Accepted Manuscript

Efficient yet robust biometric iris matching on smart cards for data high security and privacy

N. Nedjah, R.S. Wyant, L.M. Mourelle, B.B. Gupta

PII: S0167-739X(17)30108-5
DOI: http://dx.doi.org/10.1016/j.future.2017.05.008
Reference: FUTURE 3456

To appear in: *Future Generation Computer Systems*

Received date : 17 January 2017
Revised date : 6 April 2017
Accepted date : 7 May 2017

Please cite this article as: N. Nedjah, et al., Efficient yet robust biometric iris matching on smart cards for data high security and privacy, *Future Generation Computer Systems* (2017), http://dx.doi.org/10.1016/j.future.2017.05.008

Efficient Yet Robust Biometric Iris Matching on Smart Cards for Data High Security and Privacy

N. Nedjah*, R. S. Wyant*, L. M. Mourelle† and B. B. Gupta‡

*Department of Electronics Engineering and Telecommunication
†Department of System Engineering and Computation,
Engineering Faculty, State University of Rio de Janeiro, Brazil
‡Department of Computer Engineering,
National Institute of Technology Kurukshetra, India.

Abstract

Smart control access to any service and/or critical data is at the very basis of any smart project. Biometrics have been used as a solution for system access control, for many years now. However, the simple use of biometrics cannot be considered as final and perfect solution. Most problems are related to the data transmission method between the medias, where the users require access and the servers where the biometric data, captured upon registration, are stored. In this paper, we use smart cards as an effective yet efficient solution to this critical data storage problem. Furthermore, iris texture has been used as a human identifier for some time now. This biometric is considered one of the most reliable to distinguish a person from another as its unique yet perfectly stable over time. In this work, we propose an efficient implementation of iris texture verification on smart cards. For this implementation, the matching is done on-card. Thus, the biometric characteristics are always kept in the owner's card, guaranteeing the maximum security and privacy. In a first approach, the False Acceptance Rate (FAR) and False Rejection Rate (FRR) are improved using circular translations of the matched iris codes. However, after a thorough analysis of the achieved results, we show that the proposed method introduces a significant increase in terms of execution time of the matching operation. In order to mitigate this impact, we augmented the proposed technique with acceptance threshold verification, thus decreasing drastically the execution time of the matching operation, and yet achieving considerably low FAR and FRR. It is noteworthy to point out that these characteristics are at the basis of any access control successful usage.

Key words: Biometrics, iris texture, smart card, privacy, security.

1 Introduction

Biometrics represents studies related to some human characteristics that can be used in distinguishing two distinct persons. These characteristics may be physical, such as finger and palm prints, or behavioral, such as the speed of typing on the keyboard and the way one signs documents. Currently, there are many characteristics used as biometrics in real-world applications. The most commonly used include DNA, face, hand veins, fingerprints, hand geometry, iris, palm prints, voice patterns, among many others. Biometrics are considered as highly secure for identification of individuals as they are distinctive. Furthermore, it is extremely difficult to be forged. Moreover, biometrics are very convenient in their usage because they can not be stolen or forgotten somewhere [10].

Smart applications are supposed to be autonomous, having sites that are "intelligent" thanks to built-in sensors together with the control that handles the information rendered by these sensors to act on behalf of humans using the facilities. For instance, in smart cities, the number of humans involved in controlling access to buildings is usually reduced to a strict minimum. The main usage of biometrics within smart projects is mainly related to data access control, *i.e.* through the checking of biometrics, some user can be granted or denied access to the protected service or information, among others. In most cases, biometry has a big advantage when compared to other kind of identity authentication. It can really guarantee the authenticity of the claimant. It is a general belief that the usage of biometrics is the perfect solution for all identification problems [16].

Smart card technology has existed for several decades. Currently, it has a number of established sophisticated utilizations. Banking and healthcare systems often resort to the use of smart cards to secure client's and patient's private information, respectively. There is a big deal of existing works on security and privacy of data in Smart cards. However, given the scope of this work, which more about biometrics implementation on smart card than about smart cards themselves, we do not intend to go through the wealth of existing work about the privacy and security of smart card usage. Nonetheless, the most important research issues for privacy and security of electronic services, in general and in healthcare in particular can be found in the recent survey available in [31]. Moreover, even though smart cards are supposed to be resistant to logical attacks, their use is not completely safe. There have been several types of smart card vulnerabilities that could be exploited. It is noteworthy to point out that this work does not focus on improving the resistance of smart cards regarding any kind of logical attacks on the data storage of the smart card, such as Differential Power Analysis (DPA). This said, however, all nowadays smart card standards mandate DPA resistance, among others as an important component of the system's overall security requirements [10].
Besides the problems regarding security and privacy, there is also the acceptance aspect by users. The use of biometrics has been spreading rapidly and people are

starting to worry about their own safety when they are asked to register their biometrics, indiscriminately in various institutions. After all, their biometric details would be stored in many databases, which are susceptible to attacks. In such a case, biometrics, which is unique and invariable over time, could be lost forever. It is now well known that the exploitation of smart card based solutions augmented with biometrics verification provides more privacy and security when compared to biometric-only or smart card-only solutions. With the biometrics details stored in the card's local memory and executing the biometric match on-card, the privacy and security of the biometric data are enhanced as well as the system performance.

Iris textures have been used as a human identifier for some time now. They are considered one of the most reliable ways to distinguish a human from another [27]. It is considered one of the most reliable because it is an internal and practically invariant organ for a lifetime. It is a flat organ and its diameter changes only with the contraction and dilation of the pupil. It is actually a recent biometrics [13], and it is already being used in various security systems around the world, as the United Arab Emirates, Amsterdam Airport Schiphol, Canadian Air Transport Security Authority, among many others. Market researchers have predicted biometric smartphones will reach 100% adoption by 2020, and recent Samsung smartphone Galaxy Note 7 is already featuring iris recognition to market [1]. Recently some exiting research work have been published on ocular biometrics, which encompasses imaging and use of characteristic features extracted from the eyes for personal recognition [26,3]. There are also some research that aim at improving accuracy and usability via the usage of multi-biometric system, wherein features regarding multiple biometric traits are matched [22,6,15]. Note that in this works, we are only concerned with iris matching. It is noteworthy to point out that the biometrics market is very competitive and thus no technical details of commercially available solutions can be made available for study or comparison.

The iris has a texture, which is randomly determined in the embryonic stage, as well as fingerprints. This proves that it is unique, and is virtually impossible to forge. However, there are so many factors involved in the formation that the occurrence of a false validation is minimal. Figure 1 shows the parts of the eye that need to be considered when comparing irises.

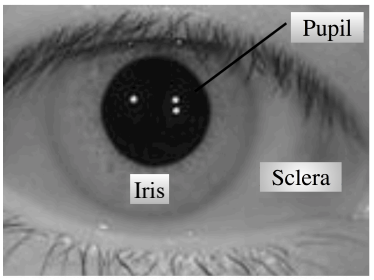


Fig. 1. Eye structure

Another benefit of iris biometrics usage is the distance of capture, making the touch of the reading equipment not necessary. Most equipment operates from 10*cm* till few meters. However, this brings a difficulty for the verification of biometrics. This distance, which on one hand, is comfortable results in an image that requires some processing before obtaining the iris template. This process usually results in error during the matching process. Another source of error comes from the number of obstacles that may prevent the correct capture of the iris as the eyelids and eyelashes.

In this paper, we propose an efficient and secure implementation of user authentication via iris texture comparison. We exploit the approach that makes use of smart cards together with iris texture as a biometric, aiming at increasing the security of access control systems, that are fundamental in smart buildings, in particular and smart cities, in general. We weigh the option of using a smart card that grants access by performing biometric comparisons. Thus, it would be possible to use a single card for several institutions and biometrics would always be stored in a single card in the possession of the owner. The biometrics details would be stored only in a unique smart card and the matching is processed on-card. Mainly, we provide an efficient yet robust implementation of iris biometric comparison on smart cards. Comparison of iris textures, as described in [32], has been selected as a reference for the proposed implementation on smart cards because it requires a reduced amount of memory and low computing effort to obtain the matching result. Nonetheless, an viable implementation of iris biometrics on smart cards would be still a big challenge given the high limitations both of storage and processing power of any state-of-the art smart card. Furthermore, as direct iris code comparisons originating from authentic users may fail due to small rotations of the iris at the time of capture, in contrast with existing implementations, the proposed method ensures robustness and more secure comparisons by allowing translations of the iris binary code. However, the introduced translations entail more processing effort. So in order to keep the proposed implementation efficient and competitive, we do only the minimal computation required to refute any mismatch of iris code, and thus guaranteeing robustness, more security and efficiency in terms of response time.

The rest of this paper is organized in six sections. First, in Section 2, we defend the case of the usage of smart cards as a way to store and match biometrics and also explain the used process in this case. After that, in Section 3, we discuss some relevant existing related to research works on using iris texture as biometric. Subsequently, in Section 4, we define the internal representation of iris texture, as used in this work, and give some details on the actual implementation on smart cards. After that, in Section 5, we present and discuss the effectiveness and performance of the proposed implementation. Last but not least, in Section 6, we draw some conclusions and point out some directions for future work.

2 Biometrics in Smart cards

This section defines the aspects that the implemented biometric systems are required to have when using smart cards. Nonetheless, recall that the focus of this work is to prove the feasibility of efficiently implementing iris biometric comparisons processed in smart cards. Biometric systems are basically composed of four components:

- A machine or mechanism responsible for the digital representation of the biometric characteristics of a person;
- A standard extraction tool that will be used in the comparison;
- A verification tool to match the stored pattern and the input pattern;
- An interface to output of the result.

Biometric systems operate in two stages: the storage of the pattern that will serve as a basis for the comparison and the matching of the stored pattern and the input pattern.

