

## IRIS RECOGNITION WITH MATCH-ON-CARD

Adam Czajka, Przemek Strzelczyk, Marcin Chochowski and Andrzej Pacut

Research and Academic Computer Network NASK  
ul. Wąwozowa 18, 02-796 Warszawa, Poland

Institute of Control and Computation Engineering  
Warsaw University of Technology  
ul. Nowowiejska 15/19, Warszawa, Poland

phone: + 48 22 3808 152, fax: + 48 22 3808 201

emails: {P.Strzelczyk,M.Chochowski}@elka.pw.edu.pl, {A.Czajka,A.Pacut}@ia.pw.edu.pl  
web: www.BiometricLabs.pl

### ABSTRACT

The paper presents a biometric smart card that supports on-card matching. The card supports two- or three-factor verification of the card holder (biometrics, smart card and optionally password or PIN). We use a novel iris coding based on Zak-Gabor transform, which may adapt to image quality. Basic properties of the proposed coding are presented in the paper. The biometric smart card is an element of Iris Recognition System, which also includes eye aliveness detection. The system, evaluated with a proprietary database of iris images, shows very favorable results.

### 1. INTRODUCTION

One-factor authentication systems, including the ones based on biometrics, may fail to guarantee a satisfactory level of access control security. To extend the base solutions to two- or more-factor verification systems, PIN or cryptography can be merged with biometrics. Typically, biometric data is kept in a central database, and must travel between the data owner and a verification unit. In some applications, however, there is no need of central biometric repositories. The data owner can be even the only subject entrusted with his or her biometric data. This solution calls for secure storage devices able to perform *biometric matching* with the desired accuracy and speed. Smart cards presently meet such conditions. They offer storage of biometric data within the secure card environment. Reading the data out of the microprocessor is cryptographically restricted or may even be blocked. Simultaneously, smart cards may also offer biometric template matching, resulting in the so called *match-on-card* solutions, in which the biometric template never leaves the card. This is in a contrast with the *match-off-card* scenarios, where the biometric template is retrieved from the card and the matching is performed outside of the card microprocessor. Certainly, biometric matching place depends on the entire system architecture and requirements. In this work we focus on match-on-card solutions.

### 2. CHOICE OF SMART CARD ARCHITECTURE

Smart cards, referred to also as microprocessor cards, undeniably prevail over the magstripe cards and became one of the most important data carriers if a secure and distributed storage of personal data are needed. Possibility of secure on-card transformation of the data enables to construct secure storage

and processing units for biometric systems. To employ efficiently a smart card in a biometric system, one must select a platform which enables to implement biometric matching procedures. This implementation should be independent of the platform manufacturer or the card proprietary hardware. JavaCards meet these requirements.

JavaCards are equipped with a card operating system and a simplified version of Java language interpreter JavaCard Virtual Machine (JCVM), Fig. 1. Due to limited hardware architecture, JavaCards do not support some functionalities fundamental to Java language, e.g., multi-threading, floating point and 32-bit integer calculations, and automatic memory management (garbage collector). Undeniable advantage of JavaCards over – for instance – the native cards is the code interoperability (“write once, run anywhere”), yet proprietary language extensions should not be used. This platform was selected to build a *biometric smart card*. Biometric applets presented in the paper were successfully launched without code adaptations on cards manufactured by Gemplus and Giesecke & Devrient.

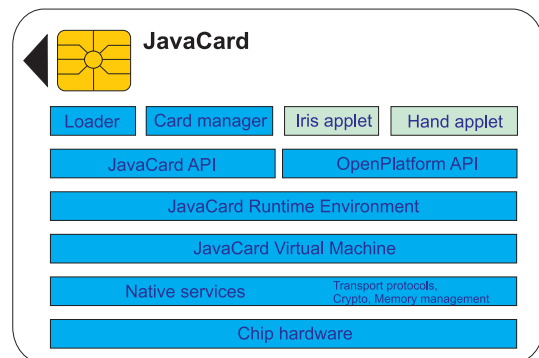


Figure 1: JavaCard architecture. Biometric applets developed by the authors are marked.

