A New Approach to Commutative Watermarking-Encryption

Roland Schmitz*, Shujun Li†, Christos Grecos‡ and Xinpeng Zhang§
* Stuttgart Media University, Stuttgart, Germany, Email: schmitz@hdm-stuttgart.de
† Surrey University, 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: xzhang@shu.edu.cn

***Abstract*— We propose a new approach of designing commutative watermarking-encryption (CWE). A permutation cipher is used to encrypt the multimedia data, which leaves the global statistics of the multimedia data intact. Therefore, any *non-localized* watermarking scheme that depends only on global statistics of the multimedia data can be combined with the permutation cipher to form a commutative watermarking-encryption scheme. We demonstrate this approach by giving a concrete implementation, which manipulates the global histogram to achieve watermark embedding/detection.**

I. Introduction

Both encryption and watermarking are important tools in protecting digital contents, e.g. in digital rights management (DRM) systems. While encryption is used to protect the contents from unauthorized access, watermarking can be deployed to serve various purposes, ranging from ensuring authenticity of content to embedding metadata, e.g. copyright or authorship information, into the contents.

The concept of commutative watermarking-encryption (CWE) was discussed in [1] with special emphasis on watermarking in the encrypted domain. Four properties are formulated in [1, Sec. 2.2] to describe watermarking in the encrypted domain:

Property 1. The marking function $\mathcal{M}$ can be performed on an encrypted image.

Property 2. The verification function $\mathcal{V}$ is able to reconstruct a mark in the encrypted domain when it has been embedded in the encrypted domain.

Property 3. The verification function $\mathcal{V}$ is able to reconstruct a mark in the encrypted domain when it has been embedded in the clear domain.

Property 4. The decryption function does not affect the integrity of the watermark.

As is pointed out in [1], Properties 2 and 3 are equivalent, if the encryption function $\mathcal{E}$ and the marking function $\mathcal{M}$ commute, that is,

$$\mathcal{M}(\mathcal{E}_K(I), m) = \mathcal{E}_K(\mathcal{M}(I, m)) \tag{1}$$

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

Previous approaches to CWE are all based on two techniques: 1) homomorphic encryption, where the encryption function is commutative to some basic arithmetic operations like addition or multiplication that can support a further watermarking step; 2) partial encryption, where only a part of the multimedia data is encrypted and only the remaining data are watermarked. In the present contribution we propose a novel approach, namely to use a cipher that leaves a feature space of the multimedia data invariant and to use this feature space to embed the watermark. As a proof of concept of this new approach, we propose a CWE scheme for digital images by combining a permutation based cipher and a "non-localized" watermarking scheme working with the global image histogram in the spatial domain.

II. Related Work

A. Commutative Watermarking-Encryption

One approach to commutative watermarking is provided by deploying homomorphic encryption techniques so that some basic algebraic operations such as addition and multiplication on the plaintexts can be transferred onto the corresponding ciphertexts, i.e., they are transparent to encryption [1, Sec. 2.1]. Especially, if both the encryption and the watermarking process consist of the same homomorphic operation, one gets a commutative watermarking-encryption scheme. Examples of homomorphic operations are exponentiation modulo n, multiplication modulo n, addition modulo n and the XOR operation. One major drawback of this approach is the influence of encryption on robustness of the watermarking algorithm because after encryption there is no visual information available for the watermark embedder to adapt itself to increase robustness while at the same time minimizing visual quality degradation [2, Sec. 9.4]. Another drawback is that the modular addition operation may cause overflow/underflow pixels that have to be handled separately, thus making the system "quasi-commutative" [3]. The XOR operation does not suffer from the overflow/underflow problem, though.

In partial encryption schemes, the plaintext multimedia data is partitioned into two disjoint parts, where one part is encrypted and the other part is watermarked. Since the encryption part is independent of the watermarking part, they are naturally commutative. Because there is some information leakage through the unencrypted parts, in order to get a high level of perceptual security, the data parts which are significant for perception are encrypted, while only the perceptually unimportant parts are watermarked, leaving the door open

for an attacker trying to remove the watermark. This lack of robustness against malicious attacks seems to be a general problem with CWE schemes (cf. section IV-A).

