2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA) | 978-1-6654-6390-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/IISA56318.2022.9904398

Digital Image Watermarking using Self Inverting Permutations - A Comparative Study

Maria Chroni
Dept. of Computer Science & Engineering
University of Ioannina
Ioannina, Greece
mchroni@uoi.gr

Stavros D. Nikolopoulos
Dept. of Computer Science & Engineering
University of Ioannina
Ioannina, Greece
stavros@cs.uoi.gr

Iosif Polenakis
Dept. of Computer Science & Engineering
University of Ioannina
Ioannina, Greece
ipolenak@cs.uoi.gr

Vasileios Vouronikos
Dept. of Computer Science & Engineering
University of Ioannina
Ioannina, Greece
v.vouronikos@uoi.gr

***Abstract*—In this work we present a comparative study of two watermarking techniques, both based on frequency domain transformations but differing on watermark construction. Many state-of-the-art approaches utilize binary images as watermarks. An alternative way of watermark construction, which incorporates error correction properties, proposes the transformation of an integer number w into a self-inverting permutation (SiP). SiP can be represented in a two dimensional (2D) object, since images are 2D structures, and that constitutes a watermark. Through this study, we highlight the importance of developing robust digital image watermarking systems through their operation in real attack scenarios. Moreover, through the conduction of a series of experiments required for the comparative study, there are provided further insights regarding the strengths and weaknesses of each approach. Several image sizes have been tested along with a range of evaluation metrics for imperceptibility evaluation. In addition, robustness of both of the techniques has also been confirmed by applying a variety of attacks on the resulting images.**

***Index Terms*—Image watermarking, Security, Digital Rights, Graph Theory**

I. Introduction

The amount of digital data circulated and the evolution of technology have led to the rapid growth of digital data interception, and by extension, copyright infringement of authors (i.e., digital object creators). The interception of intellectual property affects both the authors, who see their personal effort being illegally circulated against their will, as also the society itself, both economically and morally. The lack of reaction to issues relating to the interception of digital objects leads society as a whole to develop a sense of complacency in terms of not being aware of the risks involved, and also, on the part of the malicious users themselves, a sense of security of being non-prosecuted.

A. Information Hiding

Information hiding is used to conceal a message that is transmitted either through a physical or an intangible medium. Information hiding techniques are divided into two main categories, i.e., steganography and watermarking. The purpose of steganography differs greatly from that of watermarking. In steganography, a message is concealed in a carrier medium for the purpose of moving the message without being perceived. Watermarking, on the other hand, is used to hide a unique identifier on a carrier medium in order to secure the copyright of the rightful owner of the digital object. These two categories use the concept of information hiding as a main component but are separated in the purpose for which they are used. In information hiding it is important that the information embedded in the carrier object in question is not perceived. If the information is perceived by an unwanted recipient it is only a matter of time before the hidden message is recovered.

B. Digital Image Watermarking

Digital watermarking is a particular category of information hiding intended to protect a digital object. When watermarking a digital object, a unique identifier, called a watermark, is embedded in the digital object. The watermark, being unique, identifies uniquely a single owner and legal holder of the digital object in question. A watermarking technique is efficient when the watermark is efficiently embedded in the digital object and successfully extracted, even if the digital object has been modified by a malicious attack. Digital image watermarking has become very important since the amount of data (i.e., images) we exchange every day has increased. Nowadays digital image has become a commercial product and securing it is of utmost importance. An example is the medical digital images used for e-health, where the utilization of watermarks on medical images is of major importance [12]. The need to keep this data secure and the rapid increase in the speed of computing systems have prompted the development of more secure watermarking schemes.

C. Related Work

In [4] Kamili *et al.*, propose a watermarking system that uses two kinds of watermarks. The authors split the image into YCbCr channels (i.e. the luminance component (Y), and two chrominance components, i.e., Cb and Cr based on visual information) and by using the DCT (Discrete Cosine Transform) transform, they analyse it in the frequency domain. In the Y channel they embed a robust watermark, while in the Cb channel they embed a fragile watermark. For more

978-1-6654-6390-4/22/$31.00 ©2022 IEEE

security, the robust watermark is encoded using Chaotic and Deoxyribonucleic Acid (DNA) encryption while the fragile watermark is encoded using Chaotic coding. In [5], Roy *et al.*, propose a watermarking scheme using two transforms, IWT (Interactive Wavelet Transform) and DCT. According to the authors, the proposed technique increases PSNR by reducing the MSE (Means Square Error) value, unlike the classical DWT (Discrete Wavelet Transform) technique which has the disadvantage of high MSE values. In [6], a hybrid technique using DWT transformation is proposed by Nguyen *et al.* to optimize the robustness and capacity by dividing the image into LL, LH, HL, HH bands. Using the SVD (Singular Value Decomposition) technique the SV (Singular values) values of the watermark are embedded within the appropriate SV values of the HH band. Zhang *et al.*, in [7] propose a technique in the spatial domain with SVD transformation. A binary watermark is placed at the largest SV value of each image block. Although it is a technique that uses the spatial domain, it achieves good robustness and low computational complexity. The works [8] and [9] use the SVD transform to embed the watermark and the DWT transform to analyse the image in the frequency domain. These two works use the HVS (Human Visual System) to select the regions where the watermark is going to be embedded. In [8], Bagheri Baba Ahmadi *et al.*, utilize the PSO (Particle Swarm Optimization) algorithm to optimize the robustness of the system. The PSO algorithm belongs to the family of genetic algorithms used to minimize an objective function. Wang *et al.*, in [10] propose a system, which like most current systems, is based on the combination of DWT-SVD techniques. In this work, before the embedding process is done, the carrier image and watermark are converted to the NTSC (National Television Standards Committee) color representation. To enhance the security of the system, the carrier image and watermark are encoded with the CML (Coupled Map Lattice) algorithm. A very interesting technique, is proposed in [11] by Hussan *et al.*, where the image is divided into 4×4 blocks, and in each block the DCT transform is applied. The interesting point is that for watermark, the hash ($SHA-256$) value of DC coefficient of each 4×4 block is used. In total, 16 bits are embedded in each block, and for more security the original image is scrambled via Arnold scrambling algorithm.

