.06	46.25
Histogram stretching	49.88	49.88
Blurring 1×, 4×	49.50, 48.69	43.13, 40.19
0.7, 1.5 gamma correction	48.50, 48.75	47.19, 48.25
Median filter 3×3	49.56	44.94
Mosaic filter 6×6	48.50	40.19
Edge enhancement	48.75	35.94
Erode	44.31	38.13
StirMark 1.0 1×, 2×, 3×	45.31, 39.75, 35.13	35.94, 31.63, 29.13
Emboss	26.19	27.13

Table 1 Average number of correctly recovered bits out of 50 bits after some image processing operations.

non-smooth patterns is significantly worse especially for low-pass filtering operations and some high-pass filtering, such as sharpening or edge enhancement. Except StirMark [6], all image operations were carried out in PaintShop Pro 4.12. For StirMark, we used the default settings of version 1.0. The bit extraction algorithm can reliably extract over 48 correct bits (out of 50 bits) for the following image processing operations: 15% quality JPEG compression, additive uniform noise with amplitude of 30 gray levels, ±50% contrast adjustment, ±25% brightness adjustment, dithering to 8 colors, multiple applications of sharpening, blurring, median, and mosaic filtering, histogram equalization and stretching, edge enhancement, and gamma correction in the range 0.7−1.5. Taking the negative of the image produces all 50 correct bits as expected. Quite understandably, operations like embossing produce images from which the bits cannot be reliably extracted because the image has been flattened. Geometrical modifications, such as rotation, shift, and change of scale, need to be corrected for before the bit extraction procedure can be applied. The number of correctly extracted bit should be compared to the

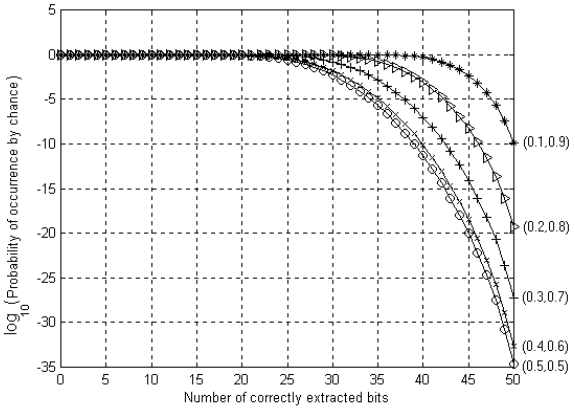

Figure 2 Probability of extracting certain number of correct bits due to pure chance for various distributions of 1's and 0's in the extracted 50-bit tuple.

number of correctly recovered bits due to pure chance given a priori probability distribution of 1's and 0's in the bit sequence (see Figure 2).

3. Generating watermark using extracted bits

Vast majority of watermarking schemes generate the watermark from a pseudo-random sequence. In this section, we explain how to synthesize a Gaussian sequence from N robustly extracted bits so that the pseudo-random sequence gradually changes with increased number of errors in the extracted N-tuple, yet sensitively depends on the secret key. In addition to that, we require that when approximately half of the bits are extracted incorrectly, the generated Gaussian sequence should not be correlated with the sequence produced from all 50 correct bits. To achieve this goal, we synthesize the pseudo-random Gaussian sequence by summing up uniformly distributed pseudo-random sequences obtained from a pseudo-random number generator (PRNG) seeded with a concatenation of the secret key, the block number, and randomly chosen q-tuples of the extracted bits ($q \approx 5$). We start by generating q random permutations π_1, π_2, …, π_q of integers between 1 and N. The permutations could be fixed for all images and blocks or change with the block. Then for each i, $1 \le i \le N$, we seed a PRNG (with uniform probability distribution on $[-1,1]$) with a seed consisting of a concatenation of the secret key K, the block number B, the number i, and q bits $\pi_1(i)$, $\pi_2(i)$, …, $\pi_q(i)$. The PRNG then generates a pseudo-random sequence $\xi^{(i)}$ of a desired length (determined by the particular watermarking technique)

$$\xi^{(i)} = PRNG(K \oplus B \oplus i \oplus b_{\pi_1(i)} \oplus b_{\pi_2(i)} \oplus \ldots \oplus b_{\pi_q(i)}).$$

