Schmitz *EURASIP Journal on Information Security* (2021) 2021:1
https://doi.org/10.1186/s13635-020-00115-w

EURASIP Journal on Information Security

RESEARCH Open Access

Use of SHDM in commutative watermarking encryption

Roland Schmitz

Abstract

SHDM stands for Sphere-Hardening Dither Modulation and is a watermarking algorithm based on quantizing the norm of a vector extracted from the cover work. We show how SHDM can be integrated into a fully commutative watermarking-encryption scheme and investigate implementations in the spatial, DCT, and DWT domain with respect to their fidelity, robustness, capacity, and security of encryption. The watermarking scheme, when applied in the DCT or DWT domain, proves to be very robust against JPEG/JPEG2000 compression. On the other hand, the spatial domain-based approach offers a large capacity. The increased robustness of the watermarking schemes, however, comes at the cost of rather weak encryption primitives, making the proposed CWE scheme suited for low to medium security applications with high robustness requirements.

Keywords: Watermarking, Encryption, DRM

1 Introduction

Encryption and watermarking are both 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 for various purposes, ranging from ensuring authenticity of content to embedding metadata, e.g., copyright or authorship information, into the contents. In the DRM context, for example, clients need to have the ability to decrypt the contents and may thus eventually misuse the ciphertext contents. The protection provided by digital watermarks, however, remains within the contents, and can serve to identify misbehaving clients.

In a buyer-seller scenario where the content owner does not trust the seller to sell copies of her own, the content owner can supply the seller with an encrypted version of the content, which is in turn individually watermarked for each buyer by the seller. In such a situation, it is important that a watermark can be embedded in the encrypted domain and detected in the cleartext domain.

Correspondence: schmitz@hdm-stuttgart.de
Department of Computer Science and Media, Stuttgart Media University, Nobelstrasse 10, D-70569 Stuttgart, Germany

Another motivation to consider watermarking in the encrypted domain are the increasing needs generated by cloud computing platforms and various privacy preserving applications.

In [1], four requirements on watermarking in the encrypted domain are formulated:

- *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 plaintext media data, and m is the mark to be embedded.

In recent years, a number of commutative watermarking-encryption (CWE) schemes have been formulated. The present paper describes a novel CWE scheme, that couples sign-bit encryption of selected pixel grey values or transform coefficients as encryption part with Sphere-Hardening Dither Modulation (SHDM) [2, 3] as the watermarking part. The encryption part can optionally be enhanced by permuting the pixels or transform coefficients, respectively. While the idea of encrypting coefficient sign-bits within a CWE scheme is not new, the use of SHDM as watermarking part is, leading to a more robust scheme than previous approaches.

The rest of the paper is organized as follows: previous CWE proposals are summarized in Section 2. SHDM is briefly reviewed in Section 3. In Section 4, we describe implementations of the proposed CWE scheme in the spatial, DCT, and DWT domain, respectively, and discuss their security. Section 5 provides experimental results on the robustness of the watermarking part in the three implementation domains, and Section 6 discusses the security aspects of the CWE schemes in terms of cryptographic and watermarking security. Section 7 concludes the paper.

2 Related work

There are currently three basic approaches to commutative watermarking encryption for raw image data. The first approach, called *partial encryption*, divides the image data into two parts and encrypts one of them (typically the perceptually more important part) and watermarks the other part. Thus, encryption part and watermarking part are completely independent and do not interfere with each other. First, important examples in this vein are provided by [4] and [5]. In [5], the basic idea is to encrypt DCT sign bits and to watermark their absolute values by means of dithered modulation. Similarly, in [4], the data are partitioned into two parts after a four-level discrete wavelet transformation. The low-level coefficients are fully encrypted, while in the medium- and high-level coefficients only the signs are encrypted and their absolute values are watermarked. In [6], encryption and watermarking happens within a secret transform domain, the Tree-Structured-Haar (TSH) Transform, which involves a secret parameter. Both the watermark embedder and encryptor need to share knowledge about the secret parameter generating the transform domain. The transform coefficients are first quantized to get B bitplanes. The N most significant bitplanes are encrypted, and $B - N - 1$ of the remaining bitplanes are watermarked. The least significant bitplane is replaced by the signs of the plaintext coefficients. In this approach, using a secret transform domain increases the security of the scheme, albeit at the cost that encryption and watermarking are not completely independent, but need to share a common secret.

Another 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, and addition modulo n (including the bitwise XOR operation). One major drawback of this approach is the influence of encryption on robustness of the watermarking algorithm: after strong encryption, there is no visual information available for the watermark embedder to adapt itself to in order to increase robustness while at the same time minimizing visual quality degradation [7, Sec. 9.4]. Another drawback is that the homomorphic watermarking operation can seriously affect the fidelity. In [8], for example, addition modulo n is used, where n is the number of grey values. However, the modular addition operation may cause overflow/underflow pixels that are handled in a preprocessing step during the encryption operation, making the system only "quasi-commutative."

Third, in *invariant encryption* schemes, the data are fully encrypted, but the encryption operation leaves a certain subspace of the data invariant, which may be used for watermarking. In [9], a permutation cipher is applied to the image, leaving the histogram of grey values invariant. The watermark is embedded by manipulating the histogram. Depending on viewpoint, schemes based on encrypting sign bits of transform coefficients may also be seen as invariant encryption schemes, as the absolute values of the coefficients form an invariant subspace.

In a another line of work, researchers concentrate their efforts on watermarking and encrypting the bitstream after encoding the data according to a certain standard. In [10], a commutative watermarking encryption scheme based on encrypting the intra-prediction modes and the sign bits of the DCT coefficients within the H.264 bitstream is presented. Here, watermarking of residual DCT coefficients is based on Quantized Index Modulation (QIM) [11]. In [12], in order to achieve the commutative property, one set of syntax elements within the HEVC bitstream is utilized for data hiding, while another set is exploited for encryption. In [13] and [14], the JPEG-LS bitstream is jointly watermarked and encrypted using the AES algorithm in Cipher Block Chaining (CBC)-mode, making this scheme suitable for scenarios with high security requirements, like in medical imaging.

