On-Board Format-Independent Security of Functional Magnetic Resonance Images

ARCANGELO CASTIGLIONE, RAFFAELE PIZZOLANTE, FRANCESCO PALMIERI, BARBARA MASUCCI, BRUNO CARPENTIERI, ALFREDO DE SANTIS, and ANIELLO CASTIGLIONE, Università degli Studi di Salerno

Functional magnetic resonance imaging (fMRI) provides an effective and noninvasive tool for researchers to understand cerebral functions and correlate them with brain activities. In addition, with the ever-increasing diffusion of the Internet, such images may be exchanged in several ways, allowing new research and medical services. On the other hand, ensuring the security of exchanged fMRI data becomes a main concern due to their special characteristics arising from strict ethics and legislative and diagnostic implications. Again, the risks increase when dealing with open environments like the Internet. For this reason, security mechanisms that ensure protection of such data are strongly required. However, we remark that the mechanisms commonly employed for data protection are doomed to fail when dealing with imaging data. In this article, we propose a novel watermarking scheme explicitly addressed for this type of imaging. Such a scheme can be used for several purposes, particularly to ensure authenticity and integrity. Moreover, we show how to integrate our scheme within commercial off-the-shelf fMRI system. Finally, the validity and the efficiency of our scheme has been assessed through testing.

1. INTRODUCTION

Research concerning brain functional activities is experiencing an ever-increasing interest in the area of neuroimaging. Functional magnetic resonance imaging (fMRI) provides an effective and noninvasive tool for researchers to understand cerebral functions and correlate them with brain activities. The fMRI-based research shares many features with the clinical practice, and hence even if an fMRI analysis may have a merely research-oriented goal, some of the data collected during such analysis might have

This work was partially supported by the Italian Ministry of Research within PRIN project GenData 2020 (2010RTFWBH).
Authors' address: A. Castiglione, R. Pizzolante, F. Palmieri, B. Masucci, B. Carpentieri, A. De Santis, and A. Castiglione, Dipartimento di Informatica, Università degli Studi di Salerno, Fisciano, Salerno, 84084, Italy; emails: {arcastiglione, rpizzolante, fpalmieri, bmasucci, bcarpentieri, ads}@unisa.it, castiglione@acm.org.

medical interest. Thus, the security of such images, as for any medical and medical-related information, is conditioned by strict ethics and legislative rules. In addition, such images may be used as evidence in a court of law. Indeed, although digital forensic techniques have been proposed to detect various traces of tampering, relatively slight modifications either cannot be detected by these techniques or make them ineffective. Therefore, it is critical to verify the integrity of such evidence.

Moreover, even if advances in communication technology have provided new ways to store, access, and distribute data in a digital format, such advances have introduced new threats, given the simplicity by which digital content can be manipulated. Therefore, it is often necessary to detect whether a digital image has been altered somehow from when it was recorded, and it is crucial to ensure the authenticity of the image. In general, an approach to protect fMRI images may be the inclusion of some other information into the image header. However, such an approach is prone to attacks, such as manipulation and tampering. In addition, information loss might occur during file format conversions [Coatrieux et al. 2009]. Finally, even if encryption can be used to protect data transmitted over insecure networks, decrypted content may be affected by unauthorized use or manipulation at the receiver's side [Boucherkha and Benmohamed 2004]. To overcome the limitations mentioned previously, it is necessary to introduce a protection level that is as near as possible to the data. Digital watermarking is a well-established technique to ensure this type of protection. More precisely, digital watermarking techniques can be used for multiple purposes. In particular, they are commonly employed to provide image integrity and authenticity [Castiglione et al. 2015a; Pizzolante et al. 2013; Pizzolante and Carpentieri 2012; Pizzolante et al. 2011]. However, watermarking techniques generally introduce a certain level of alteration to or information loss in the original image, and this cannot be tolerated when dealing with sensitive images, as in the case of fMRI. For this reason, reversible watermarking techniques have been introduced, which enable the exact recovery of the original image after the extraction of the embedded information without any alteration or information loss.

In this article, we first propose a novel fragile reversible watermarking scheme for images characterizing fMRI-based analysis, which relies on those proposed in Castiglione et al. [2015b] and Coatrieux et al. [2009]. The proposed scheme, coupled with cryptographic primitives such as digital signatures and hash functions, can be used effectively to ensure authenticity and integrity for fMRI images without using any external metadata (e.g., headers and attributes) and in a manner that is independent of the image format. In addition, the scheme that we propose may be used to include data that needs to be sent from one endpoint to another, such as information concerning a patient. Moreover, we remark that given its structure, our scheme could be modeled by means of an open-loop microcontroller [Kuo and Golnaraghi 2002], and thus it is well suited to be implemented directly on board. Furthermore, in this work, we show how to integrate the proposed scheme within fMRI systems and particularly how such a scheme can operate coupled with the DICOM standard. Again, considering that in the future the DICOM standard might be modified or other standards could be introduced, we show how the proposed scheme can be used in a manner that is independent of such a standard. Finally, to assess the performance and the effectiveness of our scheme when the hardware and software characteristics at its disposal are extremely constrained, we implemented and evaluated such a scheme on a credit card–size single-board computer.

This article is organized as follows. In Section 2, we provide some basic concepts underlying the DICOM standard. In Section 3, we describe the proposed watermark scheme and how it can be used for the protection of fMRI images. In Section 4, we show the integration of our scheme within an fMRI system. In Section 5, we show the experimental results achieved evaluating our scheme. Finally, in Section 6, we draw conclusions and offer future research directions.

