Fragile Watermarking using Chaotic Sequences

Daniel Caragata [*], Anca Livia Radu [¤], Safwan El Assad [*]
[*] *Universite de Nantes (France),* [¤] *Military Technical Academy (Romania)*

Abstract

In this article we propose a new fragile watermarking algorithm that provides the means to verify the integrity of JPEG images. We begin by presenting the cryptanalysis of CWSA (Chaotic Watermarking Scheme for the Authentication of JPEG Images), an existing fragile watermarking algorithm. We analyse its weaknesses and we propose a new version that is resistant to those attacks, faster and less perceptible. The proposed algorithm uses the JPEG quantized DCT coefficients and embeds the watermark information in their LSB. It uses robust chaotic generators for the generation of dynamic keys and watermarking information. One of the applications of the proposed algorithm is verifying data integrity for images transferred over the Internet.

1. Introduction

Digital data transmitted through the Internet can be copied and used in malicious ways. This may cause problems referring to the protection of intellectual rights and the integrity of the transmitted information. Watermarking has emerged as a new technology that can be used to resolve these problems, providing either copyright claiming mechanisms or integrity verification tools.

Watermarking techniques can be divided in various ways [2], depending on the property that is taken into account. If we refer to the type of document, the host data can be text, images, audio or video files. According to application we can refer to source based methods and destination based methods. According to the working domain there are spatial domain methods and frequency domain methods. Based on the human perception we can divide the watermarking algorithms as visible or invisible. Invisible watermarking can be divided into robust watermarking methods and fragile watermarking methods. The robust methods are focused on protecting the copyright of the owner on the digital media and the fragile methods are used to test the integrity of digital media. We can also classify watermarking methods as being standard methods or chaos based methods.

There are many types of attacks that can be used against the watermark. Removal based attacks include making modifications to the compression format (Joint Photographic Experts Group - JPEG, JPEG 2000, etc.), adding noise (Gaussian noise, Salt and Pepper noise) or various transformations that can be made to the image: rotation, scaling, translation, cropping, geometric distortions, jitter and so on. Furthermore, cryptographic attacks are based on searching the watermark by analysing the weaknesses of the algorithm. A cryptographic attack can aim to remove the watermark or to modify its content. Therefore, the robustness against this kind of attack is the most important.

There is an important number of watermarking methods that were proposed, for example, in [3] a standard watermarking method is presented, where the watermark embedding process is applied on the DC frequency components based on a quantitative analysis of the magnitudes of the Discrete Cosine Transform (DCT) components of the host images. The authors of [14] present a method which divides the image into two subsets. For the first subset the brightness is incremented by a small amount and the brightness of the other set is decremented by the same amount. The level of brightness is chosen so that the change in intensity remains imperceptible. In [15] a spread spectrum technique is applied on the frequency-domain of an image. The watermark is inserted in a certain number of low frequency coefficients, except the DC coefficient, as follows: $Y_D(i)=X_D(i)\cdot(1+a\cdot W(i))$, where $X_D(i)$ and $Y_D(i)$ are the DCT coefficients of the DCT transform of the original image X, and the watermarked image Y, respectively. Also, W(i) is an element of the watermark sequence.

Chaos based watermarking methods have also been proposed, like the one in [6], where the watermark information was generated using a 1-D chaotic map, then transformed into a 2-D watermark matrix using the "Peano Scan" and it was inserted into the spatial domain. In [5] a blind algorithm that works in the frequency domain and uses chaotic sequences was published. This algorithm divides the original image in 8 by 8 blocks and each block is DCT transformed. The obtained coefficients are quantized using a quantization matrix. Then, Cat Map is used to scramble the watermark in order to obtain a pseudo-random matrix and the Logistic Map is used to decide which DCT quantized coefficients will carry the watermark in their LSB.

