U.P.B. Sci. Bull., Series C, Vol. 74, Iss. 2, 2012 ISSN 1454-234x

SECRET COMMUNICATION IN THE DIGITAL ERA

Maria CERNAT[1], Daniel CERNAT[2]

Principalul scop al articolului de faţă este legat de dezvoltarea şi testarea unei noi metode de semnare digitală în domeniul frecvenţei. Metoda zero a distribuţiei în frecvenţă este prezentată în acest articol ca o variantă a metodei clasice a distribuţiei în frecvenţă. Această nouă metodă constă in atribuirea valorii zero pentru coeficienţii binari de frecvenţă joasă corespunzători. Un alt obiectiv, poate la fel de important, constă în dezvoltarea unui program capabil să testeze mai multe metode, incluzând-o pe cea propusă de noi. Programul în cauză se numeste DigiSign si este uşor de folosit şi de dezvoltat ulterior.

The main objective of this article is to develop and test a new method of digital signing in the frequency field. The Spread Spectrum Zero method presented here is a variation of the classical Spread Spectrum Method and it consists in introducing zero values in the corresponding low frequency binary coefficients. Another objective of this paper consists in developing a program able to test more methods, including our method. The computer program is called DigiSign and is easy to use and can be further developed.

Keywords: digital signatures, watermarking, steganography, algorithm, secret communication

1. Introduction

People have always wanted to be able to send secret messages to each other. Whether it was the need of sending an order regarding the beginning of a riot or a love letter, people have tried to hide their messages in order to limit the number of those who could understand their intentions. A whole new discipline of secret communication developed over the centuries: cryptography. The history of this discipline is shaped by very ingenious ways of transmitting secret information. Early attempts to use such methods can be traced far back in the writings of Herodotus. Histiaeus of Millet sent a secret message to Aristagoras concerning the beginning of a riot by tattooing the information on a slave scalp that was sent to Aristagoras after the slave's hair grew back. All Aristagoras had to do in order to find the content of the concealed message was to shave the slave's head [1]. Over the years the methods used for this kind of communication

[1] Lecturer, PhD, Journalism, Public Relations and Communication Department of Spiru Haret University, e-mail: macernat@gmail.com
[2] Technical Manager, BitDefender, e-mail: dcernat@gmail.com

have constantly increased their complexity. In 1499 the famous cryptographist Johannes von Heidenberg also known as Tritemius (1462-1516) published a treatise on cryptography and steganography disguised as a book on magic. Important figures of politics and science developed ways of secretly communicating information. Among them the most prominent figure is that of Francis Bacon (1561- 1626), the famous philosopher who worked, prior to the publication of his philosophical work, as a diplomatic agent for Queen Elisabeth the First. He developed a method of encrypting the messages using the binary code that anticipated the modern method used in computer science.

The historians of secret communication show there are three ways of sending secret information:

- Invisible writings
- Dissimulated writings
- Encrypted writings

This classification helps us distinguish between the two fields of secret communication: **cryptography** and **steganography** [2]. Invisible and dissimulated writings are studied by steganography while encrypted writings are studied by cryptography. As the Greek etymology shows (*crypto* – hidden, *logia* - study) cryptology, or cryptography refers to secret writings presented as such.

The need of sending secret information is far too obvious in politics and economics. The political and economic conflict is also an informational battle. For centuries classical cryptography was used mainly in politics to send concealed messages. In the digital era cryptography and steganography have applications also in atm cards, in computer passwords or electronic commerce.

2. DigiSign program

The main purpose of the program is to easily develop new methods of digital signing.

The DigiSign program was created using Microsoft Visual Studio 2005, the programming language being C++ and the interface developed using MFC classes. The program is easy to use and can be further developed. The sources of the program are available; they can be used and eventually modified but only for didactic purposes, provided that the original author of the sources is mentioned (the author of this paper) in the Info or About section of the new program.

The application works with images represented in the RGB format. As a result we have three fields of color R=red, G=green, B=blue and in each filed we have $2^8 = 256$ levels of color intensity. The digital signatures are also represented by RGB images, but those images are logically considered as black and white images, having thus only one field of color and two levels of color intensity 0=black, 1=white.

