Towards More Robust Commutative Watermarking-Encryption of Images

Roland Schmitz*, Shujun Li†, Christos Grecos‡, and Xinpeng Zhang§
* *Stuttgart Media University, Stuttgart, Germany*
Email: schmitz@hdm-stuttgart.de
†*University of Surrey, Guildford, UK*
Email: shujun.li@surrey.ac.uk
‡*University of the West of Scotland, Paisley, UK*
Email: christos.grecos@uws.ac.uk
§*Shanghai University, Shanghai, China*
Email: x.zhang@shu.edu.cn

***Abstract*—Histogram-based watermarking schemes are invariant against pixel permutations and can be combined with permutation-based ciphers. However, typical histogram-based watermarking schemes based on comparison of histogram bins are prone to de-synchronization attacks, where the whole histogram is shifted by a certain amount. In this paper we investigate the possibility of avoiding this kind of attacks by synchronizing the embedding and detection processes, using the mean of the histogram as a calibration point. The resulting watermarking scheme is resistant to three common types of shifts of the histogram, while the advantages of previous histogram-based schemes, especially commutativity of watermarking and permutation-based encryption, are preserved.**

I. Introduction

It is well known that histogram-based watermarking schemes are resistant to permutations of image pixels. In particular, this implies robustness against rotation, scaling and translation (RST) of images. Recently, this fact has been utilized to devise a commutative watermarking-encryption (CWE) scheme by choosing a permutation cipher for encryption and a histogram-based scheme for watermarking [1]. However, typical histogram-based watermarking schemes like [1], [2] work by comparing selected histogram bins, where the selection process is controlled by a watermarking key. If the whole histogram is shifted by a small amount, i.e. by adding a small number to each pixel's value, the detector will use different bin pairs for extracting the embedded watermark and will produce wrong results. To overcome this problem, in the present paper, we extend the scheme described in [1] by deploying a synchronization process between embedder and detector that is based on the global mean of the histogram. A similar approach is used in the scheme [3], but this watermarking scheme does not use a sufficiently long secret watermarking key and has a limited capacity.

The rest of the paper is organized as follows: In Section II we briefly summarize previous histogram-based watermarking algorithms. Section III describes the three types of histogram shifts we have investigated, and Section IV describes the proposed algorithm in greater detail. In Section V we discuss experimental results for the algorithm, and Section VI concludes the paper and gives directions for further work.

II. Related Work

The most widely studied approach to histogram-based watermarking is so-called exact histogram specification [4]–[6], where the histogram of the original image or a (randomly and secretly selected) sub-region of it is modified toward a target histogram, which is then used as the signature for watermark detection. However, exact histogram specification does not involve a secret embedding/detection key, and there are few other histogram-based watermarking algorithms which do.

The scheme proposed by Xiang et al. in [3] (based on earlier work on audio watermarking [7]) represents the histogram shape as the ratios of population between groups of two neighbouring bins and then modifies the ratios to carry a key-based pseudo-random sequence. Only histogram bins in the range $[(1-\lambda)\bar{A}, (1+\lambda)\bar{A}]$ are used in the process, where $\bar{A}$ is the global mean of the histogram and $\lambda \in [0.5, 0.7]$ is a public parameter. In order to withstand scaling and cropping attacks on the image that will also affect the histogram and the mean, the extraction process uses a search process based on the mean $\bar{A}'$ of the histogram of the marked image: Different ranges $[(1-\lambda)(\bar{A}'+s), (1+\lambda)(\bar{A}'+s)]$, where s is an integer running through some search space, are tried, until the correlation between the extracted sequence and the known embedded sequence reaches the maximum. The resulting scheme is very robust against geometric image modifications and lossy compression, but it suffers from two severe limitations: The parameter λ may be seen as a watermarking key if kept secret, but there are only 26 possibilities for λ. Also, the effective capacity of the scheme is only 20-30 bits. While the synchronization process deployed in [3]

is very similar to the process described here, the present approach has a much larger keyspace and capacity.

The following two histogram-based watermarking schemes do use a longer watermarking key, but are by construction prone to histogram shifting attacks: The scheme proposed by Chrysochos et al. [2] is based on the idea of (selectively) swapping two selected histogram bins a and b, where the distance between a and b is a fixed number $d < 10$. A message bit is encoded by the relative heights of the two bins (denoted by hist(a) and hist(b)): a 1-bit is encoded by hist$(a) >$ hist(b) and a 0-bit by hist$(a) <$ hist(b). Here, swapping two histogram bins a and b means changing all pixel values a to b and vice versa.