2. BACKGROUND AND PRELIMINARIES

DICOM is a software integration standard mainly used in medical and medical-related imaging. Almost all modern medical imaging systems, referred to as imaging modalities (e.g., x-ray, ultrasound, computed tomography, magnetic resonance imaging), support DICOM and use it extensively. Informally speaking, all medical and medical-related images are saved in DICOM format, and hence medical imaging equipment creates *DICOM objects*. Each of these objects holds patient information (e.g., name, ID number, gender, and date of birth), important acquisition data (e.g., the equipment used and its settings), and the context of the imaging study that has been used to bind the image to the medical treatment of which it was part. More precisely, in the DICOM model, a patient can have one or more *studies*, sometimes referred to as exams or procedures. Each study consists of one or more *series*. A series generally corresponds either to a specific type of data (*modality*) or to the position of a patient on the acquisition device. Each series contains one or more *DICOM object instances*, which most commonly are images, but they also report waveform objects and the like. Finally, the preceding information is contained in each DICOM object relative to a study. In detail, data such as patients, studies, and medical devices are viewed by DICOM as objects with their relative properties and attributes. All DICOM objects are made by *data elements*, which represent the smallest building blocks that can be grouped to build more meaningful and complex DICOM objects. DICOM denotes those hierarchically related blocks as information modules, information entities (IEs), and information object definitions (IODs). Modules form IEs, which in turn are used to build IODs. Information modules provide the first and most essential level of data element organization, gathering related data attributes (elements) in a consistent and structured manner. DICOM IEs are built from DICOM information modules. In particular, for each IE, DICOM lists the modules that such IE should include. Finally, when combined meaningfully, IEs build IODs, which are the objects used by DICOM [Pianykh 2009]. The data encoded according to DICOM standard can be transmitted and processed by DICOM devices and software, known as application entities (AEs). The AEs provide services to each other, which usually involve some data exchange over a computer network. Therefore, it becomes natural to associate particular service types to the data (IODs) that they process. Those associations are denoted as service-object pairs (SOPs) and are grouped into SOP classes. The building blocks of DICOM are a *file format* and a *networking protocol*. In particular, DICOM applications can collaborate in two ways: they can communicate over a TCP/IP network or exchange files over some physical media. Medical and medical-related information represent the application data, as shown in Figure 1.

More precisely, an AE is a DICOM implementation that organizes data into DICOM objects properly encoded. Then, such objects can be shared or sent to other DICOM applications. Finally, as shown in Figure 1, DICOM objects can be either written into DICOM files through some physical media (e.g., CD, DVD, and USB) or sent over a TCP/IP network by using DICOM commands.

3. THE FRAGILE REVERSIBLE WATERMARKING SCHEME

Our fragile reversible watermarking scheme is based on those proposed in Coatrieux et al. [2009] and Castiglione et al. [2015b]. In particular, the proposed scheme belongs to the category of *additive schemes*, where the watermark string is added directly to the image signal (i.e., to its samples). By doing this, the watermarked signal contains the original image signal, as well as the one affected by both the watermark string and secret key. The main idea underlying our scheme is to embed each bit of the watermark string into a specific block, constituted by 2×2 samples of the image. In detail, since each sample value belongs to a finite set of values (i.e., $\{0, 1, \ldots, 2^{16} - 1\}$), it is important

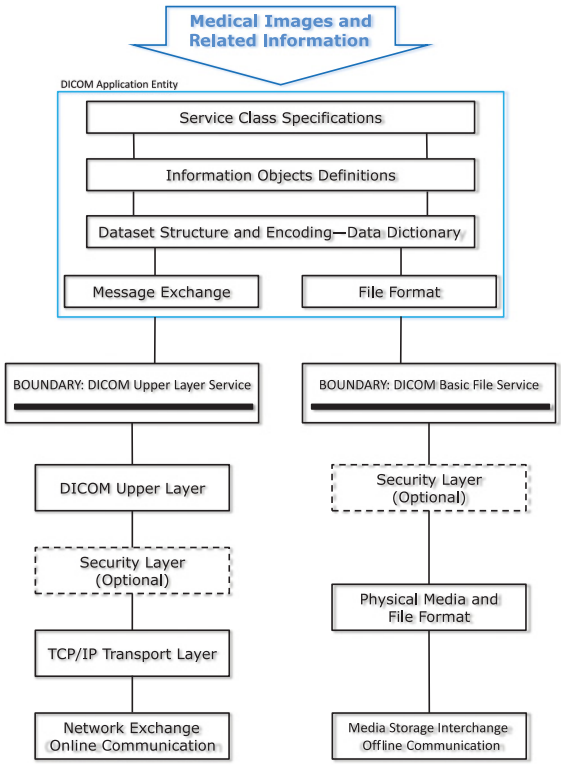


Fig. 1. DICOM general communication model [National Electrical Manufacturers Association 2015].

to add the watermark signal while avoiding underflows and overflows. For this reason, we define a *valid block* as a block that does not cause underflows/overflows, even when a single bit of the watermark string is embedded. We remark that even if, virtually, each valid block allows to embed a bit, some of them cannot be used for this purpose in practice. In fact, for some of those blocks, the extraction algorithm would not be able to extract the hidden information. Therefore, valid blocks should be further classified into *carrier blocks* and *noncarrier blocks*, where the former are able to embed a bit, whereas the latter are not. More precisely, in each carrier block, a bit can be embedded by adding or subtracting, according to the value of this bit, the watermark pattern signal W, defined by the following equation:

$$W = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

A valid block B can be detected by performing a specific estimation. Let B^e define such an estimation, where each sample is obtained through the linear combination of some samples of B. In detail, by verifying the relation between B and B^e, it is possible to decide whether B is a carrier or noncarrier block. We note that the extraction algorithm is able to classify, in two phases, the valid blocks. Furthermore, given the reversibility of our scheme, it is possible to exactly recover the original block B from the watermarked one B^W.

