Biometric Fuzzy Extractor Scheme for Iris Templates

F. Hernández Álvarez[1], L. Hernández Encinas[2], and C. Sánchez Ávila[1]
[1]*Departamento Matemática Aplicada a las Tecnologías de la Información, E.T.S.I.T.*
Universidad Politécnica de Madrid. Spain.
f.hernandez@alumnos.upm.es, csa@mat.upm.es
[2]*Departamento Tratamiento de la Información y Codificación*
Instituto de Física Aplicada, CSIC, Madrid, Spain
luis@iec.csic.es

Abstract – *Biometric recognition offers a reliable and natural solution to the problem of user authentication by means of her physical and behavioral traits.*

An iris template protection scheme which associates and retrieves a secret value with a high level of security, is proposed. The security is guaranteed thanks to the requirements of fuzzy extractors. The implementation of the scheme is done in Java and experimental results are performed to calculate its False Acceptance Rate and its False Rejection Rate.

Keywords: Cancelable biometrics, Fuzzy extractor, Fuzzy vault, Intra- and inter-user variability, Lagrange interpolation.

1. Introduction

In today's society the problem of securing information and ensuring privacy is a growing concern.

Traditionally, Cryptography is used in order to ensure the secrecy and the authenticity of information by means of different techniques and algorithms (cryptosystems). In cryptosystems one or more keys are used to transform the plaintext into a ciphertext. Without the knowledge of the correct decrypting key, the conversion of ciphertext into the plaintext is infeasible.

Although most of the cryptosystems currently used, such as AES, RSA, ECC, etc. ([13], [18]), have proven its security, some of them suffer from the key management problem. It is known that the security of these cryptosystems relies on the assumption that the keys are kept in secret for everybody but the reliable users. If the secret key, for the symmetric cryptosystems, or the private key, for the asymmetric ones, is compromised, the security is completely broken.

Another problem is related to the bitlength of the keys used. As the keys used are large (128-256 bits for symmetric cryptosystems and 1024-2048 for the asymmetric ones), it is impossible to be memorized. Thus, the keys are, in general, stored in a "secure" location such as computers, smart cards, etc., and kept in secret by a password-based authentication mechanism. The drawback of these mechanisms is that passwords can be easily stolen, forgotten or guessed using different attacks, such as brute force attacks. Consequently, a plaintext protected by means of a cryptosystem is as secure as the password used to release the key. Besides, the use of passwords does not provide non-repudiation.

Some of these limitations and weaknesses can be suppressed by the incorporation of better new methods of user authentication. Biometric authentication ([7], [12]) consists of verifying individuals based on their physiological and behavioral traits such as face, fingerprint, hand geometry, iris, voice, handwritten signature, and so on. Biometric systems offer obvious advantages over other authentication systems. They are inherently more reliable than password-based authentication, as biometric characteristics cannot be lost or forgotten. Moreover, biometric traits are extremely difficult to copy, forge, share, and distribute, and it is unlikely for a user to repudiate having accessed a particular content using Biometrics. Finally all the users have equality security level. Thus, Biometrics-based authentication can be used instead of password-based authentication.

In this work, a new biometric template protection scheme based on iris templates is proposed. This scheme permits to associate and retrieve a secret value with a high level of security. The scheme allows the authentication of a user using her own biometric template as a key. The implementation of the scheme has been developed in Java, and experiments have been performed to demonstrate its efficiency.

The rest of this work is organized as follows:

In Section II, the biometric systems are described identifying their vulnerabilities. Section III shows the main template protection schemes and the way they provide security for the biometric templates. Then, the main characteristics of fuzzy vault and fuzzy extractor schemes are commented in section IV. In section V our proposal for a biometric fuzzy extractor scheme for iris templates is presented, explaining the enrollment and the verifications phases. Finally, the main conclusions and some future works are presented in section VI.

2. Biometric systems

2.1. Description, Vulnerabilities and Security

A generic biometric system consists of five components: Sensor, feature extractor, template database, matcher, and decision module ([8]). Fig. 1 shows a basic block diagram of a biometric system ([21]).

