Iris Biometrics for Embedded Systems

Judith Liu-Jimenez, *Student Member, IEEE*, Raul Sanchez-Reillo, *Member, IEEE*, and Belen Fernandez-Saavedra, *Student Member, IEEE*

***Abstract*—In many applications user authentication has to be carried out by portable devices. Usually these devices are personal tokens carried by users, which have many constraints regarding their computational performance, occupied area, and power consumption. These kinds of devices must deal with such constraints, while also maintaining high performance rates in the authentication process. This paper provides solutions to designing such personal tokens where biometric authentication is required. In this paper, iris biometrics have been chosen to be implemented due to the low error rates and the robustness their algorithms provide. Several design alternatives are presented, and their analyses are reported. With these results, most of the needs required for the development of an innovative identification product are covered. Results indicate that the architectures proposed herein are faster (up to 20 times), and are capable of obtaining error rates equivalent to those based on computer solutions. Simultaneously, the security and cost for large quantities are also improved.**

***Index Terms*—Authentication, embedded systems, hamming distance, image processing, iris biometrics, segmentation.**

I. Introduction

BIOMETRICS is the only method capable of recognizing human beings using the real features of the user instead of his or her knowledge (e.g., passwords) or belongings (e.g., a magnetic stripe card) [1]. Among currently existing biometric modalities, iris recognition is considered to be one of the most secure and reliable technologies [2], [4], [6], [5]; however, while matching algorithms in iris recognition are straightforward, the signal processing prior to matching requires a significant amount of processing power.

Biometric applications can be classified into two major groups: identification and authentication. Identification is performed when the user identity is not provided, wherein the system must find the user from a database of biometric data from all enrolled users. In contrast, authentication, is the process of checking the identity of the user using provided biometric data. Currently, both applications are ubiquitously used; however, this paper will focus on authentication, as this application is where personal tokens play an important role.

Biometric authentication applications can be designed by following two key approaches [7]: online, which requires communication with central databases to access biometric data and offline, wherein biometric data is stored on personal tokens. The online approach must deal with serious security and privacy issues, as the communication between the system and the central database can be attacked, and the identity may be stolen or altered. For this reason, offline systems are recommended, as long as the data is kept securely in the personal token.

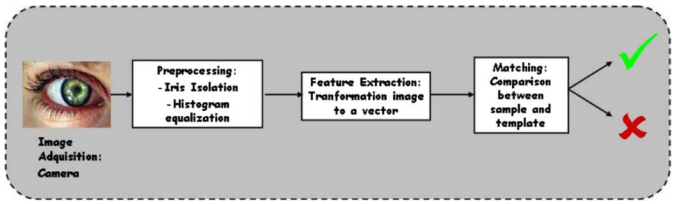

Fig. 1. Block diagram of a biometric system.

Two different strategies are followed in offline biometric authentication systems: 1) the token provides the biometric template and 2) the token performs the verification tasks and supplies the result, avoiding external access to the user's personal template. This paper recommends the second strategy for security and privacy motivations. Thus, different architecture approaches to build personal tokens will be described. These tokens are designed as tamper-proof devices, maintaining not only internal data security, but also a secure communications channel with the external world. After initial results by the authors [7], where only the matching part of the algorithm was included in the token (Match-on-Card technology), this paper will develop the work that will be included in the token to improve parts of the algorithm, achieving a higher level of simplicity in the verification devices.

In order to show this, Section II will introduce iris biometrics technology, and will discuss the state of the art for this modality. The implementations realized in this work will be described in the following section. For this purpose, Section III will briefly describe the selected algorithm, followed by details of the implementation of two potential platforms. Section IV presents the results obtained from these platforms, including a performance comparison, processing time, and the hardware area required for each platform. This paper concludes and discusses possible future work in this area.

II. State of the Art in Iris Biometrics

From a conceptual point of view, most iris recognition systems have the same block diagram as any other biometric modality (see Fig. 1). After capturing an image of the eye, the iris is located and segmented to extract its features; these features are then compared to a previously stored template [6]. This section describes each of these blocks in detail, providing information on the approaches found in previous publications.

Manuscript received November 03, 2008; revised June 04, 2009. First published November 20, 2009; current version published January 21, 2011. This work was supported by the Spanish Ministry of Science and Education (TEC2006-12365).

The authors are with the Department of Electronic Technology, University Carlos III of Madrid, 28911 Spain (e-mail: jliu@ing.uc3m.es; rsreillo@ing.uc3m.es; mbfernan@ing.uc3m.es).

Digital Object Identifier 10.1109/TVLSI.2009.2033701

A. Iris Acquisition

Contrary to popular belief, iris biometrics systems do not use laser-scans to capture the image of the human eye. Instead, an infrared photo or video camera is used at a set distance to capture a high quality image of the iris. Working in the infrared range provides many advantages when compared to the visible range: iris ridges, nerves, and crypts are more evident [31]; the border between the iris and the pupil is more pronounced; and users are not exposed to annoying flashes.

Currently, most of the work performed in this area has been dedicated to improving user-system interaction by developing cameras where the focusing system is automatic, such that users are not required to remain steady at a fixed point in front of the camera [8]–[10].

B. Iris Segmentation

The main purpose of this process is to locate the iris on the image and isolate it from the rest of the eye image for further processing. Some other important tasks that are also performed in this iris segmentation block include image quality enhancement, noise reduction, and emphasis of the ridges of the iris.

Several proposals have been made by different authors for iris location and segmentation, wherein most consider iris detection as finding two circumferences that model the iris boundaries. Daugman [11] has proposed an integro-differencial operator, which works by examining the difference in pixel levels between circles drawn in the image. Sanchez-Avila *et al.* [12] have used a similar operator, but search for the maximum difference in lines drawn crossing the entire image. Other authors [13]–[16] use the Hough transform for circle detection.

Recently, Daugman has proposed a new method for seeking the iris boundary by using active contour models [19]. Here, the iris location varies depending on preset external and internal forces until an equilibrium state is reached. Similar solutions have also been used by Ritter in [20] and Ross *et al.* in [21].

C. Feature Extraction

In the feature extraction block, different authors have presented a wide variety of proposals. The majority of these begin with a normalization of the segmented iris image. This normalization becomes necessary when considering that the pupil varies in size for different light intensities. The normalization method varies from changes to the polar coordinate system, as Daugman [11] proposed, to only considering a virtual line drawn around the pupil, known as the iris signature [12].