More information and sources (including samples) are available on the internet at the following location http://dcernat.host-ed.me/watermarking/.

3. Digital watermarking – basic elements

For almost 1000 years signing on paper has been used for identifying the author and for discouraging impostors. In the digital era proving the authenticity of an object is becoming more and more important.

Digital watermarking (the signing of digital signals) represents the process of introducing auxiliary information in a digital source signal. Following the process of watermarking we obtain the marked signal.

Below we refer strictly to signing digital images (the digital signal we want to sign is a 2D monochrome or colored digital image. We also refer to the *digital source signal* using the term *source image*, to *digital signing* using the term *signing* and to the *marked signal* using the term *signed image*.

If we use the notation X for the *source image*, W for the *signature* and Y for the *signed image*, the process of digital signing can be mathematically rendered through the function:

$$Y = \gamma(X, W) \tag{1}$$

Schematically, the process of digital signing can represented as in fig. 1 [3].

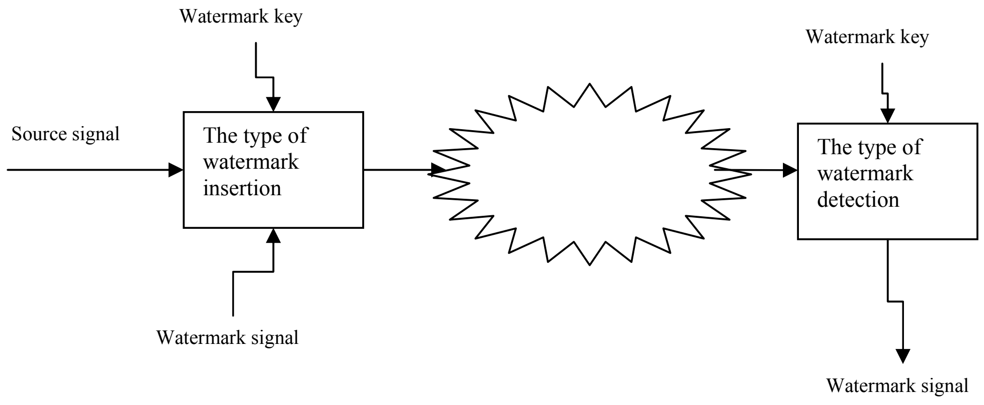

Fig. 1. The process of digital signing

The process has two parts:

- The process of signing/introducing the signature in the source image

- The process of recognition/ detection of the signature on the signed image

From the point of view of the *digital signature,* one can identify at least three types, as follows:

- the digital signature may represent a digital text,
- the digital signature may represent a digital image 2D
- the digital signature may represent a border value (the border value is used in the detection process)

For establishing the type of digital signature used in the process of digital signing we have to consider the context in which this process takes place, that is, the digital context. In this environment, any letter is represented by an octet (8 bytes). For example, letter „a" is represented as 01000001. This representation is a precise one, meaning that if we modify even a single byte, the letter is transformed into another one.

If we use a 2D digital image for the *digital signing*, even if some of the bytes representing the digital image are modified as a result of some intentional/unintentional processes, the image can still be recognized.

In the following example we have a *digital signing* represented as a digital text and a *digital signing* represented as a 2D digital image. Both signatures have been altered by 10%. As we can see in fig. 2, the digital text is beyond recognition while the 2D digital image can still be easily recognized.

1000100	*‘d’*	*1000101*	*‘e’*
1001001	*‘i’*	*1001001*	*‘i’*
1000111	*‘g’*	*1000110*	*‘f’*
1001001	*‘i’*	*1001011*	*‘k’*
1010011	*‘s’*	*1010010*	*‘r’*
1001001	*‘i’*	*1001000*	*‘h’*
1000111	*‘g’*	*1000101*	*‘e’*
1001110	*‘n’*	*1001110*	*‘n’*

Fig. 2.a. A text altered with 10% cannot be recognized.

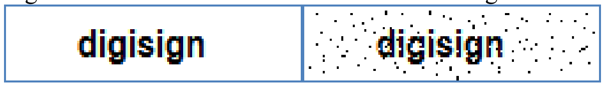

Fig. 2.b. A 2D image altered with 10% can still be recognized

Therefore, for the digital signing we shall use a 2D digital image and not a digital text.