This paper proposes a new chaos based fragile watermarking algorithm for JPEG images that can be successfully used in Internet applications. It is organised as follows. Section 2 presents the CWSA algorithm [7] and analyses the weakness of CWSA against ciphertext-only attack. We model this attack using Markov Chains in order to study the number of watermarked images needed for a successful attack. Section 3 presents our proposed watermarking algorithm. It is robust against cryptographic attacks, faster and less perceptible compared to CWSA. In section 4, we present the simulation results for the proposed algorithm and we show its effectiveness. Furthermore, a comparison with CWSA algorithm in terms of PSNR and inter-correlation coefficient demonstrates the superiority of our proposed algorithm. The conclusions and future work are presented in section 5.

2. CWSA algorithm

In this section we present the CWSA algorithm. First we introduce the JPEG standard, that is being watermarked, and then we show how the CWSA algorithm introduces the watermark information into the JPEG coefficients.

2.1 JPEG format

The JPEG standard is the most common format for image representation that is used on the Internet. It is a lossy compression standard that typically achieves a compression rate of 10:1 with very little loss in image quality.

The JPEG standard can compress any image representation, but it is mostly used with the YCbCr (Luminance Chrominance) format. This is because the human eye is more sensible at changes in the Luminance plane, Y, than in changes in the blue chrominance plane, Cb, or the red chrominance plane, Cr. This is in contrast with the RGB (Red Green Blue) format where all planes are equally important. The YCbCr representation allows a very high compression rate for the Cb and Cr planes. We are not interested in the compression of these two planes because they are not used by the CWSA algorithm. The compression of the Y plane is realized as follows:

- *Block splitting*: the image is split into 8×8 blocks;
- *DCT transform*: each block suffers a DCT transform. The low frequencies will contain information that is sensible to the human visual system while the high frequencies will contain information that is less important to the human visual system.
- *Quantization*: a quantization matrix is used to eliminate the high frequencies of the DCT transform. The quantization formula is:

$$AC_{Qkl}=\text{round}\ (AC_{kl}/Q(k,l));\ k,l \in \{1,2,\dots,8\} \quad (1)$$

where AC_{kl} represent the non quantized DCT coefficients, AC_{Qkl} represent the quantized DCT coefficients and [Q] is the quantization matrix. A typical quantization matrix is given by:

$$[Q]=\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \quad (2)$$

- *Entropy coding*: Huffman coding is used to code the useful information obtained after the quantization.

The JPEG decoding use the following steps:

- Entropy decoding.
- De-quantization.
- Inverse DCT transform.
- Image reconstruction.

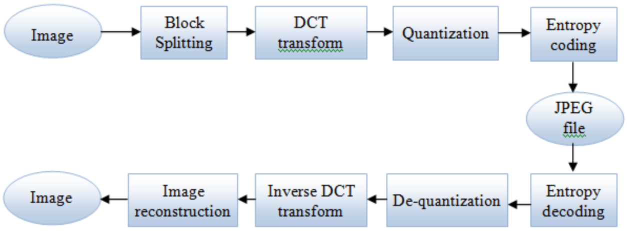


Fig. 1. JPEG standard

2.2 CWSA algorithm

W. Hang has proposed the Chaotic Watermarking Scheme for the Authentication of JPEG Images (CWSA) algorithm [7].

In this scheme, the quantized DCT（Discrete Cosine Transform）coefficients after entropy decoding, have their LSB plane set to zero and are mapped to the initial values of a chaotic system, which we will call *chaotic system 2*. Meanwhile, the watermarking key is used as the initial condition of a chaotic map, which we will call *chaotic system 1*. For each quantized AC coefficient this map is iterated one time, and the value is transformed into a positive integer number, *n*. The *chaotic system 2* is iterated *n* times and the obtained value, which is the watermark information, is transformed into the LSB of the quantized AC coefficient of the watermarked image.