3 SHDM

Sphere-Hardening Dither Modulation, or SHDM for short, was proposed by Balado in [2] and [3] as an alternative to STDM (Spread-Transform Dither Modulation), which was proposed in [11]. Both SHDM and STDM have in common that in order to embed a single bit b, a multidimensional host vector $\vec{x}$ is extracted from the cover work C_0 and modified using some dithered quantization function Q_b, where different message bits lead to different dither values. While in STDM the projection $\vec{x}^t \cdot \vec{u}$ of the host vector $\vec{x}$ onto some random vector $\vec{u}$ is quantized, in SHDM, the norm $\|\vec{x}\|$ is quantized.

More specifically, the embedding rule in SHDM is given by

$$\vec{y} = Q_b(\|x\|, \Delta, d) \cdot \frac{\vec{x}}{\|x\|}, \tag{2}$$

where for $b \in \{0, 1\}$,

$$Q_b(\|x\|, \Delta, d) = \Delta \cdot \lfloor \frac{\|x\| - d - b\Delta/2}{\Delta} \rfloor + d + b \cdot \Delta/2 \tag{3}$$

is the quantizing function. Extraction of the embedded bit is done via

$$b = \arg \min_{b \in \{0,1\}} |\|\tilde{\vec{y}}\| - Q_b\left(\|\tilde{\vec{y}}\|, \Delta, d\right)|, \tag{4}$$

where $\tilde{\vec{y}}$ is the disturbed signal vector at the detector site.

Note that the direction of the signal vector $\vec{x}$ is not changed by embedding, which is advantageous from a perceptual point of view, as opposed to related methods like STDM. As is shown in [2], SHDM offers the same level of robustness against additive white noise as STDM.

4 Using SHDM in a CWE scheme

In SHDM, the signal vector $\vec{x} = (x_1, \ldots, x_N)$ may be extracted from the host in an arbitrary fashion. In this section, we implement SHDM in the spatial (pixel) domain, the DCT domain, and the DWT domain and combine it with matching encryption schemes. As in SHDM the vector norm

$$\|\vec{x}\| = \sqrt{x_1^2 + x_2^2 + \ldots x_N^2}, \tag{5}$$

is quantized, we have the following options for encryption:

- Encrypt the sign bits of the x_i by means of a stream cipher.
- Permute the x_i.
- Apply some other norm-preserving operation on $\vec{x}$, e. g. a random rotation.

Of course, the options may be combined. In the rest of the paper, we will only explore the first two options.

4.1 Implementation in the spatial domain

In the spatial domain, we work directly with pixel grey values ranging initially between 0 and 255. In order to create sign bits, we subtract 128 from each grey value, so that the new range is $-128 \leq 0 \leq 127$.

4.1.1 *Watermarking part*

The watermarking part uses the following parameters:

- The watermarking key W_K.
- The watermark $W = (b_1, \ldots, b_n)$ to be embedded.
- The dimension of the host vectors $\vec{x}_i$, i.e., the number of coefficients N into which one bit b_i is embedded.
- The quantizing step Δ.

In the spatial domain, we assume that the watermarking key consists of two parts: $W_K = \left(W_K^{(1)}, W_K^{(2)}\right)$. After choosing a step size Δ and N, the embedding process consists of the following steps:

- For each bit b_i to be embedded, randomly select N pixels. The selection is controlled by $W_K^{(1)}$. The corresponding grey values form the signal vector $\vec{x}_i$.
- Randomly generate a dither value d_i, controlled by $W_K^{(2)}$.
- Quantize the norm of $\vec{x}_i$ according to b_i:

$$\|\vec{x}_i\|_q = Q_{b_i}\left(\|\vec{x}_i\|, \Delta, d_i\right) \tag{6}$$

- Embed the mark into $\vec{x}_i$ by changing its norm to $\|\vec{x}_i\|_q$:

$$\vec{y}_i = \|\vec{x}_i\|_q \cdot \frac{\vec{x}_i}{\|\vec{x}_i\|} \tag{7}$$

  and replace the grey values of pixels corresponding to components of $\vec{x}_i$ by the corresponding entries in $\vec{y}_i$.

For extraction of bit b_i, the disturbed signal vector $\tilde{\vec{y}}_i$ is formed from the marked image C_W in the same way as $\vec{x}_i$ was generated from the host image C_0. The norm of $\tilde{\vec{y}}_i$ is quantized and b_i is computed according to

$$b_i = \arg \min_{b_i \in \{0,1\}} |\|\tilde{\vec{y}}_i\| - Q_{b_i}\left(\|\tilde{\vec{y}}_i\|, \Delta, d_i\right)|. \tag{8}$$

Figure 1 shows two embedding examples with different resolutions (512×512 and 800×1600, respectively), where a random 64-bit watermark was embedded using a quantization step size of $\Delta = 75$.

4.1.2 *Encryption part*

Images in the spatial domain with grey values ranging between 0 and 255 can be represented by eight so-called bitplanes, where the most significant bitplane (MSB) indicates whether the grey value of a certain pixel is greater than 127 or not. Thus, after subtracting 128 from every grey value, the MSB indicates the sign of the grey values. Sign bit encryption is therefore equivalent to encrypting the MSB by means of a stream cipher.

The security of encrypting the MSB of an image has been investigated in [15]. Not only is the amount of image

Fig. 1 Embedding 64 bits in the spatial domain. **a** PSNR = 57.41 dB. **b** PSNR = 79.16 dB

quality degradation insufficient for most applications, it is also possible to estimate the encrypt sign bits based on the assumption that neighboring grey values in a natural image do not change abruptly. Both problems can be remedied if a permutation cipher is applied on the pixels in addition. However, the permutation must not mix the N pixels used for embedding b_i with the N pixels used for embedding a different bit b_j. Therefore, the permutation cipher and the watermark embedder must share knowledge of $W_K^{(1)}$, which is the part of W_K governing pixel selection. If this condition is met, watermarking and (permutation-based) ciphering commute. However, as is well-known, permutation ciphers are vulnerable to known plaintext attacks (see [16] for a quantitative analysis).

