Robust Hash Functions for Digital Watermarkin

[a)]Jiri Fridrich and [b)]Miroslav Goljan
[a)]Center for Intelligent Systems, SUNY Binghamton, Binghamton, NY 13902-6000
[a)]Mission Research Corporation, 1720 Randolph Rd SE, Albuquerque, NM 87501
[b)]Department of Electrical Engineering, SUNY Binghamton, NY 13902-6000
Phone: +1 607 777 2577, fax: +1 607 777 2577, E-mail: { fridrich, bg22976}@binghamton.edu

Abstract

Digital watermarks have recently been proposed for authentication of both video data and still images and for integrity verification of visual multimedia. In such applications, the watermark has to depend on a secret key and on the original image. It is important that the dependence on the key be sensitive, while the dependence on the image be continuous (robust). Both requirements can be satisfied using special image digest functions that return the same bit-string for a whole class of images derived from an original image using common processing operations. It is further required that two completely different images produce completely different bit-strings. In this paper, we discuss methods how such robust hash functions can be built. We describe an algorithm and evaluate its performance. We also show how the hash bits As another application, the robust image digest can be used as a search index for an efficient image database search.

1. Introduction

Hash functions are frequently called message digest functions. Their purpose is to extract a fixed-length bit-string from a message (computer file or image) of an length. Obviously, a message digest function is a many-to-one mapping. In cryptography, hash functions ar typically used for digital signatures to authenticate the message being sent so that the recipient can verify that the message is authentic and that it came from the right person. The requirements for a cryptographic hash function are [1]:

- Given a message m and a hash function H, it should be easy and fast to compute the hash $h=H(m)$
- Given h, it is hard to compute m such that $h=H(m)$ (i.e., the hash function should be one-way)
- Given m, it is hard to find another message m' such that $H(m')=H(m)$ (property of being collision free)

From the above properties it is clear that hash functions are "infinitely" sensitive in the sense that a small perturbation of the message m will give you a completely different bit-string h. In applications involving digital watermarking and authentication of digital images, the requirements on what should be a digest of an image are somewhat different. Changing the value of one pixel does not make the image different or non-trustable. Distortion introduced by lossy compression or typical image processing does not change the visual content of the image. What would be useful to have is a mechanism that would return approximately the same bit-string for all simila looking images, yet, at the same time, two completel different images would produce two uncorrelated hash strings. This is what we call in this paper a robust hash function (visual hash). One can say that we want approximately the same hash bit-strings for two images whenever the human eye can say that these two images "are the same". Obviously, this is a challenging problem that can never be solved to our complete satisfaction. This is because the fuzzy concept of two images being visually the same is inherently ill defined and difficult, if not impossible, to grasp analytically. For example, changing one pixel in the pupils of a person's eye is for all purposes a negligible change. But once we change the color of every pixel in the pupil from, say, blue to brown, an important personal characteristic has been changed. Thus, we would conclude that the two images are no longer the same. However, the pupils can occupy a very small part of the image and our robust hash, not knowing the importance of eyes, may return the same hash bit-string. Being aware of these and other limitations, nevertheless, in this paper, we attempt to meaningfully define the concept of a robust visual hash. Before we start with the definition and ideas how to construct such a function, we give a brief introduction into oblivious digital watermarking and explain how robust hash will play an important role in specific watermarking applications, such as authentication and fingerprinting.

2. Digital watermarking

Digital watermark is a perceptually invisible patter embedded in a digital image. The watermark can carry information about the owner of the image or the recipient (watermarking for copyright protection, fingerprinting, or traitor tracing), the image itself (watermarking for tamper detection and authentication), or some additional information accompanying the image (image caption embedding). Watermarking schemes can be 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 fo watermark extraction. Although this makes non-oblivious techniques more robust to attacks, 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 [2]. Such techniques are not truly oblivious because the hash needs to be exchanged prior to watermark detection.