This process is showed in Figure 2:

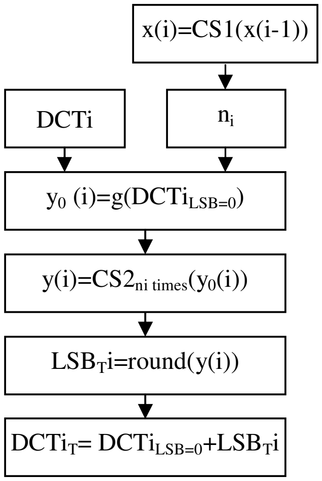


Fig. 2. Watermarking algorithm for one DCT coefficient

In Figure 2 CS1 is *chaotic system 1*, which is iterated one time for each DCT coefficient in order to obtain n_i. DCTi is the DCT coefficient that is currently being watermarked. The function g(·) maps DCTi with the LSB plane set to 0 ($\mathrm{DCTi}_{\mathrm{LSB=0}}$) on the chaotic interval of *chaotic system 2*. *Chaotic system 2* (CS2) is iterated n_i times starting from the value obtained with the function g(·). The value thus obtained is transformed into the new LSB of the DCT coefficient of the watermarked image.

For the integrity check, the watermark information is computed for each AC coefficient using the process described above and is compared with the LSB of the analysed image. If they are different, then the block containing that coefficient has been modified.

3. Cryptanalysis of CWSA algorithm

In this section we present and analyse the cryptanalysis method that we have developed against the CWSA algorithm.

3.1 Cryptanalysis algorithm

The number of chaotic iterations of *chaotic system 2*, *n*, is generated using just one iteration of *chaotic system 1*. This means that when processing an image there will be a string $\{n_i\}$, *i=1 to L*, where *L* is the number of watermarked AC coefficients. This string will not change unless the key is changed. So, when we watermark another image using the same key, the same array $\{n_i\}$ will be generated, regardless of what image is generated. On the other hand, the maximal value for any element of $\{n_i\}$, which we note n^{max}, cannot be very large. The authors of CWSA claim that this algorithm could be implemented in trust-worthy digital cameras. We have implemented the algorithm using Matlab 7.0.1 on an Intel Dual Core computer with 2 GB of RAM. The time needed to watermark a 5 mega pixels image was greater than 40 seconds with *n* fixed at 15.

An attacker can try to find the values of $\{n_i\}$ using a number of watermarked images. When the attack begins, every element of $\{n_i\}$, can have any value between 1 and n^{max}. For each of these elements, the attacker eliminates those values that would have generated another watermark value for the LSB of the DCT coefficient. He repeats this process with different watermarked images until there will be only one plausible value for each element of $\{n_i\}$. After having discovered the entire string $\{n_i\}$, the attacker can use it to watermark any image, or to modify a previously watermarked image. It is important to notice that the presented attack ignores the watermarking key. As a matter of fact, the attacker can make any modification to the image without acquiring any knowledge about the secret key. When the integrity check will be performed, the image will seem genuine.

3.2 Number of needed images for a successful cryptanalysis

We use Markov Chains to calculate the number of watermarked images that the attacker needs in order to find the string $\{n_i\}$ and thus to break the algorithm.

A Markov process is a time-varying stochastic phenomenon characterized by the property that the future state of the system is dependant only on its current state.

Suppose a stochastic process, *M*, whose states can take a discrete set of values: $S = \{S_1, S_2, ..., S_K\}$. $\mathrm{T} = \{1,2,....,m\}$ is the time domain and $X=\{X_1, X_2, ..., X_m\}$ is the list of state vectors for the system.

Each component of X, $X_j=\{x_1, x_2, x_3, ..., x_K\}$ contains K elements. Each of these elements, x_i with i=1,…,K, represents the probability that the system is in state *i* at time *j*.

