A Secure and Localizing Watermarking Technique for Image Authentication

Abdelkader H. Ouda and Mahmoud R. El-Sakka

Computer Science Department, University of Western Ontario, London, Ontario, Canada
{kader,elsakka}@csd.uwo.ca

Abstract. In this paper, a new block-based image-dependent watermarking technique is proposed. The proposed technique utilizes the correlation coefficient statistic to produce a short and unique representation (also known as hashed values or string-sequences) of the image data. These string-sequences are signed by an error-correcting-code signature scheme, which produces short and secure signatures. The image's least significant bits are utilized to embed these signatures. The used signature scheme requires string-sequences to be decodable syndromes. While the proposed correlation coefficient statistic function produces decodable syndrome string-sequences, most of the existing cryptographic hash functions do not. The results show that the proposed technique has an excellent localization property, where the resolution of the tracked tampered areas can be as small as 9x9 pixel blocks. In addition, the produced watermark has multi-level sensitivity that makes this technique well suited to the region-of-security-important approach, which increases the overall system performance.

1 Introduction

The growing development of image processing, as well as the Internet's popularization, has propelled image authentication issues to the forefront of the digital images field. Image authentication methods attempt to ensure the truthfulness of image content and its integrity. One of the best-known tools that provide reasonable solutions to this issue is Digital watermarking. Digital watermarking is a process in which signals, also known as watermarks, are embedded into digital data (images, video, or audio). These signals can be detected or extracted later to make an assertion about the data.

Over the past few years digital watermarking has received considerable attention from leading researchers around the world. Yeung *et al.* proposed a watermarking technique for image authentication [1]. In this technique the watermark is a binary image with the same size of the original image. This binary image is formed by tiling small binary images, such as a company logo, to cover the size of the original image. A key-based *lookup table* (LUT) is used in the embedding process. The LUT maps the original image pixels to the corresponding binary values in the binary image. In the verification process, every pixel of the image under question is tested by applying the same LUT to find the corresponding binary value. If the image is altered, the modified locations should appear in the extracted binary image.

The advantage of this technique is that the authentication process is done in a pixel-by-pixel basis, and image alteration can be visually detected. However the watermark is image-independent, which weakens the security of the system. Fridrich

A. Campilho, M. Kamel (Eds.): ICIAR 2004, LNCS 3212, pp. 759–769, 2004.
© Springer-Verlag Berlin Heidelberg 2004

et al. [2] has shown that if the same logo and key are reused for at least two images, it becomes very easy to accurately estimate the LUT. Hence, they proposed a solution by replacing the LUT with a public-key encryption scheme. The proposed modification becomes, however, computationally expensive because the encryption must be done at each pixel. Therefore, in practice this scheme cannot be widely implemented.

In 1998, Wong [3] proposed a block-based watermarking technique for image integrity verification, where the used block size was 8x8 pixel. In 2000, Wong and Memon [4] republished this technique with some variations that made it resist the Vector Quantition watermark attack [5]. One year later, the same authors recommended using an image block of size 12x12 [6] in order to hold the full length of the output of the MD5 hash function [7]. The watermark in this technique is constructed, as in Yeung'schnique [1], through a tiled small binary image. The original image is scanned block-by-block, each block is hashed together with some image information. These hashing values are then combined with the binary image using the binary XOR operation. The outut is encrypted and the produced ciphertext is embedded into the corresponding blocks.

Recently, Ouda et al., in [9] showed that, this technique suffers from a serious security leak. The main reason of this leak is that, the authors made an assumption that the plaintext size determines the ciphertext size in the signing process. This is no always a true assumption. On the contrary, the truth is, it is the secret key size that determines the ciphertext size. They also proposed a novel solution for these leak such that larger image block, 32x32 pixel, is used while the detection accuracy is kept almost the same as in the original technique.

In [10] Fridrich proposed another block-based image authentication technique. This technique is based on the main idea of the Wong' technique [3]. The main contribution of this technique is proposing a new solution to resist the Vector Quantization attk [5]. Instead of using a fixed binary image, as in Wong' technique, an 8´16 binary blocks are created and concatenated to form the binary image. These blocks are formed such that it is simply recognizable, and hold some information of the corresponding mage block, such as block index, the image index, and author ID. In fact the proposed solution is also suffer from the same problem of Wong' technique we mentioned above. Yet the image blocks size is 8x16 pixel, which is not enough to hold a secure signature, e.g., 1024 bits.