### 3. SMART-CARDS-BASED TRANSACTION SECURITY

We can distinguish three parties in smart-card-based transactions, namely the card, the card holder (possibly not the card owner) and the terminal (e.g., ATM), Fig. 2.

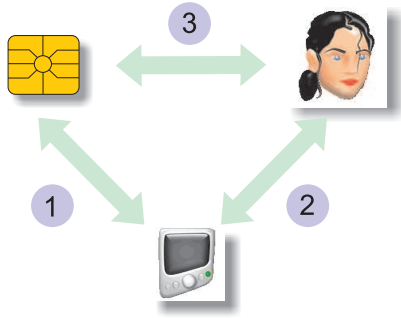


Figure 2: Three players in a smart-card-based transaction: the card, the card holder and the terminal.

A smart-card-based transaction is said to be safe if *all* the transaction players can be mutually authorized and the exchanged information was not eavesdropped or modified. We briefly describe how these requirements can be fulfilled, and show that biometric verification may be of a particular help here.

### 3.1 Terminal – smart-card authentication

Definition of secure protocols for authentication and communication with smart cards, as proposed by GlobalPlatform [1], includes three levels of security:

- mutual authentication, upon which the card and the terminal prove to possess the same secret (static cryptographic key),
- integrity and data origin authentication, which guarantees that messages are sent by a trusted terminal, they are not modified, and come in a proper order,
- confidentiality, which guarantees that the communication was not eavesdropped.

These mechanisms may guarantee security of the channel marked as “1” in Fig. 2. We briefly describe these mechanisms.

#### 3.1.1 Mutual authentication

Mutual authentication is a two-stage process initiated by the terminal that enables authenticating the card and the terminal mutually. The procedure uses static keys loaded into the terminal and into the card during personalization. Additionally, within this authentication procedure, a session key is established that may be further used in a secure messaging.

#### 3.1.2 Integrity and confidentiality of communication

Three communication security levels are allowed after successful mutual authentication. The first, and the least secure, does not require any additional security mechanisms, and the messages sent between the card and the terminal in a secure channel are not additionally protected. The second level forces the data integrity and proper order of messages by signing each message. Signature is generated by C-MAC or R-MAC algorithms with the chaining mechanism. This guarantees that the messages are received unchanged and in a proper order. The third security level, apart from data integrity, encrypts the messages content with a session key.

### 3.2 The terminal – the user authentication

Mutual authentication does not prove authenticity of the terminal to the user (channel marked as “2” in Fig. 2), since the smart card cannot present the result of the terminal verification in a simple way. Thus we propose to store on the card a secret message, known only to the legitimate card holder. This message can be displayed properly on the terminal screen, and is correct only upon a successful mutual authentication.

### 3.3 The card – the user authentication

The third channel, marked as “3” in Fig. 2, must be secured as well, i.e., it is necessary to prove the user identity to the card. If the card authenticates the user, the terminal, which already trusts the card, should also accept the user. Presently, the most popular approach to this task is to use PIN or other secret. Note however that this mechanism does not prevent from illicit smart-card usage when the PIN is stolen. To fill this gap, we propose to integrate biometrics and smart card, making the latter one a *biometric smart card*. We developed a biometric smart card with the use of a proprietary iris recognition methodology, yet the same principles are valid for other biometric modalities. In the next section we present the iris methodology applied in this work and the method of implementing the match-on-card iris recognition.

## 4. BIOMETRICS AND SMART CARDS

### 4.1 Iris recognition

We use a novel iris coding based on Zak-Gabor transformation with built-in mechanism to adapt to different iris image resolutions and quality [4]. A dedicated hardware was designed and constructed to capture the iris from a convenient distance, with the desired speed and a minimal user cooperation. The system uses the pupil position estimated in real-time to guide a person to position the eye, and to release the image capturing process. In this process several frames are captured at varying focal lengths to compensate for the small depths-of-field typical in iris imaging. The sharpest frame is selected for further analysis.

