A Joint Encryption/Watermarking System for Verifying the Reliability of Medical Images

D. Bouslimi, G. Coatrieux, *Member, IEEE*, M. Cozic and Ch. Roux, *Fellow, IEEE*

***Abstract*— In this paper, we propose a joint Encryption/Watermarking (E/W) system for the purpose of protecting medical images. This system is based on an approach which combines a substitutive watermarking algorithm, the quantization index modulation (QIM), with an encryption algorithm: a stream cipher algorithm (e.g. the RC4) or a block cipher algorithm (e.g. the AES in CBC mode of operation). Our objective is to give access to the outcomes of the image integrity and of its origin even though the image is stored encrypted. If watermarking and encryption are conducted jointly at the protection stage, watermark extraction and decryption can be applied independently. The security analysis of our scheme and experimental results achieved on 8 bit depth ultrasound images as well as on 16 bit encoded PET images demonstrate the capability of our system to securely make available security attributes in both spatial and encrypted domains while minimizing image distortion. Furthermore, by making use of the AES block cipher in CBC mode, the proposed system is compliant with or transparent to the DICOM standard.**

***Index Terms*— Block Cipher, Encryption, Medical Image Security, Quantization Index modulation, Stream Cipher, Watermarking.**

I. Introduction

THE rapid evolution of multimedia and communication technologies offers new means of sharing and remote access to patient data. In particular, medical imaging is already called to play important roles in applications like telesurgery, telediagnosis and so on. But at the same time, this ease of transmission and sharing of data increases security issues in terms of [1]:

- Confidentiality, which means that only authorized users can access patient data.
- Availability, which guarantees access to medical information in the normal scheduled conditions of access and exercise.
- Reliability, which is based on: i) Integrity - a proof that the information has not been altered or modified by non-authorized persons; ii) Authentication - a proof of the information origins and of its attachment to one patient. Reliable pieces of information can be used confidently by the physician.

In any information systems, data confidentiality, integrity and non-repudiation services are usually achieved by cryptographic means. DICOM[1], the standard of reference for medical images, allows data encryption through the triple DES[2], the AES[3] ..., as well as digitally signing a DICOM object by making use of the DSA[4] (see Part 15 of the DICOM standard). However, once decrypted or its digital signature deleted or lost, one piece of information is no longer protected and it becomes hard to verify its integrity and its origin. From this point of view, these cryptographic means, especially encryption, rather appear as an "*a priori"* protection mechanisms.

Watermarking has been proposed as a complementary mechanism to improve the security of medical images [2]. When it is applied to images, watermarking modifies or modulates the image pixels' gray level values in an imperceptible way, in order to encode or insert a message (i.e. the watermark). Thus, it allows us to intimately associate protection data with the information to be protected. Watermarking can be used for verifying the reliability of an image by asserting its integrity and its authenticity. For instance, in a transaction, patient name and physician identity can be inserted in the image [3-6]. As defined, watermarking is an "*a posteriori*" control mechanism as the image content is still available for interpretation while remaining protected.

Different approaches have been proposed in order to benefit from the complementarity of these two mechanisms in terms of *a priori*/*a posteriori* protection, essentially in the context of copyright protection. Technically, two categories of methods can be distinguished according to the way watermarking and encryption are merged:

- Joint decryption/watermarking, where watermark embedding is conducted during the decryption process [7-10].
- Joint encryption/watermarking, where watermarking and encryption step processes are merged. In this case, the watermark can be extracted: *i*) in the spatial domain, i.e. after the decryption process, or; *ii*) in the encrypted domain, or; *iii*) in both domains [11].

The system we propose in this paper belongs to the second category. It merges a substitutive watermarking algorithm, the

D. Bouslimi, G. Coatrieux and Ch. Roux are with the Institut Mines - Telecom; Telecom Bretagne; Unite INSERM 1101 Latim, Technopole Brest-Iroise, CS 83818, 29238 Brest Cedex 3 France (e-mail: {dalel.bouslimi, gouenou.coatrieux, christian.roux}@telecom-bretagne.eu).

M. Cozic is with MEDECOM, Plougastel Daoulas 29470, France (e-mail: mcozic@wanadoo.fr).

[1] DICOM: Digital Imaging and Communications in Medicine; http://medical.nema.org.

[2] Data Encryption Standard

[3] An asymetric encryption algorithm proposed by Rivest, Shamir and Adleman.

[4] Digital Signature Algorithm

Quantization Index Modulation (QIM) and an encryption algorithm which can be: a stream cipher algorithm (e.g. RC4[5]) or a block cipher algorithm (e.g. AES). Our objective is to give access to embedded security attributes in the encrypted and spatial domains for the purpose of verifying the reliability of an image.

The rest of this paper is organized as follows. In section II, we independently present the watermarking and the cipher algorithms we used, before introducing their combination in section III. We then detail our implementation in section IV. Section V presents some experimental results considering two distinct medical modalities, ultrasound and positron emission tomography and discusses some constraints of deployment. Before concluding we analyze the security of the proposed scheme in section VI.

II. Cryptographic and watermarking Primitives

A. Cryptographic primitives

Basically, there exist two types of encryption algorithms: block cipher algorithms and stream cipher algorithms. Block cipher algorithms, like the AES and the DES, operate on large blocks of plaintext whereas stream cipher algorithms, like the RC4 or the SEAL[6] [12], manipulate stream of bits/bytes of plaintext.

1) The RC4 stream cipher algorithm

As described in Figure 1, stream cipher algorithms combine the bits/bytes of plaintext $T = [t_1, \dots t_i, \dots, t_n]$ with a secret keystream of bits/bytes $K = [k_1, \dots k_i, \dots, k_n]$ issued from a pseudo random number generator (PRNG), through a *xor* operation typically. The keystream generation depends on one secret key K_e, making stream cipher algorithms as part of symmetric encryption techniques. Thus, bits/bytes of cipher text $C = [c_1, \dots c_i, \dots, c_n]$ are usually defined as:

$$c_i = t_i \oplus k_i \qquad (1)$$

Some of the main advantages of this type of algorithms are that they are simple and operate at a higher speed than block cipher algorithms [13].