After normalization, Daugman has studied the phase information by applying different Gabor filters. This was followed by the codification of this information in terms of the quadrant where the phase belongs [11]; however, Wildes, performs the extraction using Laplacian or Gaussian filters by obtaining several images of different scales for posterior comparison [14]. Sanchez–Avila *et al.* have proposed in [12] two different feature extraction approaches: one using Gabor filters weighting for small portions of the segmented iris image and another one based on the use of dyadic wavelet transformations and their zero-crossing representation. Li Ma *et al.* [13] have proposed a similar approach, but applies the dyadic wavelet transformation on a 1-D intensity signal instead of the iris signature approach used by Sanchez–Avila *et al.* Boles *et al.* [22] have also based their proposal on the dyadic wavelet transform, but on a normalized iris image (as proposed by Daugman), i.e., by using a 2-D wavelet transform on the polar scale representation of the iris, as opposed to the two previous algorithms that work in 1-D.

D. Matching

Although some authors have studied other matching algorithms [12], [14], [15], the most employed matching algorithm has been the Hamming distance, as was initially proposed by Daugman [11]. The Hamming distance is described by the following equation:

$$HD(A,B) = \frac{1}{L}\sum_{i=0}^{L}(p_i \otimes y_i) \tag{1}$$

where L is the vector length and p_i and y_i are the ith component of the template and sample vector, respectively, which are XORed in the equation. If the distance obtained is below a predefined threshold level, the studied sample is considered to belong to the user whose template is being studied. Selection of the threshold level usually depends on the final application.

III. Implementation

Previous studies have shown the viability of creating match-on-token solutions by including the comparison algorithm within the token, providing an answer that deals with the matching result [7]. In this paper, these studies have been extended to analyze the viability of integrating the feature extraction block within the personal token. With this solution, simplification of the point of service terminal is achieved, and security is improved.

The terminal, in the proposed architecture, should perform the following tasks.

- *Image Acquisition:* The iris is captured with an infrared camera, as previously mentioned. The cost and size of the electronics and lens required for this task are not commercially viable for insertion into the personal token.
- *Image Segmentation:* This preprocessing block is related to the image acquisition. The non-detection of the iris or the quality of the captured images are typical reasons for rejection of the acquired image, thus, requiring a new capture process. If this block were included in the token, many images would have to be transferred from the terminal to the token, increasing data communication and therefore the verification time.

The personal token should have the following characteristics.

- It should perform the rest of the biometric processes, i.e., feature extraction, comparison, and the matching result processing.
- It is highly recommended to be reconfigurable. Possible token robberies or user accidents would require changes in biometric data or internal token processes to avoid security holes.

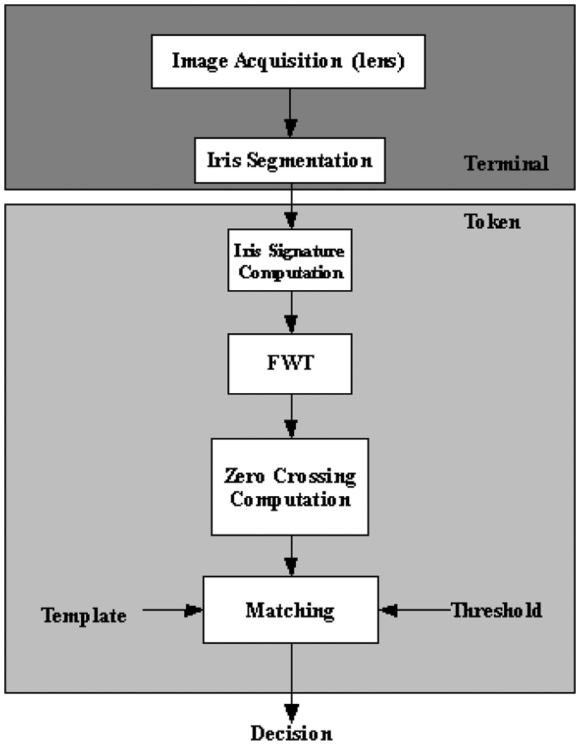

Fig. 2. Terminal and platform functionalities.

- The token should be able to build and handle a secure communication channel with the terminal.
- The token should be designed as a tamper-proof device.
- As it has to be portable, the occupied electronic area should be as small as possible.
- Although token size is limited, the processing time must be minimal to reduce user waiting time.
- Finally, the device must be cost effective, as large quantities of these devices will be manufactured.

In order to study different implementation proposals, this section is organized as follows. First, we will center on the chosen algorithm from a signal processing viewpoint. This is followed by the different implementations developed.

A. Chosen Algorithm

Iris Segmentation: The iris chosen segmentation algorithm is based on the Hough transform as proposed in [14]. This transformation works on the edge image, as information related to textures or colors is not considered. From all the different edge operators, the Canny method is used due to the shape of the edge to be detected [30]. In this application, the Hough transform is used considering a circular shape to detect the iris boundary within the sclera. Once the iris boundary is detected, the pupil boundary is found in a similar way. Finally, eyelid detection is carried out by using a separate Hough transform for elliptical figures. Reflections are eliminated during the first stages of the preprocessing block through erosion and dilation operators. These reflections have to be removed as soon as possible, as they can cause erroneous decisions in the circle detection. After the iris segmentation has been finished, a quality algorithm is applied [23] and rejects images for the following reasons:

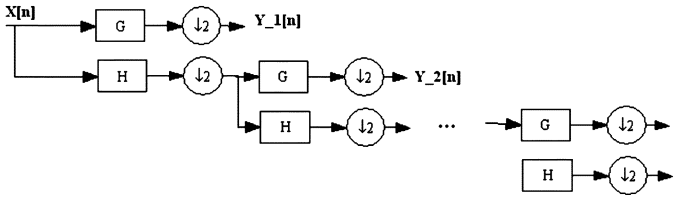

Fig. 3. FWT. The wavelet transform is computed by a hierarchical structure of filter applications. Filters G and H define which wavelet functions are used.

- low quality images;
- distance between the pupil and the iris is not higher than a percentage of the iris radius; if such distance is lower than the specified threshold, the information contained in the segmented iris is not considered to be enough to obtain a reliable feature vector;
- pupil parts are not detected within the iris area;
- eccentricity between the pupil and iris circumferences is above a certain threshold.

