IRIS VERIFICATION SYSTEM WITH SECURE TEMPLATE STORAGE

Tiago Marques Santos[1], Luís Ducla Soares[1,2], and Paulo Lobato Correia[1,3]

[1]Instituto de Telecomunicações, [2]ISCTE – Instituto Universitário de Lisboa, [3]Instituto Superior Técnico,
Torre Norte - Piso 10, Av. Rovisco Pais, 1, 1049-001, Lisboa, Portugal
phone: + (351) 218418461, fax: + (351) 218418472, email: {tms, lds, plc}@lx.it.pt

ABSTRACT

In this paper, a secure biometric recognition system, that guarantees user privacy and revocable templates, is proposed. The data stored in the system database reveals no information about the original biometric features and it is practically impossible to recover them from the stored data. Revocability is ensured by generating different templates for the same user, using the same biometric data, just by changing a parameter in the secure template scheme. To achieve this goal, a combination of an error correcting code and a hash function is used. The recognition performance of the proposed system is not significantly affected when compared to the same system without template security. The estimated number of security bits it is between 98 and 171.

1. INTRODUCTION

The use of biometrics (e.g., fingerprints, irises, faces) for recognizing individuals is becoming increasingly popular and many implementations and products are already available. These biometric systems can be divided in two different categories: *verification* and *identification* [1][2][3]. Verification systems authenticate a person's identity by comparing the captured biometric sample with that person's own biometric template previously stored in the system, while identification systems recognize an individual by searching the entire template database for a match with the captured biometric characteristic. In this paper only verification systems will be considered since this corresponds to the case where the proposed security enhancements are more relevant, as will soon become clear.

When compared to the use of passwords or personal identification numbers (PIN) in verification systems, the use of biometrics presents several advantages, the first one being that biometric characteristics are intrinsically associated with an individual and cannot be forgotten or shared with others. In addition to this, an adequately chosen biometric characteristic has much higher entropy than poorly chosen passwords and is, therefore, less susceptible to brute force attacks. Finally, systems that rely on biometric verification require very little user expertise and, therefore, can easily be widely deployed.

However, biometrics also have disadvantages when compared to passwords. A problem associated with the use of biometrics is that once a biometric characteristic is chosen, the same biometric can be used to access different systems. This means that, if it is compromised, an attacker could have access to all the accounts, services or applications that use the same biometric characteristic. This is the equivalent of using the same password across multiple systems, which can lead to some very serious problems in terms of security, as can easily be understood. For instance, it was discussed in [4] that one of the most relevant vulnerabilities of biometrics is that once a biometric image or template is stolen, it is stolen forever and cannot be reissued, updated, or destroyed.

In particular, embedded devices, such as smart cards, are especially vulnerable to eavesdropping and attack [5]. Thus, protection solutions that achieve a secure storage of the reference biometric template need to be investigated. Recently, novel cryptographic techniques, such as fuzzy commitment and fuzzy vault, were proposed [6][7]. These schemes include error correcting codes (ECC) to allow protecting data subject to acquisition noise, as is the case of biometric samples. Clancy et al. [8] employed the fuzzy vault scheme on a secure smart card system, where fingerprint authentication is used to protect the user's private key. Yang, et al. [9] further addressed the issue to develop an automatic and adaptive recognition system. Linnartz et al. [10] precisely formulated the requirements for protecting biometric authentication systems, presenting a general algorithm meeting those requirements. The feasibility of template-protected biometric authentication systems was further demonstrated in [11].

In [12], Vetro et. al proposed the usage of syndromes for the secure storage of biometric data, using as a biometric characteristic the iris. The selected error correcting code was a low-density parity-check (LDPC) code, which is used together with a hash function. In this implementation, the template length is fixed at 1806 bits. This is done because the error correcting code needs to be applied to the same number of bits for every verification attempt. Therefore, to achieve this, the bits in positions more likely to be affected by noise due to eyelids, eyelashes and misalignments in the radial orientation are discarded. This system achieves 50 bits of security, without significantly affecting the system's recognition performance when compared to the system without security.

