DOI: 10.1016/j.cose.2017.06.003

On the protection of consumer genomic data in the Internet of Living Things

Raffaele Pizzolante *, *Arcangelo Castiglione, Bruno Carpentieri, Alfredo De Santis, Francesco Palmieri, Aniello Castiglione*

Department of Computer Science, University of Salerno, Via Giovanni Paolo II, 132, I-84084 Fisciano (SA), Italy

Keywords:
Consumer genomic data
Internet of Living Things
Sequencing sensors
IoT
Cloud
Watermark
Protection
Security

ABSTRACT

Several companies have recently emerged to provide *online Direct-To-Consumer (DTC) DNA analysis and sequencing*. Those activities will be, in the near future, the foundations of the emerging *Internet of Living Things*. The concept of Internet of Living Things has been introduced to characterize networks of biological sequencing sensors, which could rely on cloud-based analysis capabilities, to support the users in deeply studying DNA or other molecules. Sequencing sensors have many fields of application and much more will likely to come. In this context, *DNA microarray images* represent the core of modern genomic data analysis, since they allow the simultaneous monitoring of many thousands of genes and represent a sort of "container", not only for storing genomics data, but also for managing, sharing and exchanging such type of data.

In this scenario, the ability to protect genomics and medical big data is a growing challenge. In particular, for what concerns DNA microarray images, the techniques commonly employed for data protection are not effective due for example to the unauthorized use or manipulation after decryption or the lost of metadata during image processing.

In this paper we address the problem of protecting such type of information, by means of watermarking techniques. In particular, we propose *reversible watermarking techniques* explicitly tailored for the characteristics of DNA microarray images to ensure the protection of such images in terms of authenticity and integrity, besides enabling the binding of those imaging data with other information related to them. We assess the effectiveness and efficiency of our techniques by means of a working prototype.

1. Introduction

The *Internet of Things (IoT)* will connect not only computers and mobile devices, but, in the near future, it will also connect Smart Infrastructures. In the context of the IoT, it is important to point out that the "things" will encompass a wider set of devices, such as devices for DNA sequencing and analysis, which will carry out several monitoring activities (Erlich, 2015). In this way, the devices for DNA sequencing and analysis will build an *Internet of Living Things (IoLT)* (Medeiros, 2017). The concept of Internet of Living Things has been introduced to characterize networks of biological sequencing sensors, which could rely on cloud-based analysis capabilities, to support the users in deeply studying DNA or other molecules (Clark, 2017; Waltz, 2017).

Given the initial mapping and open publication of *human genome*, the next step in genomic-based research will be the

* *Corresponding author.*
E-mail addresses: rpizzolante@unisa.it (R. Pizzolante), arcastiglione@unisa.it (A. Castiglione), bc@dia.unisa.it (B. Carpentieri), ads@unisa.it (A. De Santis), fpalmieri@unisa.it (F. Palmieri), castiglione@ieee.org, castiglione@acm.org (A. Castiglione).

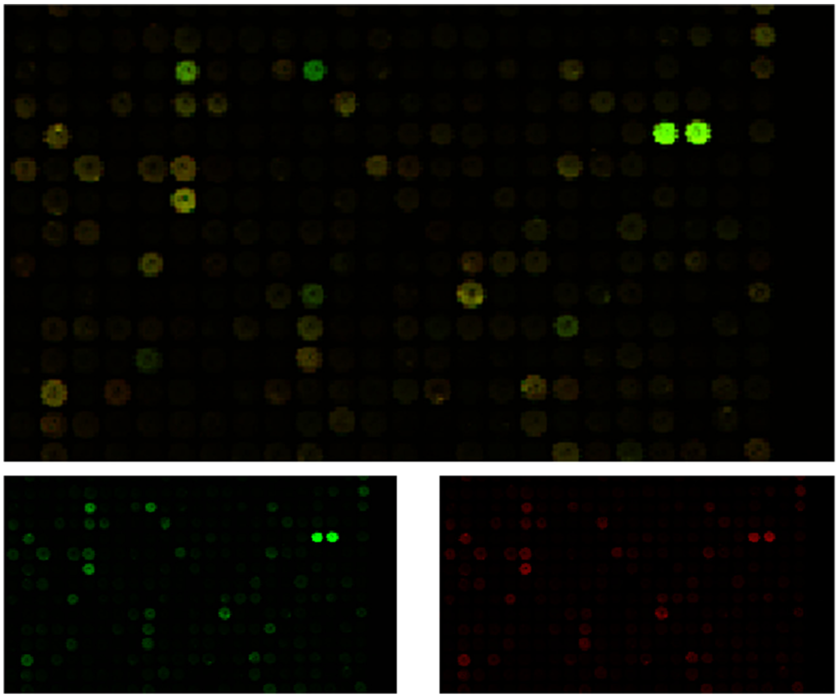


Fig. 1 – A typical DNA microarray image, along with the sub-images characterizing the green and red channels of such image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

sequencing sensors, which are tiny sequencing devices, built for real time applications, widely deployed and cheap. Sequencing sensors will be extremely tiny devices, which enable automatic sample preparation and real-time sequencing. Combining those tiny sensors in larger systems will include a DNA-awareness layer to several devices (Erlich, 2015). In addition, for what concerns genome sequencing applications, many researchers are developing streaming methods to make on-the-fly comparisons (Bhatt et al., 2017). Indeed, the upload of sequence data, produced in real time by modern sequencing devices, requires a lower bitrate than streaming a movie over the Internet (Burke, 2017).

1.1. *Motivations*

The Internet of Living Things technology has many fields of application, and much more will likely to follow (Erlich, 2015). One popular example is given by online *personalized genomic testing* (Phillips, 2016; Swan, 2012). More precisely, since the completion of the *Human Genome Project*, many companies sell DNA testing directly to consumers. Such companies exploit recent advances in genome-wide scanning and sequencing technologies, to provide their customers with a series of *personalized genetic profiles*. Online *consumer genomics* companies propose and sell a lot of DNA-based tests; in all cases, what the user needs to do is to fill a tube with his own saliva. Some companies provide genealogical information. Others offer non health-related DNA information, concerning not only the ancestry and ethnicity, but also paternity, extended relationships and individual uniqueness. Another class of companies provides genetic tests to help customers to improve their health in indirect ways, for example by means of nutrition and lifestyle. Again, a further class of companies provides disease risk testing and pharmacogenetic tests, whose aim is to support and complement regular medical care. There are also companies which sell complete personal genome sequencing, to provide users with both health-related and non-health-related information; however, such companies are still quite expensive (Nordgren and Juengst, 2009). Again, by exploiting the features offered by consumer genomics and Internet of Living Things, *personalized healthcare* will provide the ability to treat patients on a case by case basis, customized on their specific genomic blueprint (Murray, 2012). These advances are particularly relevant for cancer genomics, which is the application of genetic therapy to cancer diagnoses and treatment that is customized to people's individual circumstances. The effects of the above defined applications will be even more magnified by the introduction of the fifth-generation broadband technology (5G) (Andrews et al., 2014), which will allow the gathering of genomic data and the storing of such data on a cloud. In this way, the physicians can easily customize their treatments by accessing to detailed knowledge about genetic composition (West, 2016).

The *microarray technology* represents one of the most important components in the field of genomic data analysis. In particular, DNA microarray images (shown in Fig. 1) are a vital component of genomic data analysis, since they enable the simultaneous monitoring of many thousands of genes (Jain et al., 2002) and represent a sort of "container", not only for the storing of genomics data, but also to manage, share and exchange such type of data.

1.2. *Security issues*

The risks involved with the clinical genetic testing, concerning personal privacy, familial dynamics and genetic discrimination, are exhaustively addressed in the literature (Nordgren and Juengst, 2009). Again, as pointed out in O'Driscoll et al. (2013), the ability to protect genomics data in the era of big data and IoT is an ever growing concern. In addition, clinical sequencing must address strict regulatory requirements, primarily due to the *Health Insurance Portability and Accounting Act (HIPAA)* of 1996. Indeed, such type of data should meet the same security requirements defined for sensitive healthcare systems (Guo et al., 2016; He et al., 2016; Liu et al., 2016; Rahman et al., 2017). Thus, the adoption of cloud computing should be considered with care in such environments with appropriate measurements (Alam et al., 2017; Son et al., 2016; Xiong et al., 2017). In fact, as stated in O'Driscoll et al. (2013), several fundamental aspects of data security in clouds should be addressed before the widespread adoption of cloud-based clinical sequencing can take place. Those challenges should be addressed to build a secure and resilient IoLT infrastructure, where *Confidentiality, Integrity and Availability (CIA)* must be assured (Pacheco and Hariri, 2016).

In general, to protect DNA microarray images, which have a characterizing "spotted" internal structure, some additional information could be included into the image header, but this approach is prone to attacks, such as *tampering*-based ones. Furthermore, information loss could occur due to file format conversions. Again, even if encryption could be used to protect data transmitted over insecure channels (Pizzolante et al., 2013), decrypted content may be affected by misuse or manipulation at the receiver's side. Therefore, to address the above defined issues, it makes sense to introduce specific security approaches which enable the protection of such images in a manner that is *as independent as possible* from the specific image format and, at the same time, ensure a protection level which is *as close as possible* to the imaging data. We emphasize that by using security mechanisms which operate at pixel level, besides protecting a given image, we also ensure that the protection is resistant to file format conversion.

Digital watermarking is a well-established approach to ensure authenticity and integrity of imaging data, introducing a protection level which is the nearest as possible to such data (Albano et al., 2012, 2014; Castiglione et al., 2015; Pizzolante and Carpentieri, 2013; Pizzolante et al., 2011, 2013, 2014). However, digital watermarking schemes irreversibly distort the original image, and this could not be tolerated when the data are intended to be used for health-related (or more in general for sensitive) applications, as in the case of DNA microarray images. Therefore, since such images should be kept without any information loss, the watermark should not introduce any perceivable distortion in the image, and it should not obstruct the qualitative perception of the image.

1.3. *Our contribution*

In this paper we propose an *invisible fragile watermarking scheme* (Caldelli et al., 2010; Feng et al., 2006; Lee et al., 2007) explicitly tailored for the protection of DNA microarray images. In the proposed scheme, the original image can be completely recovered upon the extraction of the embedded information.

Moreover, we extend the proposed scheme to enable the watermark embedding in a user-defined area, i.e., a *Region Of Interest (ROI)* (Al-Qershi and Khoo, 2011; Wakatani, 2002). We emphasize that in this way the end-user can define which area he intends to protect. Finally, in order to assess the performance and the effectiveness of our proposal, even in the presence of extremely constrained hardware and software capabilities, we design and implement a working prototype of our scheme, by also testing it on a *Raspberry Pi* device. For this purpose, we rely on the *Peak Signal-to-Noise Ratio (PSNR)* and the *Q-Index (QI)* metrics to assess the imperceptibility of the watermark embedded by the proposed scheme. The results obtained by such metrics validate the fact that the embedded watermark is *not human-perceivable*. Again, the testing activity performed shows that our scheme is characterized by a low complexity and is quite efficient in terms of execution time, and the same holds for the ROI-based version. We emphasize that this confirms the applicability of our proposal directly in on-board miniaturized sensors.