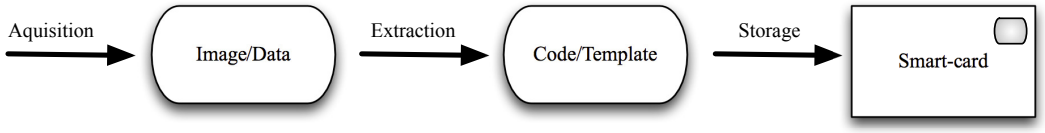


Fig. 2. Registration of a biometrics in the smart card

Figure 2 illustrates the registration process, also known as *enrollment*. The sample of the individual, who is card user, is captured. For each specific method will be used biometrics (fingerprints for scanner, microphone for voice recognition, camera for face recognition camera for iris recognition etc.). The collected data is then processed to extract the unique characteristics of the user. The extracted biometric template that will be used in future comparisons is stored on the card.

Figure 3 illustrates the biometric verification process, also known as *matching*. The applicant's biometric sample is captured similarly to the process made during enrollment stage. The unique patterns of this sample are extracted and sent to the checker. The stored pattern is retrieved from the card and sent to the checker, which then runs the verification process, resulting in a score that establish whether both biometric samples are from the same individual. Biometric system's main purpose may be identification and/or verification. The identification is the search for a person from a given biometric sample. This encompasses large databases and requires high processing power. Indexing techniques for improving search may be used to improve performance. On the other hand, verification is the validation done given two biometric samples, resulting in the identification whether the samples belong to the same person. The biometric comparisons using smart cards can occur in two ways:

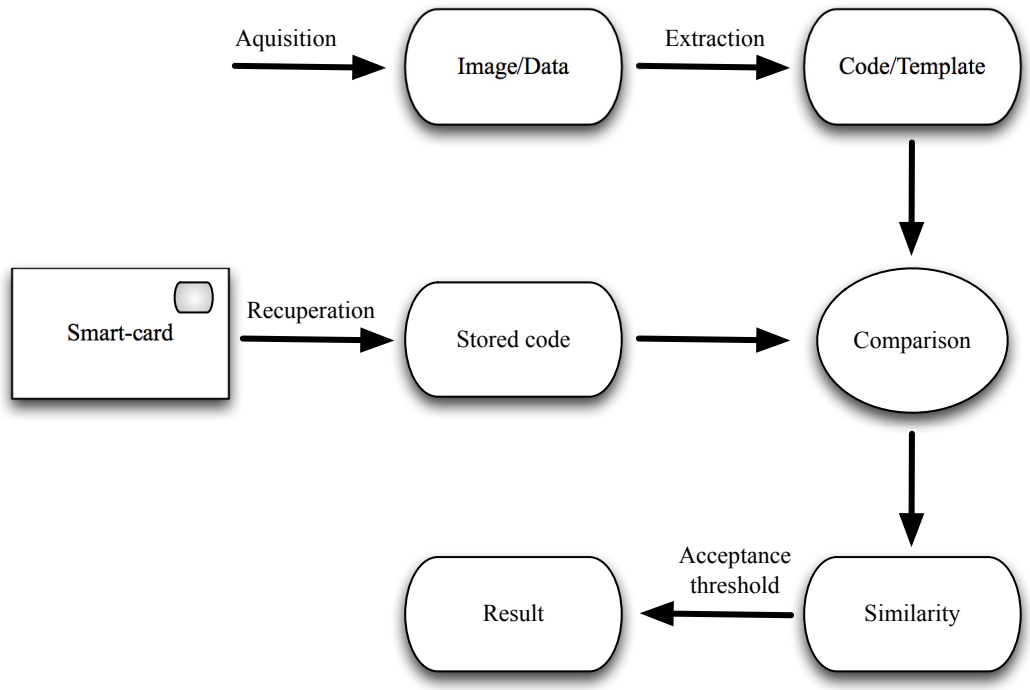


Fig. 3. Biometric verification

- *Template on Card* (ToC), where the user's biometric sample is stored in the card's memory and the comparison is done externally on another machine. This requires cards that only have memory, which are much cheaper thank cards that are endowed with a microprocessor.
- *Match on Card* (MoC), where the user's biometric sample is stored in the card's memory and the comparison is also processed on the card. In this case, the smart card needs to include at least one processor. The low processing frequency and small memory size included in today's smart cards are the biggest obstacles in these implementations. In this work, we propose an implementation MoC.

In a biometric system, when the stored sample is compared to the captured information, a score of similarity is assigned and used to confirm the identity of an individual. When this score is compared with a pre-defined threshold, two types of error rate can be observed:

- False Acceptance Rate (FAR), which indicates the rate of false entries or incorrectly accepted fraudulent data.
- False Rejection Rate (FRR), which indicate the rate of correct individual entries that were incorrectly rejected.

The aforementioned rates are extremely important in choosing the limit of the score that should define the final decision of comparisons to be declared as false or true. When it comes to embedded systems, an extremely important factor is also the choice of the algorithm to be implemented. It is necessary to determine the complexity in terms of memory usage and runtime.

3 Related work

The most important of the first publications on methods of personal recognition by iris biometrics belongs to Professor Daugman [13]. The technique proposed in [13] describes in some details the processes of segmentation, extraction and comparison. Daugman's work has become the largest benchmark in this segment and served as the basis for virtually all existing iris biometric models [4].

In Daugman's work the whole process of iris segmentation is detailed. An integral-differential operator is used to find the location of the iris as well as the regions covered by the eyelids or by reflections of light. In order to normalize the result taking into account different distances or resolutions, two coordinates are adopted: angle varying from 0° to 360° and a radial coordinate that varies from 0 to 1 independent of the image size or the dilation and contraction of the pupil, being that the deformations occurred as a result of the two movements were considered linear. The image is then transformed into a rectangle, assuming the radial coordinate as the vertical axis and the angular coordinate as the horizontal axis.

Overall image comparison would lead to many errors due to the influence of brightness. To avoid this, image convolution is performed using a two-dimensional Gabor filter to extract texture information. The result of this convolution is an array of complex numbers that are then encoded using only their phases. This process results in an array of binary numbers, which represent the phases of complex numbers, with a total length of 256 bytes.

In order to make the comparison between two codes is also made a mask that delimits which are the valid pixels, *i.e.* that are not covered by eyelids or reflexes. The comparison is made binary using the Hamming distance, a "Exclusive or" operation is applied on all the code and the number of 1s, which differ, is counted and related to the number of valid bits.

Another work has achieved some prominence without using Daugman's work as a basis. In [30], unlike Daugman's work, the Hough transform is used to detect both the inner and outer iris circles. For comparison, the Gaussian Laplacian filter was applied on multiple scales and the similarity of the images was computed. The author points out positive results regarding false acceptance, but indicates that the system is not very flexible in relation to the positioning of the iris and the luminosity of the environment. These problems are best addressed in Daugman's approach.

After the first works in the area of iris biometrics, several others appeared to improve some aspects, such as segmentation, extraction and comparison of existing methods. In [7], a modification is suggested in the Wildes's process to look for the iris in the image on another scale. The unique idea of comparing using both eyes is presented, the left one being the comparison and the right one for the correct alignment.

Regarding segmentation, the canny edge detector and the Hough transform are often used [18], but in an attempt to simplify the process, it is usually assumed that the pupil border and iris are concentric. Some images validate this conclusion. However, this finding cannot be applied to all cases, but mainly in images taken at different angles. In order to improve the segmentation process, equalization can be used via a high-pass filter [29]. Also, in [5], the authors present a segmentation method that does not use the Hough transform. The used technique is somehow similar to that used by Daugman. The coordinates and radius of the circle are modified as to find the best solution in a defined search space. Moreover, the algorithm considers a limit of the relation between the inner and outer rays of the iris. The achieved hit rate were very high for people that use glasses and about average for people wearing glasses. The proposed algorithm reduces the processing time, which is about 3.5$\times$ faster than the algorithm proposed by Daugman in [11]. However, the obtained hit rates are smaller but compatible.

Regarding extraction, different filters have been proposed to obtain the iris characteristics. In [28], a Gaussian filter is used. The field convolution of the image gradient vectors is performed using the Gaussian filter. Then, each part of the obtained result is sorted according to 6 different options, in contrast to the Daugman process, in which the convolution result is sorted in relation to the phase of the complex number. In [2], the Wavelet transform is used to extract features from the enhanced iris images. Also, in [8], the Gaussian Laplacian filter is used to implement image convolution and thus extract the so-called blobs, which are the darker areas relative to their neighborhoods. A code is then constructed based on the presence or absence of blobs. In [9], the Gaussian Laplacian filter is also used in conjunction with the Gaussian Derivative filter to determine if a given pixel is a "step" or "ridge" border. Based on these extracted characteristics, a measure of similarity between two irises can be obtained. The advantage of this kind of simpler filter when compared with Gabor's filter consists of the reduced number of parameters required, making the filter configuration much easier. Both works suggest the use of genetic algorithms to find the most adequate set of the used parameters.

Regarding comparison, some biometric methods use multiple samples as a basis to improve the results of the comparisons. This can also be done for biometric methods based on iris recognition. In [14], 1, 2 or 3 images are used in the comparison operation, achieving a hit rate of 98.5%, 99.5% and 99.8%, respectively. In [19], the authors suggest that the chosen image should be the one with the best quality while in [20], an average of the three comparisons is used. Furthermore, in [12], an experiment to determine the statistical variation of the iris texture is described. About 2.3 million comparisons are performed between different iris pairs. An average Hamming distance of 0.499 is found with a standard deviation of 0.032. This distance can vary from 0 to 1. The distribution is estimated as a binomial distribution with 244 degrees of freedom. It was also established that the comparison between the two irises of the same person and between those of different people has no statistically significant difference.

4 Proposed Iris Texture Verification on Smart cards

The iris biometry has gained significant importance in the world market. This biometry stands out from the rest because it is very safe and durable. As mentioned earlier, John Daugman is considered the pioneer of iris biometrics for presenting the first work of great acceptance in the area. His work is based on the use of the Gabor's 2D filter to extract the characteristics of the iris texture and has served as an inspiration for several subsequent works both in the area of iris biometrics as well as in other biometrics, for example the biometric method of palm print in [23].