The raw images contain the iris and its surroundings, and the iris must first be localized. To detect the boundary between the pupil and the iris, we propose a method sensitive to circular dark shapes, and unresponsive to other dark areas as well as light circles, such as specular reflections. This may be achieved by a modified Hough transform that uses the *directional image* to employ the image gradient, rather than the *edge image*, which neglects the gradient direction. The boundary between the iris and the sclera is detected using Daugman’s integro-differential operator [3].

Based on the localized occlusions, we select two opposite  $90^\circ$  wide angular iris sectors. The positions of these sectors are stored together with the iris template. This guarantees that the same iris regions of the same eye are transformed in each comparison. Each iris sector is then resampled and smoothed to a  $P \times R$  rectangle, where  $P = 512$  and  $R = 16$ . The rows of these two rectangles will be further referred to as the *iris stripes*, and will be processed independently. We thus simplify the iris 2D pattern to a set of 1D patterns, with a certain loss of information. Figure 3 illustrates the preprocessed iris image and the corresponding iris stripes.

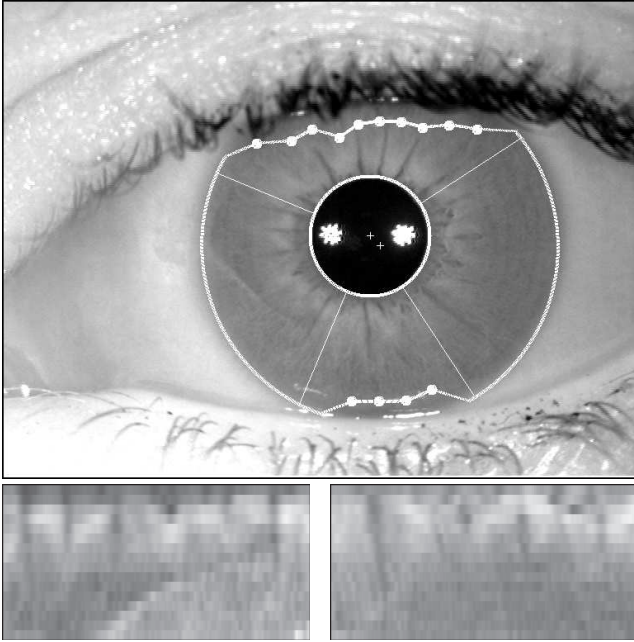


Figure 3: *Top*: Raw camera image processed by our system. The eyelids were automatically detected, and the sectors free of occlusions (marked as white full circles) are selected. Star-like shapes on the pupil are reflections of the illuminating NIR diodes, and the '+' marks represent the pupil and the iris centers. *Bottom*: Iris stripes automatically determined for the image shown on the top.

The iris stripes are not space-homogeneous. Their spatial frequency contents should be analyzed locally, with the use of *space-frequency* or *space-scale* analysis. There exist various tools to represent the signal in the mixed space-frequency domain. A family of Windowed Fourier Transforms apply Fourier Transform to windowed signals in time or space. The Gabor transform belongs to this family, and uses Gaussian windows characterized by their widths. The window width significantly influences the resulting iris features and must be carefully chosen. We use the space-frequency analysis that employs waveforms indexed by space, scale and frequency simultaneously, what results in a larger set of possible tiling in the space-frequency plane, possibly redundant. This directs our methodology towards the *wavelet packet* analysis.

We use the signs of the Gabor expansion coefficients as the iris features. Gaussian-shaped windows are not orthogonal, i.e., the inner product of any two windows is nonzero, therefore Gabor's expansion coefficients cannot be determined in a simple way. For this purpose we use the fastest method of Gabor's expansion coefficients determination applying Zak's transform [2]. This application of the Zak transform is often referred to as Zak-Gabor's transform. Our implementation of Zak-Gabor's transform is based on Fast Fourier Transform, thus the iris features calculation times are proportional to those in the FFT.