Feature Extraction: In this paper, the feature extraction method proposed by Sanchez–Avila *et al.* in [12] was selected. Here, the iris is first normalized to a virtual circle around the pupil, which is named the iris signature. Thus, the iris signature will represent the gray level values on the contour of a virtual circle, which is centered at the centroid of the pupil, with a fixed radium r and considering angular increments of $2\pi/L_s$, being $L_s = 256$, the length of the iris signature (previously fixed); i.e.,

$$I_S = I_E(x_c + r\cos\theta, y_c + r\sin\theta) \tag{2}$$

being $2n\pi/L_s \leq \theta \leq 2(n+1)\pi/L_s$, $n\epsilon\mathbb{N} \cup \{0\}$, r a fixed radium, and (x_c, y_c) the centroid of the pupil.

Afterwards, a 1-D dyadic wavelet transform is applied to the iris signature. The vector resulting for each scale is concatenated in a unique vector for computation of the zero-crossing representation, which leads to the feature vector. For computation of the wavelet transformation, Mallat's fast wavelet transform (FWT) approach has been considered [24].

Unfortunately, the down-sampling stage that follows each filter provides the worst performance results when compared to a zero-insertion in intermediate vectors. Therefore, the resulting vector does not have the same length as the initial vector, wherein its length is the initial length multiplied by the number of levels the transformation performs.

Once the dyadic wavelet is computed, the resulting vector is simplified by using its zero-crossing representation. The zero-crossing representation converts the vector into a binary representation, wherein "1" represents a positive value and "0" represents a negative value, for each vector component.

The number of levels used for the wavelet transform is a critical parameter, as it greatly influences the authentication performance results. Eight levels are used in this paper, as a tradeoff among processing time, hardware area, and vector size.

Matching: As already mentioned, the Hamming distance is used for comparing vectors. This distance, as shown in [12], offers the best results for this biometric modality and the selected algorithm.

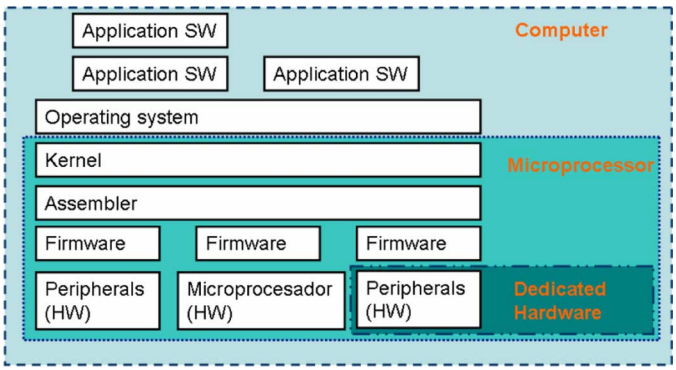

Fig. 4. Abstraction layers in a computer system and its relationship for biometric system implementation platforms.

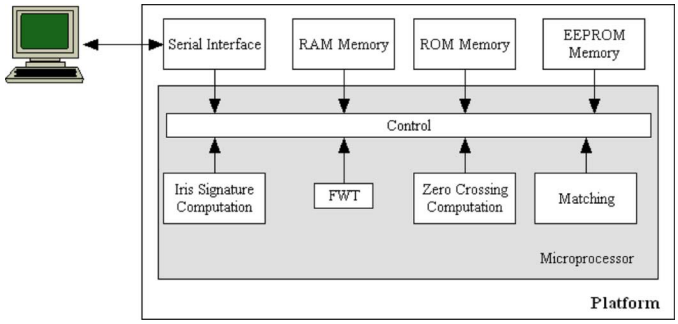

Fig. 5. Architecture of the microprocessor platform.

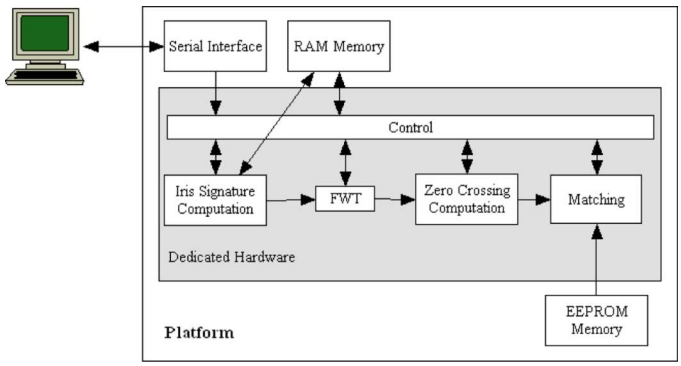

Fig. 6. Architecture of dedicated hardware implementation.

B. Platforms Considered

When designing identification tokens, several approaches may be studied. Current authentication algorithms have primarily been implemented in personal computers; however, these devices are not suitable for tokens due to their reduced size and cost. Nevertheless, the authors consider this platform as the initial stage for this study.

Computers are not the only devices that can be used to implement biometric systems. Fig. 4 represents several different approaches that have been considered.

1) Microprocessor: As Fig. 4 demonstrates, a computer is based on one or several microprocessors. Above these, several logical layers provide the user a transparent control of the electronics, which are based on an Operating System. Although this architecture eases the development of applications, these programs are not optimally translated to microprocessor instructions. Therefore, our first proposed implementation consists of a platform based on a microprocessor, which makes reasonable and optimal use of the peripherals and instructions for the functions that are to be developed (see Fig. 5).

In order to develop a biometric personal token, this platform is composed of the following peripherals.

- *Serial Interface:* Serial interface will be used so that the token can communicate with the terminal for data transfer and commands. The choice of physical interface is not crucial, i.e., from RS-232 to a USB 2.0 port.
- *RAM Memory:* As in any microprocessor system temporary memory storage is required. This memory will be used for storing data such as the segmented iris image, computational variables, etc.
- *ROM Memory:* For storing executable code and programming constants.
- *EEPROM Memory:* This stores the user template, allowing any changes if necessary. Other verification parameters, the different threshold levels can also be stored here.