1.4. *Organization*

This paper is organized as follows. In Section 2 we describe the proposed invisible and reversible fragile watermarking scheme for the protection of consumer genomics data by highlighting its main features, operation logic and advantages introduced with respect to the state of the art. Furthermore, in the same section, we present an extension of such scheme, enabling the end-user to select the ROI before the watermark embedding. In Section 3 we assess the features and performance of both the above schemes in hardware-constrained environments by testing them on a publicly available dataset. Finally, in Section 4 we draw conclusions and future research perspectives.

2. The proposed watermark schemes

In general, a digital watermark is a secret key-dependent signal inserted into digital data, which can be later detected/extracted to make an assertion about such data, e.g., integrity, identification, authentication, etc. (Barni and Bartolini, 2004). More precisely, a digital watermark can be viewed as a sort of *natural noise*. In detail, the information to be embedded is encoded into the original unwatermarked data by adding more natural noise and/or rearranging existing noise. The locations for embedding the watermark, as well as the value of the watermark itself, are usually determined by secret elements, e.g., keys. We remark that the distribution and management of such secret information is a non-trivial problem and it should be addressed with extreme care, so that only authorized users are given access to some resources; this can be achieved by properly distributing the aforementioned secret information (Castiglione et al., 2014, 2016).

2.1. *Invisible reversible fragile watermarking*

The proposed scheme enables the embedding of a watermark string W into the image I by affecting the least as possible

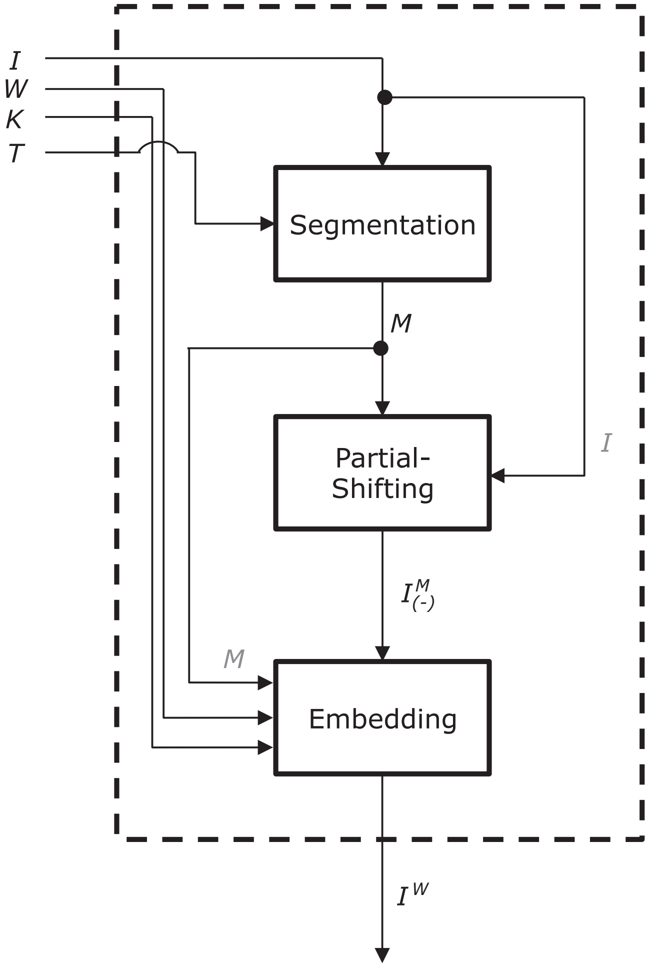


Fig. 2 – The embedding process of the proposed scheme.

the most significant parts of that image, such as the *spots*. We emphasize that through such scheme, the watermarked image might be still used to carry out some processing and analysis, which generally operate only on the meaningful parts of the image. However, as stated before, due to the reversibility of the scheme, it is always possible to restore the original unwatermarked image. The scheme belongs to the category of fragile watermarking schemes, hence, any change on the watermarked image may cause the loss of the embedded watermark, so that in order to verify whether the data integrity has been compromised, such a feature can be easily exploited. More precisely, by setting W as the digital signature of a digest computed on I, through some *cryptographically secure hash functions* (e.g., *Keccak* (Bertoni et al., 2013), SHA3-224, SHA3-384, etc.), the receiver end-point can verify both authenticity and integrity of the watermarked image. We emphasize that our scheme is specifically designed to be implemented on hardware-constrained devices and the watermark is embedded into the *spatial domain*.

In Fig. 2 we show the logical functioning of the protection scheme, and in particular the relative embedding phases. In detail, the embedding procedure takes the following parameters as its inputs:

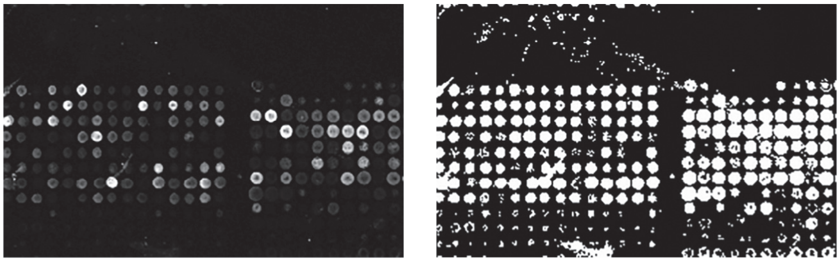
- *I*: DNA microarray image;
- W: Watermark string;
- *K*: Key used for embedding and extraction;
- *T*: Threshold used for the segmentation phase.

In the first phase, the spots are separated from the not-significant information, i.e., the background, by using the *threshold-based segmentation procedure* we propose to detect the spots, outlined in Algorithm 1.

Algorithm 1 The *Segmentation* procedure.

```
procedure SEGMENTATION($I$, $T$)
    for $x = 1$ to $I.width$ do
        for $y = 1$ to $I.height$ do
            if $I(x, y) > T$ then
                $M(x, y) = true$;
            else
                $M(x, y) = false$;
            end if
        end for
    end for
    return $M$;
end procedure
```

In particular, as it can be observed from Fig. 3a, since the spots generally show a higher intensity, they can be separated by using an appropriate threshold. In detail, the *Segmentation* procedure returns a bitmap mask *M*, in which $M(x, y) = true$ if it holds that $I(x, y) > T$, where *T* is the threshold, and $M(x, y) = false$, otherwise. Notice that *M* characterizes the points in which W will be spread, i.e., in this case the not-significant regions. For what concerns such a procedure, since each sample of the input image *I* is processed through the two *for* loops of Algorithm 1, the asymptotic time complexity depends on the size of *I* and it is $\mathcal{O}(I \cdot width \times I \cdot height)$. Again,


(a) Portion of a DNA microarray image.

(b) Segmented portion of a DNA microarray image.

Fig. 3 – Example of an output of the *Segmentation* procedure.

since such a procedure uses and returns a bitmap mask M of $I.width \times I.height$ entries, assuming that each bit of M is stored in a *cell* of memory, the asymptotic space complexity required by the *Segmentation* procedure is $\mathcal{O}(I \cdot width \times I \cdot height)$. Fig. 3a and b shows a portion of a DNA microarray image, together with the corresponding portion in the image M obtained through the *Segmentation* procedure by setting $T = 1500$. Again, in Fig. 3b, the black points represent the ones where M takes the value *false*.

It is important to note that some pixels could be increased by 1 due to the modifications for the embedding of W. In some "*borderline*" cases, such modifications can affect the correct functioning of the *Segmentation* procedure. Such a procedure is required for the extraction of W, as well as for recovering I from the watermarked image, as shown in Fig. 4. In order to ensure the correct functioning of the scheme, the *borderline* cases are adequately managed, as described in the following.

For instance, consider a scenario in which we set $T = a$ and select for the embedding a pixel $I(x, y)$, having a value equal to a. Suppose that the value of the pixel is changed to $a + 1$, then, we obtain as output $I'(x,y) = a+1$. In such a scenario, when the *Segmentation* procedure is carried out for the extraction on I', this will lead to an incoherence, since $I'(x,y) > a$. Therefore, the pixel $I'(x, y)$ is considered as not-significant by the extraction process. In order to avoid the above mentioned issue, all the not-significant pixels are modified, decreasing their values by 1.

The *PartialShifting* procedure, outlined in Algorithm 2, is responsible to perform the pixel shifting, according to the mask M. Such a procedure needs to process the whole input image I, by means of the *for* loops of Algorithm 2. Therefore, the asymptotic time complexity of such a procedure is $\mathcal{O}(I \cdot width \times I \cdot height)$. We remark that since the above mentioned procedure returns a properly modified copy of I, referred to as $I^{M}_{(-)}$, and $I^{M}_{(-)}$ has the same size as I, assuming that a sample is stored in a cell of memory, the asymptotic space complexity is $\mathcal{O}(I \cdot width \times I \cdot height)$.

Algorithm 2 The *PartialShifting* procedure.

```
procedure PartialShifting($I$, $M$)
    for $x = 1$ to $I.width$ do
        for $y = 1$ to $I.height$ do
            if $M(x, y) == false$ then
                $I^{M}_{(-)}(x, y) = I(x, y) - 1$;
            else
                $I^{M}_{(-)}(x, y) = I(x, y)$;
            end if
        end for
    end for
    return $I^{M}_{(-)}$;
end procedure
```

The *Embedding* procedure is described by Algorithm 3 and is in charge of modifying the pixel values of $I^{M}_{(-)}$ (the output of the previous phase) in order to embed W. In particular, the *Embedding* procedure implements an additive scheme, namely, W is added directly to the image signal, i.e., to its pixels, and it is based on the schemes introduced in Castiglione et al. (2015) and Coatrieux et al. (2009).

In Fig. 5a we show the unwatermarked portion of a DNA microarray image, whereas in Fig. 5b, we show the same portion of the watermarked image, which embeds a watermark string of 256 bits.