In [1], the scheme described in [2] is extended and integrated into a CWE (Commutative Watermarking-Encryption) scheme. Histogram bins a and b are randomly selected from the 256 available bins under the condition that their relative distance is smaller than 10. This leads to a significant enlargement of the key space. As this scheme also forms the basis for the present watermarking algorithm it is described in greater detail in Section IV.

III. Histogram Shift Attacks

In this section we describe simple histogram modification attacks, where the histogram as a whole is shifted on the horizontal axis by adding a fixed amount to each pixel's greyvalue. We differentiate among three ways of histogram shifting:

A. Cyclic Histogram Shifting

In a cyclic shift, the greyvalue of each pixel $P(i,j)$ is shifted by a certain amount x modulo 256:

$$P_{\text{attacked}}(i,j) = (P(i,j) + x) \mod 256,$$

where x is a positive or negative integer. Due to the wrap-up at the end of the histogram, cyclic histogram shifting may lead to severe degradation of image quality. Cyclic histogram shifts therefore constitute less relevant attacks. Moreover, cyclic shifts are invertible if the amount of shift is known.

B. Accumulated Non-Cyclic Histogram Shifting

Here, the wrap-up in cyclic shifting is avoided as the shift is only applied to those pixels whose greyvalues are sufficiently small or big. For example, a rightward shift can be defined by

$$P_{\text{attacked}}(i,j) = \begin{cases} P(i,j) + x, & \text{if } P(i,j) < 256 - x, \\ P(i,j), & \text{else.} \end{cases}$$

where x is a positive integer. This kind of histogram modification leads to an accumulation of pixels at the start or the end of the histogram. Nevertheless, the amount of image distortion remains small, if $|x|$ is sufficiently small. Note that this kind of histogram shift cannot be reverted, unless sufficient information about the original histogram is known.

C. Histogram Cropping

Histogram cropping is performed directly on the histogram bins $H(i), 0 \leq i \leq 255$. The bins are shifted to the left (or right), where bins $H(i)$ with $i < 0$ (or $i > 255$) are dropped. A rightward shift-and-crop operation can be defined as

$$H_{\text{attacked}}(i) = \begin{cases} 0, & \text{if } 0 \leq i \leq x - 1, \\ H(i - x), & \text{if } x \leq i \leq 255. \end{cases}$$

The resulting histogram for small $|x|$ is similar to the original one, but contains less pixels and therefore does not constitute a valid histogram. However, after rescaling the attacked histogram, the attacked image can be constructed from the attacked histogram by exact histogram specification [4].

D. Comparing the Shifts

Depending on the shape of the histogram, the three kinds of shift behave very similarly up to a certain amount of shift. Figure 1 shows the effects of the various kinds of shift on the same example histogram (the blue channel histogram of the baboon image, see Figure 3(a)).

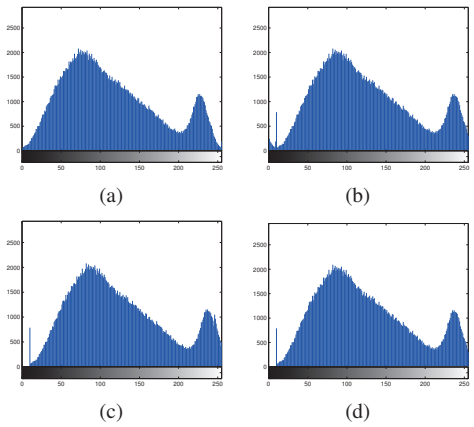
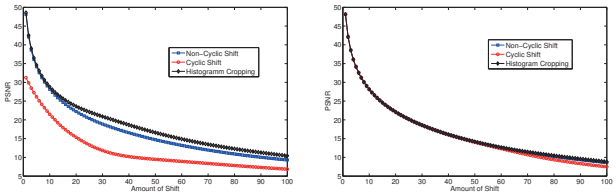


Figure 1. Effects of a histogram shift by the amount of 10 (a) Original histogram; (b) Cyclically shifted histogram; (c) Non-cyclically shifted histogram; (d) Shifted and cropped histogram

This is also verified when measuring the amount of distortion caused by the histogram shifts by calculating the PSNR between the original image and the image with shifted histograms, as Figure 2 shows.

The apparent similarity of the different kinds of shift motivates the idea of reversing the effects of non-cyclic shifts and histogram crops on the watermarked image by a cyclic histogram shift which has a similar effect on the image at the detector side. The optimal cyclic shift amount is found when the linear correlation of the detected mark and the reference mark reaches the maximum (see Section IV-B).



(a) (b)

Figure 2. The amount of distortion caused by histogram shifts (a) Blue channel of baboon image; (b) Blue Channel of Lenna image

IV. The Proposed Watermarking Algorithm

The design goal of the presented algorithm is to improve the robustness of previous histogram-oriented algorithms against simple histogram shifts, while retaining the original advantages, especially commutativity of watermarking and permutation-based encryption. Moreover, the algorithm should be able to use a watermarking key that is long enough to withstand brute-force attacks.