In the following, we describe the logical functioning of the proposed scheme by highlighting the relative embedding, extraction, and recovery phases. In particular, in Section 3.1, we describe the embedding phase of our scheme, and in Section 3.2, we

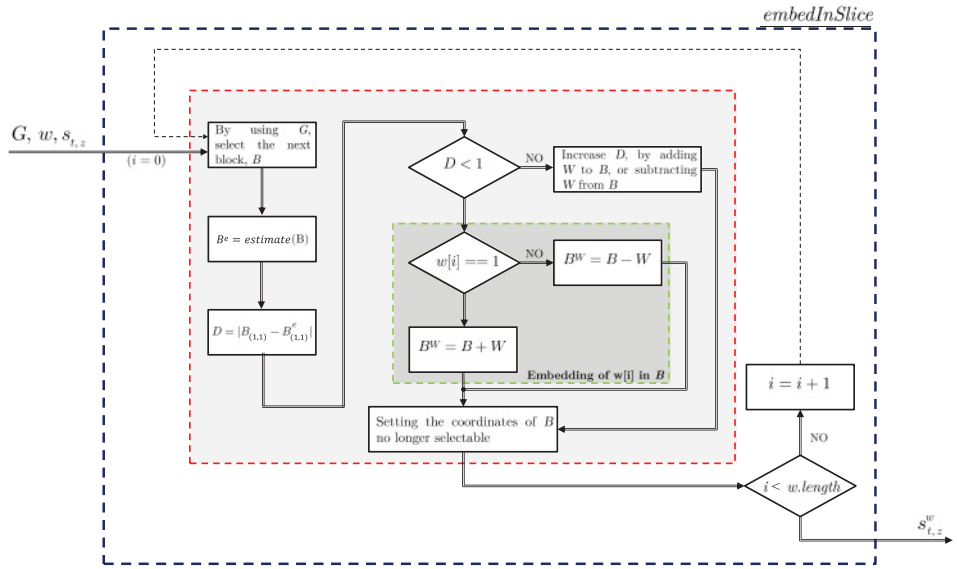


Fig. 2. An open-loop like representation of the embedding phase of the proposed watermark scheme.

highlight the key points of the extraction and recovery phases. Without loss of generality, from now on we consider all selected blocks as valid. However, we remark that valid blocks can be efficiently detected by both embedding and extraction algorithms [Coatrieux et al. 2009; Castiglione et al. 2015b].

3.1. The Embedding Phase

The embedding phase is carried out by the *embedRWInFMRI* procedure, which is outlined in Algorithm 1. Such a procedure embeds the watermark string ws in the input image *fMRI* by dividing that image into M subimages on the T-axis. Subsequently, ws is embedded independently in each subimage $fMRI_i$, with $i \in \{1, \ldots, M\}$, through the *embedRWInSubImage* procedure, which is reported in Algorithm 2. In this way, the extraction of the hidden information can be performed even when only a significant portion of the fMRI image is available. In particular, such a portion should contain at least a subimage. Finally, once ws is embedded in all of the subimages, they are merged to obtain the final watermarked image $fMRI^W$.

The *embedRWInFMRI* procedure employs a pseudorandom number generator (PRNG) G, whose *seed* is the secret used for the embedding and the extraction. Similarly to Pizzolante and Carpentieri [2012], such a procedure relies on G to select the blocks where the watermark string will be embedded. Again, with regard to the *embedRWInSubImage* procedure, the watermark string ws is split into P substrings, where P denotes the number of slices constituting $fMRI_i$. In detail, for each slice $s_{t,z}$ of $fMRI_i$, the *embedInSlice* procedure described in Algorithm 3 is invoked. Let ws_p be a substring of ws. The *embedInSlice* procedure, which is graphically explained by the open-loop like diagram in Figure 2, is used for the embedding of ws_p in $s_{t,z}$. Without loss of generality, in the figure, a substring ws_p is referred to as w. First of all, a block B is extracted from $s_{t,z}$ by using G, and B is estimated through the *estimate* procedure. Let B^e denote the output of such estimation. Subsequently, the relation between B and B^e is used to check whether B is a carrier or noncarrier block. More precisely,

ALGORITHM 1: *embedRWInFMRI*

Input: $fMRI$, ws, $seed$, M.
Output: $fMRI^W$.
Define a PRNG G and the relative $seed$
Subdivide $fMRI$ on the T-axis into $fMRI_1, fMRI_2, \ldots, fMRI_M$
for $i = 1$ *to* M **do**
    $fMRI_i^W = \underline{embedRWInSubImage}(fMRI_i, G, ws)$
end
Merge $fMRI_1^W, fMRI_2^W, \ldots, fMRI_M^W$ into $fMRI^W$
return $fMRI^W$

ALGORITHM 2: *embedRWInSubImage*

Input: $fMRI_i$, G, ws.
Output: $fMRI_i^W$.
$p = 1$
$P = fMRI_i.numOfVolumes \times fMRI_i.numOfSlicesPerVolume$
Subdivide ws into $ws_1, ws_2, \ldots, ws_P$
for $t = 1$ *to* $fMRI_i.numOfVolumes$ **do**
    for $z = 1$ *to* $fMRI_i.numOfSlicesPerVolume$ **do**
        $\underline{embedInSlice}(w_p, s_{t,z}, G)$
        $p++$
    end
end
Copy all modified slices into $fMRI_i^W$
return $fMRI_i^W$

the distance D between $B_{(1,1)}$ and $B^e_{(1,1)}$ is considered, where $B_{(i,j)}$ is the value of the sample having (i, j) as relative coordinates. If B is a carrier block, based on the value of $w[i]$, W is added to B or subtracted from B, and the relative result is stored in B^W, as highlighted by the green dotted line in Figure 2. Note that if the embedded bit is equal to 1, the relation $B^W_{(1,1)} > B^e_{(1,1)}$ is satisfied, whereas if such a bit is equal to 0, the relation $B^W_{(1,1)} < B^e_{(1,1)}$ holds. We remark that the aforementioned relations are exploited by the extraction algorithm, as it holds that $B^e = estimate(B) = estimate(B^W) = B^{eW}$. Conversely, if B is a noncarrier block, the preceding relations cannot be satisfied, and W is added to B or subtracted from B, to increase the value of D. Afterward, all modified blocks and unmodified samples of $s_{t,z}$ are stored in $s^W_{t,z}$, which is returned as output by the *embedInSlice* procedure.