Due to its great importance and robustness, the Daugman's method as described in [13] is chosen for the implementation of iris biometrics on smart cards. The binary code extraction method and the comparison algorithm as proposed in this work are discussed in Sections 4.1 and 4.2, respectively.

4.1 Extraction

The extraction of iris characteristics is performed in three main steps: segmentation, normalization and binary code formation. The latter is termed *iris code*. The inherent details of this procedure are covered in Sections 4.1.1, 4.1.2 and 4.1.3, respectively. It is noteworthy to point out that the operation regarding extraction are all carried out as a pre-processing work by the host computer, *i.e.* off-card.

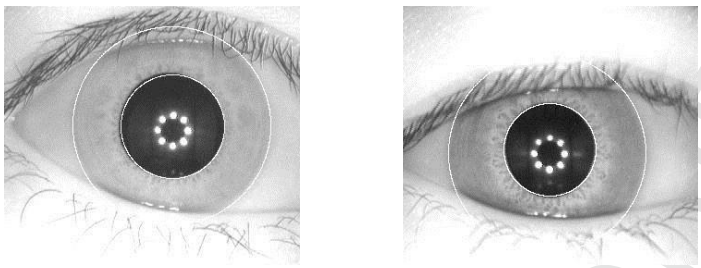
4.1.1 Segmentation

The segmentation is probably the most important and most complicated step of the extraction [17]. Basically, the expected result is the exact location of the inner and outer contours of the iris. The task can be extremely challenging when the contours are heavily covered by eyelids or eyelashes. Another difficulty can be found in very light colored irises, as they may be confused with the sclera, as illustrated in Figure 1.

The algorithm used is based on Hough transform. It is a standard algorithm in image processing used to determine parameters of simple geometric objects. The circular Hough transform can be used to deduce the points of the coordinate of the iris center and the radius of the iris contours.

Prior to the application of the transform, it is necessary to identify the edge extraction using a special filter. In this work, the derivative of the first order of the image intensity is applied to find all the possible contour points that will be used in Hough Transform. The crucial point in this step is the selection of the boundary of what will be considered as contour.

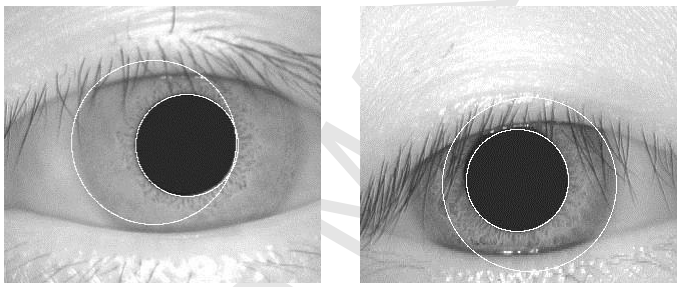
Figure 4 shows some segmentation examples that have been successfully executed. Note that even with the heavily shrouded contour, it is possible to correctly target the iris. For the detection of the eyelids, the linear Hough Transform is used.



(a) Minor difficulty (b) Major difficulty

Fig. 4. Successful segmentation examples

It is possible that errors occur during the segmentation of the eye images, causing the generation of a low quality iris codes and therefore prejudicing the comparison result. Figure 5 shows two examples of segmentation failures.



(a) (b)

Fig. 5. Examples of iris segmentation failures

4.1.2 Normalization

The circular shape of the iris does not favor comparison. Daugman proposed normalizing the iris to make it rectangular with fixed dimensions. Figure 6 illustrates the transformational process.

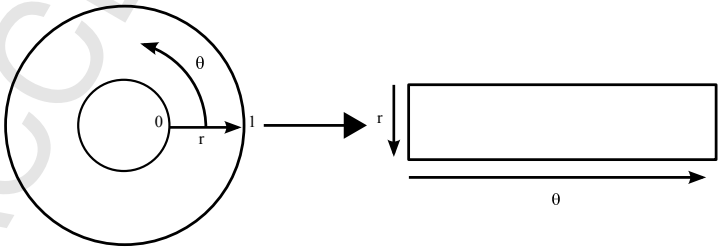


Fig. 6. Iris normalization

The normalization process consists of converting the polar coordinates into linear coordinates, adjusting the maximum and minimum to a rectangle of fixed size. As the result is a rectangle of fixed size, the difference in thickness caused by dilation and contraction of the iris is eliminated. The same normalization process is also performed for the mask, which indicates the areas that are valid to be used as iris texture.

4.1.3 Iris binary code

The normalization result still has a large amount of information and a great deal of light interference. So, it is not ideal for direct comparison. For this purpose, Daugman proposed an image convolution using the Gabor's 2D filter. The convolution result is an array of complex numbers.

Our implementation uses a similar method based on Daugman's work. However, it applies the convolution to the standard iris image using the 1D Log-Gabor wavelets [21]. The result is an array of complex numbers with dimension 8×128 bits.

The obtained matrix is then encoded according to the phase of the complex number. A complex number is replaced by 2 bits according to its location. Figure 7 shows this exchange visually. In this work, an iris code is composed of 2 parts: the code part, which is an array of binary numbers resulting from the aforementioned conversion process and it is of 8×256 bits; and the mask part, which is also as binary array that is resized to 8×256 bits, with 1 indicating the locations where the iris texture is valid and 0 indicating the locations of obstacles, such as eyelids and eyelashes.

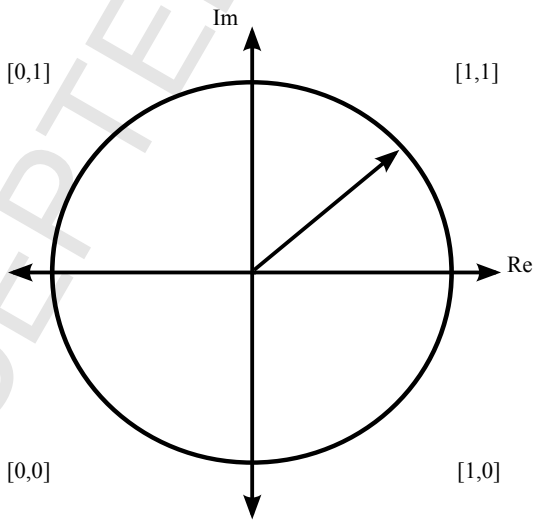


Fig. 7. Proposed codification of the iris code

4.2 Iris Comparison

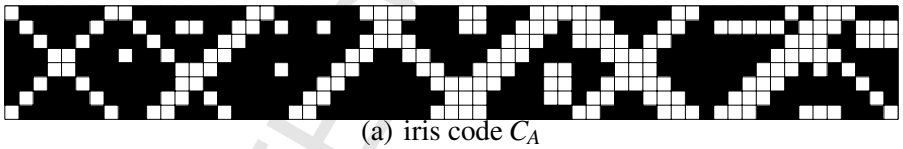
As explained earlier, an iris code consists of a code and a mask, wherein the latter indicates the points where the former is valid. Both are represented by an array of 8×256 bits, adding up to 2048 bits each. In order to perform the comparison of two iris codes, the Hamming distance between them is exploited.

The Hamming Distance is the number of different valid bits between the compared codes compared to the total of bits. Equation 1 defines the Hamming distance between two iris codes, named A and B.

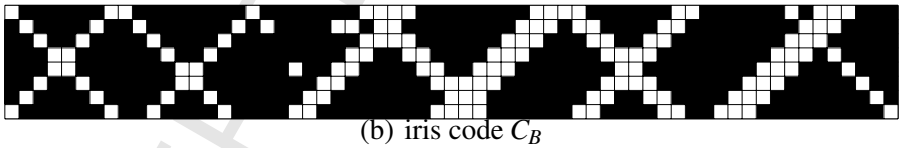
$$HD = \frac{\|(C_A \oplus C_B) \cap (M_A \cap M_B)\|}{\|M_A \cap M_B\|}, \tag{1}$$

wherein $\oplus$ is the binary operator XOR, $\cap$ the binary AND operator. Moreover, C_A and C_B represent the code part of iris codes A and B respectively while M_A and M_B represents the mask part of iris codes A and B, respectively.

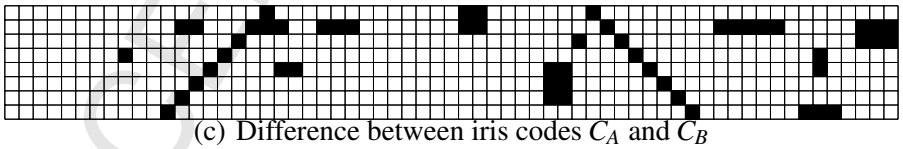
Because the process expressed in Equation 1 is at the heart of the proposed work, we give in the following an illustrative example of its main steps and the obtained results therein. Figure 8 illustrates the first step of the computation of the Hamming distance between two iris codes, wherein black cells are 1-bit while white cells represent 0-bits. So, we first compute the difference between iris code of Figure 8(a) and that of Figure 8 using a XOR operation. Note that Figure 8(c) shows the in black the cells that are of distinct value in the compared codes.