The process of *digital signing* is characterized by several proprieties [4]:

- *Imperceptiveness* refers to the fact that the digital signature introduced in the source image should not be visible by eyesight.
- *Transparency*: refers to the fact that by integrating the digital signature in the source image we lose a minimum quality factor of the source image; the image is not essentially altered.

- *Robustness*: refers to the fact that the digital signing process should withstand premeditated or unpremeditated attacks.
- *The singularity of the key* refers to the fact that if we use two different digital signatures we have to obtain two different signed images. In other words, if:

  $$W_1 \approx W_2 \rightarrow K_1 = K_2 \quad (2)$$

  where X_1 and X_2 represent the source images, W_1, and W_2 represent the digital signatures and K_1 and K_2 represent the signed images
- *Non-reversibility* refers to the fact that the function $\gamma(X,K)$ should not be reversible:

  $$\gamma^{-1}(X,K): W \rightarrow K \quad (3)$$

  does not exist or cannot be estimated or approximated out of W.
- *Computational efficiency* refers to the fact that the signing algorithm has to be implemented by hardware or software and it needs to be fast enough to allow the computation of data in real time.

The first property discussed, the *imperceptiveness,* is necessary for algorithms used to hide the digital signature so that it could not be detected in the *signed* image.

There is also a different category of algorithms whose main characteristic is non-reversibility, the *digital signature* being visible by eyesight in the signed image. In this case, the intention is to have the digital signature impossible to eliminate from the *signed image* for the purpose of obtaining an unsigned image, without visibly distorting that image.

According to the purpose of the attack, distortions are of two types: premeditated and unpremeditated.

The most common and simple type of unpremeditated distortion is represented by the jpeg based compression of an image.

There are several methods of digital signing. We will attempt to classify them. According to the types of algorithms used, the following digital signing algorithm categories can be identified [5]:

- space algorithms: applicable to certain parts of the source image;
- frequency algorithms: first they introduce a transformation in the frequency field on the source image and then they modify certain parameters. As a result, the information added for obtaining the *signed image* can be traced out throughout the image and not only in certain parts. Those algorithms are more robust than the previous ones.
- Transformation algorithms: they use certain mathematical conversions of the images (the Wavelet transform, for example).

According to their purpose algorithms can be used to:

- verify the authenticity of the image (we are not interested in determining the author);
- verify the authentication of the image (we want to know if the image is authentic, but also to identify the author of that image).

In the digital signing process two parameters are important in order to define the quality of the signing process. They are defined as follows:

$$MSE = \frac{\sum (Y(i,j) - X(i,j))^2}{N^2} \text{ (capacity of embedded signal)} \tag{4}$$

represents the capacity of integrating the information in the source signal (the total number of altered bytes needed to insert the signature)

and:

$$PSNR = 20\log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \text{ (peak signal to noise ratio)} \tag{5}$$

represents the level of noise introduced in the image.

4. Watermarking algorithms

4.1. LSB algorithm (Least significant bit)

LSB algorithm [6] (Least Significant Bit) is one of the easiest to implement algorithms used in the digital signing of an image. This is a space type algorithm and it consists in introducing signature information in the most insignificant bit of every pixel of the source image. A pixel form the source image is represented in a field of color on 8 bytes that means 256 color intensity levels. Note that value 11111111 represents maximum intensity level and corresponds to the white color.

If we modify the last bit, that is LSB (Least Significant Bit) we lose only 1 level of color, a loss unperceivable by human eye. Thus, in every pixel form the source image, on each field of color we can introduce a signature pixel.

$$Y(i,j) = \begin{cases} X(i,j) \mid 1, & if\, W(i,j) = 255 \\ X(i,j) \,\&\, 254, & if\, W(i,j) = 0 \end{cases} \tag{6}$$

where „|” (or) and “&” (and) are logical operations on bytes.

Theoretically, the algorithm should be robust to any isometric operation but it is actually robust only to framing and translation because of the number representation errors in the digital environment.

We notice that the signature can be easily detected if a difference operation followed by a normalize operation is made between the source and the signed image.

The algorithm can be expanded in order to modify the last x less significant bytes, where x=1,2,…,8. The algorithm produces a visually identical signed image if x < 4. After the bit 4 the image is visible altered and the signature can be easily detected by the human eye.