In the expression above the symbol $\oplus$ denotes concatenation. The final Gaussian sequence $\eta \in N(0,1)$ is obtained by summing up $\xi^{(i)}$ for all i and normalizing:

$$\eta = \sqrt{\frac{3}{N}} \sum_{i=1}^{N} \xi^{(i)}.$$

The process of generating the pseudo-random sequences $\xi^{(i)}$ is schematically depicted in Figure 3. If the probability of extracting 1 is the same as probability of extracting 0, we can easily estimate how many seeds will be recovered correctly for the correct secret key and similar blocks. If k bits out of N bits are recovered correctly, then approximately $(k/N)^q$ seeds (and consequently the sequences $\xi^{(i)}$) will be correct. If we use the wrong key or a dissimilar block, the number of correctly recovered seeds will be roughly $1/2^q$ which could be made much smaller than $(k/N)^q$ by choosing q appropriately (see Figure).

In our simulations, we used q=5 as a compromise between the loss of correlation due to image degradation and creating a small correlation among dissimilar blocks for the same secret key and the same fixed block. The watermarking scheme that utilizes our robust bit extraction technique is schematically depicted in Figure 5.

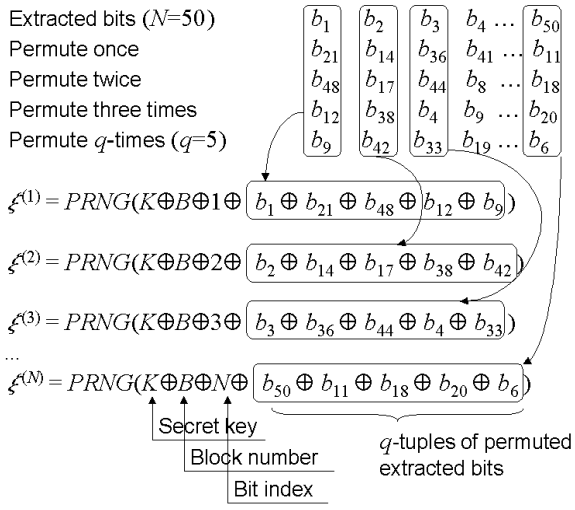

Figure 3 Synthesizing the Gaussian pseudo-random sequence from the extracted bits.

4. Conclusion and future directions

In this paper, we describe an algorithm for robust extraction of bits from image blocks. The robustness guarantees that all similar blocks will produce almost the same bits. This property is useful for generating pseudo-random watermark sequences used in many current watermarking schemes without the access to the original image or its hash. We further describe how to use the extracted bits to synthesize a Gaussian sequence that gradually changes with increasing number of errors in the extracted bits. The bit extraction algorithm depends on a secret key, which provides additional security feature. The algorithm can be used for generating pseudo-random watermark sequences that depend sensitively on a secret key yet continuously on the image. Applications include digital watermarking for tamper detection (in digital cameras) and for fingerprinting videos.

Since the random smooth patterns used in the bit extraction technique are not orthogonal to each other, the extracted bits are correlated. This means that the information content of the extracted N-tuple is not N bits

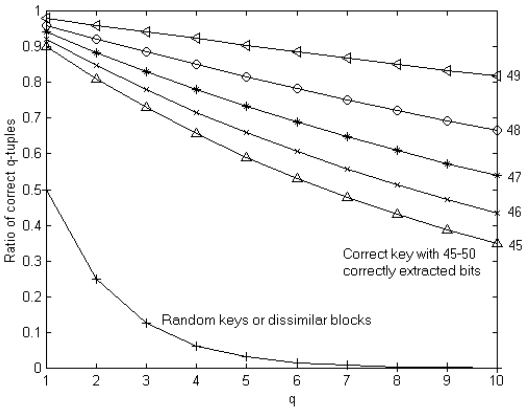

Figure 4 Ratio of correct seeds (*q*-tuples) as a function of *k* (the number of correctly recovered bits) and the factor *q*.