The specificity of such stream cipher algorithm resides in how the bit/byte keystream is generated by the PRNG. The RC4 PRNG is based on two steps:

- "Initialization", where a table of 256 bytes is filled by repeating the encryption key as often as necessary until to fill this table.
- "Byte keystream generation", where the elements of the table are combined by applying permutations and additions to generate the keystream.

More details about stream cipher algorithms can be found in [12].

2) The AES in CBC mode of operation

In this work, we use the block cipher algorithm AES in the Cipher Block Chaining (CBC) mode of operation in order to be compliant with the DICOM standard. The concept of mode of operation refers to the manner in which blocks of plaintext (sequence of bytes) are treated at the encryption stage (*resp.* decryption stage). As depicted in Figure 2, when the CBC mode is applied, a plaintext block is combined, with the previous ciphertext block through a *xor* operation before being encrypted with the AES. If we denote B_i^e the encrypted version of a block B_i and B_{i-1}^e the previous encrypted block, B_i^e is thus given by:

$$B_i^e = AES(B_i \oplus B_{i-1}^e, K_e)$$

where K_e is the encryption key. The reader may refer to [14] for a complete description of the AES.

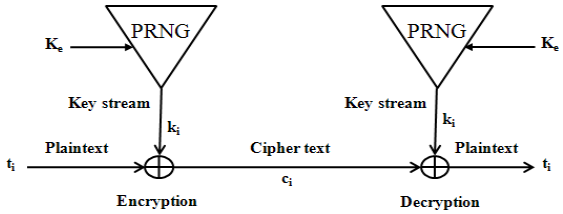


Fig.1. Encryption/Decryption processes of a stream cipher algorithm which secret key is K_e. t_i, c_i and k_i correspond to the plain text bits/bytes, the cipher text bits/bytes and the secret keystream bits/bytes respectively. k_i is issued by a pseudo random number generator (PRNG).

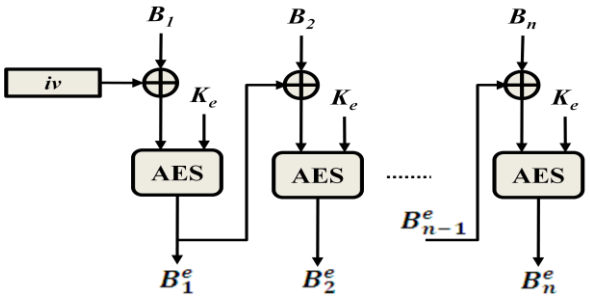


Fig.2. AES Encryption in CBC mode. B_i, B_i^e and K_e denote the plaintext block, the encrypted block and the encryption key, respectively. *iv* is a random initialization vector.

B. Watermarking primitive: the QIM modulation

The Quantization Index Modulation (QIM), proposed by Chen and Wornell [15], relies on quantifying the components of one image according to a set of quantizers based on codebooks in order to insert a message. More clearly, to each message m_{si} issued from a finite set of possible messages $Ms = \{m_{si}\}_{i=0,\dots,q_s}$, the QIM associates the elements of a codebook $C_{m_{si}}$ such as:

$$C_{m_{si}} \cap C_{m_{sj}} = \emptyset, i \neq j \qquad (2)$$

Substituting one component of the image by its nearest element in the codebook $C_{m_{si}}$ thus allows the insertion of m_{si}. Let us consider one image component such as a vector of pixels $X \in \mathbb{N}^N$ while dividing the $\mathbb{N}^N$ dimensional space into non overlapping cells of equal size. To satisfy (2), each cell is associated to a codebook $C_{m_{si}}$, $i=0,\dots,q_s$. As a consequence, one message m_{si} has several representations in $\mathbb{N}^N$. The insertion process is conducted as follows. If X belongs to the cell which encodes the message to be inserted, X_w (the watermarked version of X) corresponds then to the center of this cell, otherwise X is moved to the center of the nearest cell that encodes the desired message. During the extraction step, the knowledge of the cell to which X_w belongs is enough to identify the embedded message. Notice, that such a modulation

[5] Rivest Cipher 4
[6] Software-Optimized Encryption Algorithm

definitively alters the image. We will come back on this issue in section V.

III. Proposed Joint Encryption and Watermarking System

A. System architecture and principles

The purpose of our system is to verify the reliability of an image within the spatial domain as well as within the encrypted domain. As illustrated in Figure 3, it relies on two main procedures: protection and verification. The protection stage (fig. 3a) jointly conducts the watermarking and encryption of an image *I*. It allows us to insert two messages, Msg_s and Msg_e, which will be available in the spatial and encrypted domains respectively. The insertion and the extraction of each message depend on a watermarking key: K_w^e for the encrypted domain and K_w^s for the spatial domain.

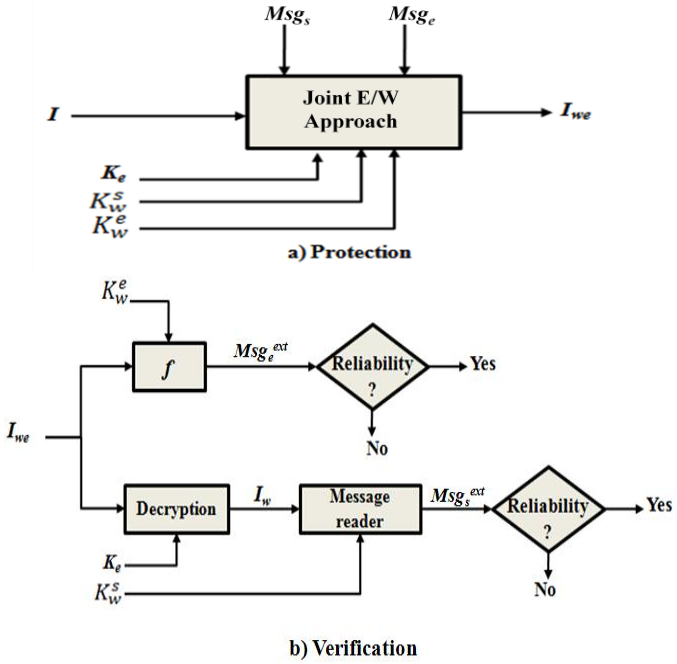


Fig.3. Architecture of the proposed system. *I*, I_{we}, I_w, K_e, K_w^s and K_w^e denote the original image, the watermarked encrypted image, the watermarked decrypted image, the encryption key and the watermarking keys for the spatial and encrypted domain respectively. Msg_e and Msg_e^{ext} are the embedded and extracted messages in the encrypted domain, respectively. Msg_s and Msg_s^{ext} denote the embedded and extracted messages in the spatial domain, respectively. *f* is the watermarking extraction function in the encrypted domain.