There is a need to select the appropriate frequencies and scales simultaneously to make Zak-Gabor's transformation sensitive to the individual features of the iris image. A systematic procedure for this purpose is embedded in our iris coding to minimize the recognition error for a given set of

iris images. The procedure relies on analysis of Fisher information (the quotient of within-eye to between-eye variabilities of iris features) calculated for each coefficient of Gabor's expansion. This approach enables our method to be applied for databases of images of various resolution, including low quality images like those originating from mobile phones. The resulting iris features are gathered in 1024-bit vectors, and the order of bits is kept constant for each eye. This enables using Hamming distance in the iris matching, applying exclusive OR operations. We stress that the resulting iris code should not be confused with the *iriscode*<sup>TM</sup> invented by Daugman [3]. The latter one is the result of Gabor's filtering, while our method uses Gabor's expansion coefficients as iris features. Table 1 presents the iris image acquisition and processing times, measured for our iris recognition system.

Table 1: Iris image acquisition and processing times, averaged for 720 iris images.

Task	Average time [s]
Head positioning by skilled volunteer	2.5
Acquisition of frames	1.0
Best frame selection	1.5
Iris boundary localization and occlusions detection	2.819
Representation of iris image as a sequence of stripes	0.586
Zak-Gabor coefficients calculation and transformation into a features vector	0.06

Small eyeball rotations in consecutive images may lead to considerable deterioration of within-eye comparison scores. Typically, the relative eyeball rotation for two iris images is close to 0. In tests using our *BioBase* database, which consists of 720 iris images taken for 180 eyes (4 images of one eye are available), the rotations did not exceed 4° for 95% of images. The maximal rotation measured for our database was 11.5°. The eyeball rotation can be corrected by maximizing the correlation within the enrolled images. However, during verification, the iris image corresponding to the template is unavailable, hence we use an iterative minimization of the comparison score between Zak-Gabor's-based features determined for a set of small artificial shifts of the iris stripes. Our iris recognition system using this method, tested in laboratory conditions with *BioBase* dataset, resulted in no sample recognition errors.

## 4.2 Biometric smart card

A biometric verification scenario with the smart card use is illustrated in Fig. 4. Since our iris matching is based on XOR and summation operations, it is relatively easy to be implemented within the card environment.

We must still correct the eyeball rotation, typically present during verification. We found that approximately 80% of the entire transaction times (measured from mutual authentication up to returning of the verification result) are consumed by the communication between the terminal and the card. Thus the application of iterative minimization, as described above, must be limited. Moreover, transfers of partial recognition results to the minimization procedure raise a risk of hill-climbing attacks. We thus introduce three other

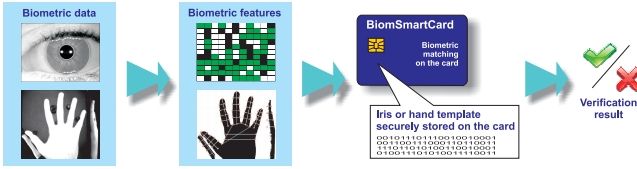


Figure 4: Biometric verification with on-card-matching. Besides the iris biometrics, a hand biometrics [5] was also included in the biometric smart card, to direct it toward multi-modal solutions.

mechanisms to compensate for the eyeball rotation when the iris features are matched on the smart card.

The first mechanism, marked M1, is to create a reference template (i.e., the template stored on the card) that contains several elementary iris feature vectors for different rotation angles of the verified image. The vectors are created for angular shifts less than  $1.5^\circ$ . In the matching process, the Hamming distance is calculated between the authentication template (i.e., the template corresponding to the iris image being verified) and each vector from the reference template stored on the card. If the distance between at least one reference template vector and the authentication template vector falls below an acceptance threshold, the verification is successful. The second mechanism (M2) is similar to M1, with a difference that the size of the authentication template is increased, instead of the reference one. The reference template feature vector is compared with all vectors included into the authentication template. The third method (M3) combines the above two, by enlarging both the authentication and reference templates and cross-comparing all vectors included into authentication and reference templates. There is also a possibility to entirely abandon the eyeball rotation compensation only if the relative rotation is less than  $1.5^\circ$ . The properties of all approaches are summarized in Table 2.