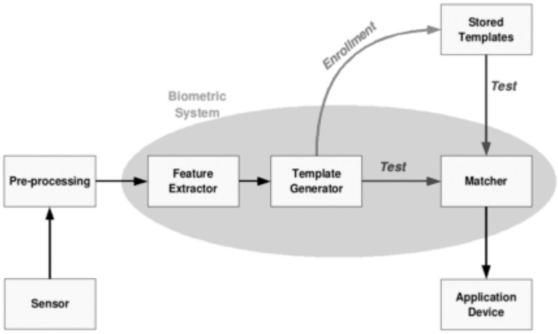

Figure 1: Basic block diagram of a biometric system.

In general, these systems run as follows: In the *enrollment phase*, the biometric templates are processed and stored in the database. Then, in the *verification phase*, the biometric query template extracted from the user in this moment is compared with the one already stored in the database. If this comparison succeeds the user identity is verified, otherwise she is rejected.

In most cases, the applications in which biometric systems are used are unimodal, i.e., they rely on the evidence of a single source of information for authentication. But these systems suffer from some problems, among them the most important are the *intra-* and *inter-user variability*. The intra-user variability measures the differences of two biometric templates extracted from the same user, while the inter-user variability measures the similarities between two biometric templates extracted from different users. These two measurements can cause not to recognize a known user or to recognize an attacker as a known user, respectively.

Some of the limitations in the use of unimodal biometric systems can be solved by including multiple sources of information, i.e., different biometric traits. Such systems are known as *multimodal biometric systems* ([16], [17]).

The most straightforward way to secure a biometric system, including the template, is to put all the system modules and the interfaces between them on a smart card. These systems are known as match-on-card and their advantage is that the biometric information never leaves the card. The drawback is that these systems are not appropriate for large-scale applications and it is possible to get the template from a stolen card. So, both the system and the template must be protected.

One desirable characteristic that the biometric templates should have is to be revoked or canceled if necessary, as PIN and passwords do.

3. Cancelable Biometrics: Template Protection Schemes

Several approaches, known as *cancelable biometrics*, have been proposed to secure biometric templates. Cancelable biometrics, proposed for the first time by Ratha *et al.* ([15]), refers to a way to inherit the protection and the replacement characteristics into biometrics. Essentially, cancelable biometrics performs a distortion of the biometric templates before matching. The variability in the distortion parameters provides the existence of different schemes.

The major challenge in the biometric template protection is the intra-user variability and, as it was explained, this is the reason of why standard protection cryptographic schemes cannot be applied into Biometrics. As the templates are not stable, a small difference in the template would lead to very large differences in the encrypted domain.

There are mainly two categories of template protection schemes ([8]): *Feature transformation approach* and *Biometric cryptosystem* (see Fig. 2).

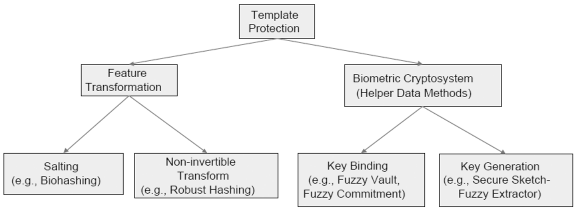
Template Protection
Feature Transformation
Biometric Cryptosystem (Helper Data Methods)
Salting (e.g., Biohashing)
Non-invertible Transform (e.g., Robust Hashing)
Key Binding (e.g., Fuzzy Vault, Fuzzy Commitment)
Key Generation (e.g., Secure Sketch-Fuzzy Extractor)

Figure 2: Template protection schemes.

In the feature transformation approach, a transformation function is applied to the biometric template and then it is store in the database. The function used can be invertible (salting) or non-invertible.

On the other hand, the basic idea for the biometric cryptosystems ([19]) is either binding the cryptographic key with the biometric templates or generating the key directly from the template. Therefore, biometric cryptosystems can be classified into two models: *Key binding*, and *Key generation*.