These two messages contain security attributes that will assess the image reliability in each domain. Indeed, each message contains an authenticity code *AC*, which identifies the image origin (e.g. about 600 bits by combining the French National Identifier with the DICOM Unique Identifier [16]), and an integrity proof. In the spatial domain, integrity is ensured by making use of a secure hash function (e.g. SHA[7]) computed on the image bit subset that is not modified by the watermarking process. We call this subset of bits *nmb*. So, the message available in the spatial domain, Msg_s, is defined as follows:

$$Msgs =< AC, SHA(nmb) > \quad (3)$$

In the encrypted domain, integrity is controlled by verifying the presence of a secret pseudo random sequence of bits generated using a secret watermarking key. As we will see and discuss in section III.C, the integrity of the watermarked-encrypted image is considered as valid if we retrieve these bits at specific locations within the SHA signature of each watermarked-encrypted block bytes. We consider this pseudo random sequence as a proof of integrity. In consequence, the verification of the image authenticity and integrity in the encrypted domain, relies on extracting Msg_e given by

$$Msg_e =< AC, PRNG(K_w^e) > (4)$$

where K_w^e represents the watermarking key in the encrypted domain. K_w^e initializes the PRNG function.

Anyway, as it can be seen in Figure 3b, protection data are made available from the encrypted image or from the decrypted image for a subsequent verification stage. If watermarking and encryption are jointly conducted, watermark extraction and image decryption are two independent processes.

B. Combination of encryption and watermarking

In this section, in order to simplify the presentation of our system we manipulate 8 bit encoded images.

1) General principles of joint E/W approach

Let us consider one block of bytes or equivalently a set of contiguous pixels. For this block, our objective is to give access to two messages: m_{si}, the message available in the spatial domain; and, m_{ej}, the message available in the encrypted domain. Similarly to m_{si} (see section II.B), m_{ej} is a message issued from a finite set of possible messages $Me = \{m_{ej}\}_{j=0,\dots,q_e}$.

In order to conduct jointly this double watermarking process and to avoid any interference between them, we propose to adapt the QIM described previously. The basic idea is to decompose each codebook $C_{m_{si}}$ into sub-codebooks $C_{m_{si}m_{ej}}$ such as

$$C_{m_{si}} = \cup_{j=0}^{q_e} C_{m_{si}m_{ej}} \quad (5)$$

$$C_{m_{si}m_{ej}} \cap C_{m_{si}m_{ek}} = \emptyset, j \neq k \quad (6)$$

Considering a vector of pixels $X \in \mathbb{N}^N$, m_{si} and m_{ej} are then embedded simultaneously by replacing X with X^w which corresponds to the nearest element of X in $C_{m_{si}m_{ej}}$. Using the Euclidian distance, X^w is given by

$$X^w = min_k(\|X - Y_{ik}\|),\ Y_{ik} \in C_{m_{si}m_{ej}} \quad (7)$$

Making the message m_{ej} available in the encrypted domain depends on the sub-codebook construction which is a process intimately linked with the encryption algorithm and also with the watermark extraction algorithm. Considering an encryption algorithm *E* and its encryption key K_e, sub-codebooks $C_{m_{si}m_{ej}}$ are built so as to verify

$$C_{m_{si}m_{ej}} = \{Y \in C_{m_{si}} / f(Y^e, K_w^e) = m_{ej}\} \text{ where } Y^e = E(Y, K_e) \quad (8)$$

where *f* is the watermark extraction function in the encrypted

[7] Secure Hash Algorithm, conceived by the US National Security Agency. The SHA-1 provides a 160 bit long signature.

domain. The choice of the function f is closely related to the goals to be achieved by our system. We will explain in subsection III.C the choice we made.

Finally, to sum up this process, m_{ej} is made available in the encrypted domain by modulating pixel values in the spatial domain. This means replacing X by X^w, its nearest element in the sub-codebook $C_{m_{si}m_{ej}}$.

2) Implementation with a cipher algorithm: the AES in CBC mode and the RC4

Depending on the selected cipher algorithm, some other constraints have to be considered when building the subcodebooks $C_{m_{si}m_{ej}}$.

In the case, E corresponds to the AES in CBC mode, Y^e is given by (see eq.8):

$$Y^e = AES(Y \oplus X^e_{-1}, K_e) \quad (9)$$

where X^e_{-1} is the previous encrypted block of bytes or set of pixels. So, the construction of $C_{m_{si}m_{ej}}$ depends also on the previous encrypted block.

Unlike the AES, the RC4 encrypts each byte separately. When it is used, Y^e is given by (see eq.8):

$$Y^e = \{y_1^e, \dots, y_i^e, \dots, y_n^e\}, \quad \text{with } y_i^e = y_i \oplus k_i \quad (10)$$

where k_i corresponds to the i^{th} byte of the keystream k generated by the RC4 according to the secret key K_e. Thus, the decomposition of each codebook into sub-codebooks depends also on the keystream bytes which are different.

From these constraints, building the subcodebooks before protecting the image is out of interest. In order to reduce computation complexity, it is more realistic to determine sub-codebooks at each block to encrypt, it means to build the cells of the sub-codebooks $C_{m_{si}m_{ej}}$ into the cell of $C_{m_{si}}$which encodes the desired message in the spatial domain, i.e. m_{si}. Moreover, in practice (see section IV.A), we sequentially test the elements of $C_{m_{si}}$ until to find the nearest element X^w of X such as:

$$f(X^{we}, K_w^e) = m_{ej}, \text{ where } X^{we} = E(X^w, K_e) \quad (11)$$

In this study, for sake of simplicity, we work with single bit messages, i.e. m_{si} ={0,1} and m_{ej}={0,1}, and consequently with two codebooks C_0 (m_{s0}=*0*) and C_1 (m_{s1}=*1*), and four sub-codebooks C_{00}, C_{01}, C_{10} and C_{11} derived from C_0 and C_1 respectively. More precisely, in one block of byte or in one pixel subset, one bit will be embedded in the spatial domain as well as in the encrypted domain.