The point of service terminal acquires the user's eye image and performs the described segmentation. The resulting image, together with other information, such as the inner and outer boundary parameters, is transmitted through the serial interface. As the token platform receives these data, it stores these data in the RAM. Once the transmission is finished, the token will begin its role by calling the normalization block. Afterwards, the feature extraction block based on the zero-crossing representation of the wavelet transform is executed. The resulting vector is then compared with the internally stored template and makes a decision on the matching result. Such decisions are transmitted back to the terminal by the serial interface. Once all of the processes are finished, the RAM is completely erased for later use.

Benefits from this approach when considering the computer platform are, as already mentioned, the optimization of all resources, no extra memory and overhead computations due to the presence of operating systems. A standalone execution is carried out, and only required functions are implemented, which is not the case with general purpose computers. The main drawbacks of this approach are related to the development and maintenance of the application. Another important issue to consider is that most microprocessors are intended for use in embedded systems, and do not use floating-point arithmetic, wherein the truncation needed for implementing these algorithms in a fixed-point arithmetic unit can cause error accumulation.

2) Dedicated Hardware: A more optimized solution than that using a commercial microprocessor previously described is to develop a specific purpose processor for the biometric personal token. This solution is implemented by developing dedicated hardware, which reduces processing time and the required electronics area. The use of dedicated hardware permits simultaneous computing processes, i.e., several processes can be computed at the same time; however, the main disadvantage of this solution is a reduction in accuracy due to the use of reduced fixed-point arithmetic, as well as the developmental costs, in terms of both time and money.

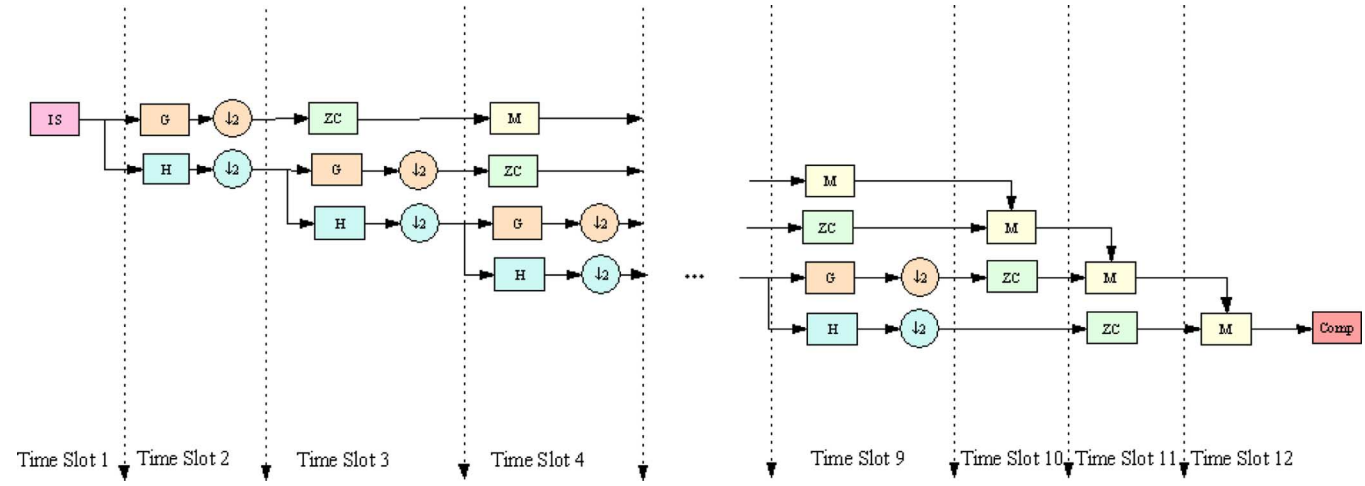

Fig. 7. Sequencing of concurrent processes in dedicated hardware implementation. In each time slot several processes occur. When all of them have finished, the system will go into another time slot.

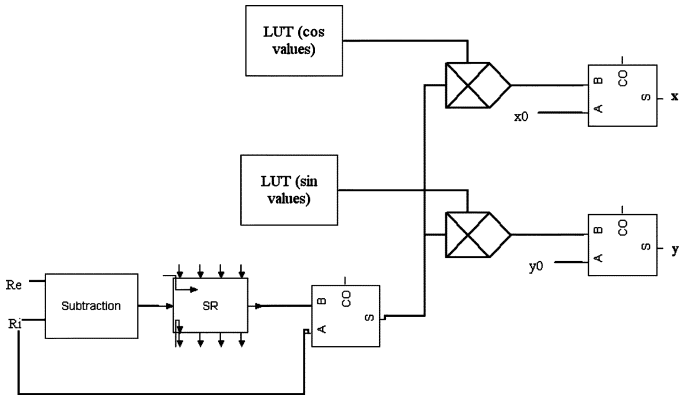

Fig. 8. Iris signature computation.

When designing dedicated hardware, several considerations should be made. First, the achievable level of parallelism needs to be evaluated, and second, the available hardware resources need to be used as efficiently as possible. For the first consideration, the designer must study the systems detailed block diagram and check which blocks require others to be finished so as to start operating, and which blocks can work in parallel. After that, the second development stage is devoted to considering which resources can be reused, so as to reduce hardware area.

The architecture of this implementation is shown in Fig. 6. Several differences between this and the previous platform can be observed.

- The communication with the terminal is still carried out using the serial interface. The segmented iris is stored in RAM, but all parameters computed in the iris segmentation block related to the inner and outer boundaries are transferred directly to the iris signature block.
- RAM is only used to store initial data.
- There is no ROM for storing executable code. The hardware is directly designed to perform all biometric functions.
- As different process implementations can be directly inter-connected, there is no need for external memory to transfer temporal information, which reduces the computation time.
- Control logic has the unique task of activating the process at the correct time. Here, it is also considered that different processes may run simultaneously.
- EEPROM is directly accessed by the matching block, as the control logic is only in charge of block timing.

The most interesting characteristic of this implementation is the concurrent functioning among blocks. This can be observed in Fig. 7, wherein it is shown that, in the first time slot, only the iris signature computation is performed, as this result is necessary for later nodes. Once this is computed and considering the fast wavelet implementation previously mentioned, H- and G-filtering can be executed in parallel. This concurrence can be clearly observed in the following time slot, where filtering of the second scale can be computed at the same time as the zero-crossing conversion is carried out for the first scale result. The forth slot will go further: filtering from the third scale, zero-crossing from the second scale, and matching from the feature vector corresponding from the first scale filtering.