We suppose that whenever the process is in state *i*, there is a fixed probability P_{ij} that it will next be in state j. The matrix of one-step transition probabilities P_{ij} is:

$$P = \begin{pmatrix} P_{0,0} & P_{0,1} & \cdots & P_{0,K} \\ P_{1,0} & P_{1,1} & \cdots & P_{1,K} \\ \vdots & & P_{i,j} & \\ P_{K,0} & P_{K,1} & \cdots & P_{K,K} \end{pmatrix} \quad (3)$$

One of the properties of Markov Chains is that $X_{j+1}=X_j*P$.

We will use Markov Chains to model the effect of the attack on a given element *n* of $\{n_i\}$. In this case, the state of the system at any time is the number of plausible values of *n*. Therefore, any element x_i of a vector state $X_j=\{x_1, x_2, ..., x_{nmax}\}$ is equal to the probability that there are *i* plausible values for *n* at moment *j*. Furthermore, *Pij* is the probability that the system will pass from a state with *i* plausible values, to a state with *j* plausible values after using one watermarked image to perform the cryptanalysis. So:

$$P_{ij} = \begin{cases} 0, j > i \\ \dfrac{C_{i-1}^{j-1}}{2^{i-1}}, j \le i \end{cases} \tag{4}$$

If j>*i* it is impossible for the system to pass from state *i* to state *j* because the attack can only eliminate certain plausible values, but it cannot add any. In the other cases, the probability P_{ij} is calculated as the ratio between the number of favorable changes (when the system passes from state *i* to state *j*) and the number of possible changes. Note that there are *i* plausible values that the attacker analyses. One of these values will always remain plausible, as it is the correct value of *n*. The other *i-1* values have a probability *p=0.5* of being eliminated from the set of plausible values.

The effect of every new watermarked image used by the attacker is modelled by a new step $X_{j+1}=X_j*P$. Therefore, the total number of iterations used to model the system is equal to the number of watermarked images used by the cryptanalyst. The first element of the state vector X_j, x_1, represents the probability that the attack has succeeded, i.e. there is only one plausible value for the analysed *n*.

In figure 3 we show the probability of breaking the algorithm against the number of watermarked images that were used for the attack: for a specific watermarked coefficient (continuous line), for a section of the watermarked image (dotted line with circles), for the entire watermarked image (dotted line). We have used an image of size 720x600 pixels and a section of size 70x70 pixels.

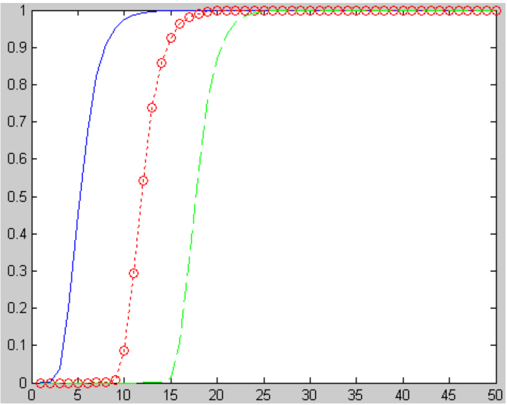


Fig. 3. Probability versus number of watermarked images

In table 1, we give the number of needed watermarked images to cryptanalyze the system in the cases described above for two specific probability values.

TABLE 1. NUMBER OF WATERMARKED IMAGES REQUIRED TO MAKE THE CRYPTANALYSIS OF THE ORIGINAL ALGORITHM

	One AC coefficient	70x70 region	760x660 image
p=0.95	9	16	23
p=0.99	12	18	25

We can see that the number of watermarked images needed for a successful attack is not very large.

3. Proposed algorithm

The study we performed in section 2 has shown the weakness of CWSA algorithm to ciphertext-only attack. Also, when CWSA algorithm was proposed, the authors did not discuss the chaotic functions that were used from a cryptographic point of view. Furthermore, the CWSA algorithm watermarks all the quantized AC coefficients of the DCT transform, which makes it slower and more perceptible, thus more vulnerable.