This paper proposes a secure biometric verification system, which will have enhanced security template storage when compared to existing systems, exploring the combined usage of distributed source coding and hash functions. The chosen biometric characteristic is the iris, since it has been reported

The authors acknowledge the support of Fundação para a Ciência e Tecnologia (FCT), under Instituto de Telecomunicações' project BIONSE.

to provide some of the best results for verification systems and it remains fairly unaltered during a person’s lifetime.

The rest of this paper is organized as follows. Section 2 presents the proposed architecture for a secure biometric verification system, while the implementation details are described in Section 3. Section 4 presents the performance evaluation of the implemented secure system. Finally, to conclude the paper, some final remarks about the strengths of this type of approach are presented in Section 5, as well as an outline of the envisioned future developments.

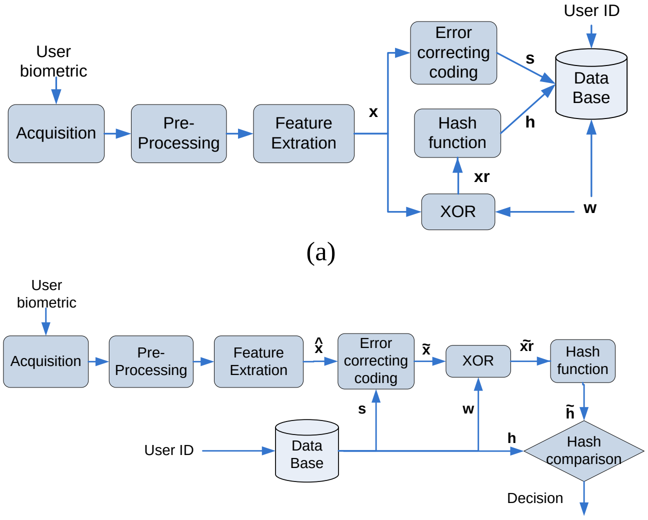
2. PROPOSED SYSTEM ARCHITECTURE

After having identified the advantages and disadvantages of using biometrics in verification systems, it would be interesting to design a verification system that preserves all the advantages of biometrics described above but manages to eliminate the mentioned disadvantages. To do so, in addition to the typical requirements imposed on biometric systems, the following requirements have to be met:

- **Different templates from the same biometric** – If a given biometric is used to access many different systems, this may lead to a highly insecure situation. After all, if one of the systems is compromised, then all the other systems that use the same biometric will also be potentially compromised. This leads to a requirement of being able to generate many different biometric templates from the same biometric trait. This way, even if one of the systems becomes compromised, the other systems will not be.
- **Cancellable templates** – The generation of cancellable biometric templates [13] is a way of dealing with the problem of stolen biometric templates. When a non-cancellable biometric template is stolen or compromised in some way, it cannot be reissued. Keeping in mind that an individual has a limited number of biometric characteristics, this becomes a mandatory requirement. This way, if an enrolled biometric template is somehow compromised, it can be simply deleted and a new one is issued, still based on the same biometric characteristic.
- **Private biometrics** – This requirement guarantees that the original biometric data cannot be recovered from the stored data, thus remaining private [14]. This is of the utmost importance, as all biometric templates are generated from the original data, which means that if an attacker has access to it, all the systems that use the same biometric characteristic could be potentially compromised.

The proposed secure biometric verification system, which relies on distributed source coding principles and cryptographic hash functions, is able to meet these requirements. This is reflected in the proposed system architecture, which is illustrated in Figure 1.

In the enrolment stage of a typical biometric verification system, after the biometric acquisition module, some processing is applied in order to obtain the biometric template, **x**, which is then stored in a database. Here, however, the biometric data is never stored in the database to prevent it from being stolen. Instead, after the acquisition and template generation, an error correcting code and a cryptographic hash function are applied to it in parallel. The hash is applied to the result of a bitwise exclusive-or between the template **x**, and a randomly generated binary string **w**. This provides the system with revocable templates, **xr**. If a user’s data is compromised, a new binary string **w’** can be generated and a new template, **xr’**, can be issued using the same biometric data. The result of these two operations, ECC and hash function, **s** and **h**, respectively, are stored in the database, together with the binary string **w**; from now on, this will be referred to as the **secure biometric template**. It should be pointed out that it is impossible to recover any biometric data from this secure template since the hash function is not invertible and **s** corresponds only to the parity bits generated by the error correcting code, the information bits not being directly stored.