Moreover, in order to have as many permutation as possible, N should be chosen as large as possible, that means

$$N = \lfloor \frac{H \cdot W}{n} \rfloor \tag{9}$$

in the spatial domain, where H and W are the height and width of the host image C_0, and n is the length of the embedded string. In order to have a minimum level of security against brute-force attacks, we need $N \geq 32$, leading to a maximum capacity of

$$n_{max} = \lfloor \frac{H \cdot W}{32} \rfloor \tag{10}$$

bits in the spatial domain, meaning, e.g., 2^{13} bits for a 512×512 image (note that the term *capacity* is used throughout this paper according to the definition given in [17]: *the watermarking capacity of digital image is the number of bits that can be embedded in a given host image.*).

Figure 2 shows encrypted versions of the marked Lena image in Fig. 1a. Thanks to the commutativity of watermarking and encryption, the mark can be extracted from both without errors.

4.2 Implementation in the DCT domain

In the DCT domain, we assume that the watermarking key consists of three parts: $W_K = \left(W_K^{(1)}, W_K^{(2)}, W_K^{(3)}\right)$. We begin by performing a block-based two-dimensional DCT on the host image C_0, i.e., we divide C_0 into non-overlapping 8×8 pixel blocks and perform a two-dimensional DCT on each block.

4.2.1 Watermarking part

- For each bit b_i to be embedded, randomly select N blocks. The selection is controlled by $W_K^{(1)}$. Each block can only be selected once. The selected blocks for bit b_i form subset T_i of the set of all blocks.
- For each T_i, randomly select a horizontal and a vertical frequency index from the medium frequencies. The selection is controlled by $W_K^{(2)}$. The corresponding DCT-coefficients from the selected blocks form an N-dimensional vector $\vec{x}_i$.
- For each T_i, randomly generate a dither value d_i under control of $W_K^{(3)}$.
- Quantize the norm of $\vec{x}_i$ according to b_i:

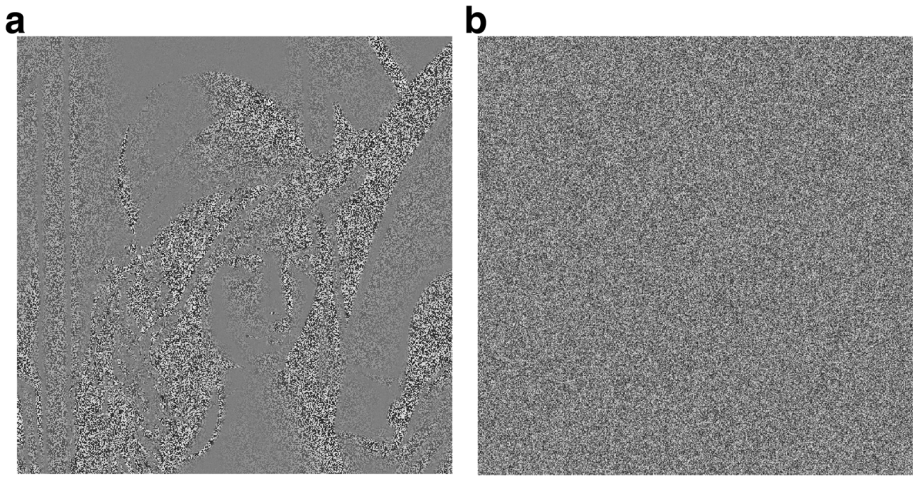

Fig. 2 Encrypting the marked Lena image in the spatial domain. **a** Sign bit encryption. **b** Permutation cipher

$$\|\vec{x}_i\|_q = Q_{b_i}\left(\|\vec{x}_i\|, \Delta_i, d_i\right) \tag{11}$$

Note that because all DCT coefficients in $\vec{x}_i$ correspond to the same horizontal and vertical frequency pair, it is possible to choose individual quantizing step sizes Δ_i according to their perceptual importance (see below).

- Embed the mark into $\vec{x}_i$ by changing its norm to $\|\vec{x}_i\|_q$:

$$\vec{y}_i = \|\vec{x}_i\|_q \cdot \frac{\vec{x}_i}{\|\vec{x}_i\|} \tag{12}$$

and replace the selected DCT coefficients in T_i with the corresponding entries in $\vec{y}_i$.

In choosing the Δ_i step sizes, we were led by the JPEG quantization matrix, which assigns a perceptual relevance to each DCT coefficient in an (8×8) block. The essential step in JPEG compression consists in quantizing DCT-coefficients according to fixed quantization tables corresponding to a certain quality factor. On the other hand, it is well known that QIM-based watermarking schemes are sensitive to re-quantization.

In order to counter the adverse effects of re-quantization, we therefore chose quantization step sizes Δ_i for the individual DCT coefficients selected for embedding bit b_i that were oriented at the actual quantization step sizes in the JPEG standard. More specifically, we used the quantization matrix

$$J = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 36 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

taken from the JPEG standard ([18]), which gives the JPEG quantization steps for the DCT coefficients within a 8×8 block referring to a 50% quality factor, and multiplied it with a constant $c > 1$. If the DCT coefficients for embedding b_i correspond to frequencies (k, ℓ), we have

$$\Delta_i = c \cdot J_{k\ell}, \tag{13}$$

the rationale behind this approach being the well-known fact that for a quantizing function Q, we have

$$Q(Q(Q(x, \Delta), \delta), \Delta) = Q(x, \Delta), \tag{14}$$

if $\Delta > \delta$ (see [19], Theorem 1). This means that quantizing some value y with step size δ can be reversed by another quantization with a larger step size Δ.