Algorithm 3 The *Embedding* procedure.

```
procedure Embedding($I^{M}_{(-)}$, $M$, $K$, $T$)
    $G = PRNG(K)$;
    $I^{W} = duplicate(I^{M}_{(-)})$;
    $wIdx = 1$;
    repeat
        $\forall l \in \{1, \cdots, 4\}$, select $(x^{(l)}, y^{(l)})$ by using $G$, so that the following conditions hold
        • $0 \le x^{(l)} \le I^{M}_{(-)}.width$;
        • $0 \le y^{(l)} \le I^{M}_{(-)}.height$;
        • $M(x^{(l)}, y^{(l)}) == false$;
        • $(x^{(l)}, y^{(l)})$ is not previously selected.
        $B = obtainBlock(I^{M}_{(-)}, (x^{(1)}, y^{(1)}), \cdots, (x^{(4)}, y^{(4)}))$;
        $B^{E} = estimateBlock(B)$;
        $D = |B[1, 1] - B^{E}[1, 1]|$;
        if $D < 1$ then
            $B^{W} = embedSymbol(B, D, W[wIdx])$;
            $wIdx = wIdx + 1$;
            $update(I^{W}, B^{W}, (x^{(1)}, y^{(1)}), \cdots, (x^{(4)}, y^{(4)}))$;
        else
            Modify $B$ to increase $D$, by adding $W$ to $B$ or subtracting $W$ from $B$;
            $update(I^{W}, B, (x^{(1)}, y^{(1)}), \cdots, (x^{(4)}, y^{(4)}))$;
        end if
        Set $(x^{(l)}, y^{(l)})$, $\forall l \in \{1, \cdots, 4\}$ as no longer selectable;
    until $(wIdx \le W.length)$
    return $I^{W}$;
end procedure
```

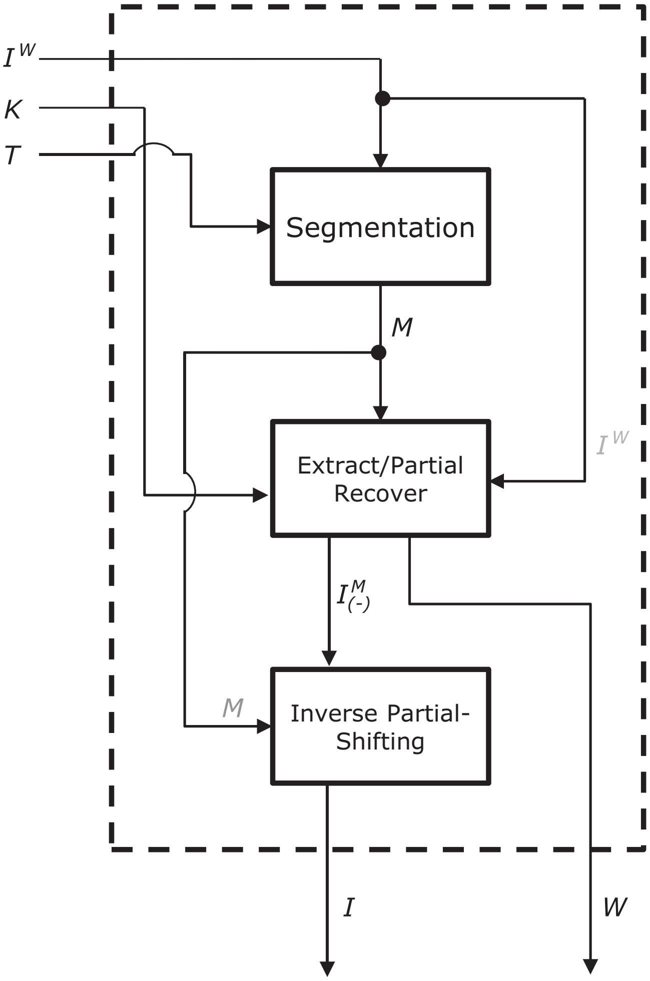


Fig. 4 – The extraction process of the proposed scheme.

In detail, each bit of W is embedded into a certain block of $I^M_{(-)}$, constituted by a matrix of 2×2 pixels. More precisely, four coordinates $(x^{(l)}, y^{(l)})$ in the not-significant area of $I^M_{(-)}$ are selected, where $1 \le l \le 4$ (lines 6–10). Subsequently, by using such coordinates, a 2×2 block, denoted as B, is obtained, through the *obtainBlock* procedure. Notice that B is referred to as *candidate block*. Candidate blocks should be further classified into two typologies: *carrier blocks*, where it is possible to embed a bit of W, and *noncarrier blocks*, in which the embedding is not possible. In particular, a bit of W can be embedded into a carrier block by adding or subtracting, according to the value of the bit, the watermark pattern signal W* (see Eq. (1)), as shown by the *embedSymbol* procedure in Algorithm 4,

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

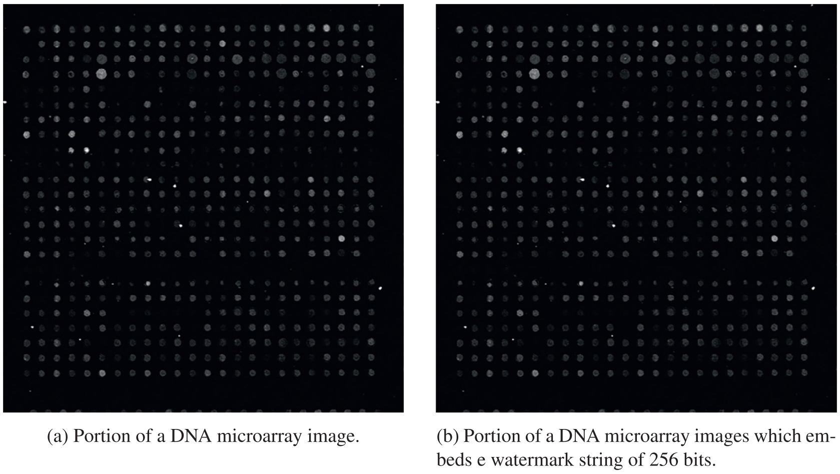
In order to classify a candidate block B, the *estimateBlock* procedure outlined in Algorithm 4 performs an estimation of each block.

Algorithm 4 The sub-procedures used by the *Embedding* procedure.

```
1:  procedure embedSymbol($B$, $Symbol$)
2:      if $Symbol == 1$ then
3:          $B^W = B + W^*$;
4:      else
5:          $B^W = B - W^*$;
6:      end if
7:      return $B^W$;
8:  end procedure
9:  procedure obtainBlock($I$, $(x^{(1)}, y^{(1)}), \cdots, (x^{(4)}, y^{(4)})$)
10:     $B[1,1] = I(x^{(1)}, y^{(1)})$;
11:     $B[1,2] = I(x^{(2)}, y^{(2)})$;
12:     $B[2,1] = I(x^{(3)}, y^{(3)})$;
13:     $B[2,2] = I(x^{(4)}, y^{(4)})$;
14:     return $B$;
15: end procedure
16: procedure estimateBlock($B$)
17:     $B^E[1,1] = \frac{(2 \times B[1,1] + B[1,2] + B[2,1])}{4}$
18:     $B^E[1,2] = \frac{(2 \times B[1,2] + B[1,1] + B[2,2])}{4}$
19:     $B^E[2,1] = \frac{(2 \times B[2,1] + B[1,1] + B[2,2])}{4}$
20:     $B^E[2,2] = \frac{(2 \times B[2,2] + B[1,2] + B[2,1])}{4}$
21:     return $B^E$
22: end procedure
```

Let B^E denote such estimation, where each pixel is obtained through the linear combination of some pixels of B. In detail, by verifying the relation between B and B^E (i.e., the difference D, line 13), it can be distinguished (lines 14–21) if B is a carrier block ($D < 1$) or not. Again, we emphasize that also the extraction algorithm can detect, in two phases, the candidate blocks. Finally, given the reversibility of our scheme, it is possible to recover the original image block, B, from the watermarked one, B^W.

The *Embedding* procedure invokes several sub-procedures, some of them are outlined in Algorithm 4. In detail, each of the aforementioned sub-procedures has asymptotic time complexity $\mathcal{O}(1)$, since the relative running time does not depend on the size of the input. More precisely, the *embedSymbol* and *estimateBlock* procedures perform some operations on the fixed-size input block B, whereas the aim of the *obtainBlock* procedure is to populate and return a block B by retrieving the values of the samples from I, according to the input coordinates $(x^{(1)}, y^{(1)})$, ···, $(x^{(4)}, y^{(4)})$. Furthermore, the *embedSymbol*, *estimateBlock* and *obtainBlock* procedures have asymptotic space complexity $\mathcal{O}(1)$ due to the fact that they use fixed-sized structures, i.e., a fixed-size block which will be returned and has the same dimension as B. After evaluating its sub-procedures, we focus on the *Embedding* procedure. In detail, we focus on the *repeat/until* loop outlined in Algorithm 3. More precisely, the number of iterations performed by the above loop is equal to the number of candidate blocks which have been considered. Let $NB_{candidate}$ be the number of candidate blocks. Recall that a candidate block can be either a carrier or a noncarrier block. More formally, let



(a) Portion of a DNA microarray image.

(b) Portion of a DNA microarray images which embeds e watermark string of 256 bits.

Fig. 5 – Example of application of our proposed scheme.

$NB_{carrier}$ and $NB_{noncarrier}$ denote the number of carrier and noncarrier blocks, respectively, it holds that $NB_{candidate} = NB_{carrier} + NB_{noncarrier}$. Notice that the value of $NB_{carrier}$ is equal to the length of the watermark string, *W.length*, since a bit of *W* is embedded into each carrier block. Without loss of generality, we can consider an *upper bound* on the number of noncarrier blocks, referred to as $NB^{max}_{noncarrier}$, which will be considered by the algorithm. More precisely, if $NB_{noncarrier} \leq NB^{max}_{noncarrier}$, the algorithm successfully embeds *W* into *I*. Otherwise, the algorithm fails the embedding process. According to the aforementioned considerations, the number of iterations performed by the *repeat/until* loop is equal to $W \cdot length + NB^{max}_{noncarrier}$, so the relative asymptotic time complexity is $\mathcal{O}(W \cdot length + NB^{max}_{noncarrier})$, since each operation and sub-procedure within the loop has asymptotic time complexity $\mathcal{O}(1)$. In addition, since for each iteration of the loop the coordinates of candidate blocks are stored and not considered any further, the computational space complexity is $\mathcal{O}(W \cdot length + NB^{max}_{noncarrier})$. The asymptotic computational time and space complexity of the *Embedding* procedure is $\mathcal{O}(I \cdot width \times I \cdot height) + \mathcal{O}(W \cdot length + NB^{max}_{noncarrier})$, where $\mathcal{O}(I \cdot width \times I \cdot height)$ is given by the *duplicate* sub-procedure. We remark that if *I* is directly modified, without creating a local copy I^W through the *duplicate* sub-procedure, the asymptotic time and space complexity is $\mathcal{O}(W \cdot length + NB^{max}_{noncarrier})$, thus obtaining an improvement of the performance. Moreover, if the upper bound $NB^{max}_{noncarrier}$ is arbitrary chosen, without considering the length of the watermark string *W.length*, and the *duplicate* sub-procedure is not used, the asymptotic computational time and space complexity is $\mathcal{O}(W \cdot length)$, since $NB^{max}_{noncarrier}$ is almost constant.

2.2. *Region of interest (ROI) watermarking*

In several scenarios it could be necessary to protect only a *Region Of Interest* (*ROI*) of a DNA microarray image. For this reason, we introduce a reversible scheme, based on the *Embedding* procedure outlined in Algorithm 3, which enables the embedding of a watermark string according to a user-defined ROI. For example, our proposed scheme enables to embed the *digital signature* of the ROI into the ROI itself. In this case, once the watermark has been extracted and the ROI has been recovered, it is possible to verify the presence of any alterations. Therefore, if the processing is focused only on the ROI, malicious alterations outside this user-defined region could not invalidate the processing of the image. However, we emphasize that before being processed, the DNA microarray image should be recovered, by extracting the watermark string, since the ROI is altered by the watermark. On the other hand, one of the most important advantages related to the embedding of the watermark outside the ROI is that such an embedding does not alter the ROI itself on the watermarked image. Thus, some processing which involves only the ROI might be still performed directly on the watermarked image. In addition, also in this case, the integrity of the ROI can be verified by checking whether the watermark contains the digital signature of the ROI. We stress that to define and select the ROI area, our scheme requires the active participation by the end-user,

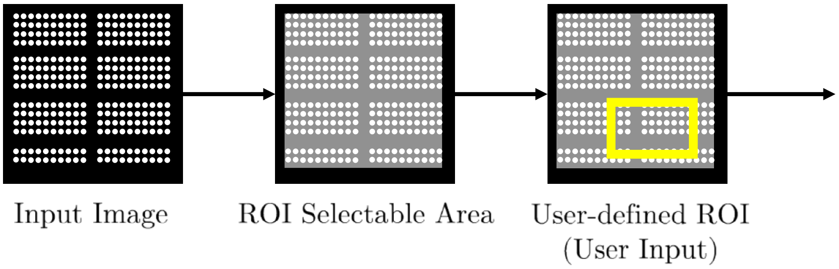


Fig. 6 – User interactions in the proposed scheme.