(a)

(b)

Figure 1 – Proposed system architecture: (a) Enrolment stage; (b) Verification stage.

During the verification stage, the probe biometric is acquired and the corresponding template, $\hat{\mathbf{x}}$, is generated. This template can be thought of as a corrupted version of **x**, if it comes from the same user, with the errors being due to acquisition noise. The error correcting decoder uses $\hat{\mathbf{x}}$, together with the parity bits stored for that user, to recover the original biometric template **x** if the user is who he claims to be, or something completely different if he is not. The comparison of $\hat{\mathbf{x}}$ and **x** has to be performed in the hashed domain, since only **h** is stored in the database. If the two hashes are equal, then the user is verified.

With this system, the three requirements above are verified. In particular, it is possible to generate many different secure biometric templates from the same biometric trait or cancel a compromised template and issue a new one; it is just a matter of using a different set of attributes for the hash function. Finally, since the biometric data is not stored in an unencrypted way in the database, information privacy is guaranteed.

3. IMPLEMENTATION DETAILS

The first decision that had to be made before starting the actual implementation of the secure biometric recognition system was to choose the biometric trait to be used. Due to its numerous advantages over other biometric traits for verification systems, the iris [15] was chosen. After this decision, it then becomes possible to decide how each of the modules in the proposed architecture will be implemented.

As shown in Figure 1, the proposed system includes five main modules: biometric data acquisition, pre-processing, feature extraction, cryptographic hash function and error correction coding.

The proposed solutions for each of these modules are described in the following subsection with more detail.

3.1 Acquisition

The acquisition module, absolutely necessary in a real biometric verification system, has not been implemented by the authors at the current simulation stage, so, problems like liveness detection, very important for such a module, are not covered in this paper. Instead, it is replaced by a large database of iris images, like the one developed by the Chinese Academy of Sciences' Institute of Automation (CASIA-IrisV3-Interval) [16]. This database consists of 2655 iris images from 249 subjects. All iris images are 8 bit gray-level JPEG files, collected under near infrared illumination and are captured in 2 sessions, separated by about a month. The database was approximately divided in half, with the first part (individuals "001" to "122") being assigned to the training set and the remainder (individuals "123" to "249") to the test set. A few images have not passed the enrolment stage (i.e., failed to enroll). The number of successfully enrolled images is presented in Table 1, together with the corresponding numbers of intra-class and inter-class comparisons that are possible.

	Training set	Test set
Number of images	1275	1030
Intra-class comparisons	8370	6362
Inter-class comparisons	1607620	1048184

Table 1 – Training and test set composition.

3.2 Pre-processing

The first step after acquisition is to extract the iris from the input eye images. The iris area is considered as a circular crown limited by two circles. The iris inner (pupil) and outer (sclera) circles are detected by applying the circular Hough transform [16], relying on edge detection information previously computed using a modified Canny edge detection algorithm [17]. The eyelids often occlude part of the iris, thus being removed using a linear Hough transform [18]. The presence of eyelashes is identified using a simple thresholding technique. The output of this segmentation step is illustrated in Figure 2 (b).

Afterwards, a normalization process is required since iris segmentation results may appear at different positions and scales. This is done with Daugman's rubber sheet model [19], which maps the circular iris image into a rectangular representation, as illustrated in Figure 3. The size of the normalized iris image is 20×240 pixels.

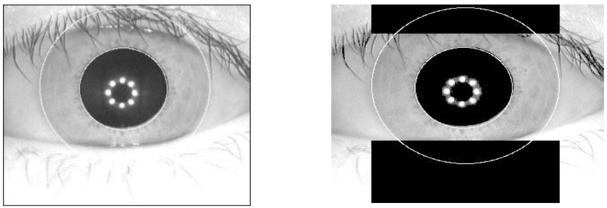
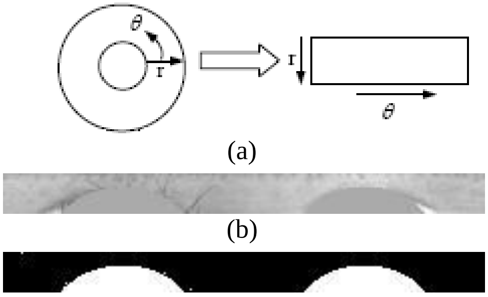

(a) (b)

Figure 2 – Iris segmentation in the presence of noise (eyelids and eyelashes): (a) Input image; (b) Iris segmentation result.