The common characteristic of these two models is that they need to generate public information, known as *helper data*, about the biometric template in order to perform the verification phase. This public information is supposed to reveal no important information about the biometric template.

4. Fuzzy vault and fuzzy extractor schemes

Biometric reference data storage must be avoided as much as possible, because if the biometric templates are compromised or stolen, there is no way to cancel or revoke them.

The first biometric system was proposed by Juels and Wattenberg in [10]. Their method is called *Fuzzy commitment* because a cryptographic key is decommitted using biometric data. In this context, fuzziness means that a value close to the original is sufficient to extract the committed value. This scheme although compensates the intra-user variability of biometric data, has some shortcomings because it makes assumptions which are not applicable in real life.

4.1. Fuzzy vault scheme

Juels and Sudan ([9]) proposed the use of *Fuzzy vault* schemes, which can be considered as an order-invariant version of the fuzzy commitment schemes. They are obtained by using a Reed-Solomon code in which they evaluate the codeword by means of a polynomial over a set of points. In this case, the process is to encode the secret data as a polynomial, $p(x)$, of degree d, by using the Reed-Solomon encoding scheme. That is, the secret message is embedded as the coefficients of the polynomial. Next, the polynomial is evaluated for different values of a set of features of the biometric data,$\mathfrak{B} = \{b_0, \ldots, b_{n-1}\}$.

Then, a genuine set of pairs of points are produced:

$$G = \{(b_0, p(b_0)), \ldots, (b_{n-1}, p(b_{n-1}))\},$$

where n refers to the size of the unordered set $\mathfrak{B}$ on $p(x)$. Moreover, a set of *chaff* points, N, which do no verify the polynomial $p(x)$, are generated in order to protect G. The value of the vault, V, i.e., the encoded message, consists of the set union $V = G \cup N$.

To decode the message a unordered set $\bar{\mathfrak{B}}$ is needed and if it is close to the set $\mathfrak{B}$, the genuine set of points G can be discriminated from V. Note that the user needs $d+1$ pairs of points $(b_i, p(b_i)) \in G$ to recover the original polynomial by using Lagrange interpolation.

A practical implementation of the fuzzy vault in a secure smart-card is proposed in [2].

In [14] it is showed that hardening the fuzzy vault scheme with a password enhances its security providing it with privacy-enhancing features such as revocability and protection against cross-matching across different biometric systems.

There are fuzzy vault implementations based on other traits different from fingerprints, such as face ([5]) and hand-written signature ([6]). Moreover, two important schemes based on the key binding model are proposed (see [11], [20]). The first scheme uses the fuzzy vault scheme to bind a secret with iris images, while the second one proposes a fuzzy extractor, according to the definitions of Dodis *et al.* ([4]), to associate, and retrieve, a committed value using a special fingerprint data representation called FingerCode.

Fuzzy vault schemes present some limitations:

1) If the same biometric data is used to construct different vaults with different polynomial and different chaff points, the genuine points can be easily identified by correlating the abscissa values from these different fuzzy vaults from different systems.

2) The set of chaff points is bigger than the set of genuine points and it is possible to substitute some points of this chaff-point set for the features of a possible attacker. In this way the attacker and the original user could be correctly identified with the same fuzzy vault.

3) The non-uniformity of the biometric features makes possible to identify the genuine set from the set of chaff points by using a statistical analysis. Chang and Li have analyzed this problem ([1])

4.2. Fuzzy extractor scheme

A *Fuzzy extractor* scheme is a biometric tool whose purpose is to authenticate a user using her own biometric template as a key. It works extracting a uniformly random string S from its input $\mathfrak{B}$, which is a biometric template, in a noise-tolerant way. That means that if the input changes to some $\bar{\mathfrak{B}}$ but remains close, the string S can still be reproduced exactly. To help in the reproduction of S, the first time the fuzzy extractor is used, i.e., in the enrollment phase, it outputs a helper string H that can safely be made public without decreasing the security of S.