We propose a new algorithm that avoids all weaknesses of the CWSA algorithm while watermarking directly the JPEG quantized DCT coefficients and using robust chaotic functions to improve its cryptographic characteristics. The main features of the proposed algorithm are (see figure 5):

1. The use of robust chaotic generators (see figure 4).
2. The watermark of only the DCT coefficients greater than a certain threshold *T*.
3. The use of a part of the key when x_0 is computed.
4. The iteration of the chaotic function a variable number of times when *n* is computed.

3.1 Chaotic generators

Two chaotic generators are used by this algorithm. Their role is very important because if they do not have strong cryptographic properties the algorithm may be vulnerable to different types of attacks that use the weaknesses of the chaotic signals [9].

We propose the use of a Piecewise Linear Chaotic Map (PWLCM) for *chaotic system 1* [10] and a cascaded recursive filter with the skew tent non-linear function for the *chaotic system 2* [11].

The PWLCM is a chaotic function composed of multiple linear segments:

$$x(n+1) = F(x(n)) = \begin{cases} x(n)\cdot\dfrac{1}{p}, 0 \le x(n) < p \\ [x(n)-p]\cdot\dfrac{1}{0.5-p}, p \le x(n) < 0.5 \\ F(1-x(n)), 0.5 \le x(n) < 1 \end{cases} \tag{5}$$

The *chaotic system 2* is comprised of two cascaded recursive filters with non linear functions [13]. Its structure is presented in fig. 4.

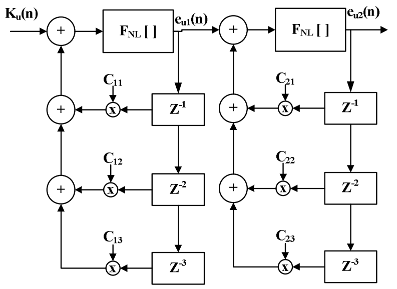


Fig. 4. Proposed chaotic generator

The equations at every layer are:

$$x_1(n) = c_{11} \cdot e_{u1}(n-1) + c_{12} \cdot e_{u1}(n-2) + c_{13} \cdot e_{u1}(n-3) \qquad (6)$$

$$e_{u1}(n) = FNL(x_1(n)) \qquad (7)$$

$$x_2(n) = e_{u1}(n) + c_{21} \cdot e_{u2}(n-1) + c_{22} \cdot e_{u2}(n-2) + c_{23} \cdot e_{u2}(n-3) \qquad (8)$$

$$e_{u2}(n) = FNL(x_2(n)) \qquad (9)$$

The coefficients C_{11}, C_{12}, C_{13}, C_{21}, C_{22}, and C_{23} are the parameters of the chaotic function and are part of the secret key.

There is a wide range of non-linear functions that can be used and it includes: *Ln(x), x·cos(x), x·exp[cos(x)],*etc. According to the studies we have performed, the best results are obtained when the skew tent map is used [11], [13]. It is given by the following equation:

$$x(n+1) = F(x(n)) = \begin{cases} \frac{1}{a}, 0 \le x(n) \le a \\ \frac{1}{a-1} \cdot x(n) + \frac{1}{1-a}, a < x(n) \le 1 \end{cases} \qquad (10)$$

3.2 Watermarked coefficients

The original CWSA algorithm embeds the watermark information on all AC coefficients. This makes it slow, because of the large number of coefficients that need to be processed and makes the modification to the image visible. We propose to embed the watermark both on the DC coefficient and on those AC coefficients with a value greater than a threshold, *T*. The advantages of using the DC coefficient have been discussed in [8].

Our watermarking technique modifies the LSB of the analyzed coefficient. This means that it changes its value by 1. If the value is small, say 2, the coefficient may become 3. In this case the coefficient has been changed by 50%. If the block contains many coefficients with small values (and that is typically the case) the changes to the image become visible. By contrast, the DC coefficient has far greater values. For example, if the DC coefficient was initially 100 and it has been changed to 101 his value was changed by 1% only.