Secure oblivious watermarking of videos fo fingerprinting or authentication requires watermarks that depend on each frame. Indeed, one watermark pattern inserted into each frame would lead to a very vulnerable watermarking scheme with a serious securit gap. It has been shown that by processing the images (frames), it is possible to statistically recover a good approximation to the watermark pattern [3]. 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 afte watermarking and attacks) will clearly lead to more useful and elegant oblivious watermarking schemes.

Tewfik et al. [2] describe a watermarking technique in which a user-defined noise-like signature is modulated with a perceptual mask calculated from small blocks using perceptual 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 we can prove image integrity or indicate blocks in the image that have been tampered with. For a comprehensive review of watermarking techniques for tamper detection and common security problems, see [4]. Again, in this particular application, using one pattern that does not depend on the image would be insecure because analyzing a relatively small number of images ma reveal the watermark pattern [3].

What is needed in both applications discussed above is a watermark W that depends sensitively on a secret key K and continuously on the image I:

1. $W(K, I)$ is uncorrelated with $W(K, I')$ whenever images I and I' are dissimilar;
2. $W(K, I)$ is strongly correlated with $W(K, I')$ whenever I and I' are similar (I' is the image I after an attack comprising of a rotation, scale, and grayscale modifications);
3. $W(K, I)$ is uncorrelated with $W(K', I)$ for $K \neq K'$.

Linnartz and Cox [5, 6] proposed simila requirements for watermarking digital video disks (DVD). The requirements 1 –3 could be satisfied provided we have a robust image digest function H (visual hash function) that returns the same N bits (or almost the same N bits) for all images I that underwent a combination of a rotation R_φ by an angle φ, scaling S_α by a factor α, and typical grayscale operations G. Noise adding, filtering, lossy JPEG compression, gamma correction, and histogram equalization are examples of typical grayscale operations. So, if the robust hash function H depends on a parameter K (secret key), we require that

$$H_K(R_\varphi \circ S_\alpha \circ G(I)) \approx const. \in \{0,1\}^N, \qquad (1)$$
$$\text{for all } \varphi,\ \alpha, \text{ and } G.$$

In the next section, we review ideas proposed by various researchers in the past (some ideas were posed in a different context). We evaluate the positive and negative properties and then outline our approach in Section 4. We present some analysis of the robustness of the hash with respect to intentional attempts to modify the hash in Section 5. In Section 6, we show how to synthesize a Gaussian sequence from the extracted hash bits so that the Gaussian sequence loses its correlation with the original sequence gradually. We conclude the paper in Section 7.

3. Image invariants and robust hash

From the definition given in the previous section, robust image hash is a bit-string that somehow captures the essentials of the digital image or block. Ou requirement is that we need a key-dependent function that returns the same bits or numbers from simila looking images. So, the question is: "What is preserved under typical image processing operations?" Image edges typically contain the essence of an image. We could also use some relative relationship between pairs of image features, such as DCT coefficients. Also, it is well known that the principal directions and principal values calculated from image blocks are resistant to all kinds of grayscale image processing [11]. However, the principal directions are publicly known and the hash built from them would not have any security element in it. One could introduce a key-dependent linear or non-linear combination of the values determined from singular value decomposition of the image block, but this would provide only marginal security since the main robust values are not protected by a key, and therefore, can be intentionally manipulated. Another possibility would be to use invariant moments [12] or their key-dependent combinations for robust extraction of bits. Again, the problem with this approach is that the invariant moments are publicly known and can be purposely modified. Thus, the watermarking technique that utilizes bits derived from those moments would be inherently less secure. In [13], the authors proposed the usual hash of an edge map of a scaled-down image as a robust way of getting key-dependent hash bits fo images. The logic is that edges are salient features of images and should be preserved for most image transformations. However, the usage of the cryptographic hash function will create a cliff-off effect that may not be desirable for robust watermarking. As long as the edge map does not change (after thresholding), the hash behaves in a robust manne with respect to small noise adding. However, once the edge map is modified, even in one pixel only, the hash returns a completely different bit-string. It would be nice to have a robust hash that deteriorates graduall rather than in an abrupt way, so that the watermark built from the hash is still highly correlated with the watermark used in watermark embedding.