The role of each variable is the following: S would be the encryption or authentication key and H would be the public data stored in the database whose function is to recover S. The user's biometric template acts as the key to recover S.

The fuzzy extractor process can be explained as a pair of efficient randomized procedures: Generate (*Gen*) and Reproduce (*Rep*). In the enrollment phase, given $\mathfrak{B}$, the procedure *Gen* outputs an extracted string $S \in \{0,1\}$ and a helper string $H \in \{0,1\}$. In the verification phase, *Rep* takes as input an element $\bar{\mathfrak{B}}$, close to $\mathfrak{B}$, and the string $H \in \{0,1\}$, and outputs the value S. The correctness of the whole procedure depends on the differences between $\mathfrak{B}$ and $\bar{\mathfrak{B}}$.

A basic tool needed in the development of fuzzy extractor is the *secure sketch*. It allows the precise reconstruction of a noisy input. On input $\mathfrak{B}$ a procedure outputs a sketch c. Then, given c and a value $\bar{\mathfrak{B}}$ close to $\mathfrak{B}$, it is possible to recover $\mathfrak{B}$. The sketch is secure in the sense that it does not reveal much information about $\mathfrak{B}$ even if c is known. Thus, it is possible to store c.

In the same way, secure sketch can de explained as a pair of efficient randomized procedures: Sketch (*Sket*) and Recover (*Rec*). The sketching procedure, *Sket*, starts with the sketch, $\mathfrak{B}$, as input and returns a string $c \in \{0,1\}^*$. The recovery procedure *Rec* takes an element $\bar{\mathfrak{B}}$ and $c \in \{0,1\}^*$ and returns the corresponding value $\mathfrak{B}$. The correctness is again depending on the distance between $\mathfrak{B}$ and $\bar{\mathfrak{B}}$.

5. A biometric fuzzy extractor scheme for iris templates

The proposed fuzzy extractor scheme can be divided into two well-defined phases: The enrollment phase and the verification phase. In the enrollment phase, the users' iris templates are processed and the public data are calculated and stored in the database. On the other hand, in the verification phase the identity of the user is verified to knowif she is really who she says to be. This phase is performed comparing the current user's template, called the query template, with the data stored in the database. If they match, the process continues, otherwise an error is displayed.

5.1. Enrollment phase

The enrollment phase (see Fig. 3) consists of obtaining specific public data from the user's *Iris Template* and from a *Secret* (or key) which she has previously selected. These public data, denoted as the sets Δ and H, are stored in the database.

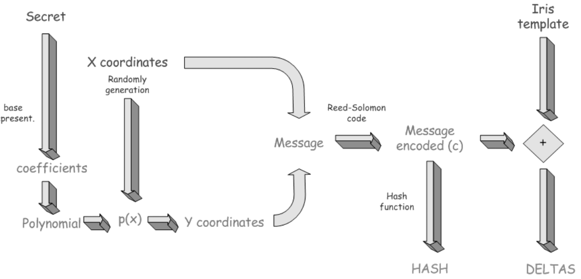

Figure 3: Enrollment phase.

The enrollment phase consists of the following: 1) The secret or the key, S, is represented in a determined *base* (base 10, 16, 256, 512, etc.).

From the expression of S in such base, the coefficients of a polynomial $p(x)$ of degree d, will be derived. So, the relationship between S and $p(x)$ is direct. For example, if S is written in the base $512 = 2^9$, each digit of S in this base can be represented as an integer between 0 and 511. Therefore, the coefficients of $p(x)$ will be numbers in the interval $[0, 511]$. In this way, if $S = \{s_0, s_1, \ldots, s_d\}$, then:

$$p(x) = s_0 + s_1 x + s_2 x^2 + \ldots + s_d x^d. \qquad (1)$$

2) Next, n random points verifying $p(x)$ are calculated, $y_i = p(x_i)$, $0 \leq i \leq n-1$. The value of n is a security parameter which controls the fuzziness of the templates allowed in the scheme. For this reason, n must be much greater than d $(n \gg d)$.