(a) iris code C_A



(b) iris code C_B

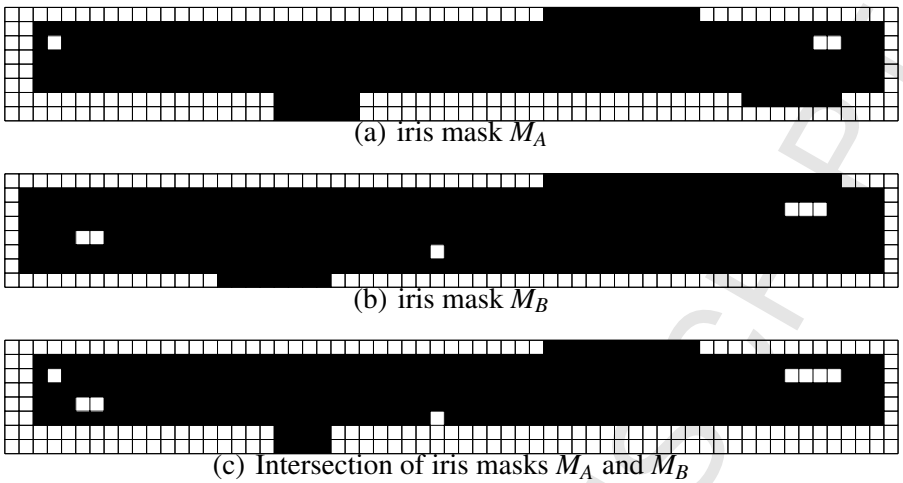


(c) Difference between iris codes C_A and C_B

Fig. 8. Comparing iris codes C_A and C_B: $D_1 = C_A \oplus C_B$

The second step is to compute the valid are of the iris considering the mask codes associated with the compared irises. Figure 9 illustrates this operation. The valid area is computed via a AND operation between the mask codes of the compared

irises M_A and M_B, shown in Figure 9(a) and Figure 9(b) respectively. Observe that Figure 9(c) shows the intersection between the mask codes.



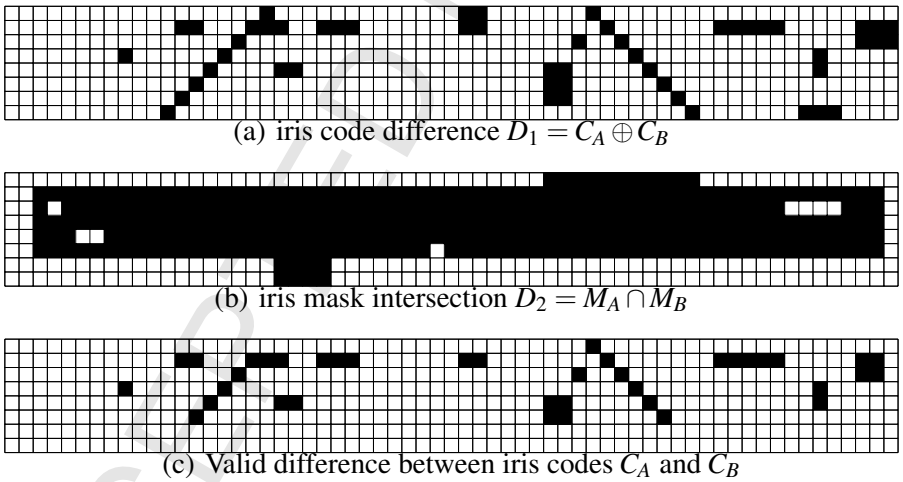
(a) iris mask M_A

(b) iris mask M_B

(c) Intersection of iris masks M_A and M_B

Fig. 9. Intersecting iris masks M_A and M_B: $D_2 = M_A \cap M_B$

Finally, the third step computes the valid differences between the compared iris codes using and the binary AND operation between the comparison results of the iris codes (D_1 in Figure 8) and that of the intersection of the associated mask codes (D_2 of Figure 9. Observe that Figure 10(c) shows only a subset of the 1-bits of Figure 8(c), *i.e.* those that are valid according to the new mask code of Figure 9(c), which is generated during the second step of the process.



(a) iris code difference $D_1 = C_A \oplus C_B$

(b) iris mask intersection $D_2 = M_A \cap M_B$

(c) Valid difference between iris codes C_A and C_B

Fig. 10. Obtaining the valid difference between iris codes $D_3 = D_1 \cap D_2$

Recall that real iris code is 8×256. However and for obvious reasons, in the illustrative example we use an iris code of 8×64. The total number of 1-bits in the valid difference between the compared codes (D_3) is 39 and that in the result of the intersection of the mask codes (D_2) is 375. Hence, the Hamming distance between the compared irises is 0.104.

The HD (Hamming Distance) represents the distance between two iris codes in percentage. However, it only considers the direct comparison between two codes. For a more secure comparison, it is possible to consider translations of the binary code. Consider a binary code from an authentic applicant to be compared to a stored code for a tentative acceptance. Direct comparison may fail due to a small rotation of the iris at the time of capture. Table 1 illustrates the array code of 8×256 bits.

Table 1
Iris code without translation

	1	2	3	4	5	6	7	8	9	...	250	251	252	253	254	255	256
1	1	0	0	0	0	1	1	0	1	...	0	1	0	1	1	0	0
2	1	0	1	0	1	0	1	0	0	...	0	1	0	1	1	1	1
3	1	1	1	0	0	1	1	0	0	...	0	1	0	1	0	1	0
4	0	1	1	0	1	0	1	1	1	...	0	1	1	1	0	0	1
5	0	0	1	0	1	0	1	0	0	...	0	0	0	1	1	1	0
6	0	1	0	0	1	0	1	0	1	...	0	0	0	1	0	1	0
7	1	1	1	0	1	0	1	1	0	...	0	1	0	1	0	1	1
8	1	0	0	0	1	0	1	1	1	...	0	0	0	1	0	0	0

4.2.1 Comparison with translation

As seen in Section 4.1.2, the extracted image of the iris passes through a normalization process, in which the patch is transformed from a circular form to a rectangular shape. Therefore, to consider iris rotation impacts on the code, it is necessary to move the columns between the limits. Table 2 shows the iris code illustrated in Table 1 with an offset of -2 bits or 2 bits leftwards. The displacement must always be done two by two, since one of them represents the real phase and the other the imaginary phase of the complex number originally extracted from the image.

Comparing the iris codes of Tables 1 to 2, one can note that the first two columns of the former have become the last column of the latter, and all columns of the former have been shifted from 2 bits leftwards in the latter. It is worth pointing out that every 2 bits represent the phase of a complex number, as seen in Section 4.1.3. The displacement of only 1 bit would cause the real part of one code to be compared to the imaginary part of another, which would certainly lead to errors.

The new rotated code of Table 2 can then be compared to stored code using Hamming distance in an attempt to obtain better results. It is important to note that the mask must pass through the same process to validate the correct bits. The translation of n bits means that the translations of $-n$ to $+n$ bits have been tested, *i.e.*, the

Table 2
Iris code com translation of 2 bits leftwards

	1	2	3	4	5	6	7	8	9	...	250	251	252	253	254	255	256
1	0	0	0	1	1	0	1	1	1	...	0	1	1	0	0	**1**	**0**
2	1	0	1	0	1	0	0	1	0	...	0	1	1	1	1	**1**	**0**
3	1	0	0	1	1	0	0	0	1	...	0	1	0	1	0	**1**	**1**
4	1	0	1	0	1	1	1	0	0	...	1	1	0	0	1	**0**	**1**
5	1	0	1	0	1	0	0	0	0	...	0	1	1	1	0	**0**	**0**
6	0	0	1	0	1	0	1	0	1	...	0	1	0	1	0	**0**	**1**
7	1	0	1	0	1	1	0	1	0	...	0	1	0	1	1	**1**	**1**
8	0	0	1	0	1	1	1	0	0	...	0	1	0	0	0	**1**	**0**

translation of 2 bits indicates that all 5 Possible translations of -2 to $+2$ will be verified.

4.2.2 Algorithm for iris code comparison

For the implementation of the iris biometrics comparisons in a smart card, the Java Card platform is used. The platform has a restriction on transferring the data to the card. The limit on the data volume for each message sent to the card is 128 bytes. However, an iris code consists of a code 2048 bit and a validation mask of the same size, resulting in a total of 512 bytes. Therefore, it would take 4 messages to download a complete iris code.

A storage form of an iris code is designed to facilitate the use of the code in general and to perform the translation operation in particular. Consider the code part of an iris code, which is an array of 8×256 bits, illustrated in Figure 11(a). The Java Card can store a maximum of 16 bits in a single variable of type short. The most usual way to allocate a matrix to a vector is to store the elements row by row. In this case, as in Figure 11(b), this consists of storing the bits from 1 to 16 of the first line in the first vector entry, the bits from 17 to 32 of the first line in the second entry and so on. Each line would be stored in a vector of 16 entries of type short ($16 \times 16 = 256$). Therefore, the array is stored in a vector of $16 \times 8 = 128$ short entries.

For this implementation, a different form of storage, as shown in Figure 11(c), is proposed and implemented. In the first vector entry, the first two columns (16 bits) are allocated, the next two columns are placed into the second and so on. This way of iris code storage also results in a vector of 128 entries of type short. Both forms of allocation will result in the same Hamming distance algorithm since the distance is computed using the XOR operation simply by comparing the stored codes in the

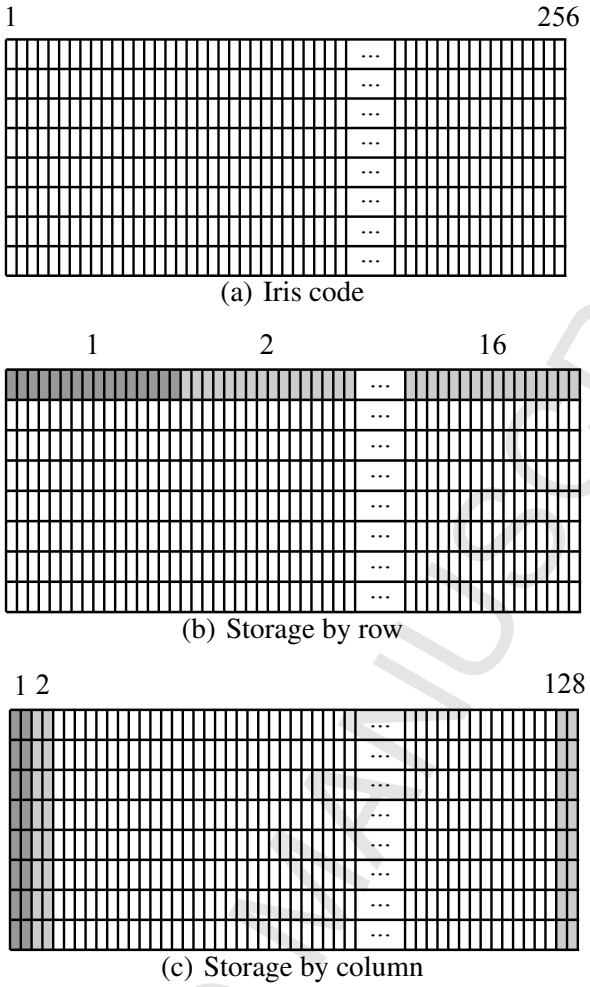


Fig. 11. Different ways of iris code storage

same way. The improvement introduced by the proposed allocation form lies in the simplicity of translating bits, as explained earlier. Each vector entry will be storing exactly the data that will undergo the displacement. If the code were stored row by row, it would be necessary to make several executions of the offset operation of bits, increasing the complexity and the number of instructions to be used, and consequently increasing the execution time.