D. Motivation and Contribution

The amount of digital information exchanged everyday, including images, audio and software, has increased the risk of malicious interceptions, especially when the digital information exchanged is not secured. The aim of this paper is to highlight the advantages and weaknesses of two digital image watermarking approaches through an experimental comparative study. The experiments focus on the robustness and the fidelity that the watermarking techniques offer under various malicious attacks, including filter attacks, compression attacks and geometrical attacks. The different quality metrics used in the proposed framework include structural similarity index metric (SSIM), bit error rate (BER), and peak signal to noise ratio (PSNR). Through this study we present the process by which a user's copyright (watermark embedding) is secured, the techniques and algorithms used by such a system and finally, and how these systems (through the experimental comparison) respond against real threats. Moreover, we show that a SiP as a watermark [1], due to its error correction properties, can be successfully extracted in all the deployed attacks, whereas multiple binary image watermark [3] is destroyed. The remaining of this paper is divided as follows: in Section 2, we present the image watermarking models. Section 3 defines presents the experimental evaluation results. Section 4 presents our concluding remarks, and future research directions.

II. The Model

In this section we discuss the image processing techniques and the corresponding watermarking methods in order to perform the information hiding procedure inside the structure of digital images.

A. Embedding and Extracting Information

For the implementation of the watermarking technique of self-inverting permutations we deploy the Discrete Fourier Transformation in order to utilize the amplitude of the frequencies to embed the watermark in the image as presented in [1]. In the proposed technique of self-inverting permutations, the watermark is embedded at specific areas of the 2DM representation, namely at positions marked with a specific symbol. Through the 2D discrete Fourier transform, the input image is transformed in the frequency domain. After the image is transformed in the frequency domain, according to the 2DM representation, the Fourier transform amplitude matrices are marked using two ellipsoidal annuli namely the 'Blue' and 'Red' respectively. During extraction we follow the reverse procedure. In particular, we find the marked cells and generate the permutation π^*. In order to embed an integer number in an image first we need to convert it into a Self Inverting Permutation (SiP), which consists the watermark, while for the extraction procedure of the SiP from the image we need to convert it back to an integer to make sure we get back the information we originally embedded, following the corresponding methodologies as they are described in [1].

B. Fundamental Requirements

The derived watermarking procedure poses two fundamental requirements in order for the watermarked image to be as close to the prototype one, i.e., to maximize the fidelity of the watermarked image, while on the other hand, regarding the hidden information, w.r.t. the resilience of the embed and extract procedures, to be as robust as possible against potential attacks. Regarding the fidelity requirement, the proposed approach is attested utilizing the *Peak to Signal Noise Ratio* (PSNR) and *Structured Similarity Index Metric* (SSIM) that we discuss next in order to attest the closeness of the watermarked image against the prototype image, ensuring so the fidelity of the image as also the level on which the information, i.e., the watermark, has been hidden adequately well. On the

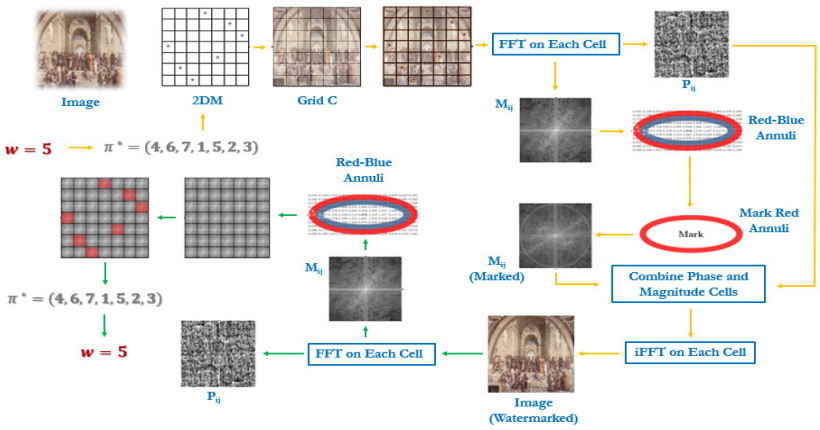


Fig. 1: System Architecture - Embed and Extract Procedures.

other hand, concerning the robustness required in order for the hidden information to be preserved and the corresponding technique to be resilient against various attacks (e.g., filter, compression, geometric, etc.) it is attested utilizing the Bit Error Rate.