Over the last few years, many image watermarking techniques are proposed. Yet these techniques have been broken [1,3,4,6,10] (i.e., proven to be cryptographically insecure), or many others have proved to be impractical [2,11,12]. In this paw practical and secure watermarking technique is proposed. This paper is organized as follows. In Section 2, the general framework of the proposed technique is presented, while the detail descriptions of the main components are shown in Sections 3, 4, and 5. Experimental results come in Section 6. The conclusion is offered in Section 7.

2 System Framework

The proposed system framework includes two main processes, namely:

- Watermark generation and embedding process
- Watermark extraction and authentication process

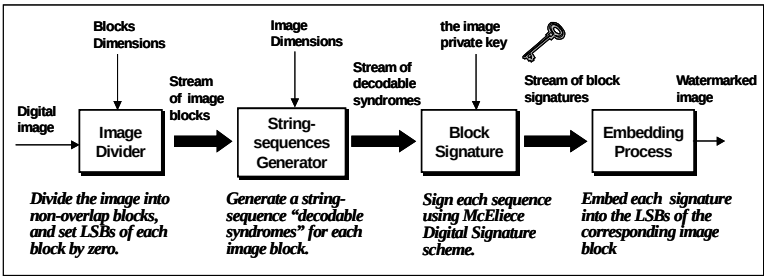

Fig. 1. The framework of the watermark generation and embedding process.

Each of these processes consists of several other units. In this section the main ideas of each unit will be demonstrated, however the detail descriptions will be given in Sections 3, 4 and 5.

2.1 Watermark Generation and Embedding Process

The watermark generation and embedding process is consists of four main units (see Fig. 1). The *image divider* unit is responsible for dividing an image into small non-overlap blocks. The image and the sub-blocks dimensions are given to this unit as an input. Section 3, will show how the image divider unit deals with an image to fulfill the demands of the *region-of-security-important* (ROSI) approach. The main function of the *string-sequence generator* unit is to produce a hashed value for each image block (see Section 4). These hashed values are generated such that they become valid (decodable) syndromes. These *decodable syndromes* are used in the *digital signature unit* to sign the image blocks, using the image private-key. The *block signature unit* is utilizing a digital signature algorithm based on error-correcting-code cryptosystem. The length of the ciphertext of each signature will be as small as 81 bits. Finally, the *embedding unit* will insert each signature (ciphertext) into the LSB of the corresponding image block using the first 81 bits only, to produce the watermarked image. Note that, the image block size might be, in some portions of the image, larger than 9x9 pixel. In this case the image divider unit will set the first 81 bits in these blocks by zeros, and leave the rest untouched. Once the watermark (the blocks signatures) is embedded into the image, the image can be securely distributed.

2.2 Watermark Extraction and Authentication Process

The watermark extraction and authentication process is responsible for extracting and verifying the signature of the image under question, which originally was signed by the watermark generation process. Fig. 2 illustrates the watermark extraction and authentication five main units. The *image divider unit* divides the image into small blocks with the same sizes as it is occurred in the embedding process. The *watermark extraction unit* will extract the block signatures from the LSB of the image. The *decryption unit* will recover the string-sequences corresponding to each image block, using the image owner public-key. At the same time, the string-*sequence generator unit* will do the same job as in the embedding process to generate a string-sequence corresponding to each image block. Finally, the sequence comparison unit will test

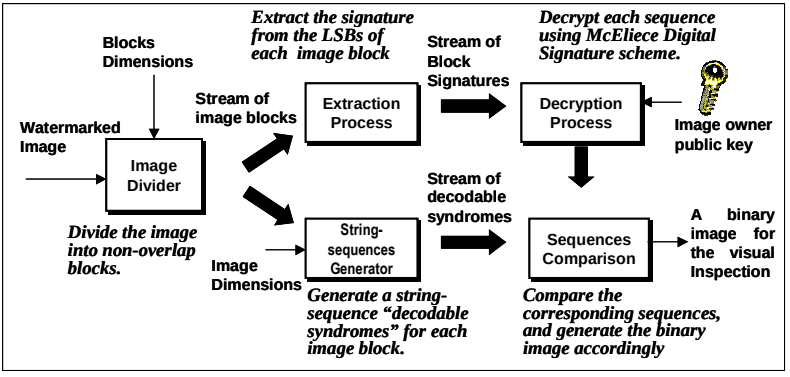

Fig. 2. The framework of the watermark extraction and authentication process.

and compare each pair of the extracted and generated string-sequences. By testing all these pairs the *sequence comparison unit* will produce a binary image to authenticate the image visually. Note that, all information needed during the extraction and verification process has already been embedded into the image, and there is no need for having the original image/watermark during this process.