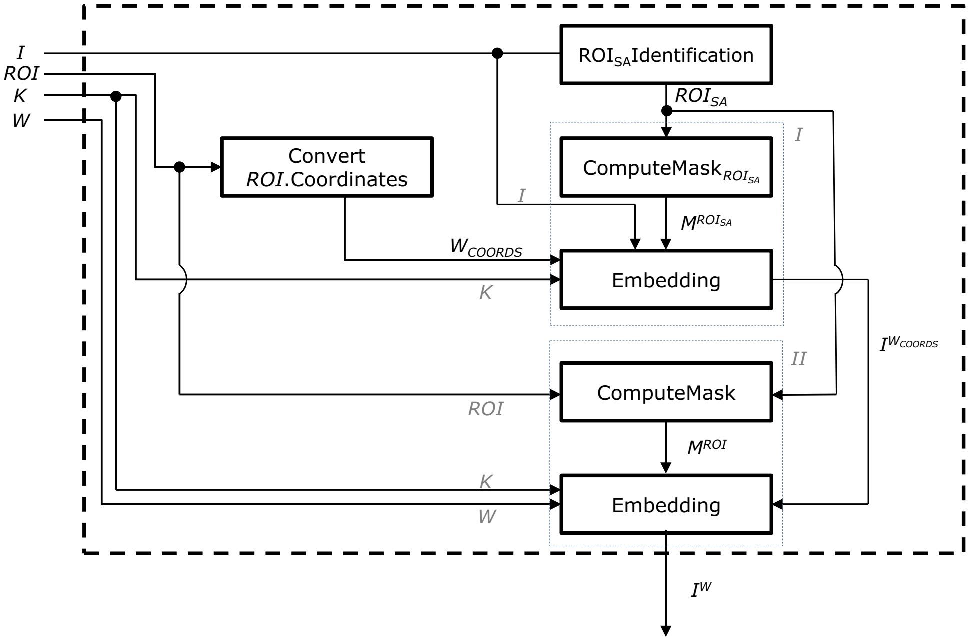


Fig. 7 – Overall logical functioning of our ROI watermarking embedding process.

as graphically outlined in Fig. 6. In detail, the end-user can interact with the system through several input devices, as for instance by using touchscreen-based devices (tablets, smartphones, embedded computers), digital pens, mouse, etc.

In Fig. 7 we show the logical functioning of the embedding scheme. In our scheme, after the selection of a ROI by the end-user, the watermark W will be embedded into the selected ROI or outside it, depending on the end-user preferences.

More precisely, in the first phase, the area of the image in which the end-user can select a ROI is identified. Basically, the selected area, referred to as *ROI Selectable Area* (ROI_{SA}), is a *rectangle* that contains all the significant portions of the image, i.e., the spots. Subsequently, by considering the ROI_{SA}, a mask referred to as $M^{ROI_{SA}}$ is obtained, through the $ComputeMask_{ROI_{SA}}$ procedure outlined in Algorithm 5, which has asymptotic time and space complexity $\mathcal{O}(I \cdot width \times I \cdot height)$. Afterward, the ROI coordinates, which are represented by means of the bit string W_{COORDS}, are embedded outside the ROI_{SA}, through the *Embedding* procedure, as shown in Fig. 8.

We denote the output of the first embedding as $I^{W_{COORDS}}$. Similarly, the watermark W is embedded into the user-defined ROI, as highlighted in Fig. 8. Again, we emphasize that the watermark W can be also embedded outside the ROI, but inside the ROI_{SA}. It is important to highlight that the *ComputeMask* procedure is used to compute the proper mask M^{ROI}, according to the user-defined ROI and ROI_{SA}. In detail, M^{ROI} is used to embed W inside or outside the ROI, based on the end-user preferences. Similar to the $ComputeMask_{ROI_{SA}}$ procedure, the asymptotic time and space complexity of *ComputeMask* is $\mathcal{O}(I \cdot width \times I \cdot height)$. Subsequently, I^W is obtained by the embedding of W into $I^{W_{COORDS}}$, according to M^{ROI}. Finally, I^W is returned as output.

Algorithm 5 The *$ComputeMask_{ROI_{SA}}$ procedure.*

```
procedure COMPUTEMASK_{ROI_{SA}}($I$, $ROI_{SA}$)
    for $x = 1$ to $I.width$ do
        for $y = 1$ to $I.height$ do
            if $(x, y) \in ROI_{SA}$ then
                $M^{ROI_{SA}}(x, y) = false$;
            else
                $M^{ROI_{SA}}(x, y) = true$;
            end if
        end for
    end for
    return $M^{ROI_{SA}}$;
end procedure
```

2.2.1. *ROI selectable area identification*

It is important to remark that DNA microarray images are highly structured, since their spots, which are characterized by higher intensity, are located on a *regular grid* (Lonardi and Luo, 2004). Starting from such consideration, to identify the grid we analyze the average pixel intensities of such images. In our scheme, the grid is substantially the area in which it is meaningful, for the end-user, to select a ROI, i.e., the ROI_{SA}. As a consequence, all the significant parts of a DNA microarray image will be contained into the grid.

Figs. 9 and 10 show an example concerning the trend of the average intensities, column-by-column and row-by-row, of a DNA microarray image, respectively.

We logically characterize, in a given DNA microarray image, the boundaries of the ROI_{SA} by considering two vertical axes, denoted as $x_{(W)}$ and $x_{(E)}$, respectively, and two horizontal axes, denoted as $y_{(N)}$ and $y_{(S)}$, respectively, as shown in Fig. 11. As it

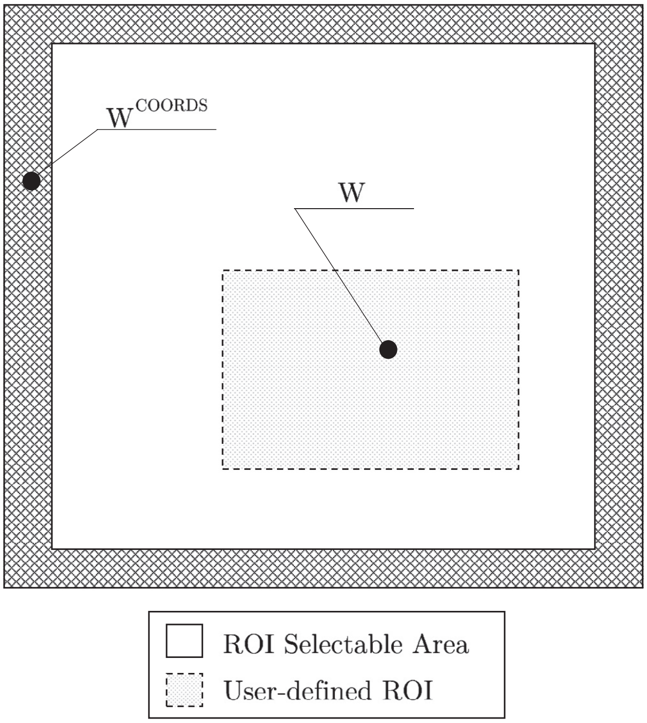


Fig. 8 – Embedding of *W* and W_{COORDS} into the image.

can be observed from Fig. 11, by using the intersections of such axes, it is possible to identify, through the points $P_1^{ROI_{SA}}$, $P_2^{ROI_{SA}}$, $P_3^{ROI_{SA}}$ and $P_4^{ROI_{SA}}$, the rectangle characterizing the ROI_{SA}. In Algorithm 6 we report the ROI_{SA}*Identification* procedure, which invokes the $Compute_{x(W)}$, $Compute_{x(E)}$, $Compute_{y(N)}$ and $Compute_{y(S)}$ sub-procedures to identify the aforementioned vertical axes and then the intersection points $P_1^{ROI_{SA}}$, $P_2^{ROI_{SA}}$, $P_3^{ROI_{SA}}$ and $P_4^{ROI_{SA}}$.

Algorithm 6 The $ROI_{SA}Identification$ procedure.

```
procedure ROI_SA IDENTIFICATION($I$, $numOfRefs$, $percentageOfPeak$)
    $x_{(W)} = \text{Compute}_{x_{(W)}}(I, numOfRefs, percentageOfPeak)$;
    $x_{(E)} = \text{Compute}_{x_{(E)}}(I, numOfRefs, percentageOfPeak)$;
    $y_{(N)} = \text{Compute}_{y_{(N)}}(I, numOfRefs, percentageOfPeak)$;
    $y_{(S)} = \text{Compute}_{y_{(S)}}(I, numOfRefs, percentageOfPeak)$;
    $P_1^{ROI_{SA}} = (x_{(W)}, y_{(N)})$;
    $P_2^{ROI_{SA}} = (x_{(E)}, y_{(N)})$;
    $P_3^{ROI_{SA}} = (x_{(W)}, y_{(S)})$;
    $P_4^{ROI_{SA}} = (x_{(E)}, y_{(S)})$;
    $ROI_{SA} = < P_1^{ROI_{SA}}, P_2^{ROI_{SA}}, P_3^{ROI_{SA}}, P_4^{ROI_{SA}} >$;
    return $ROI_{SA}$;
end procedure
```

In this section we only report the $Compute_{x(W)}$ and $Compute_{x(E)}$ sub-procedures, outlined in Algorithm 7 and Algorithm 8, respectively, which identify the two vertical axes, referred to as $x_{(W)}$ and $x_{(E)}$. We remark that the $Compute_{y(N)}$ and $Compute_{y(S)}$ sub-procedures are substantially symmetrical to $Compute_{x(W)}$ and $Compute_{x(E)}$, even if they operate on the horizontal axes, i.e., $y_{(N)}$ and $y_{(S)}$. We emphasize that the $averageIntensities_{VERTICAL}$ sub-procedure, used by the $Compute_{x(W)}$ and $Compute_{x(E)}$ sub-procedures, computes the average intensities of the pixels on the *i*-th vertical axis of *I*. Again, we point out that the average intensities of a vertical or horizontal axis, containing significant information, are higher than the average intensities of the ones that do not contain significant information. Based on the aforementioned considerations, to properly identify the first vertical axis which contains significant information, the $Compute_{x(W)}$ procedure analyzes the trend of the column-by-column average intensities of *I*. In particular, for $1 \leq i \leq I \cdot width$, given the input DNA microarray image *I*, such a procedure processes the average intensities of the *i*-th

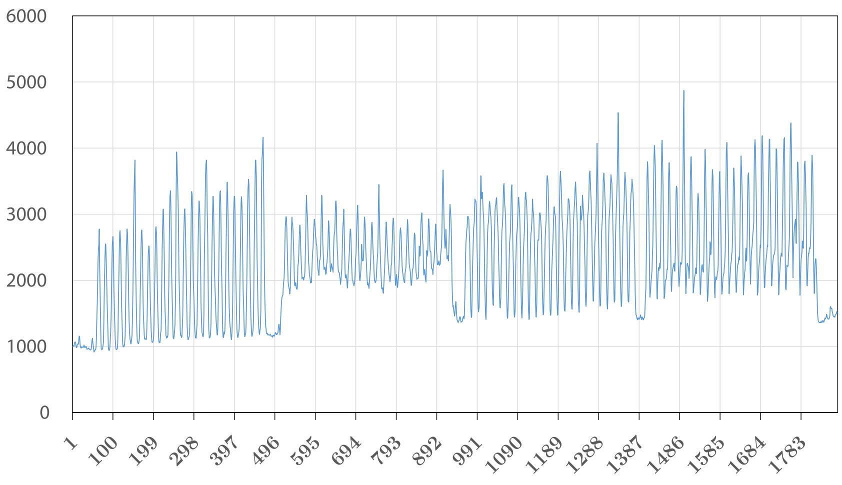


Fig. 9 – Average pixel intensities (column-by-column).