4.2. LSB random algorithm

As mentioned earlier, the LSB algorithm can be easily detected by extracting from the signed image the image corresponding to the less significant bit of each pixel. The LSB random algorithm [7] uses a random sequence of pixels that will be modified using this algorithm. The random sequence in the digital signing has to be identical to the one in the signature extraction in the signed image. Although the LSB algorithm makes it impossible to detect it, this algorithm is no longer robust to the operations of framing and translation.

$$Y(i,j)=\begin{cases} X(i,j), & if\, r(i,j)=0 \\ X(i,j)\,|\,1, & if\, r(i,j)=1\, and\, W(i,j)=255 \\ X(i,j)\,\&\,254, & i\, fr(i,j)=1\, and\, W(i,j)=0 \end{cases} \tag{7}$$

4.3. The Superposition algorithm

The Superposition algorithm introduces the digital signature in the source image using the operation of adding two images. Depending on the desired purpose, the introduced signature can or cannot be traced by the human eye in the signed image. This algorithm needs the source image during the extraction of the signature.

$$Y(i,j)=\begin{cases} X(i,j), & if\, W(i,j)=0 \\ \min(255, X(i,j)+k), & if\, W(i,j)=255 \end{cases} \tag{8}$$

where k represents the signature insertion level in the source image.

This algorithm is robust to framing and translation if we introduce in the signed image a „benchmark” that will align the signed image and the source image. For a sufficiently large k coefficient the signature is visible in the signed image and the algorithm is robust to the operation of jpeg compression.

4.4. The 2 Sets Algorithm

This algorithm divides the set of pixels from the source image in two equal sets using a random algorithm. The introduction of the signature is made by adding a quantity k to the intensity levels of the pixels in the first set and by removing the quantity k from the intensity of the pixels in the second set as shown in fig. 3. Mathematically we have:

$$Y(i,j)=\begin{cases}\min(255, X(i,j)+k), & \text{if } X(i,j) \text{ is from } 1^{th} \text{ set}\\ \max(0, X(i,j)-k), & \text{if } X(i,j) \text{ is from } 2^{th} \text{ set}\end{cases} \tag{9}$$

The detection of the signature in the signed image is made by the same division of the pixels based on the same random algorithm. We calculate the average intensities of pixels in the two sets and compare the absolute difference to the value K. If the difference exceeds k value, it means that the image has been digitally signed. (the signature has been detected).

One should notice that the algorithm is not 100% safe, in the sense that it can detect that an image has been digitally signed even if this image is an authentic one.

The algorithm can withstand operations such as framing and translation if an adequate algorithm for dividing the pixels in the two sets is used.

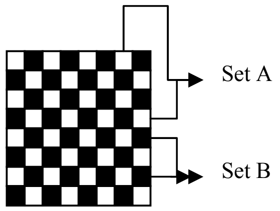

Fig. 3. The first set is represented by white points; the second set is represented by the black points

4.5. The BPCS (BitPlane Complexity Segmentation) Algorithm

The algorithms presented above belong to the category of spatial algorithms, which make no distinction between the properties of different pixels in the image. The inserted information does not take into account the position or the intensity of pixels in the source image. The BPSC algorithm [8] takes this into account and tries to introduce the information needed for the signing process in the areas with the highest level of noise in the image.

As we can see in the fig. 4, the BPSC algorithm splits the image into tiles and compute for each tile a noise level using the formula (10). The next step is to replace the noise tiles with a new ones that contains the signature information.

$$d=\sum\sum\frac{X(i,j)\, xor\, X(i',j')}{2n(n-1)} \tag{10}$$

The detection process is similar to the insertion one. The image is divided in regions, then in tiles and for each tile we calculate the noise level. If that level is higher than the established value we could extract the information previously inserted.

This digital signing technique has the advantage of introducing the information in the less visible areas. As a result, a larger quantity of information could be introduced in the source image, without visibly altering its quality.