3 Image Divider Unit

Typically, not all objects in an image have the same value, and accordingly, they do not need the same level of protection. There are many examples of such images, however if we look at the cheque image illustrated in Fig. 3, we will observe that: the courtesy amount area is very important, and it is highly targeted by the attacker. A small modification of the contents of this area would make a great change of the cheque image. Whereas, modifying some part of the background would not make that difference. This is a typical scenario of what so called region-of-security-important (ROSI) approach.

The image divider unit helps provide a practical solution to the above problem. When an image is dividing into small blocks, this unit takes into consideration some factors such as the significant part of the image, the quality (the resolution) of these areas, the overall cost and system performance. For instance, on cheque images in Fig. 4, the courtesy amount area may be divided into tiny block sizes (as small as 9x9 pixel block) to recognize small changes. The signature area of the cheque is also importance, but it always has bigger shape than that in the courtesy area, therefore it might be divided into 12x12 pixel block. The remaining areas in the cheque image will be divided into bigger sub-blocks; their dimensions are based on the image size and the computational resources available. The image divider unit is also responsible to set the LSBs of the first 81 bits of each image block to zero in order to reserve these positions for the generated watermark.

4 String-Sequence Generation

The string-sequence generator unit plays the main roll in the proposed technique. This unit utilizes the correlation coefficient statistic to produce small and unique repre-

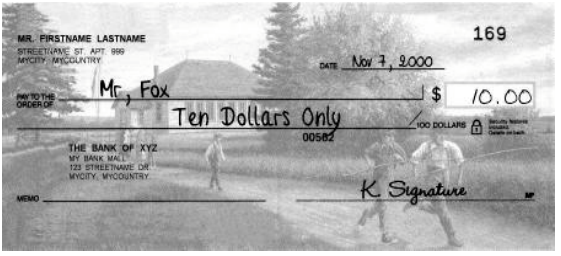

Fig. 3. A cheque image having multi-level of security importance.

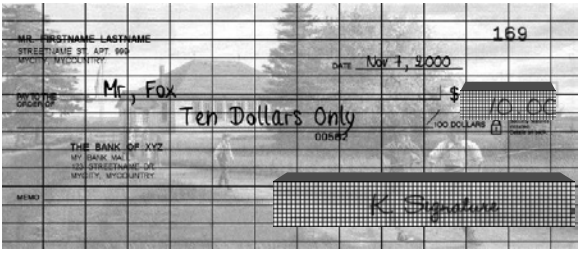

Fig. 4. The image is divided into sub-blocks in different sizes based on ROSI, where the cheque courtesy amount area is divided into 9x9 pixels blocks, the signature area is divided into 12x12 pixels, and the remaining parts is divided into 128x64 pixels.

sentation for a given image or any sub-block within it. Correlation coefficients are a useful and potentially powerful tool that statistically measures the relationship between two sets of variables, e.g., two adjacent columns or two adjacent rows in a given image. The relationships of the image-block pixels with regard to its neighbors are measured and combined all together in a way to produce one value called "string-sequence". The correlation coefficient, ρ, between any two adjacent rows, or columns, A and B, can be calculated using Eq. (1),

$$\rho = \frac{\sum_{i=1}^{n}(A_i - \overline{A})(B_i - \overline{B})}{\sqrt{\left(\sum_{i=1}^{n}(A_i - \overline{A})^2\right)\left(\sum_{i=1}^{n}(B_i - \overline{B})^2\right)}} \tag{1}$$

where A_i and B_i are two pixel values located in the same row at two adjacent columns, or located in the same column at two adjacent rows. $\overline{A}$ and $\overline{B}$ are the averages n numbers of A_i and B_i respectively.

Fig. 5, illustrates a row/column-wise correlation coefficients calculation. It shows how correlation coefficients preserve the relationship between a pixel and its 4-connected neighborhoods. For example, pixel i_{22} is compared with pixel i_{21} in $C\rho_1$, pixel i_{23} in $C\rho_2$, pixel i_{12} in $R\rho_1$, and with pixel i_{32} in $R\rho_2$.