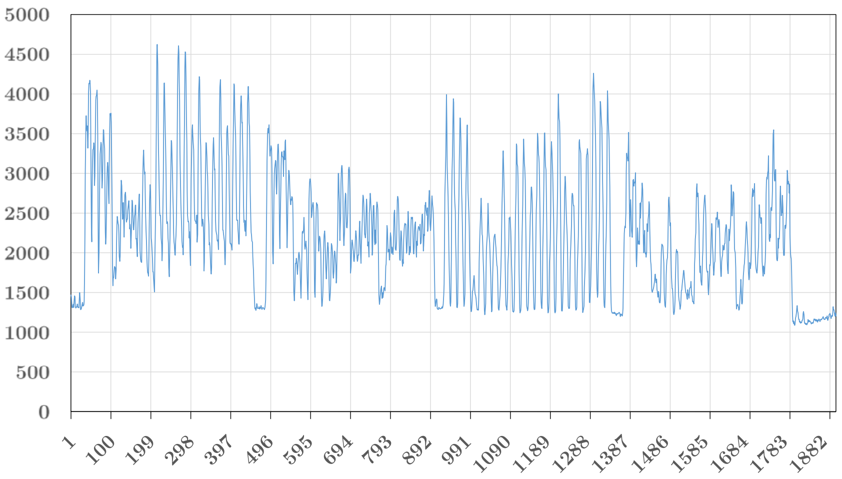


Fig. 10 – Average pixel intensities (row-by-row).

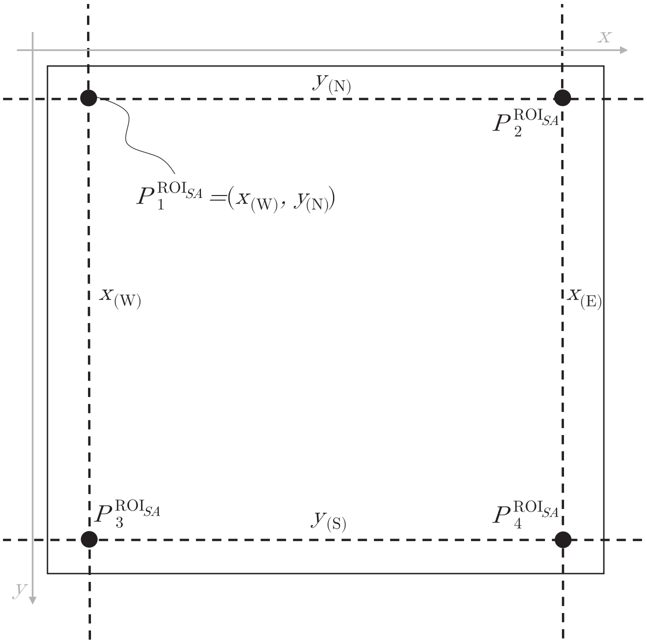


Fig. 11 – Vertical and horizontal axes characterizing the ROI$_{SA}$.

column $x_{(i)}$, as shown in Fig. 12. The *Compute*$_{x(W)}$ procedure returns the index i of the vertical axis in which the average intensities is greater (according to a given threshold) than the mean of the average intensities of a certain number of previous references, denoted as *numOfRefs*. More precisely, the *percentageOfPeak* parameter is used to define the amount, in terms of percentage, according to which the i-th column undergoing processing can be regarded as containing significant information. We stress that the *percentageOfPeak* and *numOfRefs* parameters can be specified by the end-user. Informally speaking, the *Compute*$_{x(W)}$ procedure identifies, through a *left-to-right scanning*, the first *significant* peak which occurs in the trend of the column-by-column average intensities.

Similarly, the *Compute*$_{x(E)}$ procedure analyzes the i-th column, denoted as $x_{(i)}$, concerning the trend of the column-by-column average intensities of I; the main difference is that such a procedure operates from $i = I \cdot width$ to 1 and decrement i at each step. Furthermore, such a procedure considers the subsequent *numOfRefs* references, instead of considering only the previous ones.

The asymptotic time complexity of the *Compute*$_{x(W)}$ procedure is $\mathcal{O}(I \cdot width \times I \cdot height \times numOfRefs)$. In detail, the number of iterations of the outer *for* loop is equal at the most to $I \cdot width - 1$, whereas the number of iterations of the nested *for* loop is equal at the most to *numOfRefs*. Notice that, in the nested *for* loop, the *averageIntensities*$_{VERTICAL}$ sub-procedure is invoked at each iteration. The asymptotic time complexity of this latter sub-procedure is $\mathcal{O}(I \cdot heigth)$, since it processes all the samples of a given column of I. Again, the *averageIntensities*$_{VERTICAL}$ procedure is invoked $\mathcal{O}(I \cdot width \times numOfRefs)$ times. The asymptotic space complexity of the *Compute*$_{x(W)}$ procedure is $\mathcal{O}(1)$, since such procedure, and the sub-procedures it invokes, does not use any structure dependent on the size of the input. Similarly, the *Compute*$_{x(E)}$, *Compute*$_{y(N)}$ and *Compute*$_{y(S)}$ procedures have the same asymptotic time and space complexity as *Compute*$_{x(W)}$. Therefore, the asymptotic time and space complexity of the ROI$_{SA}$*Identification* are $\mathcal{O}(I \cdot width \times I \cdot height \times numOfRefs)$ and $\mathcal{O}(1)$, respectively.

It is important to emphasize that the extraction process is able to identify, in the same manner, the ROI$_{SA}$ from the watermarked DNA microarray image, even when the embedding of the bit string W_{COORDS} has modified some pixel values outside the ROI$_{SA}$. Indeed, the trend of the average intensities results to be very similar, since the variations are not relevant. More precisely, only a sub-set of pixels in the portion of I outside the ROI$_{SA}$ will be affected by the modification of the values. Finally, the values of the modified pixels will be increased or decreased by 1.

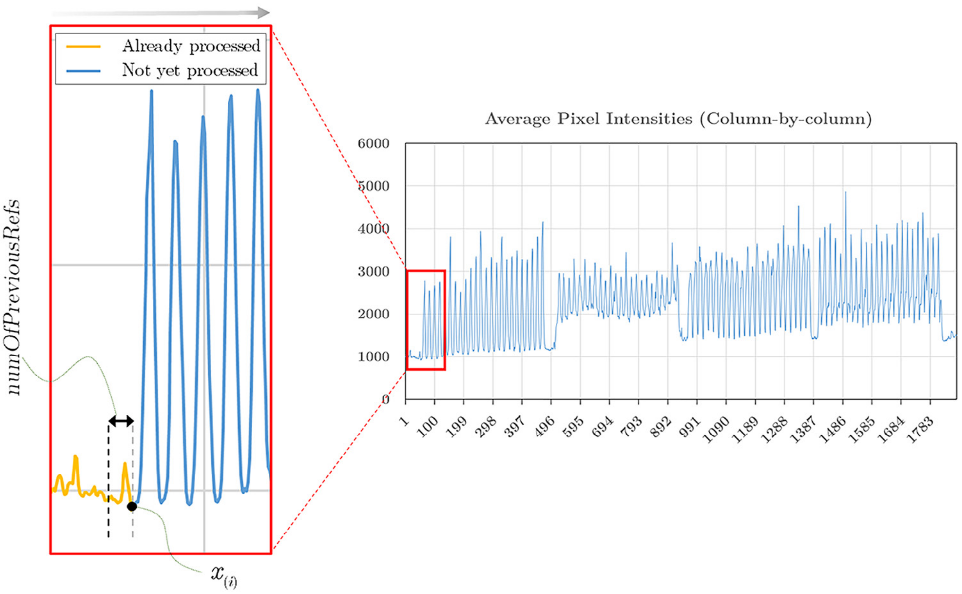


Fig. 12 – Example of the processing concerning the trend of the column-by-column average pixel intensities.

Algorithm 7 The $Compute_{x_{(W)}}$ procedure.

```
procedure COMPUTE$_{x_{(W)}}$($I$, $numOfRefs$, $percentageOfPeak$)
    for $i$ = 2 to $I.width$ do
        numOfEffectiveRefs = 0;
        mean = 0;
        for $k$ = 1 to numOfRefs do
            idx = i - k;
            if $idx \geq 1$ then
                mean = mean + $averageIntensities_{VERTICAL}$($I$, idx);
                numOfEffectiveRefs++;
            end if
        end for
        mean = mean / numOfEffectiveRefs;
        meanPercent = (mean × percentageOfPeak) / 100;
        if $averageIntensities_{VERTICAL}$($I$, i) ≥ mean + meanPercent then
            return i;
        end if
    end for
    return $nil$;
end procedure
```

Algorithm 8 The $Compute_{x_{(E)}}$ procedure.

```
procedure COMPUTE$_{x_{(E)}}$($I$, $numOfRefs$, $percentageOfPeak$)
    $i$ = $I.width$ - 1;
    repeat
        numOfEffectiveRefs = 0;
        mean = 0;
        for $k$ = 1 to numOfRefs do
            idx = i + k;
            if $idx \leq I.width$ then
                mean = mean + $averageIntensities_{VERTICAL}$($I$, idx);
                numOfEffectiveRefs++;
            end if
        end for
        mean = mean / numOfEffectiveRefs;
        meanPercent = (mean × percentageOfPeak) / 100;
        if $averageIntensities_{VERTICAL}$($I$, i) ≥ mean + meanPercent then
            return i;
        end if
        $i - -$;
    until $i \geq 1$
    return $nil$;
end procedure
```

2.2.2. *User-defined ROI selection*

After the identification of the ROI_{SA}, the end-user can select a ROI in which the watermark string W will be embedded. More precisely, a user-defined ROI is identified by four points, i.e., P_1, P_2, P_3 and P_4. In Fig. 13 we show an example of user-defined ROI.