A. Embedding

The basic process for embedding is the same as described in [1]: Given an N-bit watermark $W = \{w_i\}, 1 \leq i \leq N$, a single watermark bit w_i is embedded by pseudo-randomly selecting two histogram bins that have not been selected before, if their distance is smaller than some strength parameter $d < 10$ and if they are not of equal height. The heights of the two selected bin pairs a_i and b_i encode w_i as follows: if $w_i = 1$, $\text{hist}(a_i) < \text{hist}(b_i)$ should hold, and if $w_i = -1$, $\text{hist}(a_i) > \text{hist}(b_i)$ should hold, where $\text{hist}(x)$ denotes the height of the bin x. If this is not the case, the two bins a_i and b_i are swapped. The selection process is governed by a watermark key W_K. The theoretical maximum capacity that can be achieved by this scheme is 128 bits and can be further extended by using more than one color channel and/or subdividing the image.

In order to speed up the search for the optimal amount of cyclic shift during extraction, we devised a calibration process that uses the global mean value $\bar{A}$ of the image as a calibration point. More specifically, before selecting the histogram bins for embedding, all bins are cyclically shifted by an amount of $x = 256 - \bar{A}$ as described in Section III-A so that the bin corresponding to $\bar{A}$ becomes the first bin in the calibrated histogram. After calibrating, the embedding process proceeeds as described above.

B. Detection

Basically, the detector works by comparing the histogram bins as specified by the watermarking key. For this to work, the embedder and the detector need to be synchronized, i.e. they need to use the same ordering of histogram bins. As a histogram shift will change the global mean greyvalue of the marked image, the detector searches for the correct calibration point by cyclically shifting the histogram of the marked image by $x = 256 - \bar{A}' - s$, where $\bar{A}'$ is the mean value of the marked image, and s runs through the search space $S = \{s | -\bar{A}'/4 \leq s \leq \bar{A}'/4\}$. The detector then computes the linear correlation of the extracted mark $W_{\text{ex}} = \{\tilde{w}_i\}$ with the reference mark $W = \{w_i\}$ for each s. The detector response is

$$\max_{-\bar{A}'/4 \leq s \leq \bar{A}'/4} \left(\frac{1}{N} \sum_{i=1}^{N} \tilde{w}_i w_i \right).$$

The watermark is detected if the detector response exceeds a certain threshold T.

C. Keyspace and False Positive Probability

As the calibration process prior to selecting the histogram bin pairs does not affect the number of available bin pairs, the size of the key space $K(N)$, where N is the length of the watermark, stays the same as in the scheme [1]:

$$K(N) > N! \cdot \left(\frac{256}{N} \cdot \min\left(9, \frac{256}{N} - 1\right) \right)^N.$$

Now let $\widetilde{W}$ be a mark extracted from an unmarked image I_U. If $\widetilde{W}$ agrees with W at k positions, their linear correlation is $\frac{2k-N}{N}$. Therefore, the mark is wrongly detected if $k > \frac{N}{2}(T+1)$. In order to simplify the analysis, we assume that the bipolar bits of $\widetilde{W}$ are evenly distributed in the set $\{-1, 1\}$. Then, the probability that two single bits of W and $\widetilde{W}$ agree is $1/2$. Therefore, the false positive probability for a single detection step becomes

$$q = \left(\frac{1}{2}\right)^N \cdot \sum_{k=\lceil \frac{N}{2}(T+1) \rceil}^{N} \binom{N}{k}.$$

For $N = 64$, the bound $T = 0.7$ yields $q \leq 2.98 \times 10^{-18}$. If the detection process is carried out by running through a search space of size $|S|$, the false positive probability becomes

$$p(\text{False Positive}) = 1 - (1-q)^{|S|} \approx q|S| \text{ for small } q$$

D. Commutativity with Permutation Based Encryption

As the presented algorithm is completely histogram-based and the histogram is invariant to permutations, the commutativity property

$$\mathcal{M}(\mathcal{E}_k(I), m) = \mathcal{E}_k(\mathcal{M}(I, m))$$

holds, where $\mathcal{E}$ is the encryption function, k is the encryption key, I is the plaintext media data and m is the mark to be embedded.

V. Experimental Results

The main difference between the present algorithm and the algorithm proposed in [1] consists in the calibration step performed before embedding and detecting, which has no impact on the amount of distortion. Therefore, results on visual distortion carry over from [1].