Again, once all slices of $fMRI_i$ have been processed, all modified blocks and unmodified samples of $fMRI_i$ are then stored in $fMRI_i^W$, which is returned by the *embedRWInSubImage* procedure. Finally, the whole watermarked image $fMRI^W$ is returned by the *embedRWInFMRI* procedure.

3.2. The Extraction and Recovery Phases

The extraction and recovery phases are performed by means of the *extractAnd RecoverRWFromFMRI* procedure, which is reported in Algorithm 4. We remark that

ALGORITHM 3: *embedInSlice*

Input: G, w, $s_{t,z}$.
Output: $s^W_{t,z}$.
do
  Pseudorandomly select (x, y) by using G
  Verify the validity of (x, y)
  $B = getBlock(s_{t,z}, x, y)$
  $B^e = estimate(B)$
  $D = |B_{(1,1)} - B^e_{(1,1)}|$
  if $(D < 1)$ **then**
    if *(w[i]==1)* **then**
      $B^W = B + W$
    else
      $B^W = B - W$
    end
    i++
  else
    Add W to B or subtract W from B to increase the difference between $B_{(1,1)}$ and $B^e_{(1,1)}$
  end
  Set all coordinates of B to be not valid
while $i \leq w.length$
Copy all modified blocks and unmodified samples of $s_{t,z}$ to $s^W_{t,z}$
return $s^W_{t,z}$

the purpose of this procedure is twofold: extraction of the watermark string and the recovery of the original fMRI image. More precisely, such a procedure first splits the watermarked input image $fMRI^W$ into M subimages, each denoted by $fMRI^W_i$, with $i \in \{1, \ldots, M\}$. For each $fMRI^W_i$, the watermark string is extracted and the original subimage is recovered through the *extractAndRecoverRWFromSubImage* procedure. In detail, each invocation of such a procedure takes as input $fMRI^W_i$ and returns as output the pair $(ws^E(i), fMRI^R_i)$. Finally, once all watermark strings have been extracted and the subimages recovered, the *extractAndRecoverRWFromFMRI* procedure returns one of the following outputs:

—$(ws^E(1), fMRI^R)$, if all watermark strings extracted from the subimages are equal;
—$(nil, fMRI^W)$, otherwise.

More precisely, in the first case, the recovered fMRI image $fMRI^R$ and the extracted watermark string $ws^E(1)$ are returned, whereas in the second case, *nil* and the watermarked input $fMRI^W$ are returned. Note that in the second case, the watermark strings extracted from the subimages are different, and as a consequence, $fMRI^W$ is affected by one or more alterations.

4. ON-BOARD INTEGRATION

In this section, we describe how to integrate the proposed scheme within an fMRI system. The scheme that we propose operates in real time and is able to deal with large amounts of data. In the literature, other systems have been proposed for the real-time processing of large-size images, as in the case of satellite imagery [Muresan et al. 2006;

ALGORITHM 4: *extractAndRecoverRWFromFMRI*

Input: $fMRI^W$, $lengthWS$, $seed$, M.
Output: $(nil, fMRI^W)$ or $(nil, fMRI^W)$.
Define a PRNG G and the relative $seed$
Subdivide $fMRI^W$ on the T-axis into $fMRI_1^W, fMRI_2^W, \ldots, fMRI_M^W$
for $i = 1$ *to* M **do**
    $(ws^E(i), fMRI_i^R) =$ extractAndRecoverRWFromSubImage$(fMRI_i^W, G, ws)$
end
if $(ws^E(i) = ws^E(j), \forall i, j \in \{1, \ldots, M\})$ **then**
    Merge $fMRI_1^R, fMRI_2^R, \ldots, fMRI_M^R$ into $fMRI^R$
    return $(ws^E(1), fMRI^R)$
else
    return $(nil, fMRI^W)$
end

Pop et al. 2007; Petcu et al. 2007]. In particular, we show how our scheme can operate coupled with the DICOM standard, which defines the criteria for communication, visualization, archiving, and printing of medical and medical-related information. It is important to point out that DICOM is supported by almost all fMRI system models, produced by manufacturers such as Philips (e.g., Ingenia and Achieva), Siemens (e.g., Magnetom), and General Electric (e.g., Signa and Optima). Finally, we show how our scheme can be used effectively in a manner that is independent of the DICOM standard.

4.1. Integration with DICOM AEs

All fMRI imaging systems produce as output images conforming to the DICOM standard. More precisely, by means of the MR Image Storage SOP Class (1.2.840.10008.5.1.4.1.1.4) defined by the DICOM standard, the images created by an fMRI system are encoded as DICOM (SOP) objects, containing all information related to such images. Some fMRI systems also support derivations of that class, such as the Enhanced MR Color Image Storage SOP Class (1.2.840.10008.5.1.4.1.1.4.3). Medical and medical-related images are usually stored in uncompressed format; however, compressed formats, such as JPEG and run- length encoding, are supported. The SOP object concerning a certain image can be transmitted on the network for being displayed or stored on a PACS system. Additionally, it can be printed or stored on media devices such as USB, CD, and DVD.

It is important to emphasize that although the watermarked image produced by our scheme can be restored to its original form, it no longer complies with the MR Image Storage SOP Class. Indeed, an image being stored using that class must not be affected by any form of postprocessing. On the other hand, with regard to images affected by postprocessing, as well as those not acquired from modalities, DICOM provides the Secondary Capture Image Storage Class (1.2.840.10008.5.1.4.1.1.7) and its derivations, such as the Multi-Frame True Color Secondary Capture Image Storage Class (1.2.840.10008.5.1.4.1.1.7.4).