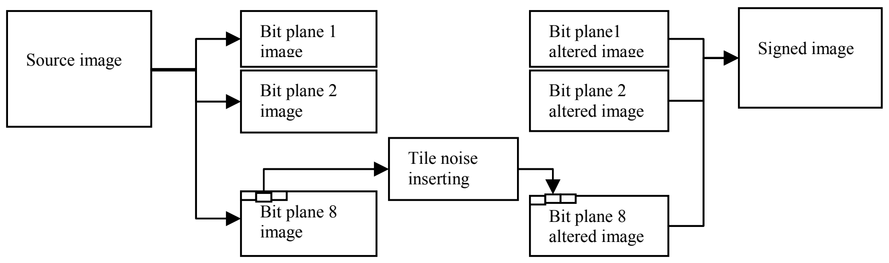

Fig. 4. The BPCS algorithm

4.6. The Spread Spectrum Algorithm

All the algorithms presented above are part of the first category of algorithms, that is, space-type algorithms. The Spread Spectrum Algorithm is an algorithm in the second category – frequency-type algorithm [9]. It is based on the direct cosine transformation and its converse.

For two dimensions (this is the case we are interested in because the digital image is represented two dimensionally, as a matrix) we have the following formulas:

- Direct cosine transformation [10]:

$$X_{k_1,k_2} = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x_{n_1,n_2} \cos\left(\tfrac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right)\cos\left(\tfrac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right), \quad k_1 = 0 \ldots N_1-1,\ k_2 = 0 \ldots N_2-1 \tag{11}$$

- Converse cosine transformation:

$$x_{k_1,k_2} = \frac{2}{N_1}\frac{2}{N_2}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}\delta(n_1)\delta(n_2)X_{n_1,n_2}\cos\left(\tfrac{\pi}{N_1}n_1\left(k_1+\frac{1}{2}\right)\right)\cos\left(\tfrac{\pi}{N_2}n_2\left(k_2+\frac{1}{2}\right)\right), \quad k_1 = 0..N_1-1,\ k_2 = 0..N_2-1 \tag{12}$$

where: $$\delta(n) = \begin{cases} \sqrt{2}, & n = 0 \\ 1, & n \neq 0 \end{cases} \tag{13}$$

The Spread Spectrum algorithm divides the source image in tiles and apply the direct cosine transformation. As a result, we obtain a set of nxn coefficients in the frequency field. The first coefficients correspond to the low frequencies, more exactly, the basic characteristics in the respective tiles. Some coefficients represent the high frequencies, that is, the details in the tile in question. The algorithm alters some of the DCT coefficients by introducing one or more data bits in every one of them. The matrix of DCD coefficients is reversed back in the space field using the converse cosine transformation and thus we obtain the altered tile that will be replaced in the original tile.

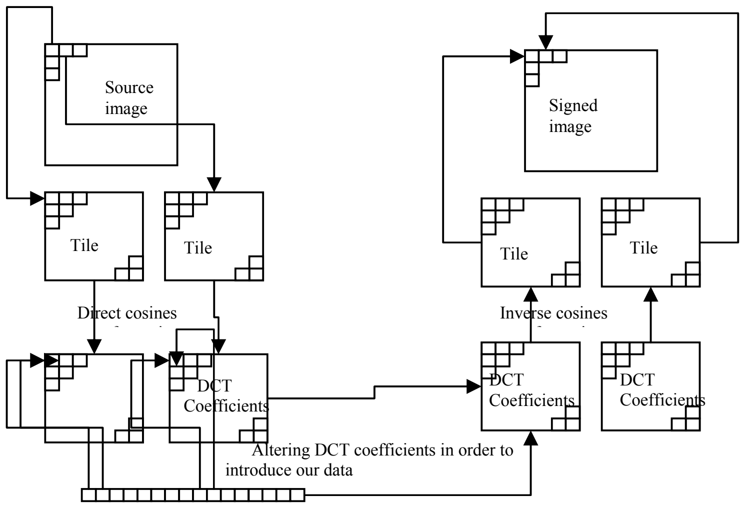

Fig. 5. The diagram of the Spread Spectrum algorithm

When we detect the signature the signed image is divided again in nxn tiles, each tile is then transformed in the frequency field using direct cosine

transformation and the signature information is extracted from the altered coefficients.

There are several variations of the Spread Spectrum method depending on the parameters in the frequency field that we want to alter. Thus, we can alter the low frequency parameters (see the example above), or the medium or high frequency parameters.