In order to evaluate the robustness of the algorithm, we embedded 64 random bits into the blue channel of all 24 images from the Kodak image database (see http://r0k.us/graphics/kodak) and three standard images from the SIPI image database (see http://sipi.usc.edu/database/). Figure 3 shows the visual effect of embedding for the three standard images.

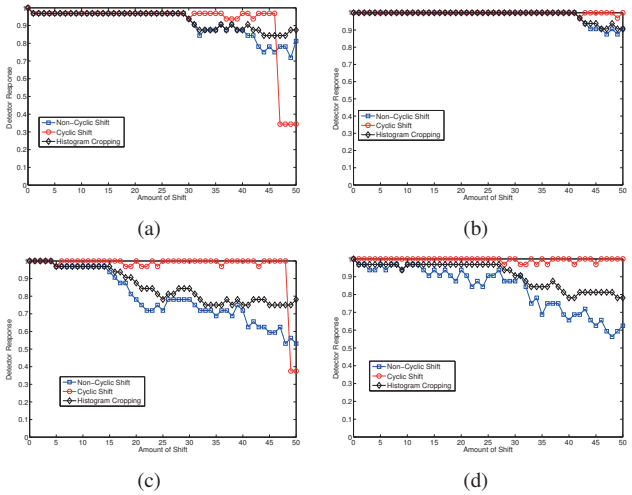
(a) (b) (c)

Figure 3. Embedding 64 random bits into test images (a) Marked baboon image (PSNR: 50.55 dB); (b) Marked sailboat image (PSNR: 58.86 dB); (c) Marked Lenna image (PSNR: 58.60 dB).

As it turned out, the three standard images show a rather prototypical behaviour with respect to robustness against the three kinds of histogram shift attacks. Figures 4(a) and 4(b) show a very good robustness for the sailboat and Lenna image, due to the fact that the corresponding histograms behave very similarly for the three kinds of shift and the shifts may thus be reversed by a suitable cyclic shift quite accurately. The relevant histogram for Figures 4(c) and 4(d) is the blue channel histogram of the baboon image (see Figure 1(a)), which behaves less favorably. In this case, the results can be significantly improved by enlarging the search space, e.g. to $S = \{s| - \bar{A}'/2 \leq s \leq \bar{A}'/2\}$, as Figure 4(d) shows. All other tested images behaved in a similar way. Thus, a robust detection strategy consists in enlarging the search space successively if the mark is not detected at first. The resulting increase of the false positive probability is negligible (see section IV-C).

VI. Conclusion

It is hard to devise a robust watermarking algorithm that can work in the encrypted domain, because there are no visually important features to use for embedding in this case. In the present paper, we have extended an earlier algorithm that is commutative with encryption by deploying a synchronization process between the embedder and the detector, making it robust against simple histogram shifts. Our further work will focus on improving robustness against other common image modifications such as lossy compression.



(a) (b) (c) (d)

Figure 4. Robustness results against histogram shifts (a) Sailboat image (b) Lenna image; (c) Baboon image; (d) Baboon image with an enlarged search space.

References

[1] R. Schmitz, S. Li, C. Grecos, and X. Zhang, "A new approach to commutative watermarking-encryption," in *Proc. Communications and Multimedia Security: 13th IFIP TC 6/TC 11 International Conf., CMS 2012*. Springer, 2012, pp. 117–130.

[2] E. Chrysochos, V. Fotopoulos, A. N. Skodras, and M. Xenos, "Reversible image watermarking based on histogram modification," in *Proc. 11th Panhellenic Conf. Informatics (PCI 2007)*, 2007, pp. 93–104.

[3] S. Xiang, H. J. Kim, and J. Huang, "Invariant image watermarking based on statistical features in the low-frequency domain," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 6, pp. 777–790, 2008.

[4] D. Coltuc and P. Bolon, "Robust watermarking by histogram specification," in *Proc. 1999 Int. Conf. Image Processing (ICIP'99)*, vol. 2. IEEE, 1999, pp. 236–239.

[5] S. Roy and E.-C. Chang, "Watermarking color histograms," in *Proc. 2004 Int. Conf. Image Processing (ICIP 2004)*. IEEE, 2004, pp. 2191–2194.

[6] C.-H. Lin, D.-Y. Chan, H. Su, and W.-S. Hsieh, "Histogram-oriented watermarking algorithm: colour image watermarking scheme robust against geometric attacks and signal processing," *IEE Proc. Vision, Image and Signal Processing*, vol. 153, no. 4, pp. 483–492, 2006.

[7] S. Xiang and J. Huang, "Histogram-based audio watermarking against time-scale modification and cropping attacks," *IEEE Transactions on Multimedia*, vol. 9, no. 7, pp. 1357–1372, 2007.