Based on the aforementioned considerations, for each fMRI image acquired, a watermarked version of the image is produced. More precisely, the image acquired from the modality is first encoded as a DICOM object through the MR Image Storage SOP Class. Afterward, our watermark scheme is applied on the image, which is encoded through the Secondary Capture Image Storage Class to produce a DICOM object containing the

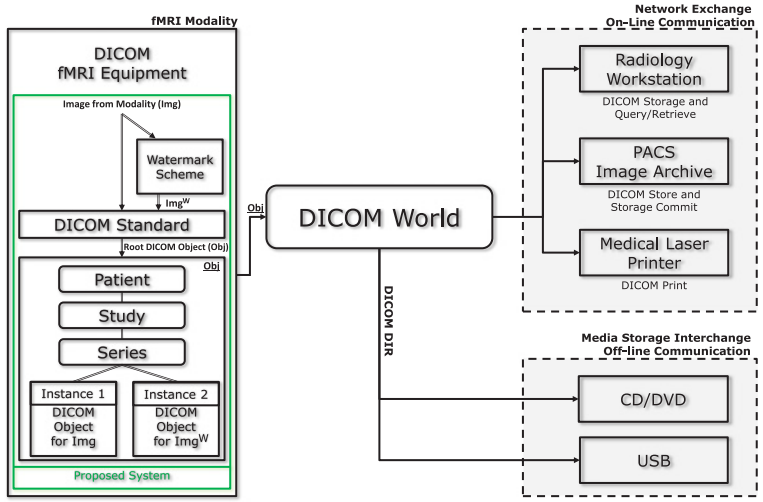


Fig. 3. Application logic of the proposed scheme within DICOM AEs.

watermarked image. In detail, the DICOM objects characterizing the watermarked and unwatermarked images have the same values for the attributes of the following DICOM modules: Patient (C.7.1.1), General Study (C.7.2.1), and General Series (C.7.3.1). In this way, using the recursion (or nesting) of DICOM objects, we create a more complex tree-like structure, which is in turn embedded in a root DICOM object.

From now on, the objects created as described earlier can follow the conventional dataflow defined by the DICOM standard. More precisely, as described by the DICOM data model shown in Figure 1, the aforementioned objects (SOP instances) can follow two paths: they can be stored offline or on a device, or they can be transmitted over the network to be analyzed, stored, or printed. Moreover, any DICOM viewer can display the watermarked image. We remark that the extraction of the watermark, as well as any eventual security check, can be easily integrated within such a viewer. In Figure 3, we show the functioning of our scheme within a typical DICOM environment.

4.2. Integration Outside the DICOM World

Considering that new standards for the management of medical and medical-related images might be introduced and changes to the DICOM standard could be carried out in the future, we also define how the proposed scheme can operate in medical and medical-related applications regardless of the DICOM standard.

However, it is important to emphasize that for diagnostic or research purposes, the presence of the invisible watermark should always be explicitly signaled. In general, in medical and medical-related fields, it is common practice to perform research on images acquired directly from modalities that have not been affected by any form of postprocessing (e.g., watermark embedding). Moreover, such signaling should always be preserved, even in the case of image format conversion.

For this purpose, we report the presence of the hidden information by means of a second watermark that is logically related to the first one but has a completely different function. In detail, the visible watermark is reversible and should not be removed until the invisible one has been completely extracted. In fact, the invisible watermark may not be extracted completely even though the visible one has been entirely removed. Therefore, the first watermark remains unreported, and this could mislead the analysis.

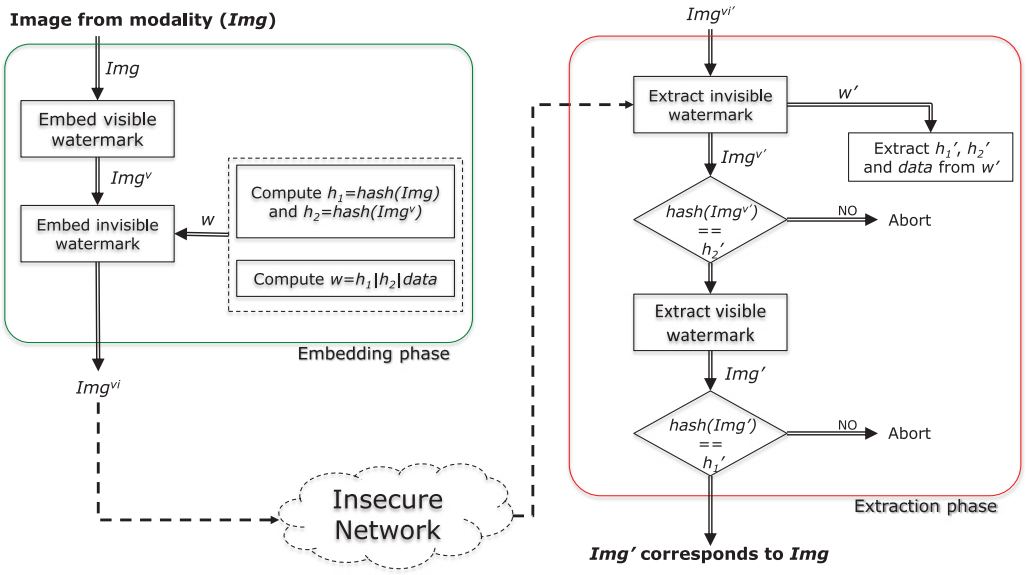


Fig. 4. Application logic of the proposed scheme outside the DICOM world.

A possible way to create the aforementioned dependence between the visible and invisible watermarks is graphically described in Figure 4. In the figure, the "|" symbol denotes the concatenation of strings, whereas the variable *data* denotes some information that may be used for subsequent computation.