3.3 Feature Extraction

Once the iris texture is available, features are extracted from it to generate a more compact representation: the biometric template. To extract this representation, the two-dimensional normalized iris pattern is convolved with a Log-Gabor wavelet. The resulting phase information is quantized, using two bits per pixel. The resulting iris template is composed of 9600 bits, stored as a 20×480 binary matrix. In the proposed system, only the 5476 most reliable bit positions of the template are used (i.e., bits that are least likely to be affected by noise such as eyelids or eyelashes). The same bit positions are considered for all templates.



(a)

(b)

(c)

Figure 3 – Illustration of the normalization process: (a) Daugman's rubber sheet model [19]; (b) Normalized iris texture image; (c) Noise mask of the normalized image.

3.4 Hash Function

Cryptographic hash functions are a deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit sequence, computed by a dispersion algorithm. The process is unidirectional, which makes it practically impossible to recover the original content from the hashed bit sequence. Moreover, a very small change in the original content will result in a considerable change in the value of the hash.

Available cryptographic hash functions include MD2, MD5, SHA-1, SHA-384 and SHA-512. In the present implementation, SHA-512 is selected due to its enhanced security characteristics.

3.5 Error Correction Coding

For dealing with acquisition noise, error correction coding is used in the proposed system. In the verification stage, the probe template of a legitimate user is (error) corrected in order to recover the original template, obtained during enrolment; this should be possible because both templates are fairly similar. However, for an illegitimate user, whose probe template is fairly different from the one originally enrolled by the legitimate user, it should not be possible to recover the original from the probe template. Therefore, the selected error correcting code should be strong enough to correct templates of legitimate users, but not so strong as to also correct the templates of illegitimate users. Therefore, the main challenge here is to find the threshold of performance needed for the error correcting codes.

To precisely determine the adequate threshold of error correcting performance, tests should be done by varying the performance of the error correcting code with enough granularity, which is not possible for all existing codes.

Since LDPC codes allow their performance to be adjusted with a very fine granularity, they were chosen here for this module. LDPC codes are a class of systematic linear block codes, which is very important here because in the proposed system the information bits have to be separated from the parity bits, since only the latter are stored in the system database. The LDPC name comes from the fact that their parity-check matrix contains only a few 1's in comparison to the amount of 0's [20]. The code performance can be defined according to the number of columns in the parity-check matrix or number of 1's per column. In addition to their granularity, the error correcting performance curve of LDPC codes is very steep when the limit is approached, which basically means that it will be possible to precisely select which templates can be corrected and which ones cannot.

With these three properties, LDPC codes are ideally suited for this type of application, allowing to choose an error correcting code whose performance closely matches the desired operation threshold. The steepness in performance and the granularity of LDPC codes is illustrated in Figure 4.

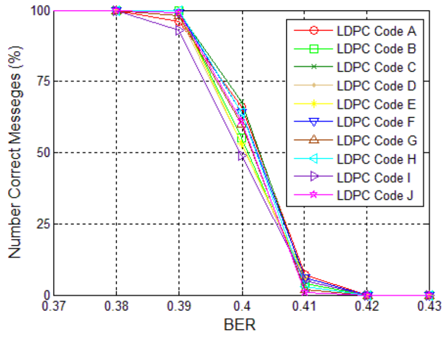


Figure 4 – LDPC codes steepness and granularity illustration.

4. PERFORMANCE EVALUATION

This section presents the performance evaluation of the implemented secure iris verification system, in terms of both recognition and security.

4.1 Recognition Performance

The simulation of various LDPC codes was done using the training set. The best performing codes were then used to test the system's recognition performance with the testing set. The simulation results are presented in Table 2 and Figure 6. False reject rates (FRR) range from 0.754% to 1.87%, whereas false accept rates (FAR) are between 0.0364% and 0.3653%. The comparison against the system without security is also included in Figure 5.