We will show in section 4 that the watermarked image with our proposed scheme is less distorted than the watermarked image using the CWSA algorithm. This is mainly due to the choice of the watermarked coefficients.

3.3 Using part of the key for the computation of x_0

In the CWSA algorithm, the initial value of the *chaotic system 2* is obtained by simply setting the LSB of the current DCT to 0. Thus it is known by the attacker, who can use this information to perform the cryptanalysis. We propose that this initial value be obtained using part of the secret key:

$$x_0 = F(f_{Qc} + k_{x0}) \qquad (9)$$

where: x_0 is the initial value of the *chaotic system 2*, f_{Qc} is the analyzed coefficient with the LSB plane set to 0 and k_{x0} is part of the secret key. In this way the attacker will not have access to the initial value of the chaotic map and in addition the key space is larger.

3.4 Variable number of iterations for *chaotic system 1*

The watermark information for each DCT coefficient is computed iterating *chaotic system 2 n* times. Because *n* is obtained after one iteration of *chaotic system 1*, the same values of *n* will be generated for all images watermarked with the same key. This can be used by the cryptanalyst to perform the attack described in section 2. To make the algorithm resistant against this type of attack, we propose that the value of *n* be computed using a variable number of iterations *L* of the *chaotic system 1*:

$$L = 1 + ((f_{Qc} + k_n) \bmod 3) \qquad (11)$$

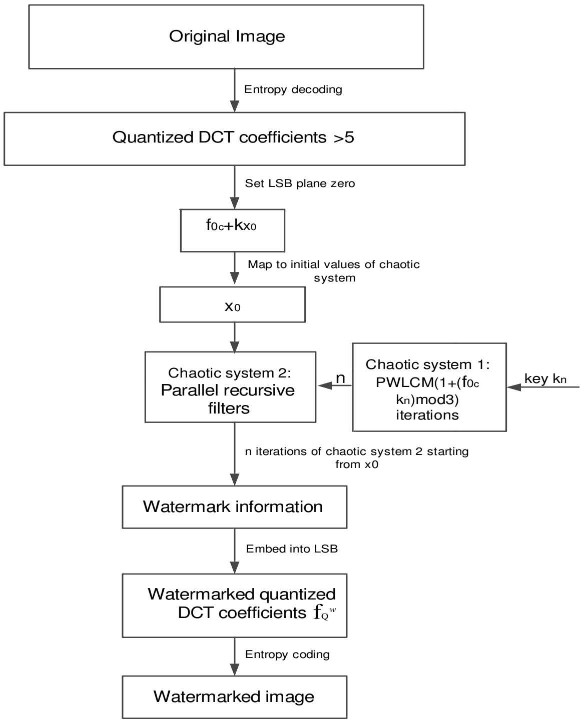
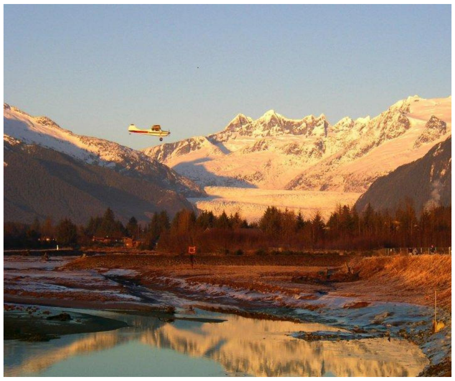

Original Image
Entropy decoding
Quantized DCT coefficients >5
Set LSB plane zero
f0c+kx0
Map to initial values of chaotic system
x0
Chaotic system 2: Parallel recursive filters
n
Chaotic system 1: PWLCM(1+(f0c kn)mod3) iterations
key kn
n iterations of chaotic system 2 starting from x0
Watermark information
Embed into LSB
Watermarked quantized DCT coefficients f_Q^w
Entropy coding
Watermarked image