C. Architecture and Deployment

The proposed image watermarking technique incorporates two basic procedures, namely, the embedding and the extraction. Next we briefly discuss the deployed procedures underlying the fundamental components of each one.

- **Embed Watermark to Image.** In order to embed a watermark into a digital image we utilize the 2DM representation of the SiP that encodes an integer in order to locate the regions of the images that will contain the hidden information, i.e., the watermark. Next, once these regions are selected, the Fast Fourier Transformation procedure is deployed over each cell of a region in order to compute the amplitude matrices as to be utilized in order to mark the amplitudes of the selected region. Finally, the inverse Fast Fourier Transformation procedure is deployed in order to reconstruct the image to its watermarked form.
- **Extract Watermark from Image.** In order to extract a watermark from a digital image that has been watermarked utilizing the proposed technique, the Fast Fourier Transformation procedure is deployed over each cell computing the amplitude matrices as to be utilized in order distinguish the marked cells. Once the marked cells are located the corresponding 2DM representations is incorporated in order to construct the SiP that then is utilized to decode the integer number.

In Figure 1 we present an illustrative example of embedding and extracting a watermark into a digital image, depicting detailed information about the insights of our proposed approach that constitute the architecture of our model.

III. Evaluation

For the evaluation of the watermarking techniques, we use metrics that compare the initial image, say I_s (start) and the image produced after the watermark has been embedded, or after some attack on it, say I_a (after). Our goal is to evaluate not only whether the generated image is close to the original image but also how qualitative it is visually.

A. Evaluation Techniques

Next we present the metrics we utilized for the evaluation of the proposed technique and conduct the comparative study taking into account the exhibited experimental results.

Peak to Signal Noise Ratio (PSNR). The PSNR value gives us, in decibels (dB), the distortion of the watermarked image relative to the original image. In the mean squared error formula, we measure the squared error between two images I_s and I_a and obtain the mean value, where m, n are the dimensions of the images that are common for both. The PSNR is computed as follows:

$$PSNR = 10 \cdot log_{10}(\frac{255^2}{MSE}), \tag{1}$$

where MSE denotes the mean squared error between images I_s and I_a, and is computed as follows:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I_s(i,j) - I_a(i,j)]^2 \tag{2}$$

Structure Similarity Index (SSIM). The SSIM metric computes the structural similarity between the watermarked image and the original one. The SSIM metric compares the similarity of the two images between three factors; Luminance (L), Contrast (C) and Structure (S) as follows:

$$SSIM(x,y) = L(I_s, I_a)C(I_s, I_a)S(I_s, I_a), \tag{3}$$

where:

$$L(I_s, I_a) = \frac{2\mu_{(I_s)}\mu_{(I_a)} + c_1}{\mu_{I_s}^2 + \mu_{I_a}^2 + c_1} \tag{4}$$

$$C(I_s, I_a) = \frac{2\sigma_{I_s}\sigma + c_2}{\sigma_{I_s}^2 + \sigma_{I_a}^2 + c_2} \tag{5}$$

$$S(I_s, I_a) = \frac{2\sigma_{I_s}\sigma + c_3}{\sigma_{I_s} + \sigma_{I_a} + c_3} \tag{6}$$

Bit Error Rate (BER). The BER metric helps us to measure the error rate between images I_s and I_a as well as between two permutations K_s and K_a. If two values at the same positions of I_s and I_a or K_s and K_a are not the same, then we consider that there is an error at the particular position of image I_a or permutation K_a. The mathematical definition of the BER metric is given below.

$$BER = \frac{ErrorBits}{TotalBits} \tag{7}$$

B. Methodology and Experimental Design

The study presented in this work compares the approaches proposed in [2] and [3] through the aspects of fidelity and robustness. In particular the experimental design focuses on the deployment of a set of metrics utilized to attest the fidelity of the watermarked images against the original ones, while on the other hand it incorporates a set of attacks (e.g., filter, compression, geometric, etc.) in order to attest the robustness of the proposed techniques after the performance of such attacks on images watermarked with these techniques.

The experiments are partitioned into two main categories concerning the attestation of fidelity and robustness, respectively. In the first category of experiments we compute the PSNR and SSIM metrics in order to measure the fidelity of the images watermarked with each technique when compared against the original ones. In the second category of experiments several attacks are deployed against the images watermarked in each case with a particular technique and the resulting image is then utilized to compute and extract the embedded watermark after the attack. In particular, when extracting watermarks (in our case SiP or Logos, respectively for each technique), we need metrics that assess the quality of the extracted watermark and to a further extent the robustness that the watermarking technique offers, where in our case it is utilized the BER measurement.

C. Discussion over the Exhibited Results

In Table I there is illustrated a comparison over seven types of filter attacks, i.e., Gamma Correction with $\gamma = 0.25$, and $\gamma = 3.0$, with Gaussian Noise with $\mu = 0, \sigma = 0.001$ and $\mu = 0, \sigma = 0.01$, with Gaussian blur with Kernel$= 3 \times 3$, Sharpening, and Histogram Equalization (HEQ) identified with numbers in the range $1 - 7$.