3) The coordinates of the n points, (x_i, y_i) are concatenated, $x_i \parallel y_i$, and encoded as n codewords by means of a Reed-Solomon code to form the set C: $C = \{c_0, c_1, \ldots, c_{n-1}\}$. Then, a hash function, $\mathfrak{h}$, is applied to the set C and a set of n hash values, H, is obtained:

$$H = \{\mathfrak{h}(c_0), \mathfrak{h}(c_1), \ldots, \mathfrak{h}(c_{n-1})\}.$$

In the proposed scheme, it is possible to use any of the usual hash functions, such as SHA-1, SHA-2, etc.

4) Now, the template of each user is necessary to calculate the set Δ. Each iris template, $\mathfrak{B}$, is divided into n parts, as many as points were computed:

$$\mathfrak{B} = b_0 \parallel b_1 \parallel \ldots \parallel b_{n-1}$$

Each value b_i, $0 \leq i \leq n-1$, is subtracted from each codeword of the set C, to obtain the elements of the set $\Delta = \{\delta_0, \delta_1, \ldots, \delta_{n-1}\}$, where

$$\delta_i = c_i - b_i, \qquad 0 \leq i \leq n-1.$$

Both sets, Δ and H, are stored in the database. Moreover, the control parameters have to be stored because they will be necessary in the verification phase. They are:

- The degree of the polynomial, d.
- The Reed-Solomon parameters: k, nRS, and m.
- The hash function used, *digest*.
- The *base* in which S is represented.

The value of n can be obtained directly from the cardinal of Δ.

5.2. Verification phase

This phase consists of the verification of a user (Fig. 4) by means of her query iris template. If the verification succeeds, her secret will be retrieved; otherwise an error is displayed. The process is as follows:

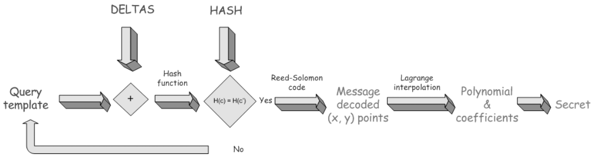

Figure 4: Verification phase.

1) All the control parameters are obtained.

2) The query iris template, $\bar{\mathfrak{B}}$, is divided into n parts:

$$\bar{\mathfrak{B}} = \bar{b}_0 \parallel \bar{b}_1 \parallel \ldots \parallel \bar{b}_{n-1}$$

3) Next, from the values of the elements in Δ and $\bar{\mathfrak{B}}$, a new set of values, $\bar{C}$ is computed:

$$\bar{C} = \{\bar{c}_0, \bar{c}_1, \ldots, \bar{c}_{n-1}\}.$$

Each element, $\bar{c}_i = \delta_i + \bar{b}_i$, are supposed to be quite similar to the values of the set C calculated in the enrollment phase, but taking into account the influence of the intra-user variability.

4) After applying the same hash function as in the enrollment phase to the elements in $\bar{C}$, they are compared with the corresponding elements in H.

In this comparison, it is necessary that at least $d+1$ values of H and $\mathfrak{h}(\bar{C})$ coincide. If it happens, the query iris template is considered as valid and the process to retrieve S continues. Otherwise, the recovery process leads to an error. This error is due because at least $d+1$ points are needed to recover a polynomial of degree d by the Lagrange interpolation.

The importance of the value n is shown in this moment. Due to the intra-user variability and as $d+1$ correct values are necessary, the value of n is a measure of the degree of fuzziness allowed by the scheme.

5) The coincident values are decoded by means of the Reed-Solomon code and, at least, $d+1$ points (x_i, y_i) are obtained.

6) The next step is to use the Lagrange interpolation method to obtain the coefficients of $p(x)$.

7) Finally, S can be easily retrieved by means of the coefficients of $p(x)$ and the *base* it was represented in.

6. Experimental results

The aim of the experiments performed with the scheme proposed is to measure how it deals with the intra-user and inter-user variability. To do this task, 25 users from the CASIA database of iris images have been chosen randomly. Each one of these users have 7 different images of their irises and by using the algorithm designed by Diez Laiz ([3]) the corresponding templates of all these $25 \cdot 7 = 175$ images have been extracted. Finally, the results obtained are used to calculate the rates of false reject, *FRR*, and false acceptance, *FAR*.