The iris signature computation block consists of the accessing those memory addresses that are related to the iris signature and the storage of the values corresponding to these addresses. These addresses can be calculated as (2). In this formula, $\Delta\theta$, values are fixed, and the computation of trigonometric functions has been substituted by lookup tables, which store the corresponding values to the sine and cosine functions. Authors have also considered using a cordic algorithm, where similar hardware areas have been used, but LUTs have been selected because they provide better processing time. Iris signature computation is shown in Fig. 8.

In regard to the fast wavelet transformation block, filters H and G are both implemented as multipliers followed by accumulators with previous shift registers. The register associated to the vector is an addressable shift register, so it allows, before filtering, reordering of the elements of the vector and the introduction of zeros required for each level [24].

The matching algorithm implementation has been thoroughly investigated by the authors [25]. Although the best results in

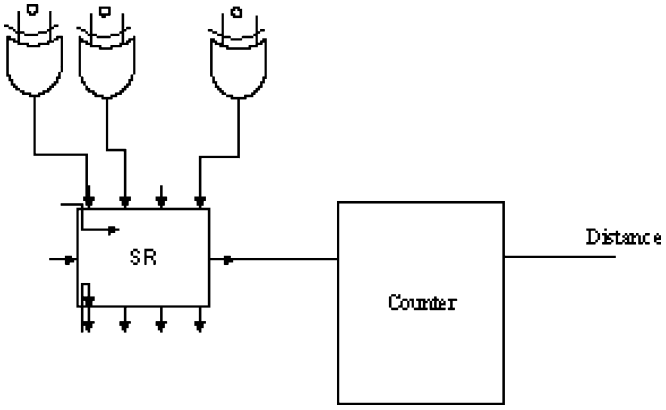

Fig. 9. Matching implementation.

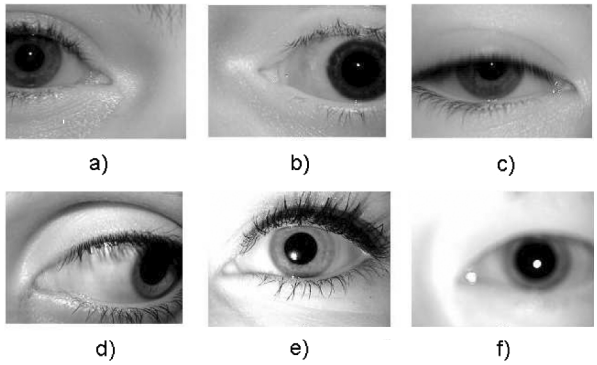

Fig. 10. Images from ICE 2005 Database.

time and hardware are obtained from a pipeline structure, the authors have selected a different solution, which is slower but smaller. The reason for this is that filtering takes much longer than matching; no improvement is achieved here by speeding up the matching process. Thus, better results are obtained by reducing the area of each of the matching blocks, resulting in longer computations that are still short when compared to the filtering process. Fig. 9 illustrates the matching implementation chosen for the dedicated hardware.

IV. Results

To present the obtained results, the database used therein will be briefly introduced. Following this, implementation details of both platforms chosen to carry out the process will be given. Finally, the results obtained from both platforms will be compared considering several viewpoints, such as performance, processing time, occupied area, cost, and security level.

A. Database

For iris recognition, there are several public databases (e.g., [26]–[28]) available for testing. There also exists a proposal for generating artificial iris images for this purpose [29]. For this work, the authors have chosen the ICE 2005 database [28]. This database is the largest public database and was proposed by NIST for the Iris Challenge Evaluation in 2005. It is composed of almost 3000 infrared images from 244 users. Assorted features can be found in this database (see Fig. 10), including different races, the usage of glasses and contact lenses, and artificially modified eyes.

As previously mentioned, quality control algorithms in the system reject some images. Fig. 10 depicts examples of some of the rejected images. Images a, b, c, and d are rejected, as they do not contain enough information due to the small iris area visible, while image f is rejected due to blurriness. Image e is accepted and processed.

B. Implementation Details for the Chosen Platforms

Several solutions for microprocessor platforms or dedicated hardware are commercially available. The decisions made on choosing the hardware have been based on several characteristics, such as cost and ease of integration. The microprocessor chosen for the first implementation was a ARM7TDMI. This microprocessor is a 16/32-bit RISC CPU designed by ARM [32]. Due to its high computational power and reduced cost, it is widely used in several commercial applications.

Regarding the dedicated hardware, field-programmable gate arrays (FPGAs) are elements that facilitate the development of application-specific integrated circuit (ASIC)-like solutions at a lower cost and at a reduced development time. FPGAs are composed of logic blocks (slices) and programmable interconnections. Each logic block is typically formed by several lookup tables, flip-flops, and RAM blocks. The FPGA used for the second platform is a Virtex4sx35. This FPGA belongs to the Virtex4 family, manufactured by Xilinx [33], and has been specially designed for digital signal processing, providing MAC units, which reduces the cost of implementing complex transforms, such as FWT.

C. Performance

Although computer and microprocessor platforms do not operate using the same type of arithmetic, mistakes arising from using 32-bit fixed point arithmetic (microprocessor) are not noticeable; hence, results obtained from both platforms are the same.

Unfortunately, using reduced fixed-point arithmetic in a dedicated hardware platform provides errors due to rounding. This rounding problem appears in the first stages of the system, such as during iris signature computation. The values of the trigonometric functions used in iris signature computation do not provide accurate values due to the length limitation in the number of bits used; however, the errors committed are not very limiting, as they refer to minimal changes in the row selection. Fig. 11 demonstrates the difference obtained from computing the address of the iris signature. When considering an image 400×400 pixels in size, values between -400 to 400 refer to errors in the same row; however, values outside of these boundaries report errors in different rows. These errors have been produced by the lookup table used for computing the trigonometric functions in the iris signature computation. The impact of these errors is further increased value due to posterior multiplication with other parameters. As Fig. 11 shows, these errors are primarily located in the areas surrounding the values of 350 and -100, demonstrating an approximate symmetry about 150 on