Observing the results from the first block of Table I we can see that for $\gamma = 0.25$ the PSNR and SSIM values are better for the technique [3] while the BER values are better for the technique [2]. The PSNR and SSIM values are quite low due to the distortions caused by the gamma filter. The BER values for [2] are zero due to either extraction of the correct permutation without the need of reconstruction or its reconstruction. For $\gamma = 3.0$ we can observe through Table III that for the image painting and clouds the sequence was extracted on the first attempt without reconstruction and the quality of the logos improved considerably compared to that of $\gamma = 0.25$ (see, Table II. The improvement of the logos is probably due to the fact that the value $\gamma = 3.0$, creates an image with a high brightness value. The brightness is closer to that of the original image than the brightness when applying the filter with $\gamma = 0.25$. This increase in brightness in turn leads to an increase in the values of the mid-frequency coefficients. Observing the results from the third block of Table I, the PSNR and SSIM values for the two watermarking techniques are quite similar. It is noteworthy to note that for sky and winter images, the BER values in the technique [2] are zero only for the blue channel. This means that we had successful extraction of the SiP only in the blue channel. In contrast to [3], such a result would imply a failure in the extraction process, as BER $= 58\%$ means that the extracted logo is no longer recognizable. This is not the case in [2], as we need the extraction to occur successfully in at least one channel of the image. From the results depicted in the fourth table of Table I and Table V, we can see that the more we increase the noise, the more the logos start to become one with noise, which is quite a problem because extracting and proving that the image in question has a specific ownership becomes impossible. In the case of [2], the advantage is that we do not rely on two channels (B, G) but on three (R, G, B). This allows us to recover the lost SiP from at least one channel. The technique [3] needs both channels for the two different logos which means that if one logo is lost the extraction fails. We can see from the sixth block Table I that the PSNR and SSIM values are quite close for the two techniques, with slightly better values in technique [3]. The BER values for technique [3] are very good and show us that the extracted logos are recognizable. In Table VII we can see that in the majority of the images, SiP's we extracted did not require reconstruction. In the seventh block Table I, the BER values are acceptable for both techniques which obviously means that we will successfully extract the watermarks. As in the above filters, the PSNR and SSIM values are better for the algorithm [3].

In Table II we present the extracted logos as well as the extracted SiP's for the corresponding images. In the majority of the images the extracted SiP's had wrong values and we had to reconstruct them. The extracted logos in the majority of the images are recognizable except for those of the dog image where they almost disappear. As for the PSNR, SSIM and BER values, the gamma filter with $\gamma = 0.25$ has quite an effect on the quality of the logos by introducing a lot of noise to them, but not to the extent that they are unrecognizable in the majority of the images. In Table VI we can see that the logos are recognizable, and the SiP's for half of the images were extracted without needing reconstruction, and for the other half they needed reconstruction. Applying the Gaussian blur filter results in an image that is duller than the original. In this way, areas with low color intensity variations are not affected by the filter, while areas with high color intensity variations are affected by the low-pass filter. In the Gaussian blur filter, the final luminance value of the pixel to which the filter is applied depends on both the luminance value of the pixel itself and the luminance values of its neighbors. In Table VIII we can see that the SiP's we extracted needed reconstruction for all

		SiP [2]				Logos [3]			
Image	**SIZE**	**PSNR**	**SSIM**	**BER-GC**	**BER-BC**	**PSNR**	**SSIM**	**BER-GC**	**BER-BC**
(1) Gamma Correction with $\gamma = 0.25$									
painting	512×512	7.77	0.29	0	0	7.89	0.31	0.04	0.06
clouds	512×512	8.5	0.33	0	0	8.5	0.33	0.16	0.18
dog	512×512	11.08	0.08	0	0	11.1	0.09	0.26	0.25
glass	512×512	8.6	0.35	0	0	8.6	0.34	0.08	0.1
sky	800×800	9.22	0.23	0	0	9.23	0.24	0.12	0.12
winter	768×768	12.07	0.31	0	0	12.09	0.34	0.1	0.1
(2) Gamma Correction with $\gamma = 3.0$									
painting	512×512	11.4	0.62	0	0	11.47	0.71	0.008	0.01
clouds	512×512	11.4	0.76	0	0	11.4	0.78	0.11	0.16
dog	512×512	9.2	0.58	0	0	9.2	0.62	0.07	0.11
glass	512×512	11.36	0.67	0	0	11.44	0.71	0.05	0.07
sky	800×800	10.4	0.67	0	0	10.4	0.72	0.11	0.19
winter	768×768	11.6	0.41	0	0	11.6	0.5	0.05	0.11
(3) Gaussian Noise with $\mu = 0$ and $\sigma = 0.001$									
painting	512×512	25.97	0.73	0	0	29.56	0.8	0.10	0.10
clouds	512×512	31.38	0.62	0	0	32.87	0.63	0.34	0.35
dog	512×512	30.94	0.59	0	0	32.55	0.61	0.32	0.34
glass	512×512	25.78	0.72	0	0	31.23	0.76	0.23	0.23
sky	800×800	29.7	0.6	0.58	0	33.31	0.67	0.36	0.36
winter	768×768	28.99	0.5	0.43	0	33.38	0.6	0.29	0.28
(4) Gaussian Noise with $\mu = 0$ and $\sigma = 0.01$									
painting	512×512	23.35	0.2	0.29	0.58	23.57	0.21	0.45	0.44
clouds	512×512	23.48	0.2	0.58	0	23.77	0.21	0.43	0.43
dog	512×512	21.86	0.39	0.86	0	21.86	0.42	0.38	0.37
glass	800×800	29.7	0.19	0	0.58	23.7	0.22	0.36	0.37
sky	768×768	23.28	0.12	0.29	0	24.11	0.16	0.4	0.42
winter	768×768	28.99	0.5	0.43	0	33.38	0.6	0.29	0.28
(5) Gaussian Blur with Kernel $= (3 \times 3)$									
painting	512×512	27.1	0.84	0	0	32.07	0.92	0.05	0.05
clouds	512×512	35.92	0.95	0	0	42.69	0.97	0.08	0.11
dog	512×512	34.82	0.94	0	0	40.82	0.97	0.07	0.1
glass	512×512	26.82	0.89	0	0	30.65	0.93	0.05	0.06
sky	800×800	32.3	0.91	0	0	46.45	0.99	0.08	0.12
winter	768×768	30.74	0.77	0	0	43.5	0.97	0.04	0.08
(6) Sharpening									
painting	512×512	19.85	0.63	0	0	19.4	0.57	0.01	0.05
clouds	512×512	29.84	0.8	0	0	29.52	0.75	0.08	0.10
dog	512×512	28.69	0.79	0	0	27.93	0.74	0.06	0.12
glass	512×512	21.86	0.39	0	0	21.86	0.42	0.04	0.05
sky	800×800	20.66	0.73	0	0	20.55	0.66	0.06	0.1
winter	768×768	27.78	0.6	0	0	30.92	0.76	0.03	0.06
(7) Histogram Equalization - HEQ									
painting	512×512	17.86	0.72	0	0	18.52	0.77	0.09	0.02
clouds	512×512	20.15	0.8	0	0	20.18	0.82	0.11	0.15
dog	512×512	11.64	0.63	0	0	11.67	0.65	0.1	0.2
glass	512×512	21.86	0.39	0	0	21.86	0.42	0.05	0.06
sky	800×800	21.69	0.87	0	0	22.15	0.96	0.09	0.16
winter	768×768	17.36	0.52	0	0	17.65	0.68	0.05	0.23