CODE	FRR	FAR	Security Bits
A	0.754%	0.3653%	98
B	0.833%	0.2750%	102
C	0.865%	0.1899%	115
D	1.037%	0.1835%	109
E	1.085%	0.1430%	119
F	1.132%	0.1182%	141
G	1.163%	0.1034%	130
H	1.320%	0.0964%	139
I	1.430%	0.0725%	146
J	1.540%	0.0665%	136
K	1.619%	0.0556%	153
L	1.666%	0.0541%	151
M	1.698%	0.0483%	160
N	1.776%	0.0376%	168
O	1.870%	0.0364%	171

Table 2 – Proposed secure system performance for the selected LDPC codes.

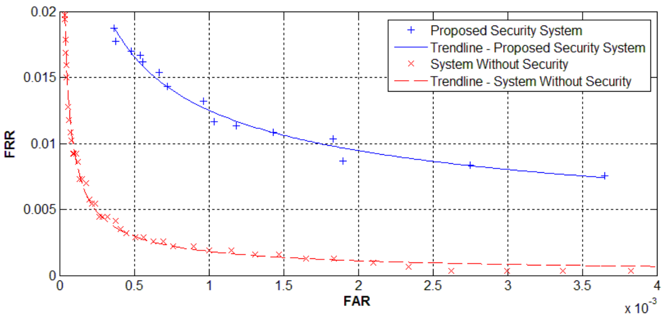


Figure 5 – Recognition performance of the proposed secure system and the system without security.

The system without security achieves a better recognition performance when compared to the proposed system. Still, the performance difference is not overwhelming, and the proposed system has considerable advantages in terms of security, as can be seen below.

4.2 Security Performance

To quantify the iris verification system security performance, a metric proposed by Hao et al. [21], is used:

$$BF \geq \frac{2^z}{\sum_{i=0}^{w}\binom{z}{w}} \cong \frac{2^z}{\binom{z}{w}} \qquad (1)$$

where BF is the mean number of attempts in a brute force attack to break the system; z is the number of independent information bits; and w is the number of bits corrected by the ECC. The number of security bits can be computed using the base two logarithm of BF.

The security performance, computed using the metric described above, for each of the considered LDPC codes, is plotted in Figure 6 and the corresponding values are included in Table 2.

The achieved security values range from 98 to 171 bits. Taking as example the lowest security value, 98 bits, this would correspond to 2^{98}, i.e.,4.24×10^{29}, necessary attempts to break the system in the case of a brute force attack. This is an extremely high value and, in particular, it is much higher than what was achieved in [12].

Additionally, as can be seen in Figure 6, in general, lower FAR values correspond to a higher security of the system.

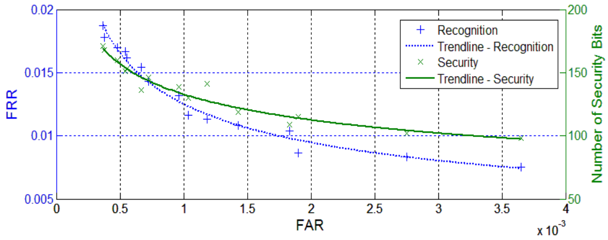


Figure 6 – Recognition and security performance for proposed secure iris verification system.

5. FINAL REMARKS

This article presents a solution to the problem of secure biometric data storage. The proposal combines cryptographic hash functions and distributed source coding principles. The proposed system guarantees that different biometric templates can be generated from the same biometric trait, that these templates can be cancelled if needed, and that the original biometric data cannot be recovered from the stored templates.

An implementation using iris as the selected biometric characteristic was described. In particular, the use of LDPC codes is an asset for this implementation, since these codes, with their granularity and correcting performance steepness, allow working with a near optimal threshold of performance for secure biometric systems.

The present implementation takes the *Iriscode* software, developed by L. Masek [21], as the basis for the traditional part of our biometric system. The proposed system's recognition performance is slightly lower than that of the Libor Masek system. However, the proposed system offers increased security, of at least 98 bits, while the Libor Masek system has no security in the storage of biometric data. The security performance achieved by the proposed security system is much higher than that of the systems currently available in the literature, particularly when compared with the system proposed by Vetro, in [12], which achieves about 50 bits of security.