Let N be the size of the vector where the code is stored, T the iris code, composed of the binary code T_C and the mask T_M, which is stored on the card, and E, composed of the code E_C and the corresponding mask E_M representing the iris code of the individual requesting authentication. Algorithm 1 details the steps used to implement the Hamming Distance computation, as defined in Equation 1, as efficiently as it is possible to be done on a smart card platform.

Algorithm 1 proceeds byte by byte in the iris code. It first computes the difference between the binary iris codes T_C and E_C via a simple XOR operation, establishing all the positions wherein the codes are distinct, which should be counted as

Algorithm 1 Hamming distance between iris codes

Require: T e E
Ensure: HD
1: $Nbits := 0$
2: $NbitsTotal := 0$
3: **for** $i := 1 \rightarrow N$ **do**
4: $xoredC = T_C(i) \oplus E_C(i)$
5: $mascTotal := T_M(i)$ AND $E_M(i)$
6: $xoredC := xoredC$ AND $mascTotal$
7: Count 1-bits in $xoredC$ and add them up to $Nbits$
8: Count 1-bits in $mascTotal$ and add them up to $NbitsTotal$
9: **end for**;
10: $Nbits := 10 \times Nbits$
11: $NbitsTotal := NbitsTotal/10$
12: $HD := Nbits/NbitsTotal$

mismatches. Then, it intersects the binary masks T_M and E_M via a simple AND operation, establishing all the valid bits in the binary iris code that need to be verified and matched to declare a hit. Subsequently, in order to identify the mismatching bits in the compared binary codes T_C and E_C that really matter, *i.e.* those that are valid according to the configuration of masks T_M and E_M, the algorithm computes the intersection of the previously computed difference with the intersection result of the masks. The Hamming distance between the compared binary codes is actually the total number of set bits in the thus obtained intersection $Nbits$. However, as we work with relative Hamming distance, the algorithm proceeds by counting the total number of bits in the intersection of the masks $NbitsTotal$ and computes the ratio between the $Nbits$ and $NbitsTotal$. Nonetheless, the strategy of multiplying the dividend $Nbits$ by 10 and dividing the divisor $NbitsTotal$ by 10 is used before calculating the ratio without manipulating float variables and without extrapolating the maximum allowed value of variables of type short, which ranges from -32768 to $+32767$. The resulting ratio is always between 0 and 100.

5 Performance Results

The aim of this work is to use the smart card to process iris matching operation and thereby increase the security level. During the card configuration, the iris code of the owner is transmitted to the card so that it is stored for future matching upon access request. In order to have access to the protected service, the card holder must provide its iris code, as an input, to confirm his/her identity through the computation of the Hamming distance to the stored template. As pointed out earlier, for implementation purposes, we used the Java Card platform. In the following, before we get to the performance results and the underlying discussion, we first present the

datasets used together with their specificities. Nonetheless, it is noteworthy to point out in advance that no comparison with third party implementations is possible due to intellectual property protection on commercial iris matching on smart cards. Note that any other kind of implementation either on general purpose or dedicated hardware would be worthless and somehow biased.

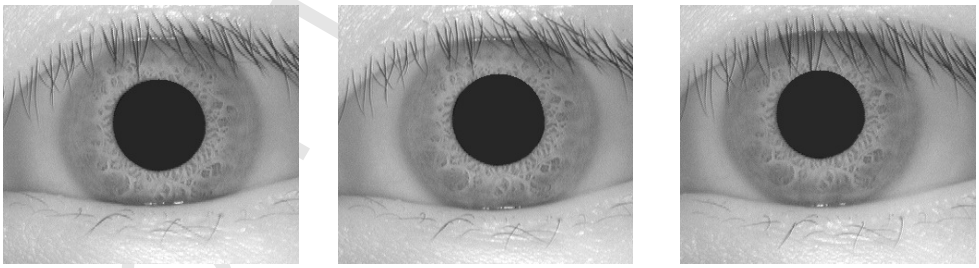
5.1 Iris texture Database

To test and evaluate the performance of the comparison algorithm implemented on the smart card, two different databases are used. These datasets have different characteristics regarding the acquisition procedure. Both of them have been collected by the Chinese Academy of Sciences, Institute of Automation (CASIA) [24].

5.2 CASIA Iris V1

The database in question is one of the first that has been made freely available to study iris biometrics. It was collected using a camera for infrared capture. It has 756 images from 108 different eyes with 7 samples each. The available images have a resolution of 320×280 pixels and are stored in BMP format.

In order to intellectually protect the procedure of the capture project, the inner part of the pupil, where the reflections of the light would appear, as necessary for the capture of the images, was replaced by a dark region of constant color. Note that this automatic processing does not affect the iris. Although very commonly used, the automatic post-processing of the images is pointed as a negative factor for the evaluation of extraction methods. This is because it makes the segmentation of the inner circle in the iris simpler [25].



(a) Sample *A* (b) Sample *B* (c) Sample *C*

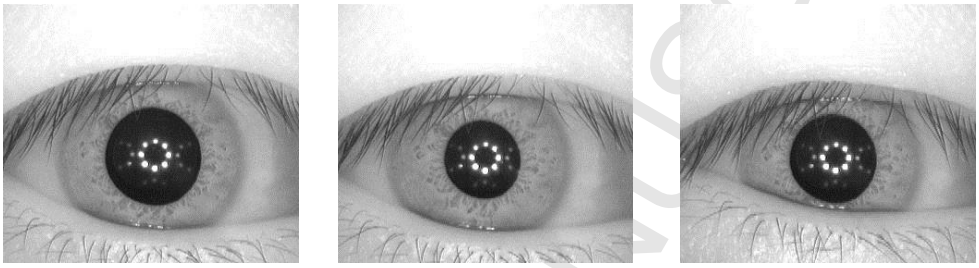
Fig. 12. Iris image samples from CASIA Iris V1

Figure 12 shows three images captured from the same eye. One of the characteristics of this database is the small variation of the eye during capture. The presented iris images show small variations. The lack of challenge makes the database ideal

for the validation of the comparison algorithm. Nonetheless, this is not the case in real-world usage of the biometry, wherein challenges are most expected.

5.3 CASIA Iris V4 Interval

After the distribution of CASIA Iris V1, the same institution constructed new databases with several characteristics for different types of iris biometry studies. The CASIA Iris V4 Interval includes 2369 images of 249 different eyes captured by a proper camera with approximation. The number of repetitions of each eye is variable. The images have also resolution of 320×280 pixels.



(a) Dilated pupil (b) Contracted pupil (c) Covered pupil

Fig. 13. Iris image samples from CASIA Iris V4 Interval

Figure 13 shows three samples from the same eye. The images were chosen in order to exemplify the big challenges provided by the samples of this database. This causes the capture to become more similar to real-world use where some small differences are expected.

Among other iris images datasets as distributed by CASIA, the Interval type is chosen because the included images have a low resolution yet a high approximation, thus generating images with high iris sharpness. Moreover, the fact that the iris images were captured in two sessions in distinct time periods makes it possible to be used to analyze the fixedness of the biometrics.

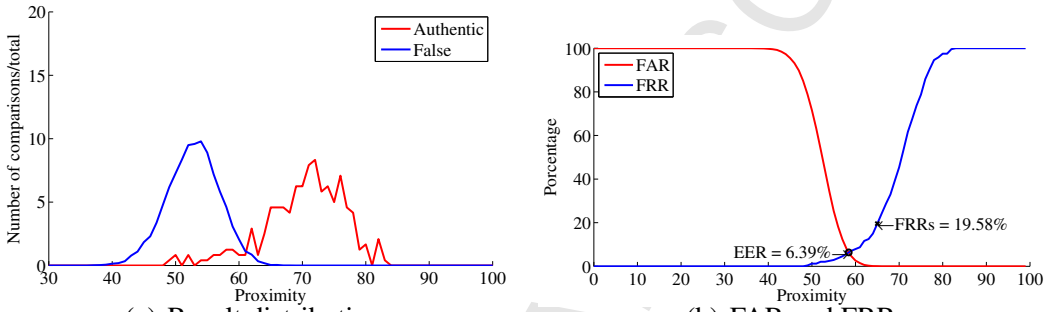
Recall that the extraction of the iris code is not the focus of this work. So, in this purpose, we used the available tool in [21]. Libor Masek compiled an extractor and comparator using the MATLAB tool based on the work published in [13]. The extractor is used to generate the iris codes for comparison on the smart card. In [4], Masek's extractor is pointed out as capable of achieving good results, which contributed to the choice of its use in the design of this work.

In the following, the performance results of the comparisons made on Java Card cards using the CASIA V1 and CASIA V4 Interval databases are presented and analyzed. The results are presented in terms of the proximity measure, which is defined as $100 - HD$.

5.4 *Results achieved for CASIA V1*

For the first test, images of 40 different eyes were randomly selected and for each eye, 4 replicates were used, resulting in a total of 160 iris images. Before the comparison, the iris codes were extracted using the tool introduced in Section 4.1.

Figure 14 shows the results of the $160 \times 160 = 25600$ comparisons performed in this test. Figure 14(a) shows the distribution of the results for authentic and false comparisons regarding their respective totals while Figure 14(b) shows the graph regarding the rates achieved for the FRR and FAR for different values of proximity.