The string-sequence for a given *m*x*n* block is calculated as follows:

1. Calculate the n-1 column-wise correlation coefficients $C\rho_i$ using Eq. (1), where $i = 1, \cdots, n-1$

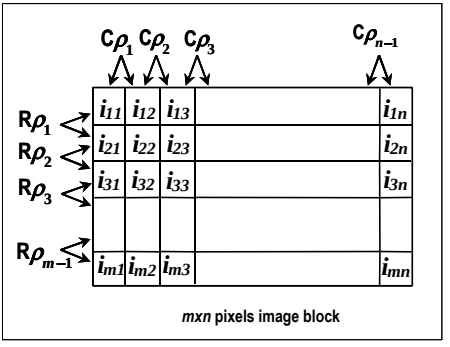

Fig. 5. Pixel representation of *m*x*n* image block, where $C\rho_i$ is column-wise correlation coefficients, and $R\rho_j$ is column-wise correlation coefficients.

2. Calculate the m-1 row-wise correlation coefficients $R\rho_j$ using Eq. (1), where $j = 1, \cdots, m-1$
3. Compute the value v using Eq. (2), i.e., is the summation of all values that produced from the above two steps.

$$v = \sum_i C\rho_i + \sum_j R\rho_j \tag{2}$$

4. Calculate the average of the image block w.
5. Calculate the string-sequence s using Eq. (3),

$$s = ddp(v^2) + ddp(w^2) + M \times N \mid i_0 \tag{3}$$

where i_0 is the smallest integer by which we can find z such that $Hz^T = s$

Note that, *ddp* (stands for *Drop Decimal Point*) is a function that drops the decimal point from a real number and makes it an integer number. For example if v=23.65908665321087 then *ddp*(v) becomes 2365908665321087. Note also that, in this work v and w are double precision variables with a 52-bits mantissa, and hence the string-sequence can be any positive integer that bounded to 16-digits in length.

From the definition of the correlation coefficients, in some special cases the output of Eq. (3) might be the same for two different image blocks. To avoid these cases, the following transformations will be made to an image data before applying Eq. (3).

Case 1: the pixels values are paired in a relation such that high values are paired with relatively high values, and low values are paired with relatively low values within a specific ratio. For example, consider the following two blocks:

Block 1:	100	200	10	70	*Block* 2:	50	45	90	5
	50	100	5	35		100	90	180	10

By applying Eq. (2) to both block 1 and block 2, we notice that v has the same values being 4. Hence, after applying Eq. (3), the string-sequences for block 1 and block 2 will be the same and will be equal to 50765641. Note that the averages of these two blocks are also the same.

Solution: a dummy variable is added to each row and column, such that the value of these variables should be in increasing order, e.g., 1,2,...,*n* mode 256. This modulus is taken in order to ensure that these variables always run within the range from 0 to 255. Now, the above blocks become:

Block 1:					
	100	200	10	70	1
	50	100	5	35	2
	1	2	3	4	

Block 2:					
	50	45	90	5	1
	100	90	180	10	2
	1	2	3	4	

In this case, block 1 has: $v = 15.6067501197491$, $w = 71.25$, and the string-sequence = 156067551963116, and Block 2 has: $v = 15.4975646811457$, $w = 71.25$, and the string-sequence = 154975697577082.

Case 2: When the image blocks having the same pixel values but not the same positions. For example, a block that is rotated, or flipped. Consider the following two blocks,

Block 1:				
	120	120	170	120
	170	170	120	120
	120	170	170	120
	120	170	170	170

Block 2:				
	170	120	170	170
	120	120	120	170
	120	170	120	120
	120	170	170	170

In this example, the value of v of both blocks is 0.333333333, and they have the same block average equal to 145. Hence the string-sequence of these blocks will be the same and will equal 1111111111132136.

Solution: transform the image block to another domain, in which the position of each pixel is preserved. This transformation is calculated by Eq. (4):

$$\hat{x}_i = \left(x_i + \left(\frac{256}{n} * i \right) - 1 \right) \bmod 256 \tag{4}$$

Where $\hat{x}_i$ is the transformation of the pixel x_i at position i in a given row or column, and n is the length of the block row (row transformation), or the length of the block column (column transformation). For example consider the two blocks mentioned above (case 2), therefore they will be transformed to the following blocks (row transformation):

Block 1:				
	183	247	105	119
	233	41	55	119
	183	41	105	119
	183	41	105	169

Block 2:				
	233	247	105	169
	183	247	55	169
	183	41	55	119
	183	41	105	169

After this transformation, block 1 has: $v = 2.14609511595264$, $w = 128$, and the string-sequence = 46057242483542, and Block 2 has: $v = 3.10488641778782$, $w = 128$, and the string-sequence = 964031966757064.