B. Histogram Based Information Hiding

In [4] it is shown how a reversible information hiding scheme can be built by hiding data within the histogram. The basic idea is to shift the grey levels of all pixels having a grey level between $g_{\min}$ and $g_{\max}$ towards $g_{\min}$, where $g_{\min}$ and $g_{\max}$ denote the grey level with the lowest and the highest heights in the histogram, respectively. Such a shift will make the histogram bin at the position $g_{\max}+1$ or $g_{\max}-1$ empty, thus "making space" for the data to be hidden.

C. Histogram Based Watermarking Schemes

Most histogram based information hiding schemes cannot be used for secret watermarking because a secret embedding/detection key cannot be used. In what follows, we describe one approach whose basic principle is used in the example of our proposed CWE framework.

The scheme proposed by Chrysochos et al. [5] is based on the idea of (selectively) swapping two selected neighboring histogram bins a and b so that a message bit is encoded by the 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 order to embed an N-bit watermark into a 8-bit grey-level image, a *public key* composed of a bin distance $1 \leq \text{step} \leq 9$ and a start bin index $0 \leq a_1 \leq 255 - \text{step}$ is needed. The i-th bin pair is selected by increasing a_1 by i but skipping those bin pairs breaking at the right boundary of the histogram. The step is upper bounded to nine in order to limit visual quality degradation. The embedding capacity of the scheme depends on the number of candidate histogram bin pairs whose heights are not equal (which is dependent on the image and the step), but it is bounded by 128 bits for 8-bit grey-level images and 384 bits for RGB images. One main problem of this scheme is the very small key space, which contains only $\sum_{\text{step}=1}^{9}(256 - \text{step}) = 2259$ different watermarking keys.

III. The Proposed CWE Framework

Neither of the existing approaches to CWE tries to preserve pixel values and distorts locations of all pixels. This led us to propose a third approach for designing a CWE scheme: using a permutation cipher to encrypt the image to preserve all pixel values intact for watermarking embedding. If a watermarking scheme only uses the global histogram of the image for embedding and detection, which we call *non-localized watermarking*, the permutation (as an encryption function) and the watermarking processes will become commutative, satisfying all the four properties listed in Sec. I. Examples include the watermarking schemes proposed in [5], [6].

An obvious advantage of this new approach is that the robustness of the watermarking algorithm remains intact because all information (global statistical statistics) required by the algorithm is not changed by encryption. In this aspect, the new approach outperforms the homomorphic cryptography based CWE approach. Compared with the partial encryption based approach, our proposed can obviously provide a higher level of security since total encryption is applied here.

A. Watermarking Part

The watermarking part is designed following the basic principle of the histogram based watermarking scheme proposed in [5]. However, since this scheme suffers from a severe limitation, namely a very small key space, we have devised a modified histogram bin pairs selection process leading to a significantly bigger key space. Rather than select the next bin pair in a sequential order, we select each bin pair randomly from all remaining candidates. The process is driven by a stream cipher that serves as a secret pseudo-random number generator. The watermark is encrypted so that the order of different bin pairs plays a role in the extraction of the watermark. Given an N-bit watermark to be embedded, the bin pairs selection, watermark embedding and detection processes can be described as follows.

Bin pairs selection: For the i-th bin pair, run the stream cipher to create a random integer $0 \leq \text{index} \leq 256 - 2i$. Then pick the index-th unused bin as the first bin a_i. Then, run the stream cipher to create a new integer $\max(-9, -a_i) \leq \text{step} \leq \min(255 - a_i, 9)$. Pick the $(a_i + \text{step})$-th bin as the second bin b_i. If b_i has been used or if the two bins have the same height, re-generate a new index and a new step until two valid bins are selected to form a new bin pair.