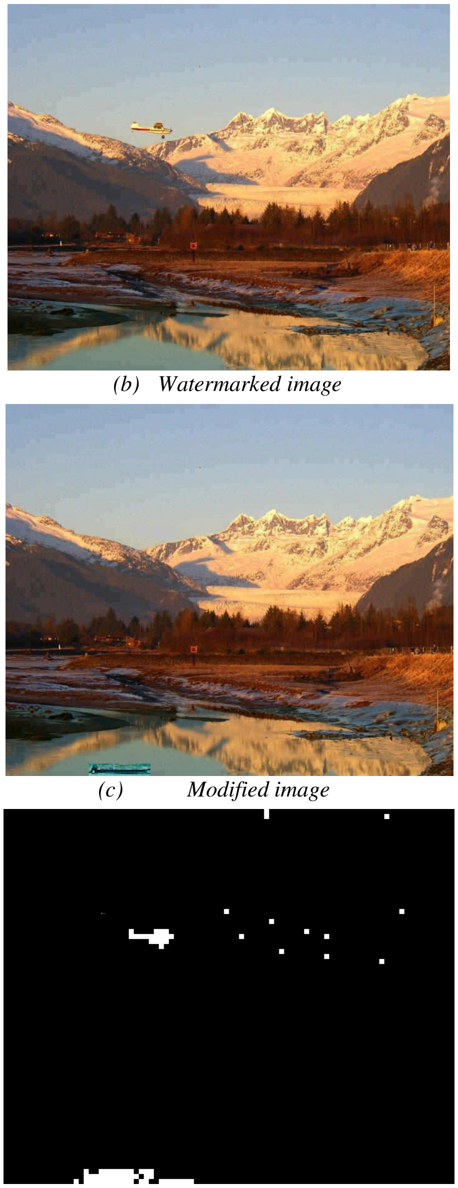
Fig. 5. Proposed watermarking scheme

4. Simulation results

We have simulated our proposed watermarking scheme and the CWSA algorithm. Figure 6 a) presents the original 720x600 Mountain JPEG image and Figure 6 b) shows the watermarked image obtained with the proposed algorithm. The watermarked coefficients have been the DC coefficients and the AC coefficients with values greater than 5. Figure 6 c) presents the modifications we have made to the watermarked image, and figure 6 d) presents the tampered regions obtained after the watermark extraction.



(a) Original JPEG image



(b) Watermarked image

(c) Modified image

(d) Tampered image

Fig. 6. Simulation results

In order to be efficient, a watermark should be imperceptible, robust and statistically invisible. This means that the watermark should not bring any visible changes on the watermarked image, the inserted watermark should be resistant to different attacks performed on the watermarked image and, also, it should be impossible to reproduce the watermarks using statistical methods.

We will use Peak Signal-To-Noise Ratio, PSNR, and the inter-correlation coefficient to measure the distortion that the watermarking process causes to the image.

Peak Signal-To-Noise Ratio PSNR is an objective criterion of imperceptibility which measures the “noise” added to the original image by inserting the watermark.

$$PSNR(f,f_w) = 10\lg\left[\frac{\max_{(m,n)} f^2(m,n)}{\frac{1}{R}\sum_{(m,n)}(f_w(m,n)-f(m,n))^2}\right] \qquad (13)$$

where f represents the original image, f_w the watermarked image, *(m,n)* represents a particular pixel location and R the number of pixels in the original/watermarked image. Typical PSNR values for high quality images exceed 30-40 dB in the case of image compression, but the values should be better for fragile watermark embedding.

Similar to the indicator presented above, the normalized inter-correlation coefficient can be used in order to measure the resemblance between the original image and the watermarked image. It has to be as close as possible to 1:

$$E(f) = \frac{1}{N}\sum_{(m,n)} f(m,n) \qquad (14)$$

$$r(f,f_W) = \frac{\frac{1}{N}\sum_{(m,n)}(f(m,n)-E(f))(f_W(m,n)-E(f_W))}{\sqrt{\frac{1}{N}\sum_{(m,n)}(f(m,n)-E(f))^2}\sqrt{\frac{1}{N}\sum_{(m,n)}(f_W(m,n)-E(f_W))^2}} \qquad (15)$$

For the human eye, the watermarked image obtained with our algorithm seems identical to the original image. The comparative results, presented in table 2, clearly indicate the effectiveness of our algorithm.