5. Spread Spectrum Zero Algorithm

This is the original part of this study. We tried to develop a new algorithm in order to prove that the Digital Watermarking is still an open research field and that this type of algorithm is robust to complex image transformation like jpeg compression.

This algorithm is a variation of the Spread Spectrum algorithm. The image is split in tiles of nxn pixels. Each tile is transformed in frequency domain using DCT transformation. For each tile, the first two DCT coefficients are altered using the following formula:

$$x_k = \begin{cases} x_k, & if\,|x_k| < t \\ [x_k]-1, & if\,|x_k| \geq t\, and\, [x_k]\ is\ odd \\ [x_k], & if\,|x_k| \geq t\, and\, [x_k]\ is\ even \end{cases} , k = 1,2; t \in N^* \tag{14}$$

The algorithm modifies only the coefficients exceeding a certain value (1,2, …) in order to eliminate the errors caused by the distortion of binary represented numbers. The modification process means to make the DCT coefficient an event integer; that means to force a “Zero” value on the last digit of the DCT coefficient in binary representation.

Detection method is also converting each tile of nxn into frequency domain. The first two coefficients are analyzed for each tile: each coefficient is transformed in binary value. We are interested only by even coefficients; that means coefficients with last digit of value „0” in a binary representation. As a result, the detection method is counting the last „0” values from the first tow DCT coefficients (binary coefficients) from each tile. Is the percent of „0” values is bigger than an established threshold (70-90%), we have a signed image.

We note that, unlike the rest of space-type algorithms, this algorithm can withstand the operation of jpeg compression. This is an expected result since the jpeg algorithm functions on the same level converting image from spatial to the frequency field. We can see the result of this algorithm implemented by DigiSign program in fig. 6.

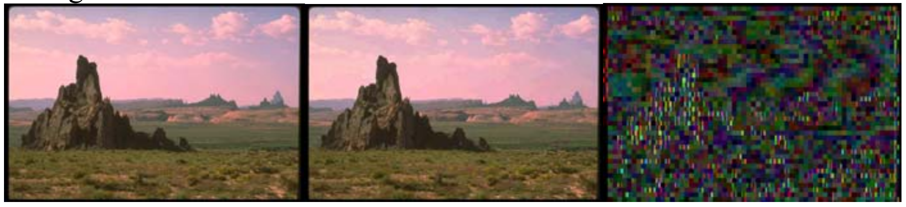
Fig. 6. Source, signed and normalized difference images.

The presence of the signature has been successfully detected in the signed image.

We have the following values for the parameters:

- MSE = 0.6%
- GL = 0.8%
- PL = 3.2%

- HL = 16.1%.

In fig. 7 we can see the result obtained after the image was compressed using jpeg algorithm with a quality factor of 75%. The presence of the signature has been successfully detected for a value of 80%.

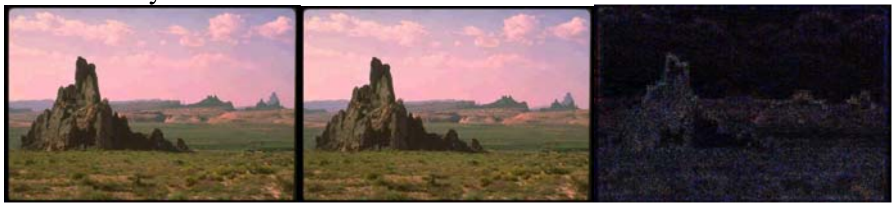
Fig. 7. The original, signed and difference image obtained after jpeg compression 75%.

In fig. 8 we can see the result obtained after the image was compressed using jpeg algorithm with a quality factor of 65%. In this case the presence of the signature has no longer been detected for minimum value of 80%.

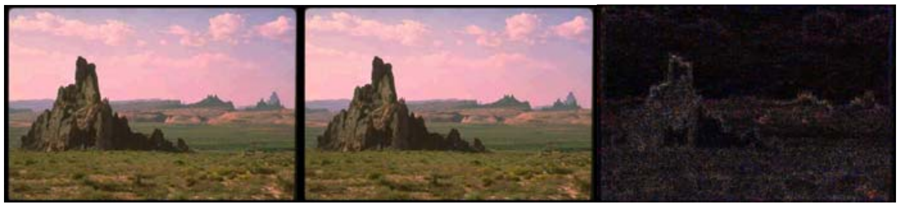
Fig. 8. The original, signed and difference image obtained after jpeg compression 65%.