Before performing the experiments, a first analysis of the characteristics of the templates of these users shows that the mean of the Hamming distance of each user is around 33.3%.

All the $25 \cdot 7 = 175$ templates are enrolled and their public data, Δ and H, are calculated and stored. The basic idea of the experiments is to count the number of coincidences between the set H and the set formed by the hash values of the set $\bar{C}$, $\mathfrak{h}(\bar{C})$.

The parameters considered in the experiment are the following: S is a secret of 192 bits, i.e, it can be a secret key for 3-DES or AES. S is represented in *base*$= 512$, the degree of the polynomial is $d = 21$, and $n = 384$. The hash function considered is *digest* = SHA-512, and the parameters for the Reed-Solomon code are: $k = 23$, $nRS = 36$, and $m = 9$.

6.1. Intra-user variability measure: *FRR*

This experiment is carried out in the following way: Each one of the 7 templates of the 25 users is considered as the input of the verification phase and it is compared with the data stored, H, of the rest of the templates of the same user. In this way, the result of these comparisons will show the level of similarity between all the templates of a single user (intra-user variability). These coincidences determines whether the secret can be retrieved or not. In our case, if there are at least $d+1 = 22$ coincidences between two different templates, the user is recognized and her secret is retrieved. The results of this experiment are used to measure the False Rejection Rate. This rate is not a secure parameter, but it is a comfort criteria.

Table 1 shows the number of comparisons with, at least, $d+1 = 22$ coincidences for each one of the 25 users compared to herself. The total number of comparisons done, in each user, is $\binom{7}{2} = 21$.

From these values, the False Rejection Rate (*FRR*) of the scheme can be computed. To do this, the

Table 1: Number of comparisons with, at least, 22 coincidences for the 25 users.

	User 1	User 2	User 3	User 4	User 5
$> d = 21$	21	20	21	20	21
	User 6	User 7	User 8	User 9	User 10
$> d = 21$	19	19	14	18	21
	User 11	User 12	User 13	User 14	User 15
$> d = 21$	13	19	15	19	20
	User 16	User 17	User 18	User 19	User 20
$> d = 21$	18	21	21	16	15
	User 21	User 22	User 23	User 24	User 25
$> d = 21$	20	21	18	17	20

Genuine Acceptance Rate (*GAR*) is calculated:

$$GAR = \frac{467}{21 \cdot 25} = 0.88952 \simeq 88.9\%$$

and then

$$FRR = 1 - GAR = 1 - 0.88952 = 0.11048 \simeq 11\%.$$

This percentage is quite good because 89% is a high rate of recognition.

6.2. Inter-user variability measure: *FAR*

This experiment measures the similarities between different users (inter-user variability). It is divided into two sub-experiments. For both sub-experiments, instead of using the 7 templates of each of the 25 users, only one template of each user is chosen randomly and these 25 templates (1 from each user) will be used.

6.2.1. Templates vs. Database. The first sub-experiment measures the similarities of the 25 templates selected with the whole database formed by the 25 users. The total number of comparisons done is $7 \cdot 24 = 168$ for each of the 25 templates. Table 2 shows the number of comparisons among the 25 templates and the 25 users with, at least, $d+1 = 22$ coincidences.

Table 2: Number of comparisons among the 25 templates and the 25 users with, at least, 22 coincidences.