In order to enable the identification of the user-defined ROI by the extraction algorithm, the coordinates of the points P_i, where $1 \leq i \leq 4$, are embedded outside the ROI_{SA}, through our modified scheme. By doing this, the extraction algorithm, after

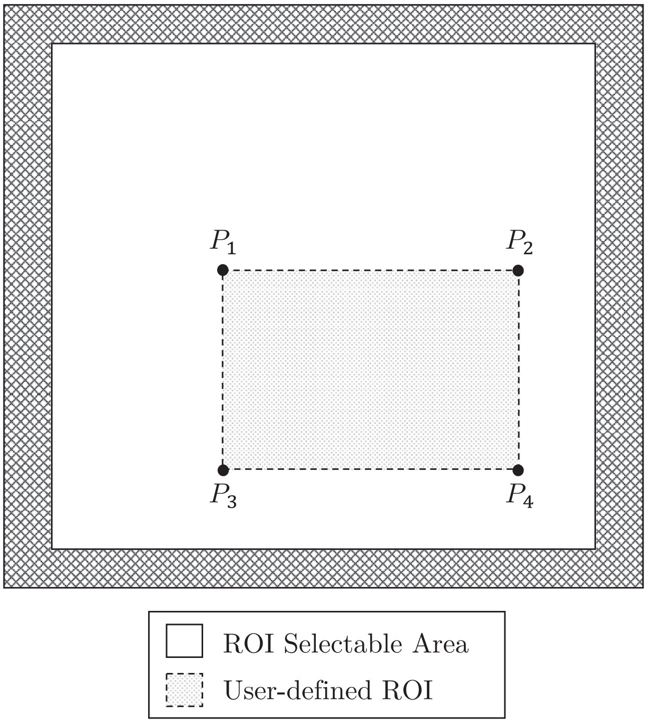


Fig. 13 – User-defined ROI.

the identification of the ROI_{SA}, can extract the points and reconstruct the user-defined ROI. We stress that no significant pixels are modified by such an embedding. In detail, let W_{COORDS} be the bit string that represents the points P_i, which will be embedded outside the ROI_{SA}, and let m be the number of bits used for the representation of a coordinate value. The representation of each point P_i has size $2 \times m$ bits, i.e., m bits for the x-coordinate and m bits for the y-coordinate, whereas W_{COORDS} has size $8 \times m$ bits, i.e., $4 \times (2 \times m)$. Notice that it is also possible to optimize the size of W_{COORDS}, by reducing its size of 50%. In detail, only the $P_1.x$, $P_1.y$, $P_2.x$ and $P_3.y$ coordinate values can be considered, since $P_1 = (P_1 \cdot x, P_1 \cdot y)$, $P_2 = (P_2 \cdot x, P_3 \cdot y)$, $P_3 = (P_1 \cdot x, P_3 \cdot y)$ and $P_4 = (P_2 \cdot x, P_3 \cdot y)$. As a consequence, by considering only such values, the bit string W_{COORDS} has size of $4 \times m$ bits, instead of $8 \times m$. Finally, we employ another bit, which will be used by the extraction process to know if the watermark has been embedded into the user-defined ROI, or outside of it. In detail, if the watermark is embedded into the user-defined ROI, the bit value will be equal to 0; otherwise, the bit value will be equal to 1. Thus, the final size of W_{COORDS} is equal to $4 \times m + 1$ bits. As a final remark, notice that the asymptotic time and space complexity of such optimization is $\mathcal{O}(1)$.

3. Experimental test results and discussion

In this section we describe and discuss the results obtained by evaluating a working prototype of our scheme. In general, we evaluate our scheme with respect to three aspects: *reversibility*, *imperceptibility of the embedded information* and *execution time*. We remark that the whole testing activity has been performed by using a publicly available dataset (Yeast Cell Cycle Analysis Project, 2017). More precisely, we first evaluate the basic version of our scheme by assessing its imperceptibility and reversibility. Afterward, we evaluate the relative ROI-based version, also assessing its performance in terms of execution time. Following an approach similar to the one used in Castiglione et al. (2017), we highlight that, as a testing environment, we used an extremely hardware-constrained device, i.e., the Raspberry Pi, to show the adequacy of our proposal on embedded devices.

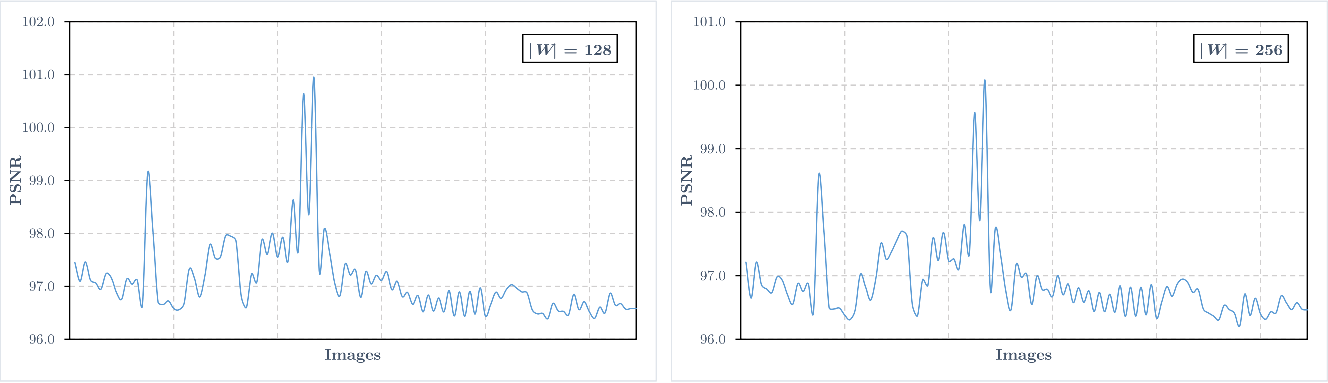
3.1. *Basic version of the proposed watermark scheme*

We evaluate the imperceptibility of the proposed scheme, that is to say, the embedded watermark should not be perceivable. For this reason, we employ the following two metrics: *Peak Signal to Noise Ratio (PSNR)* and *Q Index (QI)* (Wang and Bovik, 2002). The PSNR is a well-established 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, we also employ the QI. The QI ranges from −1 to 1. In particular, the best value for the QI is 1, which means that the compared images are exactly the same, whereas the worst value is −1, which means that the compared images are completely different.

In detail, we focus on test results achieved through several experiments, performed on a dataset composed of 109 DNA microarray images, aimed at assessing the effectiveness of the proposed scheme. In particular, such images come from the dataset referred to as *Yeast* (Yeast Cell Cycle Analysis Project, 2017), where each image is stored in 16-bit *TIFF* format and has resolution of 1024×1024.

In Fig. 14 we show the trend of the PSNR values obtained by comparing the unwatermarked image with respect to those in which a watermark has been embedded. In detail, in Fig. 14a we embed a watermark string of 128 bits, whereas in Fig. 14b we embed a watermark string of 256 bits. In both cases, we set $T = 1500$. Again, in Fig. 15, we follow the same lines followed above, as shown by Fig. 15a and b, but setting $T = 2500$. In detail, on the x-axis, we report the tested images, whereas on the y-axis, we report the PSNR value obtained by comparing the unwatermarked image with respect to the watermarked one. Furthermore, we remark that we achieve values for the QI very close to 1, i.e., around 0.99999997, when the size of W is 128 and $T = 1500$, which means that there are no perceivable differences between the two images.

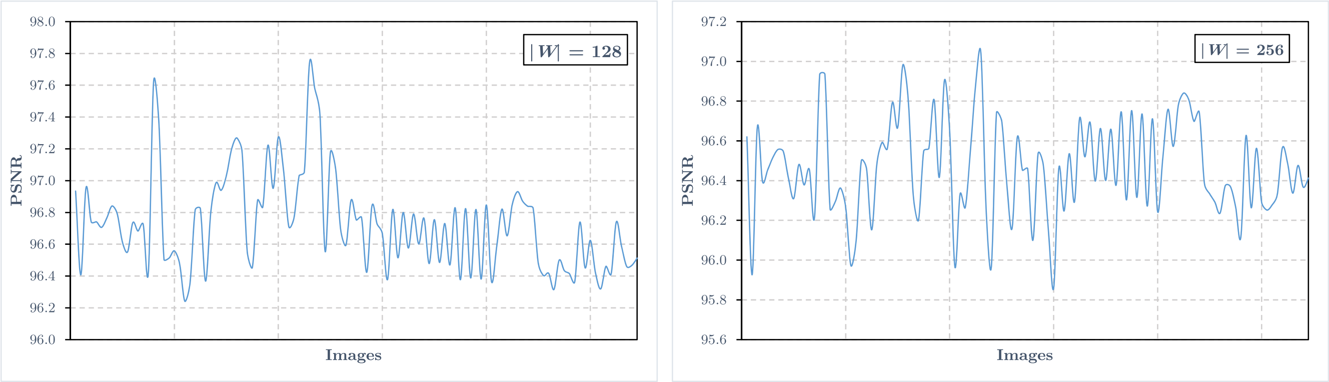
Again, by analyzing the above mentioned figures, it can be noticed that the values assumed by the PSNR are very high. Consequently, such results validate the fact that the watermark



(a) W is composed by 128 bits.

(b) W is composed by 256 bits.

Fig. 14 – Trend of the PSNR values (T = 1500).



(a) W is composed by 128 bits.

(b) W is composed by 256 bits.

Fig. 15 – Trend of the PSNR values ($T = 2500$).

is *not human-perceivable*. For this reason, non-medical consultation and online viewing might be still performed by the end-user, without perceiving any alteration of the image. Finally, as mentioned before, end-users interested in a deeper analysis/processing can recover exactly the original image by extracting the embedded information. In detail, for what concerns this aspect, we verify the *reversibility* of our scheme by comparing each recovered image with the relative unwatermarked one. We stress that in all the experiments carried out, each recovered image is the same as the relative unwatermarked one.

3.2. *ROI-based version of the proposed watermark scheme*

In this subsection we report the performance of the proposed ROI-based watermarking scheme. First, the end-user selects a specific ROI, which is used for all the experiments. Such a ROI has size of 578 × 327 (which covers about 18.03% of the whole image) and is located approximately in the center of the image. We evaluate the average execution time required by the main phases of the embedding and extraction processes, inside and outside the specified ROI, on 10 DNA microarray images of the dataset mentioned in Section 3.1. Such phases are: the *identification of the* ROI$_{\text{SA}}$, the *embedding/extraction of* W$_{\text{COORDS}}$ and the *embedding/extraction of* W. The prototype of our scheme is a *Java-based* application, which can be run on several heterogeneous hardware and software environments. In particular, in our experiments, we considered three testing environments. For what concerns such environments, the most important thing to note is that one of them is based on the *Raspberry PI B Plus*, which is a credit card-sized single-board computer with constrained hardware capabilities. In Table 1 we report, for each environment, the average execution time (in ms) required for embedding the watermark string into the user-defined ROI of 10 images. In detail, in the 7th and 12th rows, we report the average execution time regarding the ROI$_{\text{SA}}$ identification, whereas in the 8th and 13th rows, we report the average execution time concerning the embedding/extraction of W$_{\text{COORDS}}$. Again, in the 9th and 14th rows, we report the average execution time concerning the embedding/extraction of W. Furthermore, in the 10th and 15th rows, we report, for each testing environment, the average total execution time required by the embedding and extraction/recovery phases. Similar to Table 1, in Table 2 we report the average execution time, taken on 10 images, when the embedding/extraction is performed outside the user-defined ROI. We emphasize that the results are achieved by using the following parameters: $K = 23456$, $numOfRefs = 5$, $percentageOfPeak = 25$, $m = 11$ and W composed by 128 bits, respectively.

