Smart Card
Information and Operations using Biometrics

Raul Sanchez-Reillo
E.T.S.I. Telecomunicacion

ABSTRACT

Nowadays many systems need a portable media to store some sensible data, such as smart cards. The information can be protected by the user with his Personal Identification Number (PIN), or through biometrics. Unfortunately, there is not a smart card today that can verify the biometric template inside it, performing this task in the terminal. The author has developed the algorithms and data structures needed to solve this problem. Therefore, he has created a smart card with user biometric authentication, based on an Open Platform smart card (in this case, a JavaCard). To achieve these results, different biometric techniques have been studied: speaker verification, hand geometry and iris recognition. Experimental results are given to show the viability of the prototype developed.

INTRODUCTION

When an Information System needs to store some information in a distributable way, one of the possibilities is to give each user an identification token which stores his personal information, as well as any other information requested by the system designed. Some of that information, such as financial data, health care record, etc., should be protected by the user who owns the card (i.e., the cardholder). This protection is usually made with CHV-Keys (i.e., Card Holder Verification Keys), which are closely related to the user's PIN. In a smart card, security with this kind of key can involve allowing or denying not only access to some information, but also the possibility of performing some operations, such as a debit from an electronic purse. But this protection is better because a smart card has the possibility to block access for life if a determined number of wrong presentations of the PIN has been reached.

Unfortunately, PINs as passwords, can be copied by inspecting the cardholder movements as he enters his number in, for example, an ATM. The only way to perform a real user authentication, is through biometrics. But biometric verification cannot be found in any commercial smart card nowadays. The only efforts being applied in this line is to store the user's template inside a smart card, protected with Administrative Keys, and extracted from the card by the terminal to perform the verification.

This paper presents the works being carried by the author in order to achieve a new authentication system. In this authentication system, the user's template is stored inside a smart card, with the same level of protection as any other key. Then, when the user wants to authenticate himself to the card, he asks for such an operation, giving one biometric sample, which is verified inside the card. If the verification is positive, the card allows the access to the biometrically protected information and/or operations.
Fig. 1. Views of the system developed

To develop the prototype, the author has studied different biometric techniques, which will be mentioned in the next paragraph. Then the results obtained with those techniques are compared in order to obtain restrictions to its integration inside a smart card. After that, the way the integration has been performed is explained, giving the results obtained with three different Open Operating System smart cards. At the end of the paper overall conclusions will be given.

BIOMETRIC TECHNIQUES ATTEMPTED

Biometrics refers to identifying a human being through biological and/or behavioural parameters. Therefore, a lot of biometric techniques exist nowadays [4], such as fingerprint, hand geometry, voice, iris, retinal scanning, written signature, etc.

It is not really useful to consider one technique much better than any other, because each technique has its own characteristics that make it suitable for determined environments. Nevertheless, several biometric techniques have being studied and many commercial systems available have been tested by the German BKA Office, and the results have being reported by A. Zwiesele in [17]. In these results, Iris Recognition was proven to achieve the best figures, although the cost of any of the commercial systems available makes it unviable to be applied for most situations.

The author, in order to obtain general conclusions about the possibility of integrating biometric authentication inside a smart card, have deeply studied several techniques. From these, three of them were also developed by the author: Speaker Recognition, Hand Geometry and Iris Identification. This section is intended to give an overall introduction to each of these three techniques.

Speaker Recognition

In this technique, the user’s voice is captured with a microphone, then pre-processed and passed through a block that is in charge of extracting the cepstrum coefficients. From all the verification methods being applied to the human voice for Speaker Recognition [1], the author has used Gaussian Mixture Models (GMM), due to its lower memory requirements and better results [5].

In order to obtain the user’s template, 60 seconds of continuous speech from the user should be taken to train the GMM. Once trained, only 3 seconds of the user’s speech are needed to perform the biometric verification.

The user’s template is 175 bytes long, while the sampling rate that is passed to the verification algorithm, i.e., the GMM, is 3000 bytes long. Considering the computation time spent, the enrollment phase (when the user’s template is extracted) last for about 1000 seconds, while the time needed for the verification is 16 seconds. Unfortunately, the error rates obtained for a 1-time verification process (if the first verification fails, the user is rejected) are above 10%, obtaining an Equal Error Rate (EER) of 20.9%.

Hand Geometry

As seen in Figure 1, this technique is based on taking a photograph of the user’s hand [6][8][9]. The hand is placed on a platform painted in blue to increase contrast with all kinds of skin. Six tops are placed on the platform to guide the hand in front of the medium resolution digital camera. The contour of the hand is obtained using edge-extraction algorithms [3]. Thanks to a mirror located on the right of the platform, the lateral view of the hand is also obtained and its contour extracted.

Several measurements are performed on both contours, obtaining a feature vector of 25 bytes. With 5 photographs from his hand, the user’s template is computed, depending on the verification method used. The methods tested have been: a) the Euclidean Distance; b) the Hamming Distance; and c) GMM. The results obtained can be seen in Table 1.

Human Iris

From all the biometric techniques known today, the most promising is Iris Identification – affirmation of this is confirmed by the results reported in [12] – due to its low error rates, nearly null False Acceptance Rate (FAR) and without
Fig. 2. Samples taken from the database

being invasive [2][7][8]. As can be seen in Figure 2, a high resolution photograph of the user’s eye is taken and pre-processed to extract the inner and outer boundaries of the iris. After that, Gabor filters are used to extract the feature vector, which is 233 bytes long. For a deeper information about the prototype developed by the author, the reader can refer to the description given in [10].