C. How ensuring the reliability in the encrypted domain?

To control the image integrity, one common solution is to compute its digital signature and to embed it. Obviously, if the embedding is not lossless or reversible [17], this signature is computed on the image parts that are left intact by the watermarking process. Because the message available in the encrypted domain results from distortion imposed in the spatial domain, and because the impact of these distortions in the encrypted domain are not predictable, it is not possible to compute the digital signature of the encrypted block and to embed it within itself. To overcome this issue, one alternative consists in verifying the presence of a pseudo-random sequence embedded at the protection process. For instance, such a sequence can be carried by the least significant bit (LSB) of some secretly selected bytes of each watermarked-encrypted block X^{we}. In that case, the watermark extraction function is such as $f(X^{we}, K_w^e) = LSB(X^{we})$. In fact, we force these LSBs to be equal to the bits of the pseudo random sequence by modulating the pixels values in the spatial domain. Unfortunately, with this strategy, the embedded signature is independent of the content, and the verifier has no means to check the link between the pseudo-random it extracts and the rest of the encrypted content. At the same time, the detection rate is rather small. Indeed, we can only detect modification of the pseudo-random sequence. If for example, only one bit of this sequence is embedded per watermarked-encrypted block of 8 bytes X^{we}, we have at least 1/128 chance to detect X^{we} has been modified.

To solve this problem of content independency and achieve better detection performance, we propose to verify the presence of the pseudo-random sequence and access to the image authenticity code, within the SHA signatures of the watermarked-encrypted blocks. By doing so, the watermark function f used to extract the message m_{ej} from X^e, is defined as

$$f(X^{we}, K_w^e) = h_k \quad (12)$$

where h_k corresponds to the k^{th} bit of H, the SHA-1 signature of X^{we} (i.e. $H = SHA(X^{we})$). The choice of the rank k depends on the secret watermarking key K_w^e. Because the "strength" of the SHA-1 is of 80bits, if one bit of X^{we} changes then there is one-in-two chance that h_k commutes. In that way, the recipient can verify the integrity and as well as the authenticity of the image in its encrypted form. It just has to extract Msg_e from the SHA signatures of each watermarked-encrypted blocks.

IV. Implementation of the Proposed Joint E/W System

As stated earlier, our implementation works with the RC4 or with the AES in CBC mode. In the following, we firstly describe how codebooks and sub-codebooks are built and then detail the different steps of our joint E/W algorithm.

A. Codebook construction

The first step consists in constructing the set of $C_{m_{si}}$ codebooks. Let us consider block of N pixels (or bytes) and, as we stated earlier, the insertion of one bit in both the spatial and encrypted domain (i.e. m_{si} ={0,1} and m_{ej}={0,1}). The value of N depends on the image bit depth and of the adopted cipher algorithm. Indeed, in the case of an image encoded on 16 bits, because the AES works with blocks of 16, 24 or 32 bytes, N will be equal to 8, 12 and 16 pixels, respectively. In the sequel, for sake of simplicity, we consider 8 bit depth images, i.e. N equals the number of bytes in an encrypted block.

In our implementation, $C_{m_{si}}$ is built as follows:

$$C_{m_{si}} = \{Y \in \mathbb{N}^N / \left\lfloor \frac{Y_k}{\Delta} \right\rfloor mod\ 2 = m_{si}\} \quad (13)$$

where Δ represents the quantization step and where Y_k is the k^{th} byte or equivalently the k^{th} pixel of the block to encrypt. The choice of k, depends on the secret watermarking key K_w^S in the spatial domain and is different for each pixel block. As designed, only one pixel in a pixel block X is quantized in order

to encode one bit of the message in the spatial domain (i.e. Msg_s).

In order to embed the message m_{ej} along with m_{si} into one pixel block, we propose to modulate l LSBs from p secretly selected pixels other than the pixel at the location k. Again this process is based on K_w^s. By doing so, $C_{m_{si}m_{ej}}$ regroups a subset of $\left|C_{m_{si}m_{ej}}\right| = 2^{lp}$ elements of $C_{m_{si}}$.

As exposed in section III.B.1 and in. eq. 12, X will be replaced by X^w, i.e. by its nearest element in $C_{m_{si}m_{ej}}$. In order to reduce the complexity, instead of calculating the whole set of elements of $C_{m_{si}m_{ej}}$, it is preferable to test these different elements depending on their Euclidian distance with X, starting by its nearest element, until the value of X^w that satisfies eq. 11 is found.

Based on this strategy, we can determine the probability for not being able to embed m_{ej} into a block X, i.e. that $f(X^{we}, K_w^e) \neq m_{ej}$ after having tested all 2^{lp} elements of $C_{m_{si}m_{ej}}$. Indeed, based on the properties of cryptographic hash functions (see section III.C), there is one-in-two chance that the change of one bit of X leads to the correct value of m_{ej}. As a consequence, the probability the embedding of m_{ej} fails is given by: $P_{EF} = 2^{-2^{lp}}$. This probability is very small. For instance, in the case (l,p)=(2,2), i.e. we modulate the two LSBs of two pixels, this probability is already about P_{EF} ~10^{-5}.

Similarly, we can also calculate the probability for being able to embed m_{ej} within u tests. This probability is given by $P_{ES}(u) = 1 - (0.5)^u$. As it can be seen in Figure 4, P_{ES} converges rapidly to 1 with the increase of u. Considering again (l,p)=(2,2), the probability to insert m_{ej} within two tests equals 0.75. On the average, one bit of Msg_e will be embedded into a pixel block within two tests. As a consequence, the duration of our process is ad minima two times longer than simply encrypting the image (i.e. without the SHA). We will come back on this issue in section V.C.