Figure 3 shows two embedding examples, where a random 64-bit message was embedded. For the Lena image, we set $N = 64, c = 3.5$, and for the higher resolution Norba image, we set $N = 312, c = 3.5$ (cf. Section 4.2.2 for details on how N was chosen).

In order to extract message bit b_i, the disturbed marked signal vector $\tilde{\vec{y}}_i$ is extracted from the marked image C_W in the same way as the unmarked vector $\vec{x}_i$ was built from C_0 with the help of W_K. The message bit is then decoded according to

$$b_i = \arg\min_{b_i \in \{0,1\}} \left| \|\tilde{\vec{y}}_i\| - Q_{b_i}\left(\|\tilde{\vec{y}}_i\|, \Delta_i, d_i\right) \right|. \tag{15}$$

4.2.2 Encryption part

As in the spatial domain, we investigate the two options of encrypting DCT-coefficient sign bits and of permuting them. The idea of encrypting the sign bits of DCT coefficients goes back to [20] and [21], where it is proposed to encrypt sign bits of DCT coefficients and motion vectors in MPEG video. The security of this approach for still images is classified as low in [15], p. 51.

Fig. 3 Embedding 64 bits in the DCT domain. **a** $N = 64$, PSNR = 56.10 dB. **b** $N = 312$, PSNR = 63.74 dB

In order to create a larger visual distortion, instead of permuting the DCT coefficients alone, we permuted the complete (8×8)−blocks containing the coefficients (see [20] and [21]). If only those blocks containing the selected coefficients for watermarking are permuted, however, the corresponding subset T becomes visible to an attacker, who can in turn concentrate her efforts to remove the mark on T. We therefore need to permute *all* image blocks. Moreover, as in the spatial domain case, in order to make sure that the selected blocks T_i for a single bit b_i do not get mixed up with blocks for a different bit or non-selected blocks, the permutation algorithm needs to know part $W_K^{(1)}$ of the watermarking key. More specifically, each subset T_i needs to form an invariant subset of the set of all blocks under the permutation. As in the spatial domain, these subsets need to be as large as possible. We therefore have

$$N = |T_i| = \lfloor \frac{(H/8) \cdot (W/8)}{n} \rfloor \tag{16}$$

in the DCT domain. The requirement $N \geq 32$ gives a maximum capacity of

$$n_{max} = \lfloor \frac{H \cdot W}{64 \cdot 32} \rfloor, \tag{17}$$

meaning 128 bits for a 512×512 image.

Figure 4 shows encrypted versions of the marked Lena image in Fig. 4. Again, the mark can be extracted from both without errors.

4.3 Implementation in the DWT domain

4.3.1 Watermarking part

In the DWT domain, we performed a three-level DWT and embedded the mark into the level 3 approximation coefficients. This way, the number N of coefficients used to embed one bit is the same as in Section 4.2, namely

$$N = |T_i| = \lfloor \frac{(H/8) \cdot (W/8)}{n} \rfloor. \tag{18}$$

In the DWT-case, however, there are no blocks of coefficients to choose a frequency from, thus W_K consists of only two parts: $W_K = \left(W_K^{(1)}, W_K^{(2)}\right)$, where $W_K^{(1)}$ governs the selection of N coefficients for each message bit b_i, and $W_K^{(2)}$ controls the dither d_i for each bit. Likewise, a single quantization step size Δ is used for all message bits. As an example, Fig. 5 shows the results of embedding 64 bits into the Lena and Norba image, setting $\Delta = 100$.

4.3.2 Encryption part

As in the DCT case, we have the options to either encrypt the sign bits of DWT coefficients, as already proposed in [21], and/or to permute the DWT coefficients, as originally proposed in [22]. Note that in the DWT domain, permuting coefficients is not as vulnerable to known-plaintext attacks as in other domains, because the location of coefficients is image-dependent [15]. However, if only the level 3 approximation coefficients are encrypted or permuted, the image content is not rendered completely unintelligible, but fine structures are still visible (see

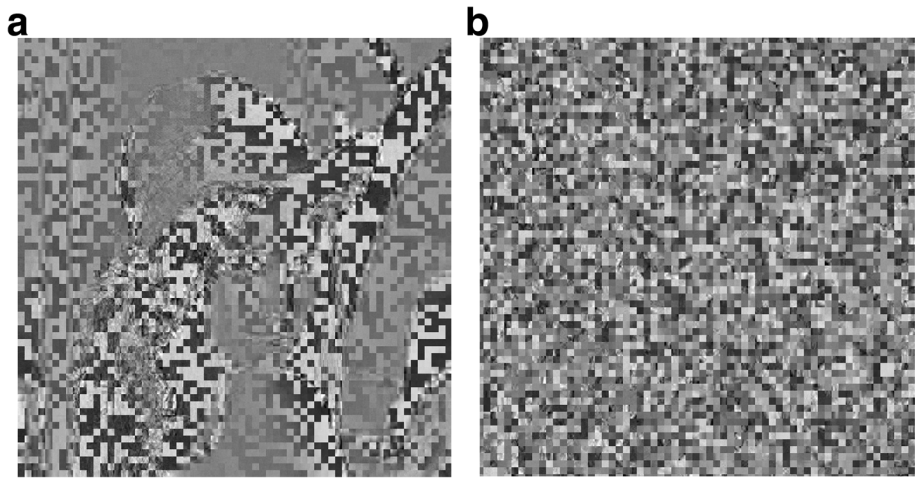

Fig. 4 Encrypting the marked Lena image in the DCT domain. **a** Coefficient sign bit encryption. **b** Permutation of 8 × 8 blocks

Figs. 6a, b). As in the DCT-case, we have the additional option of not only permuting the level 3 DWT-coefficients themselves but the complete (8 × 8) blocks leading to the level 3 approximation for a more complete obfuscation of the image content (see Fig. 6c), without sacrificing the commutativity with watermarking. The maximum capacity is the same as in the DCT-based implementation.