Figs. 16 and 17 show the percentage of execution time relative to the embedding and extraction processes in the user-defined ROI, respectively. From such figures, it can be observed that the average execution time concerning the identification of the ROI$_{\text{SA}}$ is less than 5% (ranging from 3% to 4%) of the overall execution time. Moreover, the average execution time of the embedding/extraction of W$_{\text{COORDS}}$ is around 30%, when W is embedded into the ROI, and 35%, when W is embedded outside the ROI. We emphasize that the average time for the embedding/extraction of W takes from 60% up to 68% of the

Table 1 – Average execution time for the embedding and extraction processes using the reported testing environments. The entries are reported in milliseconds (ms). W is set to be embedded inside the ROI.

Testing environments			
CPU	Intel Core i5 4200M	Intel Atom Z3735G	RaspBerry PI B +
RAM	8 GB (DDR3L)	1 GB (DDR3L)	
Memory space	1000 GB	16 GB (eMMC)	
OS	Windows 10 Home	Windows 10 Home	Raspbian Jessie
Embedding			
ROI$_{\text{SA}}$Idetintification	10	103	494
Embedding (I) – 45 bits	138	857	4552
Embedding (II) – 128 bits	537	2142	9756
Total	685	3102	14802
Extraction and recovery			
ROI$_{\text{SA}}$Idetintification	12	97	444
Extraction (I) – 45 bits	137	879	4187
Extraction (II) – 128 bits	519	2206	9275
Total	668	3182	13906

Table 2 – Average execution time for the embedding and extraction processes using the reported testing environments. The entries are reported in milliseconds (ms). *W* is set to be embedded outside the ROI.

Testing environments			
CPU	Intel Core i5 4200M	Intel Atom Z3735G	RaspBerry PI B +
RAM	8 GB (DDR3L)	1 GB (DDR3L)	
Memory space	1000 GB	16 GB (eMMC)	
OS	Windows 10 Home	Windows 10 Home	Raspbian Jessie
Embedding			
ROI$_{SA}$Idetintification	12	111	497
Embedding (I) – 45 bits	103	855	4206
Embedding (II) – 128 bits	236	1236	7732
Total	351	2202	12435
Extraction and recovery			
ROI$_{SA}$Idetintification	6	97	437
Extraction (I) – 45 bits	114	852	4165
Extraction (II) – 128 bits	217	1213	7276
Total	337	2162	11878

overall execution time. Finally, as done in Section 3.1, the reversibility of the scheme has been successfully assessed.

4. Conclusions and future research perspectives

The *DNA microarray imaging technology* represents one of the most important components in the field of genomic analysis, which can be relied on for storing, managing, sharing and exchanging genomic data. However such data may still present a lot of risks (Nordgren and Juengst, 2009), mainly when we consider the security implications of their adoption in IoLT context. Indeed, commonly employed techniques for data protection, such as encryption (e.g., the approach proposed in Ogiela and Ogiela (2010)) or the use of metadata into the image header, are doomed to fail when dealing with DNA microarray images in complex scenarios. Accordingly, we presented an invisible fragile watermarking scheme, explicitly addressed for DNA microarray images, which can be used to protect such images in a reversible manner, so that the original image can

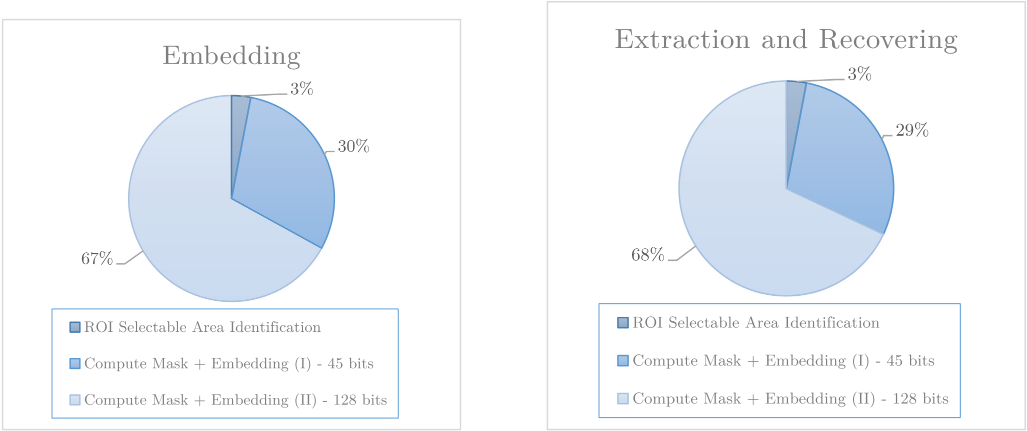


Fig. 16 – Percentage of the execution time required for each phase relative to Table 1.

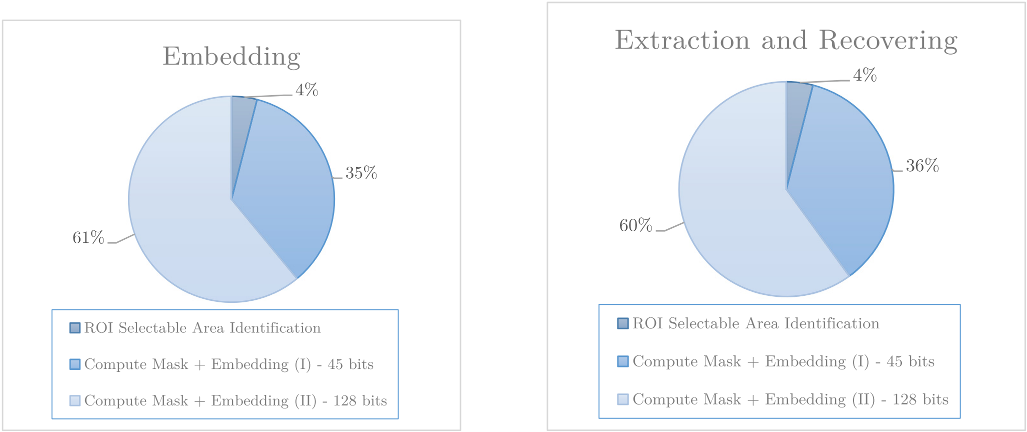


Fig. 17 – Percentage of the execution time required for each phase relative to Table 2.

be completely restored upon the extraction of the embedded information. Moreover, we extended the above mentioned scheme to enable the embedding of the watermark string into a user-defined ROI. Finally, test results proved the effectiveness of our scheme, besides showing its efficiency, even on devices characterized by constrained hardware and software capabilities. We emphasize that this confirm the applicability of our proposal directly within on-board miniaturized sensors.

In future works we intend to improve our ROI-based watermarking scheme by considering further and more complex techniques for the ROI selections. Again, we plan to consider the possibility of allowing the end-user to select more than one ROI. Finally, we intend to take into consideration other geometrical shapes for characterizing the ROI, as for example complex polygons.

REFERENCES

Al-Qershi OM, Khoo BE. Authentication and data hiding using a hybrid ROI-based watermarking scheme for DICOM images. J Digit Imaging 2011;24(1):114–25.

Alam Q, Malik SUR, Akhunzada A, Choo KKR, Tabbasum S, Alam M. A cross tenant access control (CTAC) model for cloud computing: formal specification and verification. IEEE Trans Inf Forensics Security 2017;12(6):1259–68. doi:10.1109/TIFS.2016.2646639.

Albano P, Bruno A, Carpentieri B, Castiglione A, Castiglione A, Palmieri F, et al. A secure distributed video surveillance system based on portable devices. In: International conference on availability, reliability, and security. Springer; 2012. p. 403–15.

Albano P, Bruno A, Carpentieri B, Castiglione A, Castiglione A, Palmieri F, et al. Secure and distributed video surveillance via portable devices. J Ambient Intell Hum. Comp. 2014;5(2):205–13.

Andrews JG, Buzzi S, Choi W, Hanly SV, Lozano A, Soong AC, et al. What will 5G be? IEEE J Sel Areas Commun 2014;32(6):1065–82.

Barni M, Bartolini F. Watermarking systems engineering: enabling digital assets security and other applications. CRC Press; 2004.

Bertoni G, Daemen J, Peeters M, Assche GV. Keccak. In: Johansson T, Nguyen PQ, editors. Advances in cryptology – EUROCRYPT 2013, 32nd annual international conference on the theory and applications of cryptographic techniques, Athens, Greece, May 26–30, 2013. Proceedings, vol. 7881 of Lecture Notes in Computer Science. Springer; 2013. p. 313–14.

Bhatt C, Dey N, Ashour AS. Internet of things and big data technologies for next generation healthcare. 2017.

Burke AJ. How “cloud” services democratize DNA sequencing. 2017. Available from: http://techonomy.com/2012/08/how-cloud-services-democratize-dna-sequencing/.

Caldelli R, Filippini F, Becarelli R. Reversible watermarking techniques: an overview and a classification. EURASIP J Inf Sec. 2010;2010:2.

Castiglione A, De Santis A, Masucci B. Hierarchical and shared key assignment. In: Barolli L, Xhafa F, Takizawa M, Enokido T, Castiglione A, De Santis A, editors. 17th international conference on network-based information systems (NBiS 2014). Salerno, Italy: IEEE Computer Society; September 10–12, 2014. p. 263–70.

Castiglione A, De Santis A, Pizzolante R, Castiglione A, Loia V, Palmieri F. On the protection of fMRI images in multi-domain environments. In: Barolli L, Takizawa M, Xhafa F, Enokido T, Park JH, editors. 29th IEEE international conference on advanced information networking and applications, AINA 2015. Gwangju, South Korea: IEEE Computer Society; March 24–27, 2015. p. 476–81.

Castiglione A, Pizzolante R, De Santis A, Carpentieri B, Castiglione A, Palmieri F. Cloud-based adaptive compression and secure management services for 3D healthcare data. Future Generation Comp. Syst 2015;43–44:120–34.

Castiglione A, De Santis A, Masucci B. Key indistinguishability versus strong key indistinguishability for hierarchical key assignment schemes. IEEE Trans Dependable Sec Comp. 2016;13(4):451–60.

Castiglione A, De Santis A, Masucci B, Palmieri F, Castiglione A, Huang X. Cryptographic hierarchical access control for dynamic structures. IEEE Trans Inf Forensics Security 2016;11(10).

Castiglione A, De Santis A, Masucci B, Palmieri F, Castiglione A, Li J, et al. Hierarchical and shared access control. IEEE Trans Inf Forensics Security 2016;11(4):850–65.

Castiglione A, Pizzolante R, Palmieri F, Masucci B, Carpentieri B, De Santis A, et al. On-board format-independent security of functional magnetic resonance images. ACM Trans Embed Comput Syst 2017;16(2):56:1–56:15.