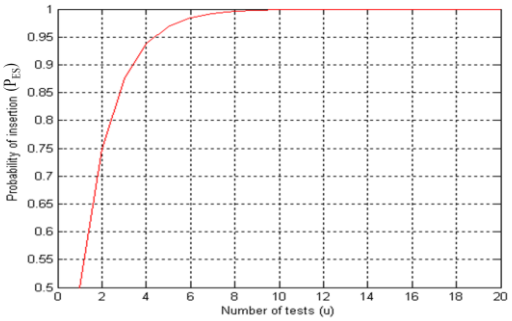


Fig.4: Probability of successful embedding m_{ej} within u tests (P_{ES}).

B. Algorithm

In the encrypted domain, bits of Msg_e will be extracted from the SHA-1 signature of these blocks. With the RC4 algorithm, it is possible to work with smaller pixel block dimension due to the fact it works on stream of bytes (see section II.A.1).

For an image I, whatever the block dimension, our joint watermarking/encryption approach acts in two steps:

1) I is splitted into non-overlapping blocks, $\{X_i\}_{i=1..U}$, of N pixels. In order to form Msg_s (see section III.A), we concatenate the image authenticity code AC with the SHA signature of nmb, which contains the bits of all the pixels that will not be modified by the insertion, i.e. the non selected pixels, as well as of the most significant bits of the selected pixels that will be modulated (see section IV.A). The message available in the encrypted domain Msg_e is also built according to eq. 4 using the secret watermarking key K_w^e.
2) Messages embedding and encryption are then conducted jointly, for each block X_i:
   a. using the sub-codebooks $C_{m_{si}m_{ei}}$, one bit m_{si} of Msg_s, and one bit m_{ei} of Msg_e, are jointly inserted X_i is replaced by X_i^w, which belongs to one cell of $C_{m_{si}}$ and which verifies:
   $$f(X_i^{we}, K_w^e) = m_{ei} \quad (14)$$
   where X_i^{we} represents the encrypted watermarked version of X_i^w.
   b. once X_i^w computed, it is encrypted through the adopted encryption algorithm (i.e. the stream cipher RC4 or the AES in CBC mode).

As stated before, at the verification stage, extraction can be conducted independently in both the encrypted and spatial domains using the corresponding secret watermarking key K_w^e or K_w^s. In the encrypted domain, the encrypted image I_{we} is decomposed in blocks of N bytes. Then, the function f is applied to each block to extract one bit of Msg_e. In the spatial domain, the message Msg_s is extracted based on principles of the QIM. Each message is used by next to verify the image reliability in one domain, it means verifying the authenticity code of the image and its integrity by comparing, in the spatial domain, the extracted SHA signature with the recomputed one and, in the encrypted domain, by checking the equality between the extracted and regenerated random sequences.

Notice that for 12 bit depth or 16 bit depth images, the principles of our algorithm remains the same. Differences stand in the codebooks construction and the pixel block dimensions. As an example, for 16 bit encoded image, we work with 8 pixel blocks instead of 16 pixel blocks; the number of bytes remains the same.

V. Performance Evaluation and Discussion

Experiments were conducted on two sets of medical images: 100 ultrasound images of 576×690 pixels of 8 bit depth, and 200 positron emission tomography (PET) images of 144×144 pixels of 16 bit depth. Some samples of our data set are given in Figure 5. Let us recall that for images encoded on 8 or 16 bits, our joint E/W system manipulates blocks of 16 or 8 pixels respectively (i.e. N=16 or N=8).

A. Image distortion

We decided to use the Peak Signal to Noise Ratio (PSNR) in order to measure the distortion between an image I and its watermarked and deciphered version I_{wd}

$$PSNR(I, I_{wd}) = 10Log_{10}\left(\frac{(2^d-1)^2}{MSE}\right) \quad (15)$$

$$MSE(I, I_{wd}) = \frac{1}{L}\sum_{k=1}^{L}(I(k) - I_{wd}(k))^2$$

where L corresponds to the number of pixels of the image I, and d to its depth. Our choice relies on the fact that the algorithm we proposed in section IV, introduces on the average the same image distortion in each block, thus spreading it over the whole image. Furthermore, it does not take advantage of a psychovisual model which is helpful to adapt the watermark amplitude locally into the image, making at the same time the PSNR not appropriate. Even though there exist some models for natural images, none of them have been proved adapted for medical imaging yet.

If we still consider our implementation, we can determine the lower bound of PSNR depending on: the image depth d, the number of modulated pixels p, the number of LSB modulated per pixel l and the quantization step Δ. Indeed, the maximum distortion one may introduce by modulating l LSBs of one pixel is $\delta = 2^l - 1$. Similarly the maximum distortion induced by the quantization of one pixel is Δ. As a consequence, considering a block B of N pixels and its decrypted-watermarked version B_{wd}, the PSNR lower-bound is given by

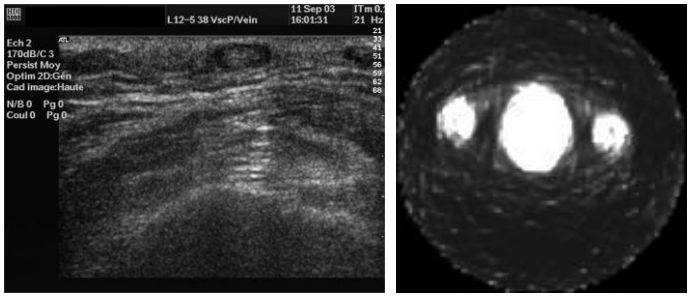
$$PSNR(B, B_{wd}) \geq 10Log_{10}\left(\frac{(2^d-1)^2}{(p\delta^2+\Delta^2)/N}\right) \quad (16)$$

We give in Figure 6 the variation of this limit for different values of p and l considering d=8/N=16 or d=16/N=8, and the smallest possible value of Δ, i.e Δ=1. In these examples, it can be seen that the PSNR limit is quite high for both 8 and 16 bit depth images.

In practice, with the same parameterization and working with the AES in CBC mode or with the RC4, achieved PSNR values are much greater (about 60 dB and 105.26 dB for our ultrasound and PET image test sets, respectively) as indicated in Table I. This can be explained by the fact that we do not have to modify all p pixels in order to make m_{ej} available in the encrypted domain. At least one or two LSB have to be changed. We are far from introducing the maximum distortion.