5 Experimental results

In our experiments, we used 50 standard images of format 512 × 512, most of them downloaded from http://decsai.ugr.es/cvg/CG/base.htm. We embedded 64 random bits and fixed all other parameters in such a way that a PSNR of about 50dB resulted for the watermarked images. In the spatial domain and the DWT domain, this meant a quantizing step size of $\Delta = 175$ (see Section 5.1 for details).

In the DCT domain, the c-Parameter (see Section 4.2) was set to 8.0. The similarity of the extracted watermarks to the originally embedded watermarks was measured using the normalized correlation of the two vectors.

Fig. 5 Embedding 64 bits in the DWT domain.**a** $N = 64$, PSNR = 54.84 dB. **b** $N = 312$, PSNR = 62.94 dB

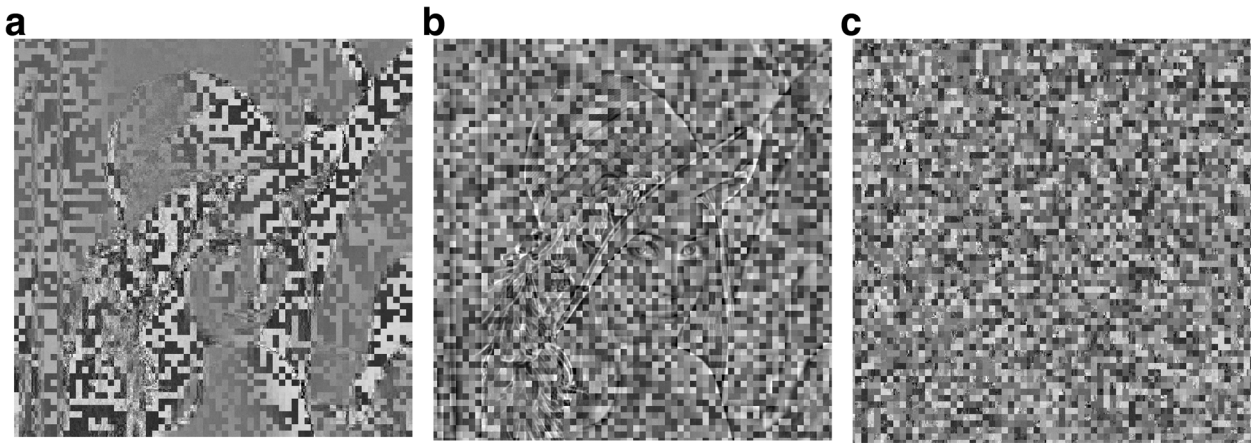

Fig. 6 Encrypting the marked Lena image in the DWT domain. **a** LL_3 coefficient sign bit encryption. **b** Permutation of LL_3 coefficients. **c** Permutation of 8×8 blocks

5.1 Fidelity

We first investigated how the Δ resp. the c-parameter affects the PSNR of the watermarked image compared to the host image. Perhaps not surprisingly, the effect of Δ on the PSNR is practically the same for the spatial domain and the DWT domain (see Fig. 7).

As the c Parameter is not directly comparable to the Δ-parameter for the other two domains, the corresponding graph is shown here in a separate diagram (Fig. 8).

Both Figs. 7 and 8 reveal that a parameter choice of $\Delta = 175$ for spatial and DWT domain and of $c = 8.0$ for the DCT domain give rise to a PSNR of about 50 dB, if 64 bits are embedded. This provides the basic setting for our further experiments.

In another fidelity experiment, we investigated the influence of the message size on the PSNR (see Fig. 9). Again, the spatial domain and DWT-based implementations show almost equal behavior, except that the spatial domain scheme has much a larger capacity.

5.2 JPEG compression

Both the DCT - and the DWT-based implementations prove to be very robust against JPEG compression (see Fig. 10). Both schemes also outperform the scheme proposed in [6] with respect to JPEG compression, which offers a normalized correlation of 0.22 at a JPEG quality factor of 50%.

5.3 JPEG2000 compression

The results of our experiments with JPEG2000 compression basically follow the same pattern as the JPEG experiments. The spatial domain implementation is the most

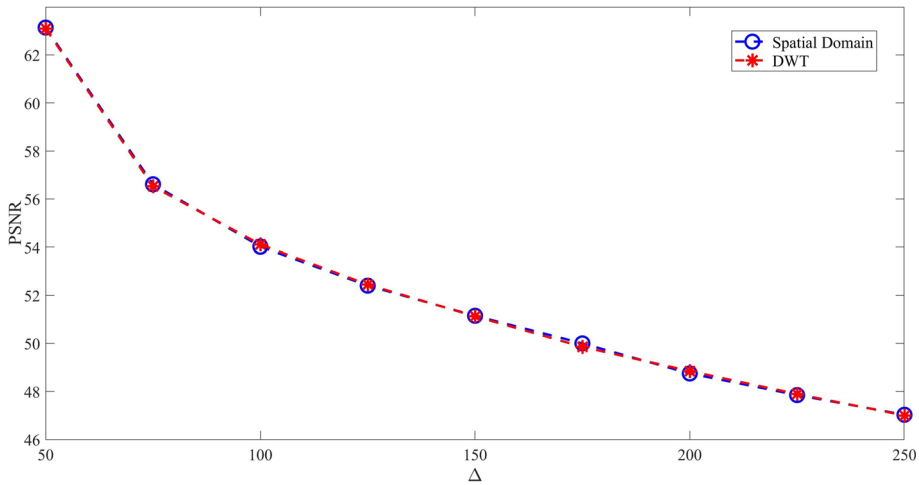

Fig. 7 PSNR versus Δ in the spatial and the DWT domain (averaged over 50 images)