Finally, we remark that the two watermarks, although dependent on each other, have completely different functions. More precisely, the visible watermark is used to provide a human-readable warning, whereas the invisible one is used to hold information necessary for an eventual subsequent processing.

5. IMPLEMENTATION AND PERFORMANCE EVALUATION

The main aim of the testing phase has been to evaluate two aspects: the imperceptibility of the proposed watermark scheme (i.e., the embedded watermark should be invisible and the execution time of our scheme on a credit card–size single-board computer. For evaluating the first aspect, the following two metrics were used: peak signal-to-noise ratio (PSNR) and the Q Index (QI) [Wang and Bovik 2002]. The PSNR is a widely used measure of similarity between the original image and the watermarked one. Such a measure is easy to compute and analytically tractable. However, it is widely known that the PSNR does not consider human visual sensitivities [Wang et al. 2002]. Consequently, to better evaluate the image quality through objective measures, the QI has been considered. The QI ranges from -1 to 1. In particular, its best value is 1, and it is achieved if and only if the compared images are exactly the same, whereas the worst value is -1, and it is achieved when the images are completely different.

With regard to the execution time of our scheme on a credit card–size single-board computer, we show the results achieved by implementing such a scheme on the Raspberry Pi B Plus shown in Figure 5, which has extremely constrained hardware and software characteristics.

The Raspberry is an open-source single-board computer based on the ARM11 family processor. This type of computer does not include any built-in hard disk, as it operates on an SD card for booting and long-term storage. In particular, the Raspberry consists of the Broadcom BCM 2835 system on chip (SoC), which integrates a processor (CPU),

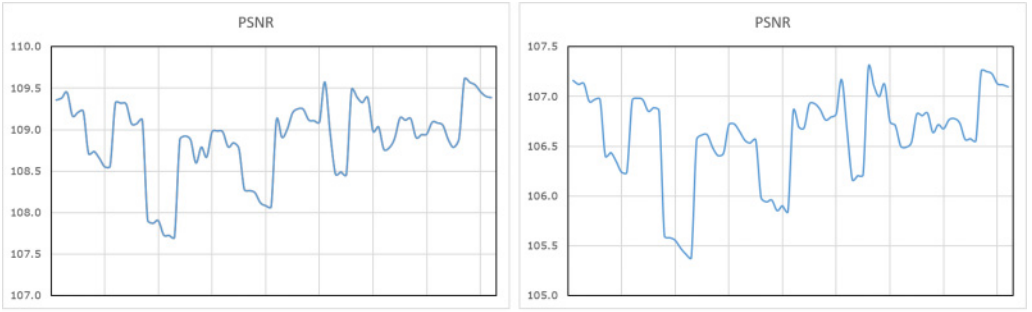
Fig. 5. Raspberry Pi B Plus used for the testing.

a graphics processing unit (GPU), and some memory into a single unit. In detail, the BCM 2835 SoC contains an ARM1176JZ-F processor running at 700MHz, 512MB of RAM, and a GPU named *Video Core IV*. Moreover, the Raspberry supports several standard communication interfaces, such as USB, IEEE 802.3, IEEE 802.11, HDMI, DVI, RCA, RS-232, andRS-485. We remark that the peak power requirements of this computer are low (i.e., 700mA at 5V), and usually the device's power consumption is much lower. The SD card can be loaded with several operating system images. For our testing phase, it was loaded with the Debian OS, called *Raspbian Wheezy*. We decided to implement our scheme on that computer for two main reasons. First, we wanted to demonstrate the efficiency of our watermark scheme even when the resources it could access were extremely constrained. As a result, our scheme is very suitable for implementation directly on board, and it may be easily integrated into the software installed on an fMRI system. Second, a single-board computer has a very small size and supports several communication interfaces, and hence it could be integrated on an fMRI system without the need to use specialized and expensive hardware components.

In this section, we focus on the test results achieved through several experiments performed on a dataset composed of 82 fMRI images and aimed at assessing the effectiveness of the proposed scheme. In detail, such images come from the dataset denoted as *Stop-signal task with unconditional and conditional stopping* provided by the OpenfMRI project [Poldrack et al. 2013].

5.1. Imperceptibility

Let I, I^W, and I^R denote the original image, the watermarked image, and the image reconstructed after the extraction process, respectively. In addition, let ws and ws^E be the embedded watermark string and the extracted one, respectively. We evaluate the distortion between I and I^W. In particular, we focus on several scenarios where an fMRI image can be consulted effectively, even if a watermark is still embedded,



(a) $M = 5$. (b) $M = 8$.

Fig. 6. PSNR values.

without affecting the result of such a consultation. For this reason, we first evaluate the distortion in terms of PSNR with respect to the unwatermarked image. More precisely, we perform the testing activity by considering the parameter M equal to 5 and 8, a watermark string σ composed of 4,096 bits, and the string "123456" as *seed*. From now on, we denote by v_t, v_t^W the t-th 3D data volume in *fMRI* and $fMRI^W$, respectively. Again, we denote by $s_{t,z}$, $s_{t,z}^W$ the z-th slice of the t-th volume in *fMRI* and $fMRI^W$, respectively. Finally, we denote by *MSE* the mean square error.