Another approach that works quite well for small distortion especially distortion introduced by JPEG compression was introduced in [14]. The authors emphasize the fact that the mutual relationship of DCT coefficients in 8×8 blocks will be preserved no matte what quantization matrix is used for coding the image. Thus, one can extract one bit of information from predetermined pairs of DCT coefficients based on the fact if the first or the second pair member is larger than the other. The extracted bits are finally processed using a one-way function to obtain the final hash. There are several disadvantages of this method for use as a robust hash. First of all, while this method works very well fo JPEG compression, its performance is less satisfactory for a different type of distortion, such as contrast enhancement. Second, as long as the mutual relationship of the coefficient pairs is not changed, the authentication technique based on this hash will not detect the change. And finally, one can purposel modify certain DCT coefficients to change the hash completely while making undetectable modifications to the image. This is because the DCT coefficients that enter the one-way function are publicly known.

4. Robust hash (our approach)

In this section, we describe a previously proposed mechanism [7,20] for robust extraction of bits from image blocks so that all similarly looking blocks, whether they are watermarked, unwatermarked or attacked by gray scale modifications, will produce almost the same bit sequence of a specified length N. We present some new results concerning the robustness of the hash bits with respect to intentional attempts to modify the hash.

The method is based 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 dependent on a key, the DCT modes are replaced 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, a threshold Th is calculated so that on average 50% of projections have absolute value larger than Th and 50% are in absolute value less than Th. This maximizes the information content of the extracted N bits.

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 uniforml distributed in the interval [0, 1]. Then, a low-pass filte is repeatedly applied to each random matrix to obtain N random smooth patterns $P^{(i)}$, $1 \le i \le N$. An example of four random patterns and their smoothened versions are shown in Fig. 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 I is projected on each pattern $P^{(i)}$, $1 \le i \le N$, and its absolute value is compared with the threshold Th to obtain N bits b_i :

$$\text{if } |B\cdot P^{(i)}| < Th \qquad b_i = 0$$
$$\text{if } |B\cdot P^{(i)}| \ge Th \qquad b_i = 1.$$

Since the patterns $P^{(i)}$ have a zero mean, the projections do not depend on the mean gray value of the block and only depend on the variations within 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 or gamma correction.

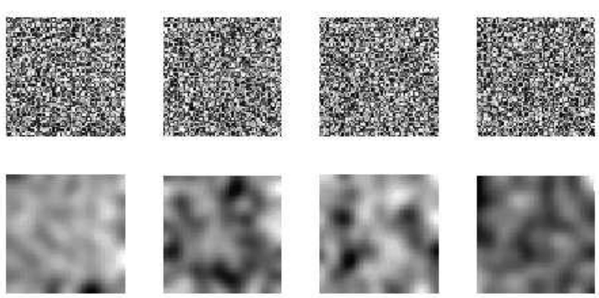

Fig. 1 Examples of four random patterns and their smoothened version

The robustness of this bit extraction technique has been tested on real imagery with very promising results (see Table 1). The bit extraction algorithm can reliabl extract over 48 correct bits (out of 50 bits) from a small 64×64 image for the following image processing operations: 15% quality JPEG compression (as in PaintShop Pro), 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 returns 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, also lead to a failure to extract the correct bits. Detailed evaluation of experiments can be found in our previous paper [7]. Modification of the scheme that should exhibit robustness to scaling and rotation has been described in [10].

5. Robustness to intentional attacks

The security of the hash is in the secrecy of the smooth patterns. An attacker who does not know the key cannot purposely modify the projections. The best he can do is to introduce noise hoping that the projections will change. In this section, we look at the possibility of changing the hash bits if the attacke knew the patterns. This is equivalent to knowing the secret key. We try to answer the question of how many hash bits can be changed using the knowledge of projections by making imperceptible changes to the pixel gray levels. The maximal allowable changes were determined by the masking model of Girod [15]. The constraints imposed by the masking model also constrain the maximal possible changes in the projections $c_i = B\cdot P^{(i)}$. Consequently, not all hash bits can be flipped.