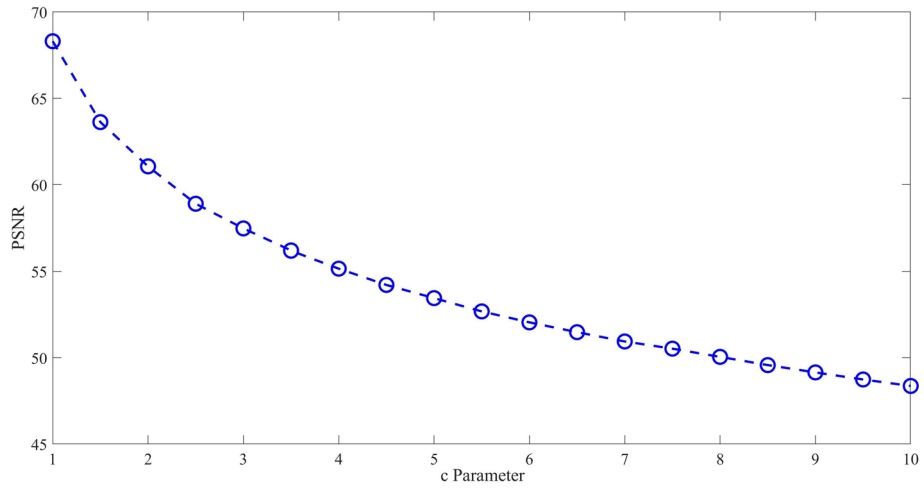

Fig. 8 PSNR vs *c* Parameter (averaged over 50 images)

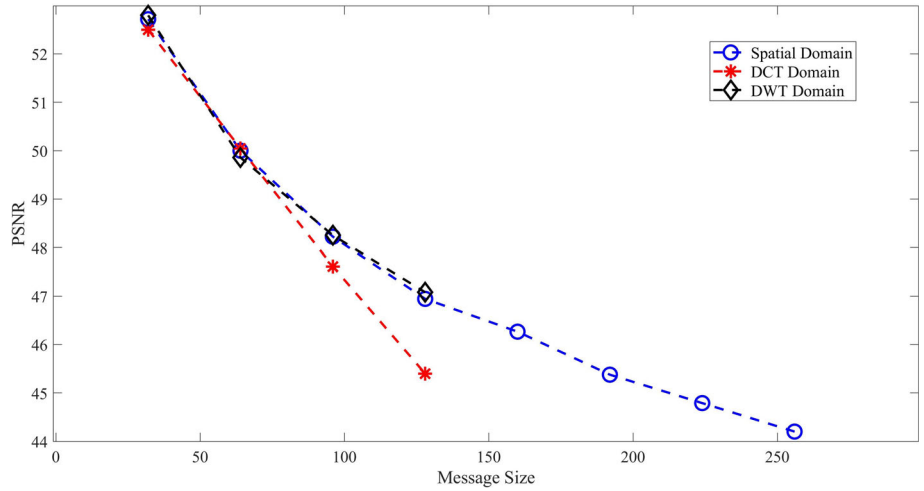

Fig. 9 PSNR vs message size (averaged over 50 images)

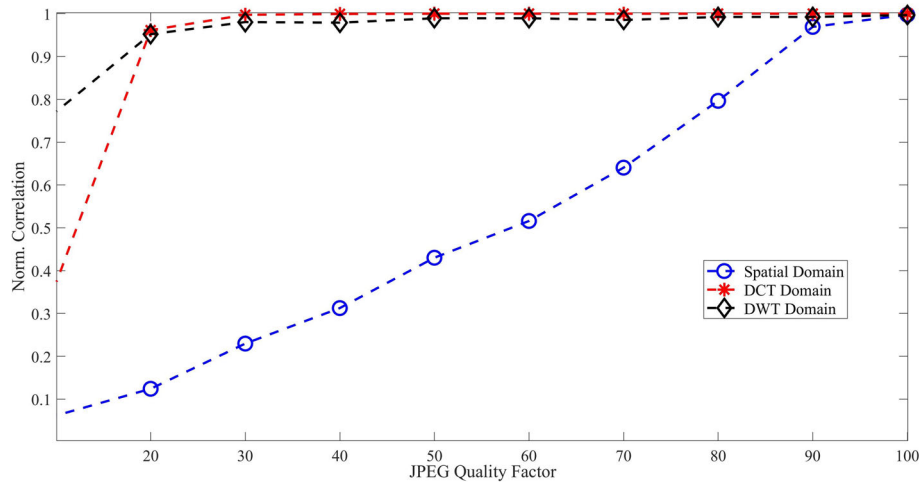

Fig. 10 Correlation value versus JPEG quality factor in three investigated domains (averaged over 50 images)

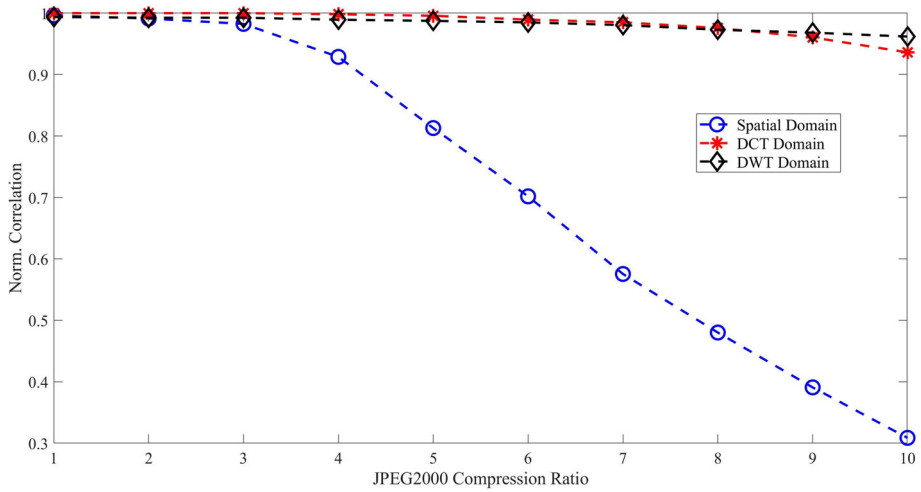

Fig. 11 Correlation value versus JPEG2000 compression ratio in three investigated domains (averaged over 50 images)

fragile one, but is still surprisingly robust, especially at low compression rates.