TABLE I: Fidelity Comparison over Filter Attacks

images. The HEQ filter enhances the brightness of the image by increasing the intensity of the pixels. This increase, affects in the same way the coefficients enclosed by the "Red" and "Blue" ellipsoidal annulus.

From the above results presented in Table X we observe that for line and column crops, the more we increase the number of lines or columns we trim, the more indistinguishable the logos become. On the other hand, the SiP's had to be reconstructed but in no case did we encounter the problem of not extracting them or the impossibility of reconstructing them. For the BER values we can say that to some extent the visual quality of the logos is reflected in them. The BER values for the logo technique range from $6\% - 19\%$, a range of values that makes the extraction of the logos successful, see, Table X. From the

Image	**SiP [2]**	***logo 1* [3]**	***logo 2* [3]**
painting	(fixed sip: [4,6,7,1,5,2,3])		
clouds	sip:[4,6,7,1,5,2,3] n1: [4,6,7,1,5,2,3] n2: [4,6,7,1,5,2,3] n3: [4,6,7,1,5,2,3]		
dog	(fixed sip: [4,6,7,1,5,2,3])		
glass	(fixed sip: [4,6,7,1,5,2,3])		
sky	(fixed sip: [4,6,7,1,5,2,3])		
winter	(fixed sip: [4,6,7,1,5,2,3])		

TABLE II: $\gamma = 0.25$

Image	SiP [2]	*logo 1* [3]	*logo 2* [3]
painting	sip:[4,6,7,1,5,2,3] s1: [4,6,7,1,5,2,3] s2: [4,6,7,1,5,2,3] s3: [4,6,7,1,5,2,3]		
clouds	sip:[4,6,7,1,5,2,3] s1: [4,6,7,1,5,2,3] s2: [4,6,7,1,5,2,3] s3: [4,6,7,1,5,2,3]		
dog	(fixed sip: [4,6,7,1,5,2,3])		
glass	(fixed sip: [4,6,7,1,5,2,3])		
sky	(fixed sip: [4,6,7,1,5,2,3])		
winter	(fixed sip: [4,6,7,1,5,2,3])		

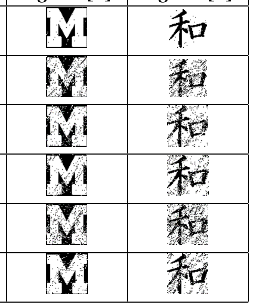


TABLE III: $\gamma = 3.0$

Image	SiP [2]	*logo 1* [3]	*logo 2* [3]
painting	(fixed sip: [4,6,7,1,5,2,3])		
clouds	(fixed sip: [4,6,7,1,5,2,3])		
dog	(fixed sip: [4,6,7,1,5,2,3])		
glass	(fixed sip: [4,6,7,1,5,2,3])		
sky	(fixed sip: [4,6,7,1,5,2,3])		
winter	(fixed sip: [4,6,7,1,5,2,3])		