5 Image Block Signature Unit

The image block signature adopts an error-correcting-code-based digital signature scheme to sign the string-sequences and produce the image watermark. This scheme is proposed by Courtois, *el. al.* [13]. Please refer to the articles in [14-18] for the theoretical background of this scheme. Courtois digital signature scheme gives short signatures of 81-bits with a security strength based on the difficulty of the syndrome decoding which was proven to be NP-complete [14]. This digital signature scheme is

based on Niederreiter's cryptosystem [17] with public key H', a scrambled parity-check matrix of a binary Goppa code.

The signature of an image data is based on the idea that we search for the first random decodable syndrome *s,* such that we can find a vector z satisfying $H'z^T = s$.

Table 1. The values of the average of smallest difference and the smallest difference of string-sequences among 8 sizes of image block.

Block Size	The average of the string-sequence differences	The minimum difference of the string-sequences
512x512	111340451142004	48970444071
256x256	120781099775258	53404490671
128x128	27790549514366	2041909336
64x64	7847346380265	6044140929
32x32	1331831487110	559798192
16x16	56782720174	119172184
12x12	14082143131	174969702
9x9	44134913	1065479

The signature will be the vector z. The probability to find a random decodable syndrome, using the Goppa code, is $1/9!$. The string-sequence generation unit is designed to provide such syndromes (see Section 4).

6 System Tests and Results

Two main experiments were conducted on a database of 680 different images. These images are scanned at a resolution of 200 dots/inch. The produced images are 1274x552 pixels each. The first experiment assessed the collision resistance of the string-sequences, whereas the second experiment tested the altering location detection property.

6.1 Collision Resistance Experiment

The string-sequences are called collision resistant, if it is hard to find two different image blocks having the same string-sequence. To test the satisfaction of this property, each image in the database is divided into non-overlapping small blocks. The sizes of these blocks are chosen to be 512x512, 256x256, 128x128, 64x64, 32x32, 16x16, 12x12, and 9x9. Note that, in some cases the block cannot completely tile the entire image, in such cases the block will be wrapped around the image boundary. The string-sequence is generated for each block in a given image. The smallest difference between any two string-sequences is calculated. Note that, if this difference is greater than zero, this means there is no collision.

The results of this experiment are summarized in Table 1. Each row shows the used image block size, the average of the string-sequences among all blocks, the minimum difference of the string-sequences. The smallest number in the third column of Table 1 is 1065479, which is the smallest difference between any two string-sequences of a given image. We conclude that, the image block in each image can be

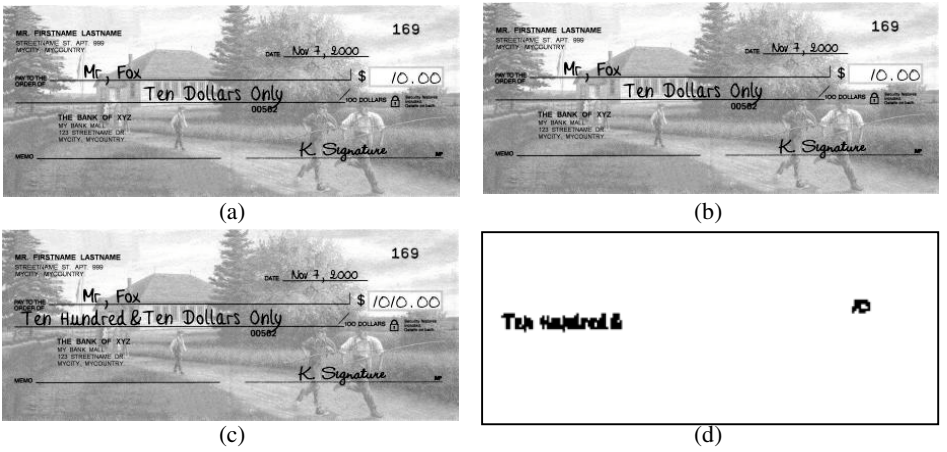

Fig. 6. (a) The original image, (b) The watermarked image, (c) some modification in the image pixels, (d) the produced binary image after editing modifications.

represented by a unique string-sequence, even in a block size 9x9. The length of the string-sequences varies based on the image block sizes. The average of the differences of the image string-sequences in Table 1 shows how far the string-sequence is from collision.