B. Capacity

Capacity rates depend on the block size N. When the AES is used with our implementation, rates achieved in each domain are both of 1/N bpp. As a consequence, capacities are about of 24,000 and 2,592 bits for ultrasound and PET images respectively. While using the AES limits the block size to some specific values, by working with the RC4 it is possible to consider smaller block size. For instance, if N=4, the capacity rate becomes of 1/4 bpp in each domain. The total amount of bits one can embed is then of 193.5 and 10.125 *K*bits for ultrasound and PET images respectively. But, this increase of capacity is accompanied with a diminution of the PSNR as shown in Table II with the parameterization (l=2, p=2, Δ=1).



(a) (b)

Fig.5: Samples of our image test sets. a) Ultrasound image b) PET image.

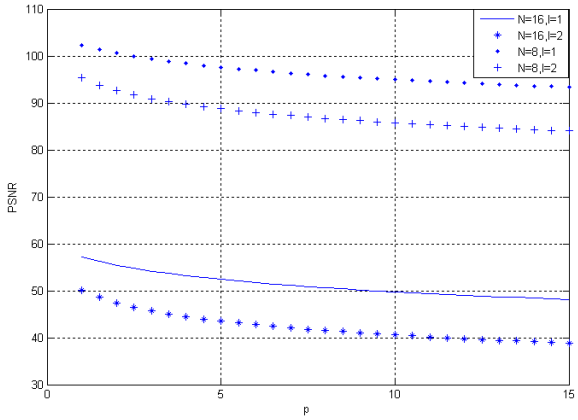


Fig.6: Lower PSNR bound for (l,N,d)=(1,16,8) , (l,N,d)=(2,16,8), (l,N,d)=(1,8,16) or (l,N,d)=(2,8,16) and Δ=1

C. Discussion

Before entering in the details of this discussion, let us recall that the main objective of our joint Encryption/watermarking system is to give access to the outcomes of the integrity and authenticity for the purpose of verifying the image reliability in the spatial domain as well as in the encrypted domain. Considering that a non-reliable image should be rejected by a medical information system, the watermark robustness constraint can be relaxed. As a consequence, messages in the encrypted and spatial domains can be fragile. That is the case with our implementation, messages will be lost after any image modifications. Nevertheless, with our approach an information loss occurs. But, if we refer to the study of Chen *et al.* in [18] about lossy JPEG compression for medical ultrasound images, this loss remains small enough to establish a good diagnosis. Indeed Chen *et al.* report [18] that some loss can be tolerated until the PSNR stays in the range of 40 and 50dB. PSNR values we obtained are much greater (more than 53 dB). There is thus some room to make Msg_s robust to image modifications (e.g. lossy compression, filtering and so on) by increasing the quantization step Δ of the QIM. However, a more complete study has to be conducted in order to evaluate how our watermarks may interfere with image interpretation.

Obtained capacities are large enough compared to the requirements for verifying the reliability of an image that we estimate about 1,000 bits (one digital signature - 160 bits - and one authenticity code - 600 bits, see section III-A and [16]). As a consequence, one can better preserve the image quality by reducing the number of watermarked pixels blocks. To the

contrary, beyond the framework of image reliability control, one can take advantage of this free space and introduce other pieces of information such as: security attributes related to a security policy like those attached to the patient consent [19]; the encryption key K_e in the encrypted domain so as to make its request by the recipient not necessary [13]; or some other medical records of the patient [5]; and so on.

Working with the AES in CBC mode makes our solution transparent or compliant to the DICOM standard. More precisely, if a system is not watermarking interoperable, it will be able to decrypt and access the image if it knows the AES encryption key. On the other hand, the RC4 which manipulates stream of bytes independently, gives us more flexibility allowing us to work on smaller blocks and consequently to embed more data. But, at the same time, reducing the size of blocks increases the complexity of our algorithm. A compromise will have to be established between capacity and computation complexity. Notice that our joint E/W system is on average two times longer than the "original" encryption algorithm (i.e. the RC4 or the AES in CBC mode). This difference is caused by the sub-codebook construction at the protection stage (see section IV). From this statement, our joint E/W system may not be suitable for real time transmission of images. However, its main advantage is that it gives access to a message in both encrypted and spatial domains. Furthermore, time computation for image decryption remains the same.

TABLE I
EXPERIMENTAL RESULTS OBTAINED WITH AES IN CBC MODE OR THE RC4

	Ultrasound		PET	
	Average	Standard deviation	Average	Standard deviation
PSNR(dB)	60.15	0.0189	105.26	0.0695
Entropy of original image (bits/pixel)	6.6664	0.0844	4.3774	0.1733
Entropy of encrypted watermarked image (bits/ pixel)	7.9995	3.58×10^{-5}	7.9897	8.88×10^{-4}
Entropy of encrypted image (bits/pixel)	7.9995	4.92×10^{-5}	7.9893	8.68×10^{-4}

TABLE II
EXPERIMENTAL RESULTS OBTAINED WITH RC4 (N=4)

	Ultrasound		PET	
	Average	Standard deviation	Average	Standard deviation
PSNR(dB)	53.94	0.0107	101.99	0.0461
Entropy of original image (bits/pixel)	6.6664	0.0844	4.3774	0.1733
Entropy of encrypted watermarked image (bits/ pixel)	7.9995	3.51×10^{-5}	7.9898	3.62×10^{-4}
Entropy of encrypted image (bits/pixel)	7.9995	2.7×10^{-5}	7.9898	3.71×10^{-4}