TABLE 2 COMPARATIVE PERFORMANCES

	CWSA	Proposed algorithm
PSNR	77.5628	79.8531
Inter-correlation coefficient	0.9993	0.9995

6. Conclusion

A new fragile watermarking scheme for JPEG image integrity has been proposed. The main features of the proposed algorithm are: robustness against cryptographical attacks, high imperceptibility of the added watermark and greater speed than the CWSA algorithm. Our algorithm can be used in some applications such as image transfer over the Internet with integrity check and watermark of images for use as forensic evidence in court of laws.

For the future work we propose an extension of the presented algorithm to include a copyright claiming mechanism.

7. References

[1] S. Katzenbeisser, F. A. Petitcolas, *Information Hiding techniques for steganography and digital watermarking,* Artech House, 2000.

[2] Saraju P. Mohanty, Digital Watermarking: A Tutorial Review

[3] J. Huang Shi, Y.Q. Yi Shi, *Embedding Image Watermarks in DC Components*, IEEE Transactions on Circuits and Systems for Video Technology, Volume 10, pp 974-979

[4] Y. Hu Kwong, S. Jiwu Huang, *Using invisible watermarks to protect visibly watermarked images*, ISCAS'04, Volume 5, pp 584-587.

[5] Zhao Yantao, Ma Yunfei, Li Zhiquan, *A robust chaos-based DCT-domain watermarking algorithm*, Proceedings of IEEE International Conference on Computer Science and Software Engineering, Volume 3, pp. 935-938, 2008.

[6] S. Tsekeridou, N. Nikolaidis, *Copyright protection of still images using self-similar chaotic watermarks*, Proceedings of IEEE International Conference on Image Processing (ICIP'00), Volume 1, pp. 411-414

[7] H. Wang, K. Ding, C. Liao, *Chaotic watermarking scheme for authentication of JPEG Images,* International Symposium on Biometrics and Security Technologies, 2008.

[8] J. Huang, Y. Q. Shi, Y. Shi, *Embedding Image Watermarks in DC Components*, IEEE transactions on circuits and systems for video technology, vol. 10, no. 6, September 2000.

[9] G. Alvarez, S. Li, *Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems*, International Journal of Bifurcation and Chaos, Volume: 16, Issue: 8 (2006) pp. 2129-2151.

[10] S. Li, G. Chen, X. Mou, "On the Dynamical Degradation of Digital Piecewise Linear Chaotic Maps", International Journal of Bifurcation and Chaos, 2005.

[11] S. El Assad, H. Noura, I. Taralova, *Design and analyses of efficient chaotic generators for cryptosystems* Lecture notes in IAENG Transactions on Electrical and Electronical Engineering, vol. 1, 2008, 10 pag.

[12] A. Tefas, A. Nikolaidis, N. Nikolaidis, V. Solachidis, S. Tsekeridou, I. Pitas, *Performance Analysis of Watermarking Schemes Based on Skew Tent Chaotic Sequences*, European Project IST, 1999.

[13] H. Noura, S. Henaff, I. Taralova et S. El Assad, *Efficient cascaded 1-D and 2-D chaotic generators*, Second IFAC Conference on Analysis and Control of Chaotic Systems Chas, 2009.

[14] I. Pitas, *A method for signature casting on digital images*, Proceedings of International Conference on Image Processing, pp. 215-218, 1996.

[15] Ingemar J. Cox , Joe Kilian, F. Thomson Leighton, Talal Shamoon, *Secure Spread Spectrum Watermarking for Multimedia,* IEEE, 1997.