The maximal allowable change for the projection c_k is determined by the expression

$$\Sigma_{ij}|P_{ij}^{(k)}|d_{ij},$$

where d_{ij} is the masking value for pixel ij from the Girod's model, and $P_{ij}^{(k)}$, $k = 1, \ldots, N$ is the pattern number, and $i,j = 1, \ldots, 64$. Based on our analysis of several test images, we have determined that on average 37 hash bits are changeable if the smooth patterns are known. We stress that all these bits cannot be changed at the same time because they require different perturbations of the image block B. A natural question to ask is how many hash bits can be changed simultaneously rather than individually.

To answer this question, we need to solve this system of equations for d

$$P^{(k)}*(B+d) = Th,\ k = 1, \ldots, N,$$

wit constraints that the maximal and minimal values of the perturbations d are integers and are determined from the masking model. Because $B \cdot P^{(k)} = c_k$, we obtain a system of linear equations

$$P^{(k)}*d = Th - c_k.$$

Our computer experiments on images indicate that as many as 13 bits (out of N = 50) on average could be chenged simultaneously while making imperceptible changes (according to the Girod's masking model). We again emphasize that this is possible to do only because we know the smooth patterns (or the secret key used for the robust hash).

6. Generating a watermark using the hash

Vast majority of watermarking schemes generates the watermark from a pseudo-random sequence. In this section, we explain how to synthesize a Gaussian sequence from N hash bits so that the pseudo-random sequence gradually changes with increased number of errors in the hash, yet sensitively depends on the secret key. In addition to that, we require that when approximately half of the hash is incorrect, 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 uniforml distributed pseudo-random sequences obtained from a pseudo-random number generator (PRNG) seeded with a concatenation of the secret key, the block number (if the watermarking is done by blocks), and randomly chosen q-tuples of the extracted bits ($q \approx 5$). We start b generating q random permutations $\pi_1, \pi_2, \ldots, \pi_q$ 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 $\pi_1(i), \pi_2(i), \ldots, \pi_q(i)$. 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 2. If the probability of extracting 1 is the same as probabilit 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.

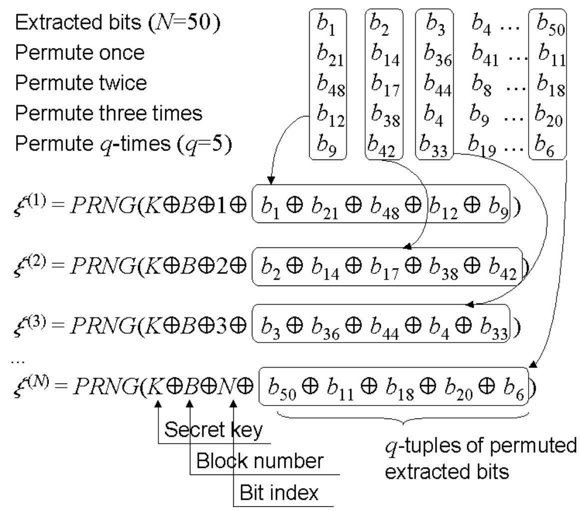

Fig. 2 Synthesizing the Gaussian pseudo-random sequence from the extracted bits

We recommend to use q=5 as a compromise between the loss of correlation due to image degradation and creating a small correlation among dissimilar blocks fo the same secret key and the same fixed block.