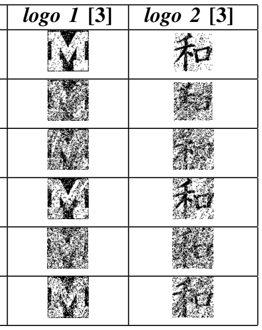


TABLE IV: Gaussian Noise with $\mu = 0$ and $\sigma = 0.001$

Image	SiP [2]	*logo 1* [3]	*logo 2* [3]
painting	(fixed sip: [4,6,7,1,5,2,3])		
clouds	(fixed sip: [4,6,7,1,5,2,3])		
dog	(fixed sip: [4,6,7,1,5,2,3])		
glass	(fixed sip: [4,6,7,1,5,2,3])		
sky	(fixed sip: [4,6,7,1,5,2,3])		
winter	(fixed sip: [4,6,7,1,5,2,3])		

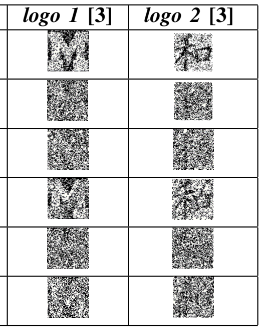


TABLE V: Gaussian Noise with $\mu = 0$ and $\sigma = 0.01$

Image	SiP [2]	*logo 1* [3]	*logo 2* [3]
painting	(fixed sip: [4,6,7,1,5,2,3])		
clouds	(fixed sip: [4,6,7,1,5,2,3])		
dog	(fixed sip: [4,6,7,1,5,2,3])		
glass	(fixed sip: [4,6,7,1,5,2,3])		
sky	(fixed sip: [4,6,7,1,5,2,3])		
winter	(fixed sip: [4,6,7,1,5,2,3])		

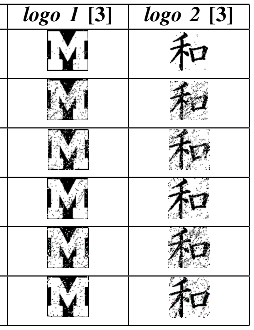


TABLE VI: Gaussian Blur with Kernel $= (3 \times 3)$

results presented in Table XII we observe similar results with those of Table X. For the BER values presented in Table XI we can say that the are quite similar to Table IX and slightly better with the maximum BER remaining under 18%.

For the compression attacks we used sub-sampling = 1 and

Image	SiP [2]	*logo 1* [3]	*logo 2* [3]
painting	(fixed sip: [4,6,7,1,5,2,3])		
clouds	(fixed sip: [4,6,7,1,5,2,3])		
dog	(fixed sip: [4,6,7,1,5,2,3])		
glass	(fixed sip: [4,6,7,1,5,2,3])		
sky	(fixed sip: [4,6,7,1,5,2,3])		
winter	(fixed sip: [4,6,7,1,5,2,3])		

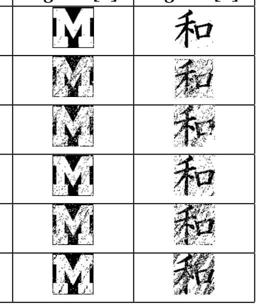


TABLE VII: Sharpening

Image	SiP [2]	*logo 1* [3]	*logo 2* [3]
painting	(fixed sip: [4,6,7,1,5,2,3])		
clouds	(fixed sip: [4,6,7,1,5,2,3])		
dog	(fixed sip: [4,6,7,1,5,2,3])		
glass	(fixed sip: [4,6,7,1,5,2,3])		
sky	(fixed sip: [4,6,7,1,5,2,3])		
winter	(fixed sip: [4,6,7,1,5,2,3])		

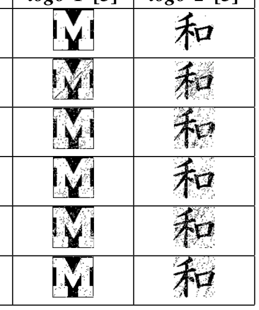


TABLE VIII: Histogram Equalization - HEQ

SiP [2]		Logos [3]		
BER-GC	BER-BC	BER-GC	BER-BC	Attack
0	0	0.10	0.13	40 colums
0	0	0.07	0.13	40 rows
0	0	0.06	0.10	64×64 white
0	0	0.06	0.10	64×64 black
0	0	0.20	0.18	128 colums
0	0	0.14	0.18	128 rows
0	0	0.11	0.13	256×256 white
0	0	0.11	0.13	256×256 black
0	0	0.19	0.16	350×350 white
0	0	0.19	0.16	350×350 black

TABLE IX: BER for the image “dog”

Image	SiP [2]	*logo 1* [3]	*logo 2* [3]
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		
	sip:[4,6,7,1,5,2,3] s1: [4,6,7,1,5,2,3] s2: [4,6,7,1,5,2,3] s3: [4,6,7,1,5,2,3]		
	sip:[4,6,7,1,5,2,3] s1: [4,6,7,1,5,2,3] s2: [4,6,7,1,5,2,3] s3: [4,6,7,1,5,2,3]		
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		

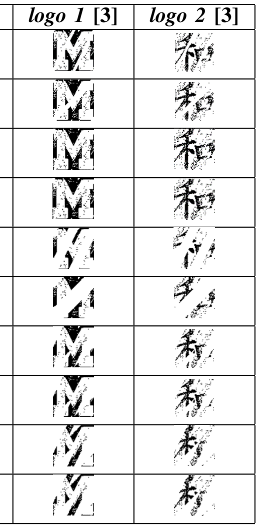


TABLE X: Geometrical attacks for the image “dog”