In particular, for each pair $(fMRI, fMRI^W)$, where *fMRI* is a tested image and $fMRI^W$ is the relative watermarked image, we measure the $PSNR_{(4-D)}$ value, obtained by means of the following equations:

$$PSNR_{(4-D)}(fMRI, fMRI^W) = \frac{1}{T} \times \sum_{t=1}^{T} PSNR_{(3-D)}\left(v_t, v_t^W\right), \tag{1}$$

$$PSNR_{(3-D)}\left(v_t, v_t^W\right) = \frac{1}{Z} \times \sum_{z=1}^{Z} PSNR_{(2-D)}\left(s_{t,z}, s_{t,z}^W\right), \tag{2}$$

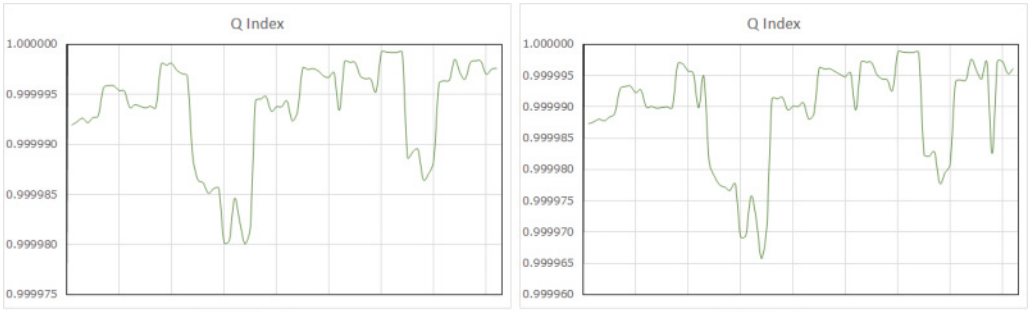
$$PSNR_{(2-D)}\left(s_{t,z}, s_{t,z}^W\right) = 10 \log_{10}\left(\frac{(2^{16}-1)^2}{MSE(s_{t,z}, s_{t,z}^W)}\right), \tag{3}$$

$$MSE\left(s_{t,z}, s_{t,z}^W\right) = \frac{1}{X \times Y} \times \sum_{x=1}^{X} \sum_{y=1}^{Y} (fMRI_s(t,z,x,y) - fMRI_s^W(t,z,x,y))^2. \tag{4}$$

In Figure 6, we show the PSNR trend when $M = 5$ and $M = 8$, respectively. In particular, on the x-axis, the evaluated images are reported, whereas on the y-axis, the relative PSNR values are reported. In detail, we use such values for the M parameter since one of the main aims of the testing phase is to protect subimages of the whole fMRI image, which are sufficiently wide and meaningful.

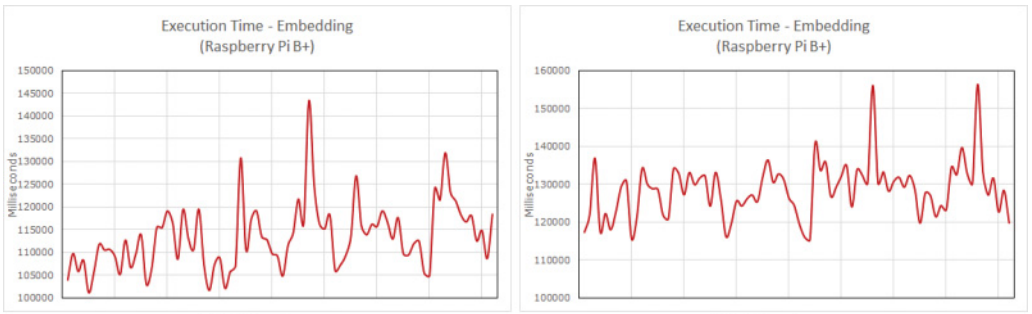
Furthermore, in Figure 7, we graphically report the QI trend when $M = 5$ and $M = 8$, respectively.

By carefully analyzing the preceding figures, it can be noted that the values assumed by the PSNR are very high and the QI is close to 1. Therefore, such results validate the fact that the watermark is not human perceivable. For this reason, nonmedical



(a) $M = 5$. (b) $M = 8$.

Fig. 7. QI values.



(a) $M = 5$. (b) $M = 8$.

Fig. 8. Execution time of the embedding phase.

consultation and online viewing could be still performed by the end user without perceiving any alteration of the image. Finally, as mentioned previously, users interested in a deeper analysis/processing can exactly recover the original image by extracting the embedded watermark string.

5.2. Execution Time

To execute our scheme on multiplatform environments, we implemented it in the Java programming language. More precisely, our experiments are carried out using a Raspberry Pi B Plus equipped with a Class 4 MicroSD of 8GB and version $1.8.0 - b132$ of the Java Runtime Environment (JRT).

In Figure 8(a), we report a graphical representation concerning the execution time of the embedding phase of our scheme when $M = 5$. In detail, on the x-axis, we report the tested images, whereas on the y-axis, we report the execution time in milliseconds. In Figure 8(a), the execution time ranges from around 101,000ms (101 seconds) to around 143,000ms (143 seconds).

Similarly, in Figure 8(b), we graphically report the execution time when $M = 8$. In this case, the execution time ranges from around 115,000ms (115 seconds) to around 156,000ms (156 seconds). On the other hand, the average execution time concerning both the extraction and reconstruction phases is around 240 seconds and 265 seconds when $M = 5$ and $M = 8$, respectively.

6. CONCLUSIONS AND FUTURE WORKS

fMRI is a technology that has a wide range of applications both in medical and research fields. This imaging technique is more and more used in so-called multidomain

environments, where several different entities cooperate by exchanging information. Therefore, it may be useful to detect whether fMRI images have been altered somehow from when they were recorded, and it is crucial to ensure the authenticity of such images. In general, an approach to protect such images may be the inclusion of some information into the image header. However, such an approach is prone to attacks (e.g., manipulation and tampering). In addition, information loss might occur during file format conversions. Finally, even if data encryption can be used to protect data transmitted over insecure networks, decrypted content may be affected by unauthorized use or manipulation at the receiver's side. Therefore, to overcome the aforementioned limitations, we introduce a protection level that is as near as possible to the data. In particular, in this article, we introduced a scheme for fragile reversible watermarking based on the ones proposed in Castiglione et al. [2015b] and Coatrieux et al. [2009]. Such a scheme is explicitly addressed to operate on images characterizing fMRI-based analysis. More precisely, this scheme can be used to ensure authenticity and integrity of fMRI images in a manner that is independent of the image format and without using any external metadata, such as headers and attributes. Moreover, we remark that due to its structure, our scheme may be modeled by means of an open-loop microcontroller. Furthermore, we show how to integrate our scheme within fMRI systems, particularly how such a scheme can operate coupled with the DICOM standard. In addition, we show how our scheme can operate in a manner that is independent of such a standard. Finally, we implement and test the proposed scheme on a credit card–size single-board computer to show the performance and effectiveness of such a scheme even when the hardware and software characteristics at its disposal are extremely constrained.