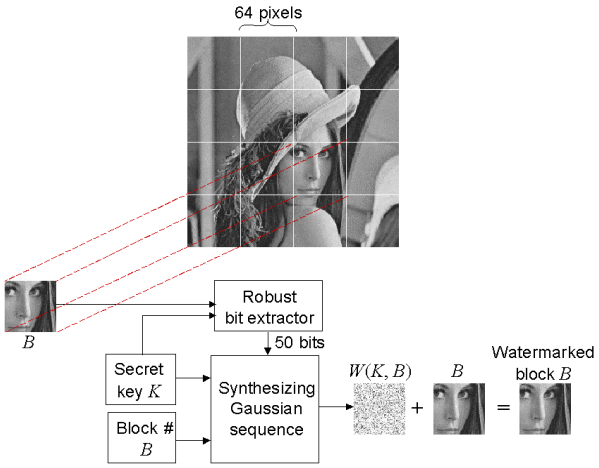

Figure 5 General scheme for watermarking using the proposed bit extraction procedure.

but somewhat less. If the correlation between the bits is an important issue, the algorithm can be modified to remove the correlation. For example, one can use orthogonal random smooth patterns for the bit extraction. One solution would be to use the Gram-Schmidt orthogonalization procedure and make the random smooth patterns orthogonal. If N patterns are used on blocks of $B \times B$ pixels, the computational complexity of the procedure would be $O(N^2B^2)$, which could become prohibitively large. A computationally much more efficient way would be to first generate $k = \sqrt{N}$ vectors of dimension B, smooth them, and orthogonalize them using the Gram-Schmidt algorithm. Having calculated k random, smooth orthogonal column vectors $x_1, x_2, \ldots, x_k$, one can obtain $N = k^2$ matrices using the tensor products $x_i \times x'_j$, $1 \le i, j \le k$, where the prime means transposition. The matrices obtained in this way will form a set of mutually orthogonal matrices (if considered as vectors). The computational complexity associated with this algorithm is only $O(NB)$, which is a significant saving. Detailed investigation of the robustness properties of the scheme based on orthogonalized random smooth patterns will be a part of future research.

In the future, we will investigate other principles for robust bit extraction, such as projecting random, key-dependent vectors onto principal directions of image blocks, utilizing the robustness of those pricipal directions for bit extraction. Another research direction is robust bit extraction that is translationaly invariant.

5. Acknowledgements

The work on this paper was supported by Air Force Research Laboratory, Air Force Material Command, USAF, under a Phase II SBIR grant number F30602-98-C-0049. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, or the U. S. Government.

6. References

[1] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Data Hiding for Video in Video", *Proc. ICIP '97*, vol. II, pp. 676–679.

[2] J. Fridrich, "Methods for Detecting Changes in Digital Images", *Proc. of The 6th IEEE International Workshop on Intelligent Signal Processing and Communication Systems* (*ISPACS'98*), Melbourne, Australia, 4–6 November 1998.

[3] J. Fridrich, "Image Watermarking for Tamper Detection", *Proc. ICIP '98*, Chicago, Oct 1998.

[4] I. J. Cox and J.-P. M. G. Linnartz, "Public watermarks and resistance to tampering", *ICIP'97*, Santa Barbara, California, October 1997. Paper appears only in CD version of proceedings.

[5] I. J. Cox and J.-P. M. G. Linnartz, "Some general methods for tampering with watermarks", preprint, 1998.

[6] F. Petitcolas, M. Kuhn, "StirMark watermark removing software", http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/