SiP [2]		Logos [3]		
BER-GC	**BER-BC**	**BER-GC**	**BER-BC**	**Attack**
0	0	0.05	0.07	40 columns
0	0	0.04	0.08	40 rows
0	0	0.04	0.07	64×64 white
0	0	0.04	0.07	64×64 black
0	0	0.11	0.10	128 colums
0	0	0.06	0.08	128 rows
0	0	0.09	0.10	256×256 white
0	0	0.09	0.10	256×256 black
0	0	0.17	0.13	350×350 white
0	0	0.17	0.13	350×350 black

TABLE XI: BER for the image "temple"

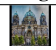
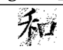
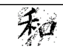
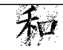
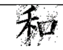
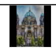
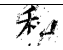
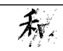
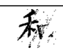
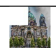
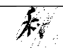
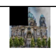
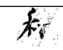
Image	SiP [2]	*logo 1* [3]	*logo 2* [3]
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		
	sip:[4,6,7,1,5,2,3] s1: [4,6,7,1,5,2,3] s2: [4,6,7,1,5,2,3] s3: [4,6,7,1,5,2,3]		
	sip:[4,6,7,1,5,2,3] s1: [4,6,7,1,5,2,3] s2: [4,6,7,1,5,2,3] s3: [4,6,7,1,5,2,3]		
	(fixed sip: [4,6,7,1,5,2,3])		
	(fixed sip: [4,6,7,1,5,2,3])		

TABLE XII: Geometrical attacks for the image "temple"

then we used sub-sampling= 2. For sub-sampling= 1 what we observe from Table XIV is that for all quality values the logo embedded in the blue channel is lost and in its place we extract the logo of the green channel. This happens even for quality equal to 95 where is the minimum compression quality. The PSNR and SSIM values shown in the first block Table XIII decrease proportionally with the decrease in compression quality for both algorithms. This is because information is removed from the image during compression to reduce its size. This information removal leads to a large MSE value and consequently small PSNR values. The BER values are quite high for the technique [3], as can be observed from Table XIII This is due to the subsampling applied during compression. This problem does not occur in technique [2] because of the way the watermark is embedded in the image. We need to reconstruct the SiP's in compression attacks as the watermark is embedded in the high frequency coefficients. Compression algorithms truncate the high frequency coefficients as they do not contain important information about the image. The second block of Table XIII shows that in channel B the BER values for the logo technique are quite high which shows us that the logo in this channel is either lost or the noise is high and the logo is now indistinguishable. The PSNR and SSIM values in The second block of Table XIII follow the same pattern that we reported in the first block of Table XIII What is noteworthy to notice from the above results in Table XV is that although we extract the logo in the first channel, in the second channel we extract the logo that we put in the first channel just as it happened in sub-sampling $= 1$. As mentioned above during compression, some sub-sampling algorithm is used to reduce the image size. In the compression attack with a sub-sampling parameter equal to 2, the ratio $4 : 2 : 0$ is applied while with sub-sampling= 1 the ratio $4 : 2 : 2$ is applied.

IV. Conclusion

In this work, we presented a comparative study between the watermarking techniques proposed in [2] and [3], utilizing various metrics for this purpose, so as to evaluate the robustness and fidelity provided by each approach. The technique proposed in [2] uses Integers encoded as self-inverting permutations as watermarks, while proposed in [3] uses two binary images as watermarks. The watermarks of both techniques are embedded in the frequency domain coefficients. The technique proposed in [2], uses the 2DM representation to select the cells in which the watermark will be embedded, while on the other hand, the technique proposed in [3] embeds $1-$bit of each logo in each 8×8 DCT amplitude coefficient matrix using repetition code.

A. Remarks

To compare the techniques we used the PSNR, SSIM and BER metrics after subjecting the images to a series of attacks. The attacks included filter attacks (Gamma, Gaussian, HEQ, etc.), geometric attacks (cut 40 rows, cut 128 rows, etc.) and finally compression attacks with different compression qualities and sub-sampling values. We compared the watermarked image after the attack, with the original image, and tabulated the results of the metrics for inference. Our results showed that in all attacks, in the self-inverting permutation algorithm using the properties of self-inverting permutation, we can recover the watermark even if enough information is lost from it. The PSNR and SSIM values are better for the technique [3], mainly due to the way the watermark is embedded. Increasing the values of the coefficients enclosed by the "Red" ellipsoidal annuli, in the corresponding cells denoted by the 2DM representation, reduces the PSNR and SSIM values compared to simply swapping the pairwise coefficients in the embedding algorithm of [3]. This exchange of values in the embedding algorithm of [3] often results in watermarked images with high distortion, which is not the case in [2]. The approach presented in [2] is more flexible in terms of the size of the images it supports, while the approach presented in [3], using binary images, supports a smaller range of image sizes.