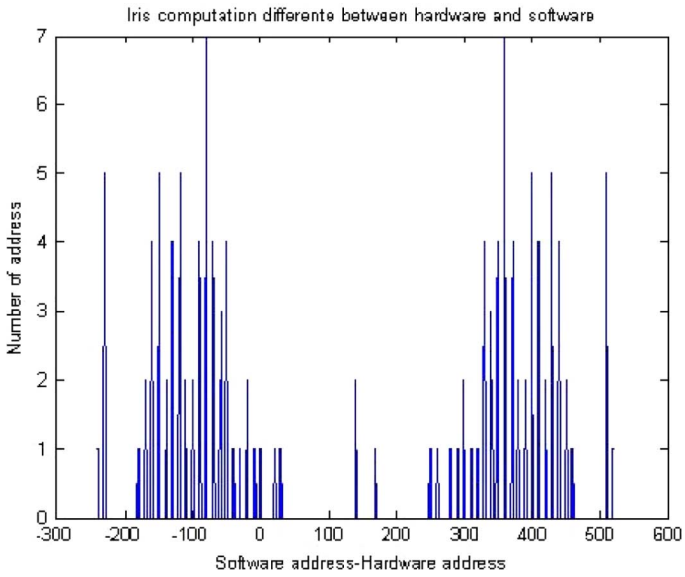

Fig. 11. Histogram of the differences shown in Iris Signature addresses computation.

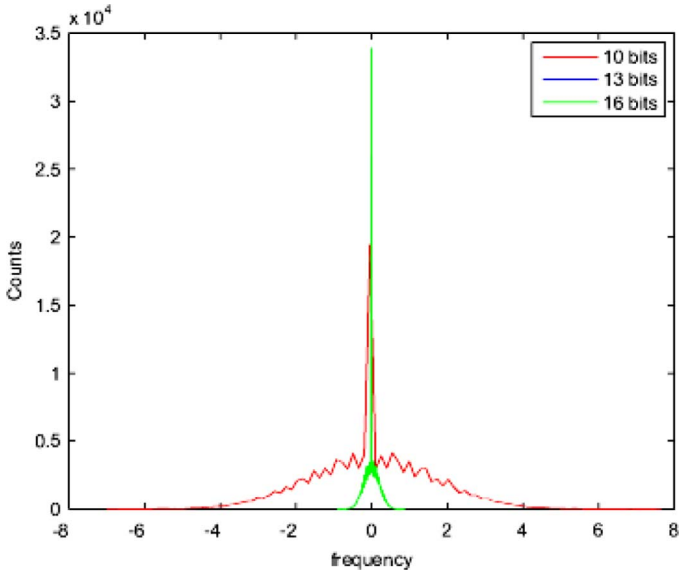

Fig. 12. Histogram of the different between values obtained with different hardware implementation of wavelet transformation and PC computation.

the abscissa. This symmetry and error figure are caused by two phenomena: first, the accuracy of the LUT used for the trigonometric functions, since we have fixed the output value of these LUT to 16 bits, and second, the posterior truncation performed in the iris signature computation after multiplying those LUT values with the pupil radium.

Considering wavelet transformation, several tests have been performed according to the different data input lengths for the filters and the number of bits dedicated to the binary point values has been modified. We have considered three values: 10, 13, and 16 bits, fixing the integer part to 10 bits.

The histogram of the differences found between the obtained PC values and hardware values can be observed in Fig. 12. As it can be seen, using integer arithmetic (red line, 10-bit width with no bits dedicated to the binary point) produces errors with

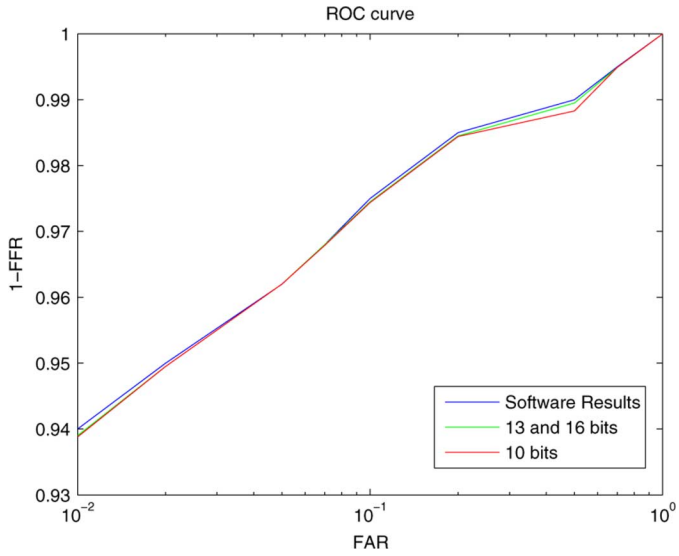

Fig. 13. ROC curve.

TABLE I
MICROPROCESSOR PLATFORM AREA

Data Size (bytes)	Constant Size (bytes)	Code Size (bytes)
5031	324	2640

values that range from −4 to 4, thus leading to errors in the feature computation; however, in the case of using 13 or 16 bits, these errors are reduced to less than unity, and the feature vectors obtained, due to the algorithm's robustness, are the same as those obtained from the software. In addition, it can be seen in this figure that using 13 or 16 bits does not improve results, as the deviation from the software results in both cases is the same.

In biometrics, the most common parameters used for evaluating system performance include the false acceptance rate (FAR), which is a measure of the number of potential intruders that access the system, and the false rejection rate (FRR), which measures the number of authorized users who will be rejected by the system. These two parameters are usually expressed in a single curve called the receiver operating characteristic (ROC). Fig. 13 depicts the obtained ROC curve for the proposed system. As it can be observed, the differences between the software curve and the hardware implementation curve are minimal. These differences are provoked by the LUT granularity used in the iris signature computation, which, as Fig. 11 points out, provokes some errors in this addressing, as well as by the wavelet transformation.

D. Area

When comparing the occupied area, it is not possible to find a unique parameter for comparative purposes between these platforms. The most commonly used parameter in microprocessor-based solutions is code size; however, in dedicated hardware, the area is usually specified by the slices used.

Table I summarizes the area occupied by the microprocessor solution. The code size is determined by program instructions. In addition, the size plus the defined constants determine the ROM size. The RAM is determined by the data size, i.e., temporary stored data.