REFERENCES

[1] A. K. Jain, S. Pankanti, S. Prabhakar, L. Hong, A. Ross, "Biometrics: A Grand Challenge", *Proc. of the International Conference on Pattern Recognition*, Vol. 2, pp. 935–942, August 2004.

[2] J. Wayman, A. Jain, D. Maltoni, D. Maio, *Biometric Systems: Technology, Design and Performance Evaluation*, Springer-Verlag, 2005.

[3] D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer, 2003.

[4] B. Schneier, "Inside Risks: The Uses and Abuses of Biometrics", *Communications of the ACM*, Vol. 42, No. 8, pp. 136, August 1999.

[5] J. Jeong, M. Chung, H. Choo, "Integrated OTP-based User Authentication Scheme Using Smart Cards in Home Networks", *Proc. of the International Conference on System Sciences*, Waikoloa, HI, USA, January 2008.

[6] A. Juels, M. Sudan, "A Fuzzy Vault Scheme", *Proc. of the International Symposium on Information Theory*, p. 408, Lausanne, Switzerland, June 2002.

[7] A. Juels, M. Wattenberg, "A Fuzzy Commitment Scheme," *Proc. of the 6th ACM Conference on Computer and Communications Security*, pp. 28-36, New York, NY, USA, 1999.

[8] T. C. Clancy, N. Kiyavash, D. J. Lin, "Secure Smartcard-based Fingerprint Authentication," *Proc. of the ACM Workshop on Biometrics: Methods and Applications*, pp. 45-52, Berkeley, CA, USA, 2003.

[9] S. Yang, I. Verbauwhede, "Automatic Secure Fingerprint Verification System Based on Fuzzy Vault Scheme", *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 609-612, Philadelphia, PA, USA, 2005.

[10] J.-P. Linnartz, P. Tuyls, "New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates", *Proc. of the 4th International Conference on Audio- and Video-Based Personal Authentication*, pp. 393-402, Guildford, U.K., 2003.

[11] P. Tuyls, A. H. M. Akkermans, T. A. M. Kevenaar, G.-J. Schrijen, A. M. Bazen, R. N. J. Veldhuis, "Practical Biometric Authentication with Template Protection," *Proc of the 5th International Conference on Audio- and Video-Based Personal Authentication*, pp. 436-441, Rye Brook, NY, USA, 2005.

[12] A. Vetro, S. Rane, and J. Yedidia, "Securing Biometric Data," in *Distributed Source Coding: Theory, Algorithms and Applications*. Elsevier, January 2009.

[13] N. K. Ratha, J. Connell, R. Bolle, "Enhancing Security and Privacy in Biometrics-based Authentication Systems", *IBM Systems Journal*, Vol. 40, No. 3, pp. 614-634, 2001.

[14] G. I. Davida, Y. Frankel, B. J. Matt, "On Enabling Secure Applications Through Off-Line Biometric Identification", *Proc. of IEEE the Symposium on Privacy and Security*, pp. 148-157, Oakland, CA, USA, May 1998.

[15] S. Yang, I. Verbauwhede. "Secure Iris Verification", *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. II-15-20, Honolulu, HI, April 2007.

[16] Chinese Academy of Sciences' Institute of Automation (CASIA). iris image database CASIA-Iris-V3. http://www.cbsr.ia.ac.cn/IrisDatabase.htm.

[17] J. Canny, "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 679-714, 1986.

[18] R. Duda, P. Hart, "Use of Hough Transformation to Detect Lines and Curves in Pictures: Graphics and Image Processing", *Communications of the ACM*, Vol. 15, pp. 11-15, 1972.

[19] J. G. Daugman, "How Iris Recognition Works", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 1, pp. 21–30, January 2004.

[20] R. G. Gallager, *Low Density Parity-Check Codes*, MIT Press, Cambridge, MA, USA, 1963.

[21] F. Hao, R. Anderson, J. Daugman, "Combining crypto with biometrics effectively", *IEEE Transactions on Computers*, 55(9):1081-1088, 2006.

[22] L. Masek, P. Kovesi, *MATLAB Source Code for a Biometric Identification System Based on Iris Patterns*, School of Computer Science and Software Engineering, University of Western Australia, Australia, 2003.