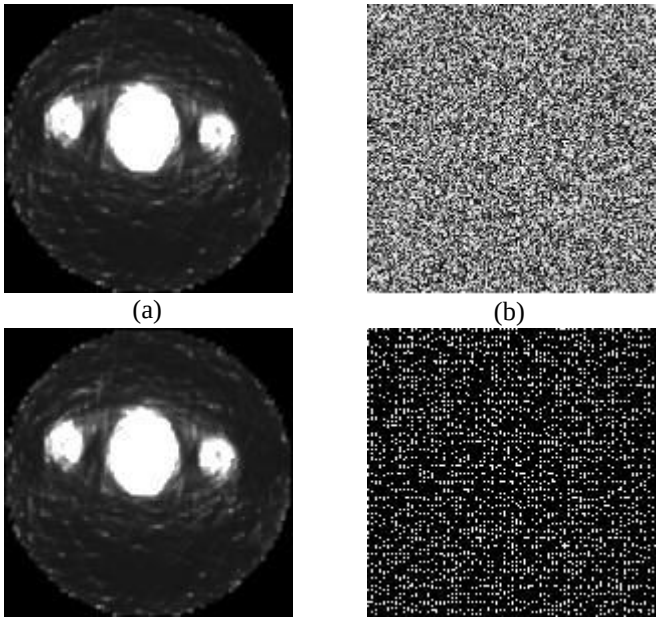


Fig.7: Examples of the images used for evaluation (using AES). a) Original PET image b) joint watermarked/ciphered image c) deciphered watermarked image d) difference between the original image and the decrypted watermarked image.

VI. SECURITY ANALYSIS

The security of our joint E/W system partly relies on the watermarking-encryption relationships we introduced and on the application framework. Let us recall that Msg_e and Msg_s serve the same purpose which is the protection of the reliability of an image. Thus they contain some common pieces of information. In this section, we first start by looking at cryptographic attacks, which aim is to break confidentiality, before focusing on watermarking attacks.

A) Cryptographic attacks

In our joint E/W system, we work with popular encryption algorithms (the AES and the RC4) which security performance are well known. Due to the fact we do not intrinsically modify them, without the knowledge about the watermarking keys, their performance are preserved against common cryptographic attacks like the ones based on ciphertext-only, known plaintext, chosen plaintext, or/and on chosen ciphertext attacks.

If K_w^e or Msg_e are known from the attacker, he has no other additional means than a regular cryptographic attack to get K_e or to have an idea about the clear watermarked image (i.e. I_w). This is due to the fact Msg_e is embedded within the SHA signatures of watermarked-encrypted blocks and not directly into the encrypted bit stream. Furthermore, Msg_e appears "encrypted" in I_{we} and its presence does not reduce the entropy of the watermarked-encrypted image as compared with the simple encryption of the image in tables I and II.

If K_w^s and Δ are known, codebooks $C_{m_{si}}$ can be computed but the sub-codebooks $C_{m_{si}m_{ej}}$ cannot be derived even if I_w is known. The attacker has no clues about K_e. If now, he knows also K_w^e or/and Msg_e, we retrieve the cryptographic attack based on known plaintext and known ciphertext. Nevertheless,

if the attacker complete this set of data with the original image *I*, he can get an idea about to the sub-codebooks and consequently find the encryption key K_e.

B) Watermark attacks

In [20], Zhou et *al.* have defined three types of watermark attacks we analyze in this sub-section: unauthorized message embedding, unauthorized message detection/extraction and unauthorized watermark removal.

1) Unauthorized detection/extraction of messages

Let us start with the encrypted domain. The location of the bits of Msg_e within the SHA signatures of the watermarked-encrypted blocks depends on K_w^e. Without this key, an attacker cannot distinguish the bits of Msg_e from the others. Even though the structure of Msg_e is known (it is the concatenation of a pseudo random sequence with an authenticity code *AC* - see section III.A), an attacker can only try an exhaustive search until he finds a valid *AC*, which does not mean this code is the one of the image. Moreover, in the case the attacker knows K_w^s, he will only have a partial knowledge about Msg_e (Msg_s and Msg_e contain the image authentication code). If K_w^s gives some knowledge about the sub-codebooks, without K_w^e he cannot access Msg_e (nor modify it - see below). If at the same time the encryption K_e becomes available, even though the sub-codebooks can be reconstructed, it remains not possible to find exactly the locations of the bits of Msg_e.

The security of Msg_s lies on the QIM which efficiency has been recently studied [21-22]. In our system, the knowledge of the encryption key K_e, has no impact on the QIM security due to the fact it does not contribute to the construction of $C_{m_{si}}$. However, if we consider our application framework, the knowledge of K_w^e or equivalently of Msg_e allows a clear-text attack in the spatial domains as it gives some partial knowledge about Msg_s to one attacker.

2) Unauthorized embedding attack

The insertion of a message available in the encrypted domain depends on K_e and K_w^e (see eq. 8). One can try to modify arbitrarily the watermarked-encrypted bit stream in order to introduce a falsified message Msg_e' but this will be detected at the decryption stage; it will not be possible to reconstruct the image. At the same time, the watermarked-encrypted image will be claimed as unreliable due to the fact that Msg_e depends on K_w^e (see section III). If now I_w and K_w^s are available, one attacker may insert a new message in the encrypted domain but it will be difficult for him to make the message compliant with K_w^e. This is almost the case if at the same time K_e is known. One can completely reconstruct the codebooks, but he does not which bits to focus on in the SHA signatures of the watermarked-encrypted blocks. This information is given by K_w^e.

In the spatial domain, a pirate will have to break the QIM. The knowledge of K_w^e or of K_e gives him no advantage. To the contrary, it will be possible to detect that a pirate has modified Msg_s based on Msg_e, thus at the condition K_w^e is unknown from him.

3) Unauthorized removal attack

In the framework we consider, an image is rejected if it is not reliable, i.e. if it is not possible to verify its integrity and its origins. As a consequence, valid watermarks should be present in all images and in their encrypted versions. If someone changes embedded messages, he should also provide a new set of encryption and watermarking keys.

VII. Conclusion

In this paper, we have proposed a new joint watermarking/encryption system, which guarantees an *a priori* and *a posteriori* protection of medical images. It merges the QIM and a cipher algorithm or a block cipher algorithm. Our system gives access to two distinct messages in the spatial domain and in the encrypted domain, respectively. These two messages are used for verifying the image reliability even though it is encrypted. The AES in CBC mode makes our system compliant with the DICOM standard. Experimental results show that the image distortion is very low and that the achieved capacity is enough to embed a reliability proof as well as some other data. Obviously, our joint watermarking/encryption system is slower than simply encrypting the image but it provides reliability control functionalities. On the other hand, the execution time for image decryption is not impacted. We have also shown that the way we combine encryption and watermarking does not interfere with the security of the encryption algorithm and that the security of our system depends on the knowledge of the encryption and watermarking keys. Future works will focus on making our scheme more robust to attacks like lossy image compression (ex. JPEG) and reducing the complexity of our algorithm.