6.2 Altering Location-Detection Property

When a watermarking scheme is able to identify a modified pixel region in a given image, it satisfies the *altering location detection property*. To test this property, the image in Fig. 6(a) is used as an original image that is needed to be protected. The produced watermarked image is shown in Fig. 6(b). The watermarked image is modified as shown in Fig. 6(c). The extraction process produced the binary image shown in Fig. 6(d). This image shows that the modified areas have been successfully identified, and located.

7 Conclusion

In this paper, a new block-based image-dependant watermarking technique is proposed. In this technique a correlation coefficient statistic is utilized to produce a small and unique representation (string-sequence) for a given image or any sub-block within it. These string-sequences are generated such that it easily converted to be decodable syndromes. An error-correcting-code digital signature scheme is used to sign the image data. Experimental results showed that the produced string-sequences are collision resistant. More precisely even if, after exhaustive search of the string-sequences, a collision were occurred then the two input image data will differ in what the human eyes cannot distinguish. This is because the string-sequence is produced from an image-depended hashing function. The experiments also showed that the performance of the proposed technique, both in terms of cryptographic security and the localization property, is superior to other counterparts available today.

Acknowledgement. This work was partially supported by the Ontario Graduate Scholarship (OGS). This support is greatly appreciated. Special thanks belong to Nicolas Sendrier, and Matthieu Finiasz for the useful support for testing their signature scheme.

References

1. M. Yeung, F.Mintzer: An Invisible Watermarking Technique for Image Verification, IEEE International Conference on Image Processing, vol. 2, pp. 680–683, 1997.
2. J. Fridrich, M. Goljan, and N. Memon: Further attacks on the Yeung-Mintzer fragile watermark, SPIE Photonics West, Electronic Imaging 2001, Security and Watermarking of Multimedia Contents II, vol. 3971, pp. 428–437, 2000.
3. P. Wong: A Public Key Watermark for Image Verification and Authentication, IEEE International Conference on Image Processing, vol. I, pp. 455–459, 1998.
4. N. Memon and P. Wong: Secret and Public Key Authentication Watermarking Schemes that Resist Vector Quantization Attack, SPIE International Conference on Security and Watermarking of Multimedia Contents II, vol. 3971, pp. 471–427, 2000.
5. M. Holliman and N. Memon: Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes, IEEE Transactions on Image Processing, vol. 9, no. 3, pp. 432-441, 2000.
6. P. Wong and N. Memon: Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification, IEEE Transactions On Image Processing, vol. 10, no. 10, pp. 1593–1601, 2001.
7. R. Rivest: The MD5 message digest algorithm, Technical Report RFC1321, Internet Engineering Task Force, 1992.
8. R. Rivest, A. Shamir, and L. Adleman: A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.
9. A. Ouda and M. El-Sakka: Technical report on methods to correct the Wong-Memon image watermaking scheme, London, Ontario, University of Western Ontario, Allyn and Betty Taylor Library, no QA76.5.L653 no.603, 2003.
10. J. Fridrich: Security of Fragile Authentication Watermarks with Localization, SPIE Photonic West, Electronic Imaging 2002, Security and Watermarking of Multimedia Contents, vol. 4675, pp. 691–700, 2002.
11. J. Fridrich, M. Goljan, and A. Baldoza: New Fragile Authentication Watermark for Images, IEEE International Conference on Image Processing, vol. 1, pp. 446–449, 2000.
12. M. Costa, Writing on dirty paper, IEEE Transactions on Information Theory, vol. 29, no. 3, pp. 439–441, 1983.
13. N. Courtois, M. Finiasz, and N Sendrier: How to achieve a McEliece-based digital signature scheme, Advances in Cryptology - ASIACRYPT 2001, LNCS 2248, pp. 157–174, Springer-Verlag, 2001.
14. R. McEliece E. Berlekamp and H. Tilborg: On the inherent intractability of certain coding problems, IEEE Transactions on Information Theory, vol. 24, no.3, pp. 384–386, 1978.
15. R. McEliece: A public-key cryptosystem based on algebraic coding theory, *Jet* Propulsion Lab. DSN Progress Report, 1978.

16. R. Deng, Y. Li and X. Wang: On the equivalence of mceliece's and niederreiter's public key cryptosystems, IEEE Transactions on Information Theory, vol. 40, no. 1, pp. 271–273, 1994.
17. H. Niederreiter: Knapsack-type cryptosystems and algebraic coding theory, In Problem, Contribution and Information Theory, vol. 15, pp. 159–166, 1986.
18. T. Cover: Enumerative source encoding, IEEE Transactions on Information Theory, vol. 19, pp. 73–77, 1973.