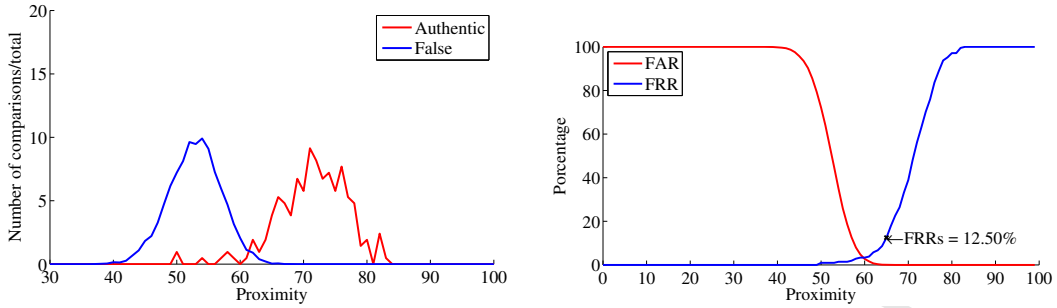
(a) Result distribution (b) FAR and FRR

Fig. 14. Comparison results as achieved for CASIA V1

The distribution graph shows that authentic comparisons are concentrated on higher proximity values while false comparisons were concentrated at lower values. The graphs that relate errors to proximities indicate that the point where errors are equal (EER) approaches 6%, but this is a very high value for an acceptable FAR, since an individual other than the owner of the card could need 20 attempts to be granted access. The point at which the FAR rate is less than 0.1%, which known as secure FRR, is chosen as a safe boundary of proximity, and for that proximity the FRR is around 20%. This means that the authentic card user would have their access denied once every 5 attempts.

The CASIA V1 database has no challenging eye images (see Section 5.2), but there are still possible errors in the segmentation of images. Figure 15 shows the comparison results for the CASIA V1 database wherein iris images with segmentation failures are excluded. As expected, the results are significantly better, showing a clear distinction between the distribution of concentrations of authentic and false comparisons, as it can be clearly observed in Figure 15(a). Similarly, the EER decreased and the FRR also secured. This can be seen in Figure 15(b).

Figure 15(b) shows the comparison between the rates achieved. From the presented results it is possible to conclude that the extraction has a great impact on the comparison process result, since it is responsible for doubling the error rate. Although

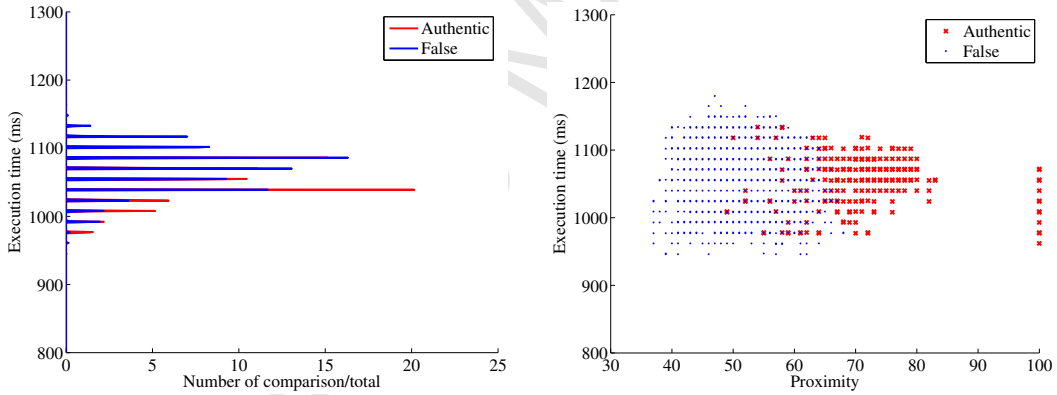


(a) Result distribution (b) FAR and FRR

Fig. 15. Comparison results as achieved for CASIA V1 wherein iris with segmentation failure are not considered

implying an increase in error rate, failures during segmentation cannot be eliminated but we can only attempt to reduce their impact.

In order to reduce the impact of segmentation failures, we proceed with the translation of the iris code as explained in Section 4.2.1. The obtained distributions of the execution time in terms of the total performed comparisons and in term of the proximity are illustrated in Figure 16. Note that the result shows no dependency between the proximity result and the execution time.



(a) Execution time (b) Proximity *vs.* time

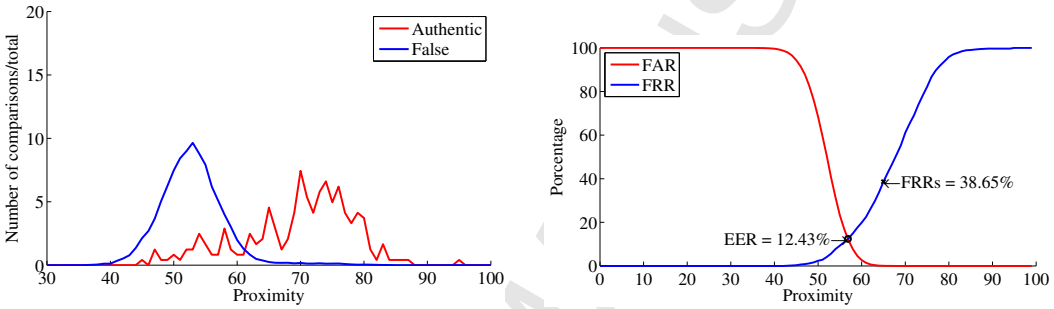
Fig. 16. Execution time when a translation of 1 bit is applied

For authentic comparisons, the average execution time is 1052.48ms while the standard deviation is 51.93. On the other hand, for false comparisons, the average execution time is 1071.36ms with a standard deviation of 33.06. AS it was explained earlier, 4 message exchanges are required to completely transfer a given iris code. Each message exchange takes 180ms, summing up 720ms for a complete transfer. It is noteworthy to point out that approximately 70% of the total execution time is due to data transfer.

5.5 *Results achieved for CASIA V4 Interval*

The CASIA V4 Interval database was introduced in Section 5.3. It is a database with images captured in a more realistic way. It is has some real problems related to eye rotation, iris occlusion among other difficulties. It includes 200 images of several eyes. Several rotations were chosen randomly due to the fact that the database does not have a fixed number of repetitions for each considered eye. All iris codes were extracted and no image was disregarded.

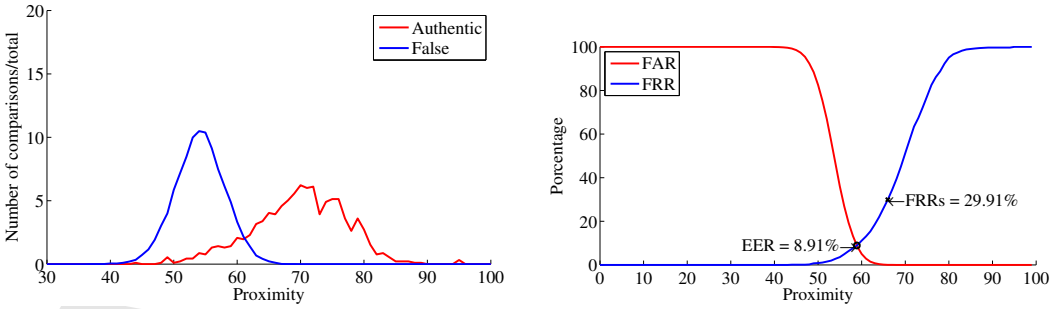
Figure 17 shows the results of proximity comparisons. Compared with the results of the comparisons using CASIA V1, as presented in Figure 14, it is easy to observe that the result is much lower, illustrating once again the different characteristics between the databases.



(a) Result distribution (b) FAR and FRR

Fig. 17. Comparison results for CASIA V4 Interval

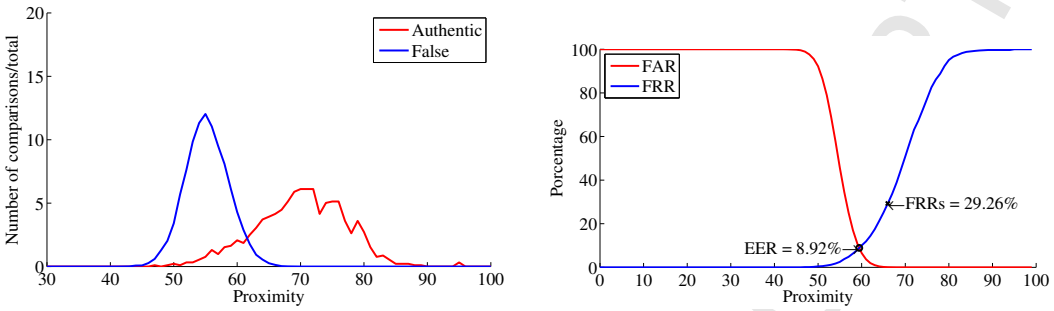
The aforementioned bad performance raises the need to apply a method that improves the effectiveness of the comparison algorithm. As explained earlier, we will apply the bits translation proposed method to the iris codes of the dataset. Using this method it is possible to improve the results thus making it more suitable for real-world use. Figure 18 shows the result of the comparisons using the translation of 2 bits.



(a) Result distribution (b) FAR and FRR

Fig. 18. Comparison results as achieved for CASIA V4 considering translation of 2 bits

The safe FRR result shows a large improvement, reducing from 38.65% to 29.91%. This improvement illustrates the need to use translations when there is efficacy during authentic comparisons is required. Figure 19 presents the results of the comparisons with 4-bits translation.