The DCT-based implementation and the DWT-based implementation perform almost equally well. Only for higher compression rates, the DWT-based implementation has a slight advantage. (see Fig. 11). Again, both transform domain based schemes outperform the scheme presented in [6] and have roughly the same robustness against JPEG2000 compression as the scheme in [4], which works in the LL_4-subband.

5.4 Adding noise

All three implementation domains perform equally well in the presence of low- or medium-density additive white noise. For higher noise densities, the DWT-based implementation is the most robust (see Fig. 12).

6 Security considerations

6.1 Security of encryption

In this subsection, we summarize and enhance the security assessments made in Section 4 for the three implementation domains.

As sign bit encryption in the spatial domain can be attacked directly [15] to reveal part of the image contents, this approach seems to be weakest of all options. Combining it with a permutation cipher makes for a cryptographically and visually stronger cipher, although the permutation cipher is in turn vulnerable to known plaintext attacks. This means, however, to share part of the watermarking key between content owner and seller.

Sign bit encryption in the DCT domain has been attacked by Wu and Kuo [23], who could recover some visual information from the encrypted image by setting

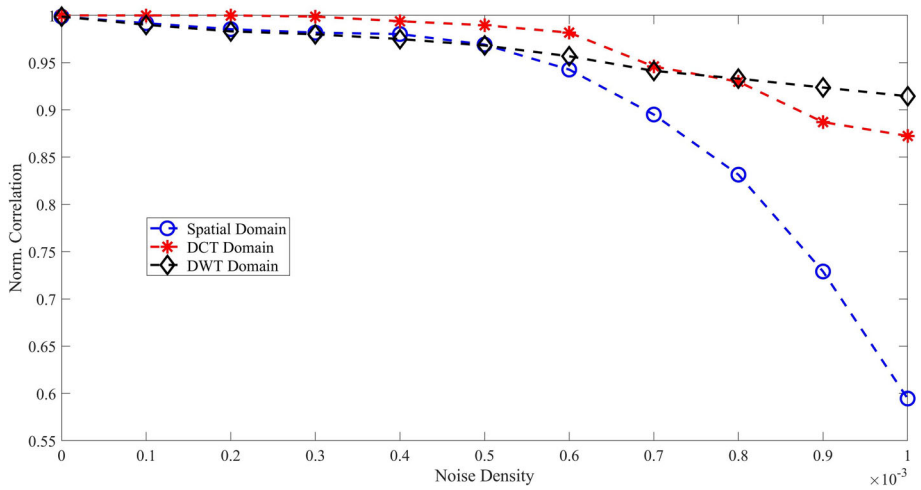

Fig. 12 Correlation value versus noise density in the three investigated domains (averaged over 50 images)

the DC coefficient to 128 and giving all AC coefficients a positive sign. Again, a combination with a block-based permutation will strengthen the security of the cipher (note that we do not recommend to permute DCT coefficients directly, as the DC coefficient will normally stick out as the one with the largest absolute value).

For the DWT domain, there are, to the best of our knowledge, no analogous attacks on sign-bit encryption in the literature. Still, it is recommended to encrypt not only the watermarked DWT coefficients of the LL_3 subband, but all subbands, and combine the sign bit encryption with permutation, if a secret sharing between content owner and seller is possible. If this is not the case, the content owner can resort to permute all subbands excluding the LL_3 subband.

6.2 Watermarking security

According to [24], watermarking security means the occurrence of an adversary trying to break the system, as opposed to random modifications of a marked image due to benign image processing. In the following discussion, we assume that an attacker has access to the unencrypted, marked image C_W, but not to the original host image C_0 or the watermarking key W_K. In this context, breaking the system means that the attacker is either able to insert a mark of her own, or to detect a mark, or to remove the mark from C_W without rendering the image unusable.

In order to successfully detect a watermark or embed a watermark of her own without knowledge of the watermarking key W_K, an attacker would have to guess how the signal vectors are formed as a first step. If the mark is embedded in the DCT- or DWT domain, there are $\binom{(H/8)\cdot(W/8)}{N}$ possibilities for the first bit, where W and H are the dimensions of the image and N is the dimension of the signal vector. For typical values ($H = W = 512, N = 32$), this means about 10^{80} possibilities.

As is shown in Section 5, it is rather hard for an attacker to remove the watermark from a marked image by adding white noise or compression, especially if the mark was embedded in the DCT or DWT domain. Without knowledge of the correct watermarking key, depending on the implementation domain, an attacker would have to modify the value of the pixel grey values or transform coefficients in such way that the norm of each possible signal vector is changed by an amount of at least $\Delta/2$.

7 Conclusion

We have presented a novel commutative watermarking encryption (CWE) scheme, which is very robust to common attacks like JPEG/JPEG2000 compression and noise addition, especially when implemented in some transform domain (Discrete-Cosine or Discrete-Wavelet). On the other hand, the spatial domain implementation has the advantage of a much higher capacity. However, the robustness comes at the cost of relatively weak encryption primitives, especially if the scheme is applied in the spatial or DCT domain. The implementation in the DWT domain offers the best tradeoff between robustness and security of the cipher.

Nevertheless, because of the leakage of visual contents if sign bit encryption is used exclusively, and because of the inherent weaknesses of permutation ciphers, the proposed scheme is recommended for scenarios with low to medium security requirements with regard to the image contents, where robustness of the watermark has the highest priority. For many commercial application scenarios, this seems to be a good fit. In future research, we will explore ways to further enhance the security of the encryption primitives by using norm-preserving operations.