TABLE II
HARDWARE AREA FOR DIFFERENT FILTER CONFIGURATIONS

Filter Input Length	LUTs	Flip Flops	MACs	Slices
10 bits	1137(3%)	546 (1%)	9 (5%)	640 (4%)
13 bits	1208 (3%)	591 (1%)	9 (5%)	677 (4%)
15 bits	1245 (4%)	609 (1%)	9 (5%)	697 (4%)

TABLE III
PROCESSING TIME

	Maximum Frequency	Total time (μs)
Computer Solution	2.8 GHz	5,660
Microprocessor	60 MHz	56,600
FPGA 10 bits	122.264 MHz	295.459
FPGA 13 bits	118.034 MHz	306.047
FPGA 16 bits	112.300MHz	321.674

Table II summarizes the results obtained for the three investigated hardware configurations when the area instead of the processing speed is optimized.

Although the best performance results have been obtained using 16- and 13-bit filter input lengths, these solutions do not show a significant difference in the area; 13-bit input lengths require fewer hardware area resources. As it can be seen, variability in the filter input length does not affect the number of DSP slices used (i.e., MACs), but undoubtedly affects the number of LUTs and flip-flops.

E. Processing Time

Table III summarizes a comparative analysis of the processing time. From the results shown in this table, the work cycles and working frequency have been considered when analyzing this data, wherein it can be seen that the microprocessor solution requires more processing time when compared to the computer solution, as expected. The main reason for this is the clock frequency of both platforms, wherein the computer uses a clock speed that is almost 50 times faster than that of the microprocessor; however, the execution time of the microprocessor solution is only 10 times slower, so in relative terms, the microprocessor platform presents a more optimized solution. The dedicated hardware produces the best results in terms of processing time due to the possibility of parallel processing. The results obtained for this platform are approximately 20 times faster than the computer platform. Furthermore, these results are even more impressive since the dedicated hardware clock frequency is twice that of the microprocessor and 20 times less than the frequency used by the computer. Table III also depicts the time difference among the different investigated hardware configurations. The number of cycles used for obtaining the identification results is the same in all cases: 36 124 clock cycles; however, the maximum working frequency slightly varies among all of the configurations. This variation is caused in the synthesis process when routing the slices.

F. Security

An indirect advantage of both proposed platforms is the achieved level of security. When comparing the microprocessor to the computer-based solution, less potential attack points are accessible. The microprocessor does not have an open operating system that can be programmed with new applications; hence, no malicious code can be programmed into the system. Furthermore, when using the hardware-based solutions, no programs exist, which again reduces potential attack points when compared to the microprocessor-based solution.

Obviously, all of the solutions proposed herein do not provide a 100% level of security. Potential attack points still exist, excluding reverse engineering and tempest attacks, which are primarily solved by manufacturing processes (pseudo-random memory organization, Faraday cases, tamper-proof techniques, etc.). Major attacks arise from the exploitation of the data transferred between the token and the external world. This can be solved by using cryptographic solutions (ciphering, authentication, time stamping, etc.). All of these solutions are not included within the scope of this paper, but certainly may be recommended for future investigation.

G. Cost

In addition, another indirect result from a hardware development, is cost reduction. It is clear that development tools are not as simple as those used for computers, and therefore, the development time is longer. This affects the initial fixed cost, which is much higher than the variable costs when production is in progress. This fact is more important when considering dedicated hardware solutions instead of a microprocessor solution.

When considering the development of personal tokens and the mass production of these units, economical benefits emerge at several different levels. First, since a considerable number of units are manufactured, the economy of large-scale production decreases variable costs. This is especially important when considering dedicated hardware, as no further elements are required, and the electronic board will be the smallest possible design.

Indirectly, another cost benefit is related to the cost of the terminal. Since most biometric algorithms are included within the token, the complexity of the terminal is greatly reduced, requiring cheaper components and increasing performance. When preprocessing is included in the biometric sensor, the terminal developer will not have to acquire the know-how or licences of biometric algorithms, thus reducing terminal development costs.

Finally, the long term benefits, i.e., the reduction in security breaches, will increase user acceptance and reduce maintenance costs, increasing sales and producing more economical benefits.

V. CONCLUSION AND FUTURE WORK

Different platforms were studied for biometric authentication scenarios. Two platforms have been designed and developed: a microprocessor-based architecture and a dedicated hardware design. Each platform exhibits benefits when compared to general purpose computer systems. Selecting one of these platforms depends on system and authentication application requirements. In the case of high security environments, where low error rates are extremely important, the microprocessor solution is recommended, especially when the number of users in the system is relatively high; however, if the number of users is low or size and execution times are significant constraints, the dedicated hardware solution should be chosen.

The obtained processing times exhibit the best results for the dedicated hardware solution, improving by over 200 times over

microprocessor-based solutions, and the request of a clock rate two times faster.

The results obtained in this study direct future research into the integration of cryptographic modules that would secure all data transmission. Another research area would explore optimal hardware solutions for identification tokens that combine the benefits of both platforms developed herein (i.e., using HW/SW codesign).

REFERENCES

[1] A. Jain, R. Bolle, and S. Pankanti, , S. P. A. Jain and R. Bolle, Eds., *Biometrics: Personal Identification in a Networked Society*. Norwell, MA: Kluwer, 1999.

[2] M. Faundez-Zanuy, "Biometric security technology," *IEEE A&E Syst. Mag.*, vol. 21, no. 6, pp. 15–26, Jun. 2006.

[3] J. Mansfield and J. L. Wayman, Best practices in testing and reporting performance of biometric devices U.K. Goverment Biometrics Working Group, 2002. [Online]. Available: http://www.npl.co.uk/upload/pdf/biometrics_bestprac_v2_1.pdf

[4] P. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott, and M. Sharpe, "FRVT 2006 and ICE 2006 large-scale results," Nat. Inst. Standards Technol., 2007. [Online]. Available: http://www.frvt.org/FRVT2006/docs/FRVT2006andICE2006LargeScaleReport.pdf

[5] Independent Biometric Group, "Comparative biometric testing round 6 public report," 2006 [Online]. Available: http://www.biometricgroup.com/reports/public/comparative_biometric_testing.html

[6] K. Bowyer, K. Hollingsworth, and P. Flynn, "Image understanding for iris biometrics: A survey," *Comput. Vision Image Understand.*, vol. 110, no. 2, pp. 281–307, 2008.

[7] R. Sanchez-Reillo, J. Liu-Jimenez, and L. Entrena, "Architectures for biometric match-on-token solutions," in *Proc. ECCV Workshop BioAW*, 2004, pp. 195–204.