(a) Result distribution (b) FAR and FRR

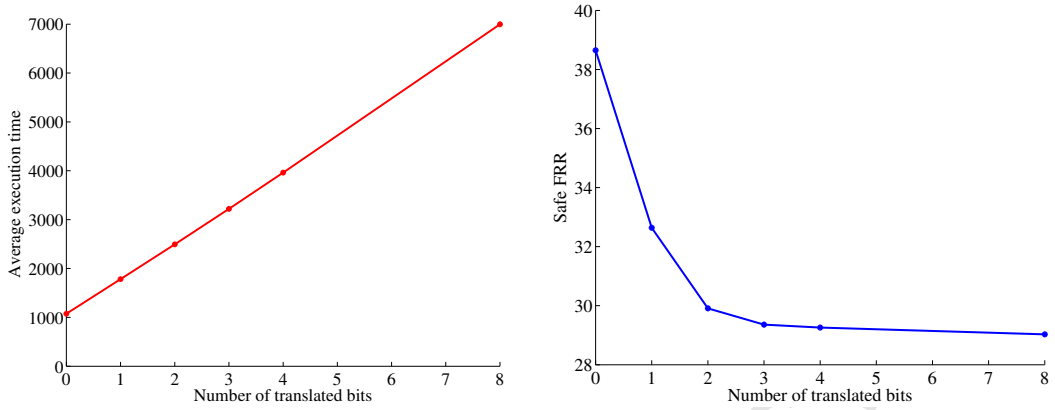
Fig. 19. Comparison results as achieved for CASIA V4 considering translation of 4 bits

It is important to note that with the translation of 4 bits, only slight improvement of the results is achieved. The improvement regarding the safe FRR is about 0.65%, *i.e.* reducing from 29.91% to 29.26%. In fact, it does not seem very meaningful as it certainly entails a large increase in terms of execution time. In order to analyze the results of the proximity *vs.* execution time using different translations, tests are performed using translations of 1, 2, 3, 4 and 8 bits. The Table 3 summarizes the obtained results for all the considered translations.

Table 3
Comparison results for different translations

Translation	Safe FRR	Average Execution time (ms)	Standard deviation
0	38.65	1,074.07	49.84
1	32.64	1,782.24	86.92
2	29.91	2,495.32	139.73
3	29.36	3,220.02	193.78
4	29.26	3,963.08	249.67
8	29.03	6,998.28	503.63

Figure 20 depicts the results presented in Table 3. So, Figure 20(a) is regarding the relates the average execution time using different translations of bits. The rising graph has a similar slope between all points indicating a linear relationship, in which 700ms is added for each additional translation. Figure 20(b) depicts the secure FRR achieved for the different translations of bits. Note that in the translations of 1 and 2 bits, a large decrease of the safe FRR occurs. However, with the subsequent bit translations, only a limited improvement is introduced.



(a) Execution time (b) Safe FRR

Fig. 20. Achieved results for CASIA V4 considering different translations

From the translation of 2 bits, the safe FRR reaches no more than 29% while the execution time increases linearly. To mitigate the time impact, it is possible to improve the time of authentic comparisons stipulating an acceptance threshold so that the result is anticipated without the requirement of all comparison's completion. In the next section, we first define what we consider as acceptance threshold; then we present and discuss the results of the comparisons using this improvement.

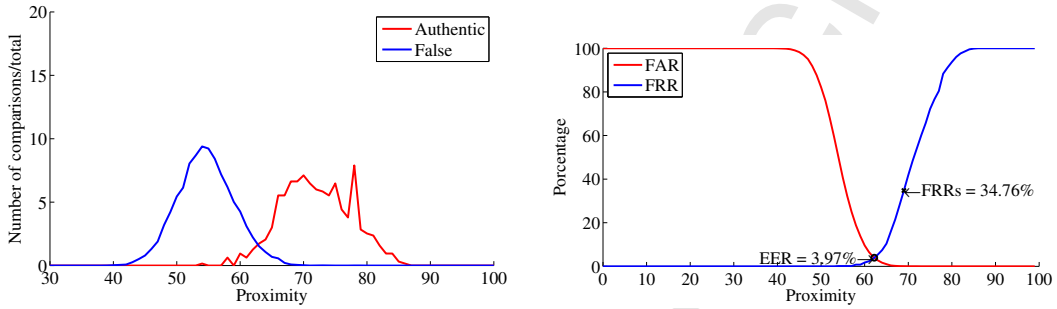
5.6 Comparison with acceptance threshold

The acceptance threshold is nothing more than a given proximity value from which the comparisons should be considered correct, *i.e.*, after a predefined proximity has been reached, the card processing unit will already have the comparison response, and thus may terminate the comparison process regarding direct comparisons followed by comparisons with translations and hence return the produced result. It is noteworthy to emphasize that using an acceptance threshold to abort the execution of the remaining comparisons can only reduce the execution time regarding authentic cases, but does not affect the execution time of false comparisons since in this case the execution will not be interrupted. The comparisons with no bit translations perform the calculation of *HD* only once, and thus cannot be interrupted. The acceptance threshold is implemented for comparisons with translation of 1 and 2 bits.

Based on the results of Section 5.5, the comparison that considers an acceptance threshold is implemented using as a basis the comparison of 2-bit translation. This choice is based on the observation, in Fig. 20, of the execution time and safe FRR achieved in this case compared to other translations. For the 2-bit translation, the execution time increases but the secure FRR remains practically the same. The acceptance threshold can thus be safely set as the proximity of 66 because for all comparisons considering all translations, the FAR is less than 0.1%, *i. e.*, the se-

lected proximity is superior to the situation wherein the safe FRR is achieved. In order to make the tests more realistic and reliable, 200 new eye images of the CASIA V4 Interval database are introduced.

Figure 21 shows the results of the comparisons. The achieved FRR is about 34%. However, for the chosen proximity as the acceptance threshold, the resulted FAR is about 0.42% while the FRR of about 15.95%. This is mainly due to the change in the analyzed set of images. The result shows that for every 1000 false comparisons, 4 will be considered authentic and for every 100 authentic comparisons, 16 will be declared as false.



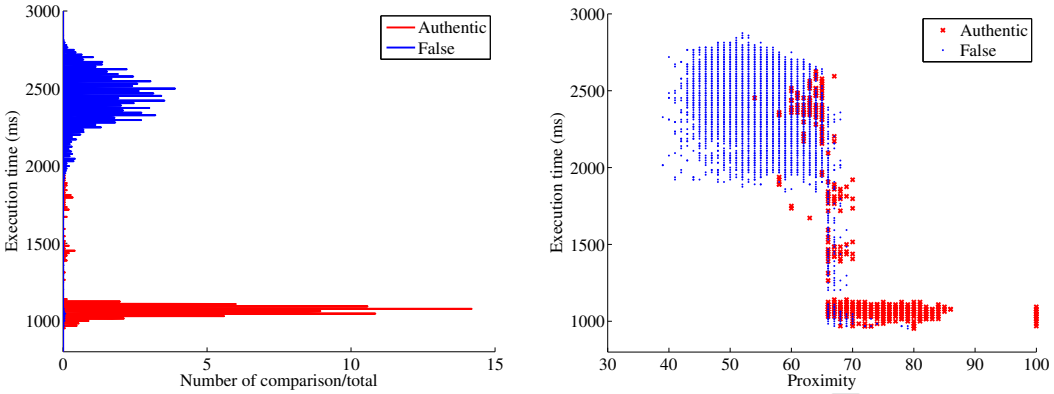
(a) Result distribution (b) FAR and FRR

Fig. 21. Comparison results considering a acceptance threshold

Figure 22 shows the execution times, which benefited a big deal from the use the acceptance threshold. The average execution time for authentic comparisons is 1210ms with a standard deviation of 395, while the average time for false comparisons is 2430ms with a standard deviation of 189. The results indicate that the proposed method is capable of dramatically decreasing the average time in case of authentic comparisons. Note that in a real system, this type of comparisons are probably the most requested. Figure 22(b) illustrates the relationship between execution *vs.* proximity. It indicates, together, Figure 22(a), that the highest concentration of authentic comparisons did not complete the whole cycle of translations.

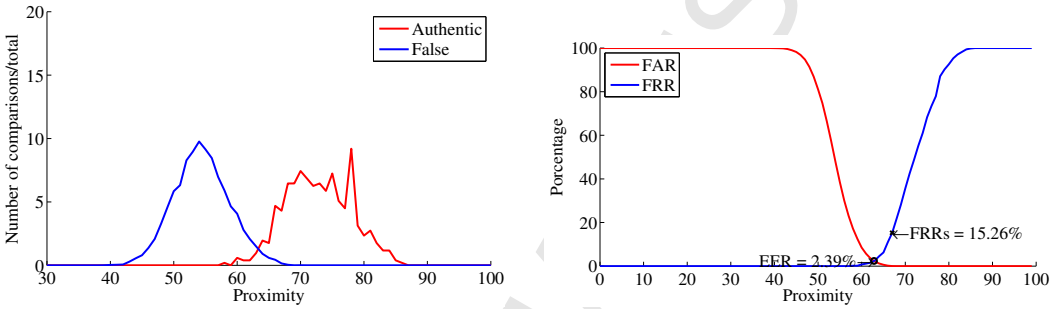
In order to better understand the results of the comparison when an acceptance threshold is used, taking into account only the cases where errors occurred, we removed from this analysis the cases related to eye images that did not have a satisfactory extraction. Thus, the results regarding comparisons in the case of only good quality iris codes are illustrated in Figure 23. These prove that the safe FRR has greatly decreased in relation to the result using iris codes, including the faulty samples.

Because the proposed method uses an acceptance threshold, it would be more correct to analyze the result based on this limit, since the execution of the verification is always halted when the proximity reaches the imposed acceptance limit. For the



(a) Execution time (b) Proximity *vs.* time

Fig. 22. Execution time for comparison with acceptance threshold



(a) *Result distribution* (b) FAR and FRR

Fig. 23. Comparison results when ignoring extraction failures

proximity acceptability limit of 66, the achieved FAR is 0.2% while the corresponding FRR is about 10.96% in contrast to a 0.42% FAR and 15.95% FRR, as obtained in the case wherein the segmentation failures were included. There is a decrease in both FRR and FAR. Again, the results show the importance of the extraction process as it impacts the achieved result significantly. Despite that iris recognition is a challenging biometrics due mainly to the fact that it has so many obstacles, the method used is robust and capable of providing very good results.