The user’s template is obtained from a single sample, and the Hamming Distance is used to verify the samples with the template stored. Both, the template and the sample size are 233 bytes. The enrollment time is determined by the pre-processing and feature extraction blocks, which can be optimized from the results obtained in the prototype developed (142 seconds). The verification lasts 9 ms. The False Rejection Rate for 1 time verification is 3.51% with a null FAR.

It is important to note that the results given about these three techniques, especially those referring to the time spent in the algorithms, have been obtained with the execution of the algorithms on a PC through the programs developed in a mathematical development software called MATLAB. Unfortunately, this development system, although having many advantages, does not offer a good performance in computation time, and those results can be easily bettered by coding those algorithms in C or C++.

COMPARISON AMONG RESULTS

From the results achieved with the above mentioned techniques, several conclusions can be obtained focusing on the possibility of integrating biometric authentication into a smart card:

- Due to the sample vector size and verification times, Speaker Recognition is not considered as a viable technique for the purposes stated in this paper. Also, the error rates achieved should be improved.

- Error rates obtained with the Euclidean Distance in the Hand Geometry technique are not good enough to consider it for a medium/high security system. However, due to its simple verification algorithm, it could be interesting to integrate this technique in a 3-time retry authentication method, improving, therefore, the error rates.

- Hand Geometry and Iris Identification, both with Hamming Distances, seem to fit perfectly with the purposes of this paper.

- Hand Geometry with GMMs achieve really good results. However, its computation cost and the need of using floating point operations, will disable - as stated below - the possibility of integrating it inside a smart card. Further work should be applied to enable this possibility.

INTEGRATION INSIDE A SMART CARD

In order to perform the integration, two possibilities exist:

1) building a whole new mask, which is a very high cost process but is able to achieve the best results;

2) using an Open Operating System smart card, such as a JavaCard [14], which is a low cost process but has certain constraints given by the platform used.

The author has chosen the second way to implement the biometric authentication inside a smart card, using three different products from two different manufacturers.

The main difference between one of the cards of one manufacturer and the other two cards used, is the processor architecture used. While the other two use a 16-bit CISC processor, that one uses a 32-bit RISC processor.

The prototypes implemented were built with the following elements:

- Some environmental variables that indicate blocking, successful verification, number of maximum and remaining tries, possibility of unblocking, etc. of the biometric template.

- CREATE-TEMPLATE-FIle function, which creates the file where the user’s template is going to be stored, giving some parameters, such as the algorithm that applies to perform the verification.

- WRITE-TEMPLATE-FIle function, that writes the data corresponding to the user’s template, including the maximum number of erroneous verifications allowed. This function is based on the UPDATE-BINARY standard function.

- PERSONAL-AUTHENTICATION function. This function was coded as the standard function accepted by the industry.

- TEMPLATE-UNBLOCK function, which can be executed only if the template file is defined as having the possibility of being unblocked. This will depend on the level of security needed by the application where the card is going to be used.

- Several testing commands (APIs) to verify the right functioning of the prototype.

All these elements have been developed following the available standards, such as the ISO 7816. The results obtained with the three cards mentioned above (sorted in two categories according to the architecture of its processor) can be seen in Table 2, on next page.

Three main considerations should be made from these results. The first one is that Hand Geometry with GMMs has not been covered. This has been impossible because JavaCard
specifications do not accept floating-point operations. The second consideration is that times obtained with the RISC processor are much lower than the ones obtained with the CISC processor, enabling sophisticated verification methods in the near future. Unfortunately the cost of the RISC processor manufacturing has led us to forget, for awhile, this version of the card. The last consideration is that verification time obtained with the CISC processor, which is higher than an acceptable time of half a second, can be easily lowered by optimising the binary functions in the JavaCard platform.

CONCLUSIONS

A prototype for performing biometric authentications inside a smart card was presented. Three biometric techniques have been studied to analyse their viability. These techniques are Speaker Recognition, Hand Geometry and Iris Identification. The results obtained show the possibility of integrating biometrics as a Card Holder Verification method, therefore, improving user authentication in smart card based applications. Better results can be obtained building a new smart card mask, instead of using an Open Operating System card, such as the JavaCards used in the prototypes developed. If this last possibility is not possible, results with the RISC based JavaCard are good enough for a commercial product. Further efforts will be applied to integrate other biometric techniques in the prototypes developed, such as fingerprint or facial recognition.

ACKNOWLEDGEMENTS

The authors thank the work being carried out by Carmen Sanchez-Avila, Ana Pilar Gonzalez-Marcos and Jose A. Martin-Pareda.

We also thank the Spanish CICYT TIC98-1195 for funding for the works being carried out.

REFERENCES


Table 2.

<table>
<thead>
<tr>
<th></th>
<th>RISC</th>
<th>CISC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of the prototype code</td>
<td>2121</td>
<td>1511</td>
<td>B</td>
</tr>
<tr>
<td>Authentication Time (Hand Geometry, Euclidean D2)</td>
<td>5.11</td>
<td>121</td>
<td>ms</td>
</tr>
<tr>
<td>Authentication Time (Hand Geometry, Hamming D2)</td>
<td>30.1</td>
<td>127</td>
<td>ms</td>
</tr>
<tr>
<td>Authentication Time (Iris Identification)</td>
<td>7.11</td>
<td>1230</td>
<td>ms</td>
</tr>
</tbody>
</table>