As a future research direction, we intend to investigate the possibility of protecting this type of image even when multiple trials are produced. Additionally, we remark that the scheme we introduced could be virtually extended to consider other types of sensitive images in which any kind of alteration cannot be tolerated.

ACKNOWLEDGMENTS

The authors would like to thank Professor Ing. Mario Magliulo (Istituto di Biostrutture e Bioimmagini—Consiglio Nazionale delle Ricerche, IBB-CNR, Napoli, Italy) for his valuable support.

REFERENCES

Samia Boucherkha and Mohamed Benmohamed. 2004. A lossless watermarking based authentication system for medical images. In *Proceedings of the International Conference on Computational Intelligence (ICCI'04)*. 240–243.

Arcangelo Castiglione, Raffaele Pizzolante, Alfredo De Santis, Bruno Carpentieri, Aniello Castiglione, and Francesco Palmieri. 2015a. Cloud-based adaptive compression and secure management services for 3D healthcare data. *Future Generation Computer Systems* 43–44, 120–134. DOI: http://dx.doi.org/10.1016/j.future.2014.07.001

Arcangelo Castiglione, Alfredo De Santis, Raffaele Pizzolante, Aniello Castiglione, Vincenzo Loia, and Francesco Palmieri. 2015b. On the protection of fMRI images in multi-domain environments. In *Proceedings of the 29th IEEE International Conference on Advanced Information Networking and Applications (AINA'15)*. IEEE, Los Alamitos, CA, 476–481. DOI: http://dx.doi.org/10.1109/AINA.2015.224

G. Coatrieux, C. Le Guillou, J.-M. Cauvin, and C, Roux. 2009. Reversible watermarking for knowledge digest embedding and reliability control in medical images. *IEEE Transactions on Information Technology in Biomedicine* 13, 2, 158–165.

Benjamin C. Kuo and Farid Golnaraghi. 2002. *Automatic Control Systems* (8th ed.). John Wiley & Sons, New York, NY.

Ovidiu Muresan, Florin Pop, Dorian Gorgan, and Valentin Cristea. 2006. Satellite image processing applications in MedioGRID. In *Proceedings of the 5th International Symposium on Parallel and Distributed Computing (ISPDC'06)*. IEEE, Los Alamitos, CA, 253–262. DOI: http://dx.doi.org/10.1109/ISPDC.2006.42

National Electrical Manufacturers Association. 2015. NEMA PS3.1 2015b/ISO 12052, Digital Imaging and Communications in Medicine (DICOM) Standard. Retrieved November 15, 2016, from http://dicom.nema.org/medical/dicom/current/output/pdf/part01.pdf.

D. Petcu, D. Zaharie, D. Gorgan, F. Pop, and D. Tudor. 2007. MedioGrid: A grid-based platform for satellite image processing. In *Proceedings of the 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'07)*. IEEE, Los Alamitos, CA, 137–142. DOI: http://dx.doi.org/10.1109/IDAACS.2007.4488392

Oleg S. Pianykh. 2009. *Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide*. Springer Science & Business Media.

Raffaele Pizzolante and Bruno Carpentieri. 2012. Copyright protection for images on mobile devices. In *Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'12)*. IEEE, Los Alamitos, CA, 585–590. DOI: http://dx.doi.org/10.1109/IMIS.2012.73

Raffaele Pizzolante, Bruno Carpentieri, and Arcangelo Castiglione. 2013. A secure low complexity approach for compression and transmission of 3-D medical images. In *Proceedings of the2013 8th International Conference on Broadband and Wireless Computing, Communication, and Applications*. IEEE, Los Alamitos, CA, 387–392. DOI: http://dx.doi.org/10.1109/BWCCA.2013.68

Raffaele Pizzolante, Bruno Carpentieri, Aniello Castiglione, and Giancarlo De Maio. 2011. The AVQ algorithm: Watermarking and compression performances. In *Proceedings of the 2011 3rd International Conference on Intelligent Networking and Collaborative Systems (INCoS'11)*. IEEE, Los Alamitos, CA, 698–702. DOI: http://dx.doi.org/10.1109/INCoS.2011.153

Russell A. Poldrack, Deanna M. Barch, Jason P. Mitchell, Tor D. Wager, Anthony D. Wagner, Joseph T. Devlin, Chad Cumba, Oluwasanmi Koyejo, and Michael P. Milham. 2013. Toward open sharing of task-based fMRI data: The OpenfMRI project. *Frontiers in Neuroinformatics* 7, 1–12.

Florin Pop, Claudiu Gruia, and Valentin Cristea. 2007. Distributed algorithm for change detection in satellite images for grid environments. In *Proceedings of the 6th International Symposium on Parallel and Distributed Computing (ISPDC'07)*. IEEE, Los Alamitos, CA, 303–308. DOI: http://dx.doi.org/10.1109/ISPDC.2007.13

Zhou Wang and Alan C. Bovik. 2002. A universal image quality index. *IEEE Signal Processing Letters* 9, 3, 81–84.

Zhou Wang, Alan C. Bovik, and Ligang Lu. 2002. Why is image quality assessment so difficult? In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*. IEEE, Los Alamitos, CA, 3313–3316. DOI: http://dx.doi.org/10.1109/ICASSP.2002.5745362