The algorithm presented above is not robust to the geometrical operations as a result of the fact that it divides the image in several tiles with the dimension of 8x8. The algorithm can be made robust if we introduce some „benchmarks" in the image, which will lead to the correct detection of the tile position in the signed image.

6. Conclusions

In this paper we analyzed the most significant categories of digital signing. We also analyzed the most important properties that a digital signing process must have (imperceivness, robustness, and transparency). The first result of this analysis allowed us to establish that the type of information introduced in the source image has to be another binary image because this type of image is more resilient to errors than a digital text.

The second result of our analysis allowed us to make a hierarchy of the digital signing methods. Our study revealed that the methods used in the spatial field are easier to implement but they are not robust enough. All the spatial methods fail to the simplest, unpremeditated attack: the jpeg compression. The frequency methods are more difficult to implement, but they are more robust also against the jpeg compression. The frequency methods have also the advantage of introducing the information globally and not locally, like the spatial methods. Consequently, the inserted information is harder to detect and eliminate.

We also determined that the spatial methods that do not divide the image into tiles are robust to the geometrical transformations of image. Theoretically, those methods could stand any type of isometric image transformation.

One could also notice that the spatial methods which do not divide the image into tiles are resilient to different types of geometrical transformations applied to the image. Theoretically, these methods can withstand any isometric operation (an operation which preserves the distances). Practically, due to the number representation errors in the digital field, these methods are robust only to translation and framing. The methods dividing the image into tiles can be made robust by inserting "benchmarks" in the marked image, to allow the recognition of the initial position of the tiles.

In addition to the algorithms presented above there is also a different type of algorithms. The field of digital signing is still expanding. At present, the best digital signing method seems to be the one implemented with the use of the Wavelet transform. This method (which has not been presented in this paper) is more robust than other methods presented here, including the frequency methods.

The purpose of this paper was:

- To enumerate and classify different techniques of digital signing that exists at present.

- To develop/test a new digital signing method in the frequency field. The Spread Spectrum Method presented in the paper represents a variation of the classical Spread Spectrum method and consists in inserting 0 values in coefficients corresponding to low frequencies. Only the coefficients with a value exceeding a pre-established minimum value are modified, thus eliminating most of the errors caused by the representation of numbers in the digital field.

- To implement DigiSign program that can be easily used to develop and test new watermarking algorithms.

R E F E R E N C E S

[1] *J.A.S. Evans*, The American Journal of Philology, Vol. 84, No. 2, Apr. 1963

[2] *Ingemar Cox, Matthew Miller, Jeffrey Bloom and Jessica Fridrich*, Digital Watermarking and Steganography, 2nd Ed, Nov. 27, 2007

[3] *Juergen Seitz*, Digital Watermarking for Digital Media, Jun. 2005

[4] *G. Langelaar, I. Setyawan and R. Lagendijk*, Watermarking digital image and video data, IEEE Signal Processing Magazine, 17:20–46, 2000

[5] *Mitchell D. Swanson, Mei Kobayashi and Ahmed H. Tewfik*, Multimedia data embedding and watermarking technologies, Proceedings of the IEEE, 86(6):1064 1087, June 1998

[6] *Deepa Kundur and Dimitrios Hatzinakos*, A Robust Digital Image Watermarking Method using Wavelet-Based Fusion,Department of Electrical and Computer Engineering, University of Toronto

[7] *Itai Katz*, A survey of digital watermarking techniques, March 2006

[8] *H. Zhang, X. Zang, and J. Liu*, “A Secure BPCS Steganography against Statistical Analysis” IEEE CNF. Digital Object Identifier

[9] *C. Rey and JL. Dugelay*, A survey of watermarking algorithms for image authentication, EURASIP Journal on Applied Signal Processing, 6:613–621, 2002

[10] *Frank Y Shih*, Digital Watermarking and Steganography: Fundamentals and Techniques, Dec. 17, 2007.