	Tpl. 1	Tpl. 2	Tpl. 3	Tpl. 4	Tpl. 5
$> d = 21$	0	2	2	0	0
	Tpl. 6	Tpl. 7	Tpl. 8	Tpl. 9	Tpl. 10
$> d = 21$	2	4	0	0	1
	Tpl. 11	Tpl. 12	Tpl. 13	Tpl. 14	Tpl. 15
$> d = 21$	0	1	2	2	6
	Tpl. 16	Tpl. 17	Tpl. 18	Tpl. 19	Tpl. 20
$> d = 21$	7	4	8	0	0
	Tpl. 21	Tpl. 22	Tpl. 23	Tpl. 24	Tpl. 25
$> d = 21$	0	9	1	2	4

The results shown in Table 2 are used to measure the False Acceptance Rate (FAR_1) as this rate indicates if the secret of the original user is retrieved to a person who is not really the original user. This ratio is critical because it is a security relevant measure since the system recognizes an attacker as a known user.

$$FAR_1 = \frac{57}{7 \cdot 24 \cdot 25} = 0.01357 \simeq 1.35\%.$$

This value is not excellent, but it can be acceptable for some applications.

6.2.2. Templates vs. Templates. The second sub-experiment consists of comparing the 25 templates chosen among themselves. This sub-experiment measures a more specific False Acceptance Rate (FAR_2).

The total number of comparisons in this occasion is $\binom{25}{2} = 300$. Only 2 of all these comparisons were $\geq d+1 = 22$, thus:

$$FAR_2 = \frac{2}{300} = 0.00667 \simeq 0.67\%,$$

which is a good value and it is closer to a real situation with a real biometric system.

7. Conclusions and future work

The conclusions of this work are divided in two parts: Those related with the limitations of the fuzzy vault schemes (see §4.1), and those related with the two main problems of the biometric systems: the secure storage of the biometric template and the intra- and inter-user variability.

1) The first limitation of the fuzzy vault scheme is not longer a problem with our scheme because the biometric data is not used in the creation of the polynomial. Each time the process is executed a different set of random points, *x*- and *y*-coordinates, is created. Thus, it is not feasible a cross-matching of templates among different vaults from different systems.

2) Another limitation solved with our scheme is the possible identification of two people with the same vault due to the set of chaff points. In our scheme, as the points (x_i, y_i) are chosen randomly, there is no option to insert the biometric template of an attacker. The attacker can use her template trying to obtain the set $\bar{C}$, but at this point the hash function assures the security of the system against this attack.

3) In our scheme the third limitation is avoided because only the random-generated values are used in the polynomial.

In relation to the problems of secure storage of the biometric template, our scheme fits with the four main characteristic that a biometric template protection scheme should have, namely: 1) *Diversity*. Cross-matching between databases are not feasible with our scheme since the points and the secret are generated randomly every time. The user's privacy is assured.

2) *Revocability*. If an iris template is compromised, it can be changed for a new one based on the same biometric trait. In our scheme, as the iris templates are not stored directly anywhere they cannot be stolen or compromised.

3) *Security*. It is computationally hard to obtain the original iris template from the secure template thanks to the use of hash functions.

3) *Performance*. The recognition performance (*FAR* and *FRR*) of the system is not degraded.

And related to the intra- and inter-user variability:

1) The results obtained in the second experiment of inter-user variability shows (see §6.2) two values of False Acceptance Rate (*FAR*). Firstly comparing a single iris template of each user with the whole database has a $FAR_1 \simeq 1,3\%$; and secondly comparing only the 25 templates selected at random has a $FAR_2 \simeq 0.67\%$.

The big difference between these two values, $FAR_1 \simeq 2\cdot\ FAR_2$, proves that the intra-user variability is very high (recall that the mean of Hamming distances between the templates of the same user is around 33.3%).

2) The results of the first experiment performed show that the False Rejection Rate (*FRR*) has a percentage of $\simeq 11\%$ (see §6.1). In other words, the Genuine Acceptance Rate is $GAR = 1 - FRR = 89\%$.

To sum up, our scheme has a good behavior related to the intra- and inter-user variability since it is capable of recognizing users with a high percentage, 89%, although their intra-user variability is a bit high, $FAR_1 \simeq 1,3\%$ versus $FAR_2 \simeq 0.67\%$.