Clark L. Oxford Nanopore: we want to create the internet of living things. 2017. Available from: http://www.wired.co.uk/article/clive-brown-oxford-nanopore-technologies-wired-health-2015.

Coatrieux G, Le Guillou C, Cauvin JM, Roux C. Reversible watermarking for knowledge digest embedding and reliability control in medical images. IEEE Trans Inf Technol Biomed 2009;13(2):158–65.

Erlich Y. A vision for ubiquitous sequencing. Genome Res 2015;25(10):1411–16.

Feng JB, Lin IC, Tsai CS, Chu YP. Reversible watermarking: current status and key issues. Int J Net. Security 2006;2(3):161–70.

Guo C, Zhuang R, Jie Y, Ren Y, Wu T, Choo KKR. Fine-grained database field search using attribute-based encryption for E-healthcare clouds. J Med Syst 2016;40(11):235. doi:10.1007/s10916-016-0588-0. http://dx.doi.org/10.1007/s10916-016-0588-0.

He D, Kumar N, Wang H, Wang L, Choo KKR, Vinel A. A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network. IEEE Trans Dependable Secure Comput 2016;PP(99):1. doi:10.1109/TDSC.2016.2596286.

Jain AN, Tokuyasu TA, Snijders AM, Segraves R, Albertson DG, Pinkel D. Fully automatic quantification of microarray image data. Genome Res 2002;12(2):325–32.

Lee S, Yoo CD, Kalker T. Reversible image watermarking based on integer-to-integer wavelet transform. IEEE Trans Inf Forensics Security 2007;2(3):321–30.

Liu Z, Choo KKR, Zhao M. Practical-oriented protocols for privacy-preserving outsourced big data analysis: challenges and future research directions. Comp. Security 2016; Available from: http://www.sciencedirect.com/science/article/pii/S0167404816301778, http://dx.doi.org/10.1016/j.cose.2016.12.006.

Lonardi S, Luo Y. Gridding and compression of microarray images. In: Proceedings of the computational systems bioinformatics conference, 2004 (CSB 2004). IEEE; 2004. p. 122–30.

Medeiros J. DNA analysis will build an internet of living things; 2017. Available from: http://www.wired.co.uk/article/dna-analysis-internet-living-things.

Murray JF. Personalized medicine: been there, done that, always needs work! 2012.

Nordgren A, Juengst ET. Can genomics tell me who I am? Essentialistic rhetoric in direct-to-consumer DNA testing. New Genet Soc 2009;28(2):157–72.

O'Driscoll A, Daugelaite J, Sleator RD. "Big data", Hadoop and cloud computing in genomics. J Biomed Inform 2013;46(5):774–81.

Ogiela MR, Ogiela U. Grammar encoding in DNA-like secret sharing infrastructure. In: Advances in computer science and information technology. Springer; 2010. p. 175–82.

Pacheco J, Hariri S. IoT security framework for smart cyber infrastructures. In: IEEE international workshops on foundations and applications of self* systems. IEEE; 2016. p. 242–7.

Phillips AM. Only a click away – DTC genetics for ancestry, health, love… and more: a view of the business and regulatory landscape. Appl Transl Genomics 2016;8:16–22.

Pizzolante R, Carpentieri B. Lossless, low-complexity, compression of three-dimensional volumetric medical images via linear prediction. In: 2013 18th international conference on digital signal processing (DSP). IEEE; 2013. p. 1–6.

Pizzolante R, Carpentieri B, Castiglione A, De Maio G. The avq algorithm: watermarking and compression performances. In: 2011 third international conference on intelligent networking and collaborative systems (INCoS). IEEE; 2011. p. 698–702.

Pizzolante R, Carpentieri B, Castiglione A. A secure low complexity approach for compression and transmission of 3-D medical images. In: 2013 eighth international conference on broadband and wireless computing, communication and applications, Compiegne, France. IEEE; October 28–30, 2013. p. 387–92.

Pizzolante R, Carpentieri B, Castiglione A, Castiglione A, Palmieri F. Text compression and encryption through smart devices for mobile communication. In: 2013 seventh international conference on innovative mobile and internet services in ubiquitous computing (IMIS). IEEE; 2013. p. 672–7.

Pizzolante R, Castiglione A, Carpentieri B, De Santis A, Castiglione A. Protection of microscopy images through digital watermarking techniques. In: Xhafa F, Barolli L, Palmieri F, Koeppen M, Loia V, editors. 2014 international conference on intelligent networking and collaborative systems. Salerno, Italy: IEEE; September 10–12, 2014. p. 65–72.

Rahman F, Bhuiyan MZA, Ahamed SI. A privacy preserving framework for RFID based healthcare systems. Future Generation Comp. Syst 2017;72:339–52.

Rahman MS, Basu A, Kiyomoto S, Bhuiyan MA. Privacy-friendly secure bidding for smart grid demand-response. Inf Sci (Ny) 2017;379:229–40.

Son J, Kim D, Bhuiyan MZA, Hussain R, Oh H. A new outsourcing conditional proxy re-encryption suitable for mobile cloud environment. Concurrency Comp. Pract Exp 2016;1–16.

Swan M. Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0. J Sens Actuator Netw 2012;1(3):217–53.

Wakatani A. Digital watermarking for ROI medical images by using compressed signature image. In: Proceedings of the 35th annual Hawaii international conference on system sciences (HICSS); 2002. p. 2043–8.

Waltz E. Portable DNA sequencer MinION helps build the internet of living things. 2017. Available from: http://spectrum.ieee.org/the-human-os/biomedical/devices/portable-dna-sequencer-minion-help-build-the-internet-of-living-things.

Wang Z, Bovik AC. A universal image quality index. IEEE Signal Processing Lett 2002;9(3):81–4.

Wang Z, Bovik AC, Lu L. Why is image quality assessment so difficult? In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP 2002). Orlando, Florida, USA: IEEE; May 13–17, 2002. p. 3313–16.

West DM. How 5G technology enables the health internet of things. Brookings Center for Technology Innovation; 2016. p. 3.

Xiong H, Choo KKR, Vasilakos AV. Revocable identity-based access control for big data with verifiable outsourced computing. IEEE Trans Big Data 2017;PP(99):1. doi:10.1109/TBDATA.2017.2697448.

Yeast Cell Cycle Analysis Project. Yeast microarray image set; 2017. Available from: http://genome-www.stanford.edu/cellcycle/data/rawdata/individual.html.

Raffaele Pizzolante Ph.D. is a research fellow at University of Salerno (Italy). He received the Laurea and Laurea Specialistica degrees cum laude in Computer Science from University of Salerno (Italy). In 2015 he received the Ph.D. degree in Computer Science from University of Salerno (Italy). He regularly serves as a peer reviewer for scientific conferences and journals and is also active in writing and publishing. His research interests include Data Compression, Image Processing, Digital Watermarking and Information Hiding.

Arcangelo Castiglione received the B.S., M.S. and Ph.D. degrees in computer science from the University of Salerno. Currently he is a post-doctoral fellow with the Department of Computer Science, University of Salerno. His research mainly focuses on cryptography, multimedia data protection and network security. He is associate editor of an international journal (Journal of High Speed Networks), guest editor for several special issues and volume editor for international conference proceedings. He is a member of various program committees for international conferences and reviewer for several scientific journals and conferences.

Bruno Carpentieri received the "Laurea" degree in Computer Science from the University of Salerno, Salerno, Italy, and the M.A. and Ph.D. degrees in Computer Science from the Brandeis University, Waltham, MA, U.S.A. Since 1991, he has been first Assistant Professor and then Associate Professor of Computer Science at the University of Salerno (Italy). His research interests include lossless and lossy image compression, video compression and motion estimation, information hiding. He has been, from 2002 to 2008, Associate Editor of the journal IEEE Trans. on Image Processing. He was recently chair and organizer of the International Conference on Data Compression, Communication and Processing 2011, co-chair of the International Conference on Compression and Complexity of Sequences, and, for many years, program committee member of the IEEE Data Compression Conference and of other international Conferences in the field. He has been responsible for various European Commission contracts regarding image and video compression.

Alfredo De Santis received the degree in computer science from the University of Salerno. Since 1984, he has been with the Dipartimento di Informatica, University of Salerno. Since 1990, he has been a Professor of Computer Science. From 1991 to 1995 and from 1998 to 2001, he was the Chairman of the Dipartimento di Informatica ed Applicazioni, University of Salerno. Since 2015, he has been the Chairman with the Dipartimento di Informatica, University of Salerno. He was the Chairman of the Graduate Program in Computer Science with the University of Salerno: ciclo XII (1996–2000), ciclo XIII (1997–2001), ciclo XIV (1998–2002), ciclo XV (1999–2002), and ciclo XVI (2000–2003). He was the Chairman of the Graduate Program in Computer Science and Information Engineering with the University of Salerno: ciclo XXIX (2013–2016) and ciclo XXX (2014–2017). From 1987 to 1990, he was a Visiting Scientist with the IBM T. J. Watson Research Center, Yorktown Heights, NY. He was a Visiting Scientist with the International Computer Science Institute, Berkeley CA, USA, in 1994. He is an Associate Editor of Applied Soft Computing. He was an Associate Editor of the IEEE Transactions on Information Forensics and Security. From 2009 to

2012, he was a member of the Board of Directors (Consiglio di Amministrazione) of Consortium Garr (Gestione Ampliamento delle Reti di Ricerca) for the management of the communication network of the public research in Italy. From 2011 to 2013, he was a member of the Group of Experts for the Evaluation Area 01 (mathematics and computer science), selected by the National Agency for the Evaluation of Universities and Research Institutes, for the Italian research assessment from 2004 to 2010 (VQR 2004–2010). His research interests include algorithms, data security, digital forensics, cryptography, communication networks, information theory, and data compression.

Francesco Palmieri received the M.S. and Ph.D. degrees in computer science from the University of Salerno. Currently, he is an Associate Professor with the University of Salerno. He was an Assistant Professor with the Second University of Napoli. He was the Director of the Networking Division with the Federico II University of Napoli and contributed to the development of the Internet in Italy as a Senior Member of the Technical-Scientific Advisory Committee and of the CSIRT of the Italian NREN GARR. His research interests include advanced networking protocols and architectures and network security. He serves as the Editor-in-Chief of an international journal and participates on the editorial board of other ones.

Aniello Castiglione received the Ph.D. degree in computer science from the University of Salerno, Italy. Actually, he is an Adjunct Professor with the University of Salerno, Italy, and University of Naples Federico II, Italy. He received the Italian national habilitation as an Associate Professor of Computer Science. He serves as a Reviewer for several international journals and is a Managing Editor of two international journals. He also acts as a reviewer for around 50 international journals. He served as a Program Chair and TPC Member of around 90 international conferences. He acted as a Guest Editor several journals and serves as an Editor of several editorial boards of international journals. He has authored more than 130 papers in international journals and conferences. One of his papers has been selected as Featured Article in the IEEE Cybersecurity initiative. He has been involved in forensic investigations, collaborating with several Law Enforcement Agencies as a Consultant. He is a member of several associations, including the ACM. His current research interests include information forensics, digital forensics, security and privacy on cloud, communication networks, and applied cryptography.