Watermark embedding: First encrypt the watermark $W = \{w_i\}_{i=1}^{N}$ by the stream cipher to get $W^* = \{w_i^*\}_{i=1}^{N}$. The heights of the two selected bin pairs a_i and b_i should encode w_i^* as follows: if $w_i^* = 1$, hist$(a_i) <$ hist(b_i) should hold, and if $w_i^* = 0$, hist$(a_i) >$ hist(b_i) should hold, where 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.

Watermark extraction: First, reconstruct the same sequence of bin pairs $\{a_i, b_i\}_{i=1}^{N}$ at the detector side. Then, extract the encrypted watermark as follows: $W^* = \{w_i^*\}_{i=1}^{N}$, where $w_i^* = 0$ if hist$(a_i) >$ hist(b_i) and $w_i^* = 1$ if hist$(a_i) <$ hist(b_i). Finally, decrypt W^* to recover the plaintext watermark W.

The capacity of the above modified scheme is limited to the number of candidate bin pairs, which is upper bounded by 128 bits. Figure 1 shows the results of embedding a 64-bit watermark "12345678" into the blue channel of the test image "baboon" by using the modified watermarking scheme and by the original watermarking scheme.

We ran both watermarking schemes on the Kodak color image database[1] and measured the quality of the watermarked images by using ten objective visual quality assessment (VQA) metrics included in the MeTriX MuX VQA Package[2]. The results show that our new change to the bin pair selection

[1]Available online at `http://r0k.us/graphics/kodak`

[2]Available online at `http://foulard.ece.cornell.edu/gaubatz/metrix_mux`

(a) (b) (c)

Fig. 1. The result of applying the two watermarking watermarking schemes to embed a 64-bit watermark "12345678": (a) plaintext image; (b) image watermarked by the modified scheme (PSNR $= 42.57$); (c) image watermarked by the original scheme (PSNR $= 42.36$).

process does not compromise the visual quality of the watermarked image. To be more exact, the mean of the visual quality measured by all the ten VQA metrics remains similar for both schemes but our scheme seems to have a smaller variance in the measured visual quality, which can be partly explained by the stronger random effect of the bin selection process. See Fig. 2 for the visual quality indices of 24 images watermarked by the two schemes, measured by two typical objective VQA metrics – PSNR and SSIM.

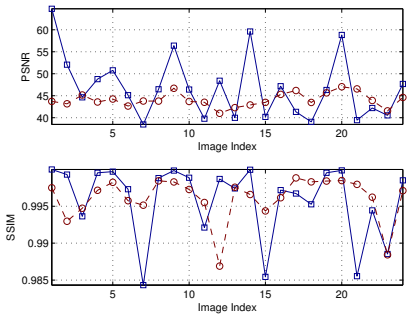


Fig. 2. Visual quality comparison of the modified watermarking scheme and the original one, measured by PSNR and SSIM. The solid line marked with squares corresponds to the original scheme, and the dashed line marked with circles corresponds to the modified scheme.

B. Encryption Part

A permutation cipher acting on an $W \times H$ image can be modeled by a $W \times H$ permutation matrix $\boldsymbol{M} = \{m(x,y) = (x',y')\}_{\substack{0\le x,x'\le W-1\\ 0\le y,y'\le H-1}}$, where (i',j') denotes the new location of the pixel (i,j) after permutation [7, Sec. 2].

Many researchers have suggested iterating a parameterized 2-D discrete map to generate the permutation matrix. The average and worst-case complexity of such an approach is always $O(nWH)$, where n is the number of iterations.

For our prototype implementation of the proposed CWE framework, we choose Arnold's cat map [8], which was proposed by several researchers for encrypting square images [9], [10]. Given an $H \times H$ square image, one of its discretized versions [10] is defined as follows:

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix} \mod H. \qquad (2)$$

where a and b are parameters that can serve as the secret key if the function is used for encryption purposes.