B. Future Research

We have seen that self-inverting permutations offer useful properties that allow us to reconstruct the watermark after various malicious attacks. To this point, it is worth-noting that most filters were first used as an attack for the self-inverting permutation algorithm with complete success. This fact opens a new way for us to develop more synthetic attacks

		SiP [2]				Logos [3]				
Image	**SIZE**	**PSNR**	**SSIM**	**BER-GC**	**BER-BC**	**PSNR**	**SSIM**	**BER-GC**	**BER-BC**	**Quality**
(1) Compression with $sub-sampling=1$										
painting	512×512	26.85	0.84	0	0	31.85	0.93	0.02	0.39	95
		26.73	0.85	0	0	30.97	0.91	0.23	0.41	80
		26.66	0.83	0.42	0	30.56	0.91	0.30	0.47	70
		26.58	0.82	0.28	0	30.23	0.9	0.36	0.47	60
		26.51	0.82	0	0	29.91	0.89	0.39	0.48	50
(2) Compression with $sub-sampling=2$										
painting	512×512	26.85	0.84	0	0	31.85	0.93	0.03	0.40	95
		26.73	0.85	0	0	30.97	0.91	0.24	0.42	80
		26.65	0.83	0	0	30.56	0.91	0.31	0.43	70
		26.54	0.82	0	0	30.23	0.9	0.38	0.43	60
		26.53	0.82	0	0	29.91	0.89	0.42	0.45	50

TABLE XIII: Compression attacks

Image	**SiP [2]**	***logo 1* [3]**	***logo 2* [3]**	**Quality**
painting	(fixed sip: [4,6,7,1,5,2,3])			95
painting	(fixed sip: [4,6,7,1,5,2,3])			80
painting	(fixed sip: [4,6,7,1,5,2,3])			70
painting	(fixed sip: [4,6,7,1,5,2,3])			60
painting	(fixed sip: [4,6,7,1,5,2,3])			50

TABLE XIV: Compression with $sub-sampling=1$

Image	**SiP [2]**	***logo 1* [3]**	***logo 2* [3]**	**Quality**
painting	(fixed sip: [4,6,7,1,5,2,3])			95
painting	(fixed sip: [4,6,7,1,5,2,3])			80
painting	(fixed sip: [4,6,7,1,5,2,3])			70
painting	(fixed sip: [4,6,7,1,5,2,3])			60
painting	(fixed sip: [4,6,7,1,5,2,3])			50

TABLE XV: Compression with $sub-sampling=2$

using modern and widely used image processing filters. We would like to extend existing algorithms in the future to protect images used as textures in digital games. The interception of such images is quite widespread and many of them are used in digital games available for payment. Research to develop more secure methods of information hiding by extending existing algorithms is still a future goal for copyright protection. The problem we encounter in the state of the art watermarking techniques, is that the watermarks are embedded either in specific areas of the image or small parts of the watermark are embedded in the whole image. The problem with those approaches is that easily some malicious user can cut an area of interest of an image where no information is embedded and claim ownership of that part despite that it belongs to a watermarked image.

ACKNOWLEDGMENT

This research was supported by project "Dioni: Computing Infrastructure for Big-Data Processing and Analysis" (MIS No. 5047222) co-funded by European Union (ERDF) and Greece through Operational Program "Competitiveness, Entrepreneurship and Innovation", NSRF 2014-2020.

REFERENCES

[1] Chroni, M., Nikolopoulos, S. D., and Palios, L.: Encoding watermark numbers as reducible permutation graphs using self-inverting permutations. Discrete Applied Mathematics, 250, 145-164, (2018).

[2] Chroni, M., Fylakis, A., and Nikolopoulos, S. D.: Watermarking images in the frequency domain by exploiting self-inverting permutations. WEBIST 2013, 45-54, (2013).

[3] Roy, S., and Pal, A. K.: A blind DCT based color watermarking algorithm for embedding multiple watermarks. AEU-International Journal of Electronics and Communications, 72, 149-161, (2017).

[4] Kamili, A., Hurrah, N. N., Parah, S. A., Bhat, G. M., and Muhammad, K.: DWFCAT: dual watermarking framework for industrial image authentication and tamper localization. IEEE Transactions on Industrial Informatics, 17(7), 5108-5117m (2020).

[5] Agrwal, S. L., Yadav, A., Kumar, U., and Gupta, S. K.: Improved invisible watermarking technique using IWT-DCT. In 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO) (pp. 283-285), IEEE, (2016).

[6] Nguyen, T. H., Duong, D. M., and Duong, D. A.: Robust and high capacity watermarking for image based on DWT-SVD. In The 2015 IEEE RIVF International Conference on Computing and Communication Technologies-Research, Innovation, and Vision for Future (RIVF) (pp. 83-88), IEEE, (2015).

[7] Zhang, H., Wang, C., and Zhou, X.: A robust image watermarking scheme based on SVD in the spatial domain. Future Internet, 9(3), 45, (2017).

[8] Bagheri Baba Ahmadi, S., Zhang, G., Wei, S., and Boukela, L.: An intelligent and blind image watermarking scheme based on hybrid SVD transforms using human visual system characteristics. The Visual Computer, 37(2), 385-409, (2021).

[9] Ernawan, F., and Kabir, M. N.: A block-based RDWT-SVD image watermarking method using human visual system characteristics. The visual computer, 36(1), 19-37, (2020).

[10] Wang, K., Gao, T., You, D., Wu, X., and Kan, H.: A secure dual-color image watermarking scheme based 2D DWT, SVD and Chaotic map. Multimedia Tools and Applications, 1-32, (2022).

[11] Hussan, M., Parah, S. A., Jan, A., and Qureshi, G. J.: Hash-based image watermarking technique for tamper detection and localization. Health and Technology, 1-16, (2022).

[12] Vaidya, S. P., and Kishore, V. R.: Adaptive Medical Image Watermarking System For E-Health Care Applications. SN Computer Science, 3(2), 1-10, (2022).