Abbreviations

CBC: Cipher block chaining; CWE: Commutative watermarking encryption; DCT: Discrete cosine transform; DRM: Digital rights management; DWT: Discrete wavelet transform; JPEG: Joint photographic expert group; PSNR: Peak signal-to-noise ratio; QIM: Quantized index modulation; SHDM: Sphere hardening dither modulation; STDM: Spread-transform dither modulation

Acknowledgements

The research described in this article was done during a sabbatical semester granted by the Stuttgart Media University. The author gratefully acknowledges having been given this opportunity. He also thanks the anonymous reviewers for their helpful comments.

Authors' contributions

The entire manuscript is a sole contribution of the author. The author read and approved the final manuscript.

Funding

The author did not receive any funding for this research. Open Access funding enabled and organized by Projekt DEAL.

Availability of data and materials

The experimental results of this study are based on the image data set available at `http://decsai.ugr.es/cvg/CG/base.htm`. The corresponding code is available from the author on request.

Competing interests

The author declares that they have no competing interests.

Received: 7 July 2020 Accepted: 17 September 2020
Published online: 05 January 2021

References

1. J. Herrera-Joancomartí, S. Katzenbeisser, D. Megías, J. Minguillón, A. Pommer, M. Steinebach, A. Uhl, Ecrypt European network of excellence in cryptology, first summary report on hybrid systems (2005). http://www.ecrypt.eu.org/ecrypt1/documents/D.WVL.5-1.0.pdf
2. F. Balado, in *International Workshop on Digital Watermarking*. New geometric analysis of spread-spectrum data hiding with repetition coding, with implications for side-informed schemes Springer, (2005), pp. 336–350
3. F. Balado, N. Hurley, G. Silvestre, in *Security, Steganography, and Watermarking of Multimedia Contents VIII*. Sphere-hardening dither modulation, vol. 6072 International Society for Optics and Photonics, (2006), p. 60720
4. S. Lian, Z. Liu, R. Zhen, H. Wang, Commutative watermarking and encryption for media data. Opt. Eng. **45**(8), 080510 (2006)

5. S. Lian, Z. Liu, Z. Ren, H. Wang, Commutative encryption and watermarking in video compression. IEEE Trans. Circ. Syst. Video Technol. **17**(6), 774–778 (2007)
6. M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. De Natale, A. Neri, A commutative digital image watermarking and encryption method in the tree structured Haar transform domain. Signal Process. Image Commun. **26**(1), 1–12 (2011)
7. S. Lian, *Multimedia content encryption*. (CRC Press, 2009)
8. S. Lian, Quasi-commutative watermarking and encryption for secure media content distribution. Multimedia Tools Appl. **43**(1), 91–107 (2009)
9. R. Schmitz, S. Li, C. Grecos, X. Zhang, in *IFIP International Conference on Communications and Multimedia Security, Lecture Notes in Computer Science*, ed. by B. De Decker, D. Chadwick. A new approach to commutative watermarking encryption, vol. 7394 Springer, (2012), pp. 117–130
10. A. Boho, G. Van Wallendael, A. Dooms, J. De Cock, G. Braeckman, P. Schelkens, B. Preneel, R. Van de Walle, End-to-end security for video distribution: the combination of encryption, watermarking, and video adaptation. IEEE Signal Proc. Mag. **30**(2), 97–107 (2013)
11. B. Chen, G. W. Wornell, Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Trans. Inf. Theory. **47**(4), 1423–1443 (2001)
12. B. Guan, D. Xu, Q. Li, An efficient commutative encryption and data hiding scheme for HEVC video. IEEE Access. **8**, 60232–60245 (2020)
13. S. Haddad, G. Coatrieux, M. Cozic, in *2018 25th IEEE International Conference on Image Processing (ICIP)*. A new joint watermarking-encryption-JPEG-LS compression method for a priori & a posteriori image protection, pp. 1688–1692
14. S. Haddad, G. Coatrieux, A. Moreau-Gaudry, M. Cozic, Joint watermarking-encryption-JPEG-LS for medical image reliability control in encrypted and compressed domains. IEEE Trans. Inf. Forensics Secur. **15**, 2556–2569 (2020)
15. A. Uhl, A. Pommer, *Image and video encryption: from digital rights management to secured personal communication*, vol. 15. (Springer, 2004)
16. S. Li, C. Li, G. Chen, N. G. Bourbakis, K.-T. Lo, A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. Signal Process. Image Commun. **23**(3), 212–223 (2008)
17. F. Zhang, in *Handbook of Research on Secure Multimedia Distribution*. Digital watermarking capacity and detection error rate IGI Global, (2009), pp. 257–276
18. J.-D. Huang, The JPEG standard. Graduate Institute of Communication Engineering National Taiwan University (2006). http://disp.ee.ntu.edu.tw/meeting/%E4%BF%8A%E5%BE%B7/JPEG/JPEG.doc
19. C.-Y. Lin, S.-F. Chang, in *Security and Watermarking of Multimedia Contents II*. Semifragile watermarking for authenticating JPEG visual content, vol. 3971 International Society for Optics and Photonics, (2000), pp. 140–151
20. W. Zeng, S. Lei, in *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)*. Efficient frequency domain video scrambling for content access control, (1999), pp. 285–294
21. W. Zeng, S. Lei, Efficient frequency domain selective scrambling of digital video. IEEE Trans. Multimedia. **5**(1), 118–129 (2003)
22. T. Uehara, R. Safavi-Naini, P. Ogunbona, in *First IEEE Pacific-Rim Conference on Multimedia*. Securing wavelet compression with random permutations IEEE, (2000), pp. 332–335
23. C.-P. Wu, C.-C. Kuo, Design of integrated multimedia compression and encryption systems. IEEE Trans. Multimedia. **7**(5), 828–839 (2005)
24. P. Bas, T. Furon, F. Cayre, G. Doërr, B. Mathon, *Watermarking security: fundamentals, secure designs and attacks*. (Springer, 2016)

Publisher's Note