IV. Security Analysis

As the watermarking and encryption schemes deployed are completely independent, they do not interfere with each other and their security can be assessed separately.

A. Watermarking Part

It is well known that histogram based watermarks are resistant against geometric attacks since the histogram is largely invariant to geometric transformations. More precisely, according to [11], the histogram is preserved by all image transformations $\Psi_t : D \to \mathbb{R}^2$, where $D \subset \mathbb{R}^2$ is the domain of the image and $t \in \mathbb{R}$ is the parameter of the transformation, with the property $\operatorname{div}\left(\frac{d}{dt}\Psi_t\right) = 0$.

The keyspace of the watermarking scheme described in Sec. III-A depends on the length N of the embedded watermark. It can be shown that for $N \geq 8$, the keyspace contains more than 2^{89} keys. For larger N, this number increases rapidly . Unfortunately, the watermark cannot withstand a malicious attacker removing it by randomly swapping neighbouring histogram bins. Such an attack resembles the original technique to embed the watermark and will not visibly decrease image quality. This problem seems unavoidable, however, when watermarking strongly encrypted images. Since all perceptual information is destroyed by the encryption, the watermark embedder can only modify image features that are likely to be unimportant for perception. The same thing can be done by an attacker, however.

B. Encryption Part

A quantitative study of plaintext attacks against permutation-based ciphers has been performed in [7], where it is shown that for an $H \times H$ square image with L grey levels $O(\log_L H^2)$ known plaintexts are sufficient to recover half of the ciphertext pixels. The computational complexity of these attacks is $O(p \cdot H^4)$, where p is the number of known ciphertexts used, making these attacks practical. Therefore, we propose to use image-varying keys, e.g. image-dependent keys derived from the (normal or visual) hash of the image. The key is divided into one long-term secret master key and one short-term public image-dependent session key. The latter can be embedded into the encrypted image by using a reversible information hiding scheme such as those described in Sec. II-B. It is combined with the secret master key to form the key for decrypting the image.

V. Complexity Analysis

Without loss of generality, we assume that the plaintext image is an $H \times H$ image with L grey levels, that the watermark is an N-bit pattern, and that N is much smaller than H^2, so that the derived complexity can be more compact. In addition, we only consider the average complexity because the worst-case complexity can be quite different and less meaningful.

A. Watermarking Complexity

Generating the histogram corresponds to H^2 operations. To select N bin pairs from the histogram, $\approx 2N$ operations are needed. To embed all the N bits, averagely $N/2$ bin pairs need swapping (i.e., N bins needs processing), whose complexity is NH^2/L. To detect the watermark, only N comparisons of bin heights are needed. To sum up, the overall computational complexity of the watermark embedding process is $O(2N + (N/L+1)H^2) \approx O((N/L+1)H^2)$ and that of the watermark detection process is $O(3N + H^2) \approx O(H^2)$.

B. Encryption/Decryption Complexity

Since any permutation cipher can be represented by an $H \times H$ permutation matrix, the complexity of encrypting an image requires merely H^2 look-up table operations and H^2 assignments. Iterating the discrete 2-D cat map n times requires nH^2 look-up table operations and pixel value assignments. Generating the permutation matrix requires $3H^2$ multiplications and $3H^2$ addition/assignments. We can ignore the $3H^2$ additions/assignments, because multiplications are computationally much heavier. Thus, the overall complexity becomes $O((n+5)H^2) \approx O(nH^2)$. When the sorting based approach is used to generate the permutation matrix, the overall complexity is $O(2(\log_2(H)+1)H^2) \approx O(\log_2(H)H^2)$. Since the decryption can be done by using the same matrix, the computational complexity remains the same.

C. Comparison with Existing Schemes

Table I shows the complexity comparison of our proposed CWE scheme and some existing schemes in the literature. These schemes follow the other two approaches to CWE.