Table 2: Different on-card matching variants depending on eyeball correction mechanisms (M1 – M3). Variant with no eyeball rotation correction is also shown. The results were obtained with a BioBase dataset.

Feature	M1	M2	M3	No correction
Reference template	Multiple vectors	Single vector	Multiple vectors	Single vector
Authentication template	Single vector	Multiple vectors	Multiple vectors	Single vector
Communication time overhead*	$T$	$NT$	$NT$	$T$
Memory usage	2.2kB**	154B	2.2kB**	154B
Average on-card matching time (sec.)	0.5	0.6	1.2	0.2
Maximum on-card matching time (sec.)	1.7	1.9	30	0.2
Performance	Good EER=0%	Good EER=0%	Good EER=0%	Poor EER=8.6%

\*  $T$  – time of sending the feature vector to the card (depends on security mechanisms, typically  $T \leq 10$  sec);  $N$  – number of elementary feature vectors in the template

\*\* the memory usage depends on  $N$ , in this paper we set  $N = 17$

Due to the flexible design, the iris biometric applet allows to use any of the presented mechanisms. In the results shown above the multi-vector templates contain  $N = 17$  elementary vectors. When the quality of the reference template is high, there are limited differences in performance among the first three approaches. Because the largest size of the template (2.2kB) is acceptable for the smart cards that have 32-128kB of memory, the best choice is M1 mechanism, which combines good performance with the least communication overhead.

## 5. SUMMARY

The iris biometrics was selected to develop a biometric smart card supporting on-card matching. Thanks to flexible implementation in JavaCard technology, the biometric applets can be ported to any current smart card chip and Card Operating System (COS) compliant with JavaCard technology. The multi-factor authentication biometric smart card presented in this paper may belong to one of the first biometric cards that employ iris biometrics.

There are several straightforward extensions of the biometric smart card presented here. The applet can be enriched with interactive authentication to select iris sectors for verification. Selection of different sectors in each transaction may form a countermeasure against the biometric replay-attacks, since the comparison of iris codes for different sectors gives the rejection. It is possible to enrich the approach with additional biometric modalities (e.g., hand), thus making the card a multimodal solution. We also see a straightforward application of the biometric smart card in undeniable electronic signature generation. In this case, this functionality is unlocked only after a successful biometric authentication.

## 6. ACKNOWLEDGEMENT

Part of the above results were obtained during the BioSec European integrated project IST-2002-001766.

## REFERENCES

- [1] *GlobalPlatform Card Specification*. Version 2.1.1, March 2003.
- [2] Martin J. Bastiaans, “Gabor’s Expansion and the Zak Transform for Continuous-Time and Discrete-Time Signals”, in Josh Zeevi and Ronald Coifman (Eds.), *Signal and Image Representation in Combined Spaces*, pp. 1–43, Academic Press, Inc., 1995
- [3] John Daugman, “How Iris Recognition Works”, *IEEE Transactions on circuits and systems for video technology*, Vol. 14, No. 1, January 2004
- [4] Andrzej Pacut, Adam Czajka, Przemek Strzelczyk, “Iris biometrics for secure remote access”, in: J.S. Kowalik *et al.* (Eds.), *Cyberspace Security and Defense: Research Issues*, pp. 259–278, Springer, 2005
- [5] Łukasz Stasiak, “Support vector machine for hand geometry-based identity verification system”, in: Ryszard S. Romaniuk (Ed.), *Proceedings of SPIE - Volume 6347, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2006*, Vol. 634725, Oct. 12, 2006