7. Conclusions

In this paper, we introduce the concept of a robust hash function with applications to digital image watermarking for authentication and integrit verification of video data and still images. The robust image digest can also be used as a search index for efficient database searches. The hash function depends on a parameter K (a secret key) in a sensitive manne and on the image in a robust manner. The hash function is designed to retur N = 50 bits from a 64×64 image block. The bits obtained from two different images or for two different keys K will generally be different (uncorrelated). However, for the same key K, two images that can be matched after applying gra scale operations, such as lossy compression, recoloring, filtering, noise adding, gamma correction, and simple geometrical operations including rotation and scaling, the extracted N-tuple will be almost the same except fo a few bits. In [7,10], it is explained how the extracted N-tuple can be further utilized for synthesizing a Gaussian sequence that gradually changes with increasing number of errors in the extracted bits. Thus the robust hash function can be used for generating pseudo-random watermark sequences that depend sensitively on a secret key yet continuously on the image. This robustness enables us to construct watermarks that depend on the original unwatermarked image in a non-trivial manner while making it possible to recover the watermark without having to access an information about the original image (oblivious watermarking). Such watermarks play an important role for authenticating videos or still images taken with a digital camera [4].

As another application of robust hash functions, we mention indices for efficient image database search. There are many quantities that could be derived from images using which one can search a database in an efficient manner. Many indices are based on colo information that can be extracted from a histogram. However, such indices are not useful if the image has been processed using histogram equalization, o recolored. The essence of an image can be well captured using its edges. Our method captures the mutual spatial relationship among edges rather than color information. This relationship is independent of the image orientation and size and on typical non-destructive image processing operations, such as recoloring, brightness adjustment, filtering, loss compression, or small noise adding. Thus, it is computationally much more efficient to search an extensive image database by matching the extracted bit-string rather than the whole images.

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 ther on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, eithe expressed or implied, of Air Force Research Laboratory, or the U. S. Government.

References

[1] B. Schneier, *Applied Cryptography*, John Wiley&Sons, New York, 1996.

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

[3] M. Holliman, N. Memon, and M. M. Yeung, “On the Need for Image Dependent Keys for Watermarking”, *Proc. Content Security and Data Hiding in Digital Media*, Newark, NJ, May 14, 1999.

[4] J. Fridrich, “Methods for Tamper Detection in Digital Images”, *Proc. ACM Multimedia 1999, Workshop on Multimedia and Security*, October 30 – November 5, 1999.

[5] 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.

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

[7] J. Fridrich, “Robust Bit Extraction From Images”, *ICMCS'99*, Florence, Italy, June 7–11, 1999.

[8] R. D. Brandt and F. Lin, "Representations that uniquely characterize images modulo translation, rotation and scaling", *Pattern Recognition Letters* **17**, pp. 1001–1015, August 1996.

[9] J. J. K. Ó Ruanaidh and T. Pun, “Rotation, scale and translation invariant digital image watermarking”, *Proc. of the ICIP'97*, vol. 1, pp. 536–539, Santa Barbara, California, 1997.

[10] J. Fridrich, "Visual Hash for Oblivious Watermarking", *Proc. SPIE Photonic West Electronic Imaging 2000, Security and Watermarking of Multimedia Contents* , San Jose, California, January 24–26, 2000.

[11] M. Alghoniemy and A. H. Tewfik, "Progressive Quantized Projection Watermarking Scheme", Proc. ACM Multimedia '99, Orlando, Florida, November 2–5, 1999, pp. 295–298.

[12] Ming Kuei-Hu, "Visual Pattern Recognition by Moment Invariants", *IRE Transactions on Information Theory*, Vol. 8, pp. 179–187, February 1962.

[13] L. Xie and G. R. Arce, "A Class of Authentication Digital Watermarks for Secure Multimedia Communication", preprint, submitted to *IEEE Transactions on Image Processing*, December 1999.

[14] Ching-Yung Lin and Shih-Fu Chang, "Generating Robust Digital Signature for Image/Video Authentication", *Proc. ACM Multimedia 1999, Proc. Multimedia and Security Workshop at ACM Multimedia '98,* U.K., September 1998.

[15] B. Girod, “The information theoretical significance of spatial and temporal masking in video signals”, *Proc. of the SPIE Human Vision, Visual Processing, and Digital Display*, vol. 1077, pp. 178–187, 1989.