TABLE I
COMPLEXITY COMPARISON

Scheme	Watermarking embedding complexity	Watermarking detection complexity	encryption complexity
Proposed CWE scheme*	$O((N/L+1)H^2)$	$O(H^2)$	$O(nH^2)$
Homomorphic CWE schemes in [2, Sec. 9.3]	$O(H^2)$	$O(H^2)$	$O(H^2)$
Partial encryption based CWE scheme in [12]**	$O(mH^2)$	$O(mH^2)$	$O(mH^2)$

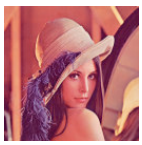
*: n is the number of iterations of the cat map
**: m denotes the length of the low-pass and high-pass wavelet filters.

VI. Experimental Results

Table II shows the results of applying the proposed CWE scheme to the Lenna image. The difference image between the watermarked-encrypted image and the encrypted-watermarked image is an all-zero image, as to be expected. For watermarking, a 64-bit message "12345678" was embedded. For encryption, Arnold's cat map with parameters $a = 8, b = 7$, $n = 5$ was used.

The watermark could be successfully extracted either from the encrypted marked image $\mathcal{E}_K(\mathcal{M}(I,m))$ (Property 3) or from the marked encrypted image $\mathcal{M}(\mathcal{E}_K(I),m)$ (Property 2).

TABLE II
EXPERIMENTAL RESULTS

Plaintext Image I	$\mathcal{E}_K(\mathcal{M}(I,m))$	$\mathcal{M}(\mathcal{E}_K(I),m)$	**Difference Image**
			

In all cases, decrypting either $\mathcal{M}(\mathcal{E}_K(I),m)$ or $\mathcal{E}_K(\mathcal{M}(I,m))$ leads to the marked plaintext image $\mathcal{M}(I,m)$, from which the watermark could still be successfully extracted (Property 4).

VII. Conclusion and Further Work

We have presented a novel approach to building commutative watermarking-encryption schemes which is based on permutation-only ciphers and non-localized watermarking schemes. A concrete CWE scheme was designed by combining a histogram based watermarking scheme with a permutation cipher based on a discrete 2-D chaotic map.

In our future work, we will study a possible generalization of the proposed CWE scheme to the compressed domain, where the key questions will be how to apply permutations without compromising compression efficiency and how to make the watermarking scheme more robust to lossy compression. We will also investigate if reversible CWE schemes can be designed within the proposed framework.

References

[1] J. Herrera-Joancomartí, S. Katzenbeisser, D. Megías, J. Minguillón, A. Pommer, M. Steinebach, and A. Uhl, "ECRYPT European Network of Excellence in Cryptology, First Summary Report on Hybrid Systems," 2005. [Online]. Available: http://www.ecrypt.eu.org/ecrypt1/documents/D.WVL.5-1.0.pdf

[2] S. Lian, *Multimedia Content Encryption*. CRC Press, 2009.

[3] ——, "Quasi-commutative watermarking and encryption for secure media content distribution," *Multimedia Tools and Applications*, vol. 43, no. 1, pp. 91–107, 2009.

[4] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–361, 2006.

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

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

[7] S. Li, C. Li, G. Chen, N. G. Bourbakis, and K.-T. Lo, "A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Processing: Image Communication*, vol. 23, no. 3, pp. 212–223, 2008.

[8] V. I. Arnold and A. Avez, *Ergodic Problems of Classical Mechanics*. Benjamin, 1968.

[9] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. of Bifurcation and Chaos*, vol. 8, pp. 1259–1284, 1998.

[10] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons and Fractals*, vol. 21, no. 3, pp. 749–761, 2004.

[11] E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar, "Histogram preserving image transformations," *Int. J. of Computer Vision*, vol. 45, no. 1, pp. 5–23, 2001.

[12] S. Lian, Z. Liu, R. Zhen, and H. Wang, "Commutative watermarking and encryption for media data," *Optical Engineering*, vol. 45, no. 8, 2006.