6 Conclusions

In this paper, the different possibilities of comparison iris codes using different translations are discussed and analyzed. The use of an acceptance threshold has proven to be a good decision based on the translation of 2 bits. Considering the performed tests, the best results achieved yielded an error rate of approximately 10%, proving that the iris biometry is robust and easy to use as a biometric. It also becomes clear that a good extraction process is of paramount importance for the effectiveness of iris-based authentication.

As future work, we intend to study other types of biometrics, such as fingerprint and palm print comparison in order to verify the possibility of their implementations on smart cards. Moreover, as a multi-application card, it is also possible that the same smart card offers more than one type of biometric verification. Thus, we plan also to enhance privacy and security by the combining or even merging the implementation of different biometrics on the same smart card. We expect that this make the offered protection very strong and quasi impossible to breach.

Acknowledgments

We are eternally grateful to the reviewers and editor that allowed for a great deal of improvement of the content and contribution of this paper. We also would like to thank CNPq, FAPERJ and CAPES, which are basic research sponsoring agencies in Brazil, for their continuous financial support.

References

[1] Biometric smartphones to reach 100% adoption as samsung brings iris biometrics to market. *Biometric Technology Today*, 2016(9):1, 2016.

[2] M. A. M. Abdullah, F. H. A. Al-Dulaimi, W. Al-Nuaimy, and A. Al-Ataby. Smart card with iris recognition for high security access environment. In *1st Middle East Conference on Biomedical Engineering*, pages 382–385, Sharjah, United Arab Emirates, 2011.

[3] Fernando Alonso-Fernandez and Josef Bigun. A survey on periocular biometrics research. *Pattern Recognition Letters*, 82, Part 2(10):92–105, 2016.

[4] Kevin W Bowyer, Karen Hollingsworth, and Patrick J Flynn. Image understanding for iris biometrics: A survey. *Computer vision and image understanding*, 110(2):281–307, 2008.

[5] Theodore A Camus and Richard Wildes. Reliable and fast eye finding in close-up images. In *16th International Conference on Pattern Recognition*, volume 1, pages 389–394, Quebec, Canada, 2002. IEEE.

[6] Yarui Chen, Jucheng Yang, Chao Wang, and Na Liu. Multimodal biometrics recognition based on local fusion visual features and variational bayesian extreme learning machine. *Expert Systems with Applications*, 64(12):93–103, 2016.

[7] Yi Chen, Sarat C Dass, and Anil K Jain. Localized iris image quality using 2-d wavelets. In *Advances in Biometrics*, pages 373–381. Springer, Beijing, China, 2005.

[8] Lu Chenhong and Lu Zhaoyang. Efficient iris recognition by computing discriminable textons. In *International Conference on Neural Networks and Bra*, volume 2, pages 1164–1167, Beijing, China, 2005. IEEE.

[9] Chia-Te Chou, Sheng-Wen Shih, Wen-Shiung Chen, and Victor W Cheng. Iris recognition with multi-scale edge-type matching. In *18th International Conference on Pattern Recognition*, volume 4, pages 545–548, Hong Kong, China, 2006. IEEE.

[10] Healthcare Council. Smart cards and biometrics in healthcare identity applications. Technical report HCC-12001, Smart Card Alliance, May 2012.

[11] John Daugman. Statistical richness of visual phase information: update on recognizing persons by iris patterns. *International Journal of Computer Vision*, 45(1):25–38, 2001.

[12] John Daugman and Cathryn Downing. Epigenetic randomness, complexity and singularity of human iris patterns. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1477):1737–1740, 2001.

[13] John G Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1148–1161, 1993.

[14] Y. Du. Using 2D log-Gabor spatial filters for iris recognition. *Proc. SPIE, Biometric Technology for Human Identification III*, 6202(62020F), 2006.

[15] Anjith George and Aurobinda Routray. A score level fusion method for eye movement biometrics. *Pattern Recognition Letters*, 82, Part 2(10):207–215, 2016.

[16] Gael Hachez, Jean-Jacques Quisquater, and Francois Koeune. Biometrics, access control, smart cards: a not so simple combination. In A. Watson J. Domingo-Ferrer, D. Chan, editor, *Smart Card Research and Advanced Applications*, pages 273–288. Springer, Bristol, Inglaterra, 2000.

[17] Farmanullah Jan. Segmentation and localization schemes for non-ideal iris biometric systems. *Signal Processing*, 133(4):192–212, 2017.

[18] Yuanning Liu, Senmiao Yuan, Xiaodong Zhu, and Qingliang Cui. A practical iris acquisition system and a fast edges locating algorithm in iris recognition. In *Instrumentation and Measurement Technology Conference*, volume 1, pages 166–169, Vail, CO, USA, 2003. IEEE.

[19] Li Ma, Tieniu Tan, Yunhong Wang, and Dexin Zhang. Personal identification based on iris texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1519–1533, 2003.

[20] Li Ma, Tieniu Tan, Yunhong Wang, and Dexin Zhang. Efficient iris recognition by characterizing key local variations. *IEEE Transactions on Image Processing*, 13(6):739–750, 2004.

[21] Libor Masek et al. Recognition of human iris patterns for biometric identification. Master's thesis, Masters thesis University of Western Australia, 2003.

[22] Di Miao, Man Zhang, Zhenan Sun, Tieniu Tan, and Zhaofeng He. Bin-based classifier fusion of iris and face biometrics. *Neurocomputing*, 224(2):105–118, 2017.

[23] Nadia Nedjah, Rafael Soares Wyant, and Luiza de Macedo Mourelle. Efficient biometric palm-print matching on smart-cards for high security and privacy. *Multimedia Tools and Applications*, pages 10.1007/s11042–016–4271–8, 2017.

[24] Chinese Academy of Sciences Institute of Automation. CASIA iris database. http://biometrics.idealtest.org/, 2004. Last access: October in 2012.

[25] P Jonathon Phillips, Kevin W Bowyer, and Patrick J Flynn. Comments on the CASIA version 1.0 iris data set. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1869–1870, 2007.

[26] Ajita Rattani and Reza Derakhshani. Ocular biometrics in the visible spectrum: A survey. *Image and Vision Computing*, 59(3):1–16, 2017.

[27] Wei Shu and David Zhang. Automated personal identification by palmprint. *Optical Engineering*, 37(8):2359–2362, 1998.

[28] Zhenan Sun, Tieniu Tan, and Yunhong Wang. Robust encoding of local ordinal measures: A general framework of iris recognition. In *International Workshop on Biometric Authentication*, pages 270–282, Beijing, China, 2004. Springer.

[29] Hanho Sung, Jaekyung Lim, Ji-hyun Park, and Yillbyung Lee. Iris recognition using collarette boundary localization. In *17th International Conference on Pattern Recognition*, volume 4, pages 857–860, Cambridge, England, 2004. IEEE.

[30] Richard P Wildes. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, 85(9):1348–1363, 1997.

[31] Buket Yüksel, Alptekin Küpçü, and Öznur Özkasap. Research issues for privacy and security of electronic health services. *Future Generation Computer Systems*, 68:1–13, 2017.

[32] David Zhang, Wai-Kin Kong, Jane You, and Michael Wong. Online palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1041–1050, 2003.

Nadia Nedjah is an associate professor in the Department of Electronics Engineering and Telecommunications at the Faculty of Engineering, State University of Rio de Janeiro, Brazil. Her research interests include functional programming, embedded systems and reconfigurable hardware design as well as cryptography. Nedjah received her Ph.D. in Computation from the University of Manchester - Institute of Science and Technology (UMIST), England, her M.Sc. in System Engineering and Computation from the University of Annaba, Algeria and her Engineering degree in Computer Science also from the University of Annaba, Algeria. Contact her at nadia@eng.uerj.br.

Raphael Wyant Soares holds a Master degree in Electronics Engineering from State University of Rio de Janeiro. His research interests include hardware architecture, FPGA in particular, and intelligent systems in general. Contact him at rapahel.wyant.soares@gmal.com.

Luiza de Macedo Mourelle is an associate professor in the Department of System Engineering and Computation at the Faculty of Engineering, State University of Rio de Janeiro, Brazil. Her research interests include computer architecture, embedded systems design, hardware/software co-design and reconfigurable hardware. Mourelle received her PhD in Computation from the University of Manchester - Institute of Science and Technology (UMIST), England, her MSc in System Engineering and Computation from the Federal University of Rio de Janeiro (UFRJ), Brazil and her Engineering degree in Electronics also from UFRJ, Brazil. Contact her at ldmm@eng.uerj.br.

Brij B. Gupta received PhD degree from Indian Institute of Technology Roorkee, India in the area of Information and Cyber Security. In 2009, he was selected for Canadian Commonwealth Scholarship and awarded by Government of Canada Award ($10,000). He spent more than six months in University of Saskatchewan (UofS), Canada to complete a portion of his research work. He has published more than 80 research papers (including 01 book and 08 chapters) in International Journals and Conferences of high repute including IEEE, Elsevier, ACM, Springer, Wiley Inderscience, etc. He has visited several countries, i.e. Canada, Japan, Malaysia, Hong-Kong, etc. to present his research work. At present, Dr. Gupta is working as Assistant Professor in the Department of Computer Engineering, National Institute of Technology Kurukshetra, India. His research interest includes Information security, Cyber Security, Mobile Security, Cloud Computing, Web security, Intrusion detection, Computer networks and Phishing.

Highlights

1. An efficient implementation of iris texture verification on smart-cards.
2. For this implementation, the matching is done on-card.
3. The biometric characteristics are always kept in the owner's card, guaranteeing the maximum security and privacy.
4. The False Acceptance Rate (FAR) and False Rejection Rate (FRR) are improved using circular translations of the matched iris-codes.
5. The proposed technique is augmented with acceptance threshold verification, thus decreasing drastically the execution time of the matching operation.
6. Achievement of a considerably low FAR and FRR.