[8] K. Ryoung Park and J. Kim, "A real-time focusing algorithm for iris recogntion camera," *IEEE Trans. Syst., Man Cybern. C, Cybern.*, vol. 35, no. 3, pp. 441–444, Aug. 2005.

[9] Y. He, J. Cui, T. Tan, and Y. Wang, "Key techniques and methods for imaging iris in focus," in *Proc. 18th Int. Conf. Pattern Recog.*, Aug. 2006, vol. 4, pp. 557–561.

[10] J. R. Matey, O. Naroditsky, K. Hanna, R. Kolczynski, D. J. LoIacono, S. Mangru, M. Tinker, T. M. Zappia, and W. Y. Zhao, "Iris on the move: Acquision on images for iris recognation in less contrained enviroments," *Proc. IEEE*, vol. 94, no. 11, pp. 1936–1947, Nov. 2006.

[11] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independece," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, Nov. 1993.

[12] C. Sanchez-Avila and R. Sanchez-Reillo, "Two different approaches for iris recognition using gabor filters and multiscale zero-crossing," *Patt. Recog.*, vol. 38, no. 2, pp. 231–240, 2005.

[13] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Efficient iris recognition based characterizing key local variations," *IEEE Trans. Image Process.*, vol. 13, no. 16, pp. 739–750, Jun. 2004.

[14] R. P. Wildes, "Iris recognition: An emerging biometric technology," *Proc. IEEE*, vol. 85, no. 9, pp. 1348–1363, Sep. 1997.

[15] C. Tisse, L. Martin, L. Torres, and M. Robert, "Personal identification on technique using human iris recognition," in *Proc. Vision Interface*, 2002, pp. 294–299.

[16] L. Masek, "Recognition of human iris patterns for biometric identification," Master's thesis, Sch. Comput. Sci. Softw. Eng., Univ. Western Australia, Perth, 2003.

[17] J. Daugman, "Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparison," *Proc. IEEE*, vol. 94, no. 11, pp. 1927–1935, Nov. 2006.

[18] J. Daugman, "Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparison," *Proc. IEEE*, vol. 94, no. 11, pp. 1927–1935, Nov. 2006.

[19] J. Daugman, "New methods in iris recognition," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 37, no. 5, pp. 1167–1175, Oct. 2007.

[20] N. Ritter, R. Owens, J. Cooper, and P. P. Van Saarloos, "Location of the pupil-iris boder in slit-lamp images of the cornea," in *Proc. Int. Conf. Image Anal. Process.*, Sep. 1999, pp. 740–745.

[21] A. Ross and S. Shah, "Segmenting non-ideal irises using geodesic active contours," in *Proc. Biometr. Symp.*, 2006, pp. 1–6.

[22] W. W. Boles and B. Boashash, "A human identification technique unisg images of the iris and wavelet transform," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 1185–1188, Apr. 1998.

[23] B. Fernandez-Saavedra, J. Liu-Jimenez, and C. Sanchez-Avila, "Quality measurements for iris images for biometrics," in *Proc. IEEE EUROCON Int. Conf. "Computer as a Tool"*, Sep. 2007, pp. 759–764.

[24] S. Mallat, "Zero-crossing of wavelet transform," *IEEE Trans. Inf. Theory*, vol. 37, no. 4, pp. 1019–1033, Jul. 1991.

[25] J. Liu-Jimenez, R. Sanchez-Reillo, A. Lindoso, and O. Miguel-Hurtado, "FPGA implementation for an iris biometric processor," in *Proc. IEEE Int. Conf. Field Program. Technol.*, Dec. 2006, pp. 265–268.

[26] Center for Biometrics and Security Research, Beijing, China, "Casia database," 2005. [Online]. Available: http://www.cbsr.ia.ac.cn/iris-database.htm

[27] Dept. Comput. Sci., Palacky University in Olomouc, Czech Republic, "Upol database," 2005. [Online]. Available: http://phoenix.inf.upol.cz/iris/

[28] National Institute of Standards and Technology, Arlington, VA, "Nist ice database," 2005. [Online]. Available: http://iris.nist.gov/ice/

[29] J. Zuo, N. A. Schmid, and X. Chen, "On generation and analysis of syntetic eyes iris images," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 1, pp. 77–90, Mar. 2007.

[30] R. E. Gonzalez and R. C. Woods, *Digital Image Processing*, 3rd ed. Reading, MA: Addison-Wesley, 1992.

[31] H. Davson, *The Physiology of the Eye*. New York: Little, Brown and Company, 1963.

[32] ARM, Ltd., U.K., "ARM LTD website," 2005. [Online]. Available: http://www.arm.com/

[33] Xilinx, San Jose, CA, "Xilinx website," 1984. [Online]. Available: http://www.xilinx.com/

Judith Liu-Jimenez (S'07) received the M.S. degree in electrical engineering from the Universidad Politecnica de Madrid, Madrid, Spain, in 2003. She is currently pursuing the Ph.D. degree in hardware/software codesign applied to Biometrics from the Universidad Carlos III of Madrid, Madrid, Spain.

She is an Assistant Teacher with the Electronic Technology Department, Universidad Carlos III of Madrid. Her research interests focus on iris biometrics, hardware/software codesign, and security. She has been actively involved in several European projects related to these fields, such as eEpoch and Biosec.

Raul Sanchez-Reillo (M'00) received the Ph.D. degree in telecommunication engineering from the Universidad Politecnica de Madrid, Madrid, Spain, in 2000.

He is currently an Associate Professor with the Universidad Carlos III de Madrid, Madrid, Spain. Since 1994, he has been working with the University Group for Identification Technologies (GUTI—formerly known as University Group of Smart Cards), and has been involved in project development and management concerning a broad range of applications, from social security services to financial payment methods. He has participated in several European Projects, such as eEpoch and BioSec, as the WP leader. He took the leadership of GUTI in 2000. He is also an expert in security and biometrics, with several published articles and conference presentations. His experience has allowed him to become a member of the ISO SC17, SC27, and SC37 standardization committees.

Belen Fernandez-Saavedra (S'07) received the M.S. degree in robotics and electronic systems from the University Carlos III of Madrid, Madrid, Spain, in 2006, where she is currently pursuing the Ph.D. degree with a focus on evaluating the security of biometric systems, following the works completed by Common Criteria.

She is an a R&D engineer with the University Group for Identification Technologies (GUTI), University Carlos III of Madrid, Madrid, Spain.