Acknowledgment

The work has been supported by the Agence Nationale de la Recherche (ANR) through the ANR ARPEGE SELKIS project.

References

[1] G. Coatrieux, H. Maître, B. Sankur, Y. Rolland, R. Collorec, "Relevance of watermarking in medical imaging," *in proc. of Int. Conf. on IEEE EMBS* ITAB, USA, pp. 250-255, 2000.

[2] G. Coatrieux, C. Le Guillou, J.-M. Cauvin, C. Roux, "Reversible watermarking for knowledge digest embedding and reliability control in medical images," IEEE Trans. Inf. Technol. Biomed., 2009 Mar., 13(2):158-165.

[3] U. Rajendra Acharya, D. Acharya, P. Subbanna Bhat, and U.C. Niranjan, "Compact Storage of Medical Images with Patient Information," IEEE Trans. on Inf. Tech. in Biomed. Dec. 2001, vol. 5, n°. 4, pp. 320-323.

[4] W. Pan, G. Coatrieux, N. Cuppens-Boulahia, F. Cuppens, and C. Roux, "Medical Image Integrity Control Combining Digital Signature and Lossless Watermarking," Data Privacy management and autonomous spontaneous security, LNCS, 2010, vol. 5939/2010, pp. 153-162.

[5] U. Rajendra Acharya, U.C. Niranjan, S.S. Iyengar, N. Kannathal, and Lim Choo Min, "Simultaneous storage of patient information with medical images in the frequency domain," Computer Methods and Programs in Biomedicine (2004) 76, 13-19.

[6] J.M. Rodrigues, W. Puech, C. Fiorio, "Lossless crypto-data hiding in medical images without increasing the original image size," in Proc. of

the 2nd International Conference on Advances in Medical Signal and Information Processing, Sliema, Maltes, Sep. 2004, pp. 358–365.

[7] R. Anderson, C. Manifavas, "Chameleon- A New Kind of Stream Cipher," in Proc. of the 4th International Workshop on Fast Software Encryption (FSE '97), Haifa, Israel, Jan. 1997, vol. 1267, pp. 107–113.

[8] A. Adelsbach, U. Huber, A.S. Sadeghi, "Fingercasting–Joint Fingerprinting and Decryption of Broadcast Messags," in Proc. of the 11th Australasian Conference on Information Security and Privacy (ACISP '06), Melbourne, Australia, Jul. 2006, vol. 4058 of Lecture Notes in Computer Science, pp. 136–147.

[9] M. Celik, A.N. Lemma, S. Katzenbeisser, M. van der Veen, "Secure Embedding of Spread Spectrum Watermarks Using Look-up-Tables," in Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP '07), IEEE Press, Hawaii, USA, Apr. 2007, vol. 2, pp. 153-156.

[10] L. Shiguo, L. Zhongxuan, R. Zhen, W. Haila, "Joint fingerprint embedding and decryption for video distribution," in *Proc. of IEEE International Conference on Multimedia and Expo, Jul. 2007, pp.1523-1526.*

[11] L. Shiguo, L. Zhongxuan, R. Zhen, W. Haila, "Commutative encryption and watermarking in video compression," IEEE Transactions on Circuits and Systems for Video Technology, Jun. 2007, vol. 17, n°. 6, pp.774-778.

[12] B. Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C," Paris: International Thomson Publishing, 1997.

[13] W. Puech, J.M. Rodrigues, "A new crypto-watermarking method for medical images safe transfer," in Proc. of the 12th European Signal Processing Conference (EUSIPCO'04),Vienna, Austria, Sep 2004, pp. 1481-1484.

[14] J. Daemen, V. Rijmen, AES Proposal: The Rijndael Block Cipher. Technical report, Proton World Int.1, Katholieke Universiteit Leuven, ESAT-COSIC, Belgique, 2002.

[15] B. Chen, G.W. Wornell, "Quantization Index Modulation: A Class of Provably Good Methods for Digital watermarking and information embedding," IEEE Trans. on Information Theory, May 2001, vol. 47, n°. 4, pp. 1423- 1443.

[16] G. Coatrieux, C. Quantin, J. Montagner, M. Fassa, F-.A. Allaërt, Ch. Roux, "Watermarking medical images with anonymous patient identification to verify authenticity," Studies in health technology and informatics, 2008, vol. 136, pp. 667-672.

[17] W. Pan, G. Coatrieux, N. Cuppens-Boulahia, F. Cuppens, C. Roux, "Reversible Watermarking Based on Invariant Image Classification and Dynamical Error Histogram Shifting," IEEE EMBS (2011), Boston, USA, pp. 4477-4480.

[18] K. Chen, T.V. Ramabadran, "Near-lossless compression of medical images through entropy coded DPCM," IEEE Transactions on Medical Imaging, 1994, vol. 13, n°. 3, pp. 538-548.

[19] W. Pan, G. Coatrieux, N. Cuppens-Boulahia, F. Cuppens, C. Roux, " Watermarking to enforce medical image access and usage control policy," SITIS 2010, Kuala Lampur, Malaysia, Dec. 2010, pp. 251-260.

[20] X. Zhou, W. Zhao, Z. Wang, and L. Pan, "Security theory and attack analysis for text watermarking", in Proceedings of International Conference on E-Business and Information System Security,(EBISS 2009), May 2009, pp. 1-6

[21] L. Perez-Freire, F. Perez-Gonzalez, "Security of Lattice-Based Data Hiding Against the Known Message Attack", IEEE Transactions on Information Forensics and Security, Dec. 2006, vol. 1, n°4, pp. 421– 439.

[22] S. Braci, R. Boyer, C. Delpha, "Security evaluation of informed watermarking schemes", 16th IEEE International Conference on Image Processing (ICIP), Nov. 2009, pp. 117 - 120