Some of future works are the following:

- Use a cryptographic frame to protect the biometric template until it is inserted in the system.
- Improve the iris template extraction to reduce the intra-user variability.
- Implement the scheme in a more efficiency way to allow using secrets with a bigger bitlength.
- Extend the experiments to the whole CASIA database or even others databases.
- Use different template extraction algorithms.

Acknowledgement. This work has been partially supported by Ministerio de Industria, Turismo y Comercio (Spain), in collaboration with CDTI and Telefónica I+D under the project SEGUR@ (CENIT-2007 2004).

References

[1] E.-C. Chang, and Q. Li, Hiding secret points amidst chaff, *Proc. EURO-CRYPT 2006*, *Lecture Notes in Computer Science*, **4004**, (2006), 59–72.

[2] T. Clancy, D. Lin, and N. Kiyavash, Secure smartcard-based fingerprint authentication, *Proc. ACM SIGMM Workshop on Biometric Methods and Applications*, (2003), 45–52.

[3] E. Diez Laiz, Master Thesis, Universidad Politécnica de Madrid (publishing pending).

[4] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data, *SIAM Journal Computing* **38**, 1 (2008), 97–139.

[5] Y.C. Feng and P.C. Yuen, Protecting face Biometric data on smart-card with Reed-Solomon code, *Proc. CVPR Workshop Privacy Research in Vision*, (2006), p.29.

[6] M. Freire-Santos, J. Fierrez-Aguilar, and J. Ortega-Garca, Cryptographic key generation using handwritting signature, *Proc. Biometrics Technologies for Human Identifications III*, **6202**, (2006), 225–231.

[7] A.K. Jain, R. Bolle, and S. Pankanti, *Biometrics: Personal Identification in Networked Society*, Springer, New York, 1999.

[8] A.K. Jain, K. Nandakumar, and A. Nagar, Biometric Template Security, *Journal on Advances in Signal Processing* **8**, 2 (2008), 17 pp.

[9] A. Juels and M. Sudan, A fuzzy vault scheme, *Designs, Codes and Cryptography*, **38**, 2 (2006), 237–257.

[10] A. Juels and M. Wattenberg, A fuzzy commitment scheme, *Proc. of the 6th ACM conference on Computer and Communications Security*, (1999), 28–36.

[11] Y.J. Lee, K. Bae, S.J. Lee, K.R. Park, and J. Kim, Biometric Key Binding: Fuzzy Vault Based on Iris Images, *Lecture Notes in Computer Science*, **4642**, (2007), 800–808.

[12] D. Maltoni, D.Maio, A.K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer, New York, 2003.

[13] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, 1997.

[14] K. Nandakumar, A. Nagar, and A.J. Jain, Hardening Fingerprint Fuzzy Vault using Passwrod, *Proc. of ICB*, (2007), 927–937.

[15] N.K. Ratha, J.H. Connell, and R.M. Bolle, Enhancing security and privacy in biometrics-based authentication systems, *IBM Systems Journal*, **40**, (2001), 614–634.

[16] A. Ross and A. Jain, Multimodal biometrics: an overview, *Proc. of 12th European Signal Processing Conference (EUSIPCO)*, (2004), 1221–1224.

[17] A. Ross, A. Jain, and J.Z. Qian, Information Fusion in Biometrics, *Proc. of 3rd Int'l Conference on Audio- and Video-Based Person Authentication (AVBPA)*, (2001), 354–359.

[18] D.R. Stinson, *Cryptography: Theory and Practice*, 3^{rd} ed., Chapman & Hall/CRC, Boca Raton, FL, 2006.

[19] U. Uludag, S. Pankanti, S. Prabhakar, and A.K. Jain, Biometric Cryptosystems: Issues and Challenges, *Proc. of the IEEE*, **92**, 6 (2004), 948–960.

[20] V.V. Triem Tong, H. Sibert, J. Lecoeur, and M. Girault, Biometric Fuzzy extractors made practical: A proposal based on FingerCodes, *Advances in Biometrics, Lecture Notes in Computer Science*, **4642**, (2007), 604-613.

[21] Wikipedia. Biometrics. http://en.wikipedia.org/wiki/Biometrics