Security Efficiency Analysis of a Biometric Fuzzy Extractor for Iris Templates

F. Hernández Álvarez and L. Hernández Encinas

Department of Information Processing and Coding
Applied Physics Institute, CSIC
C/ Serrano 144, 28006-Madrid, Spain
{fernando.hernandez, luis}@iec.csic.es

Abstract. A Biometric fuzzy extractor scheme for iris templates was recently presented in [3]. This fuzzy extractor binds a cryptographic key with the iris template of a user, allowing to recover such cryptographic key by authenticating the user by means of a new iris template from her. In this work, an analysis of the security efficiency of this fuzzy extractor is carried out by means of a study about the behavior of the scheme with cryptographic keys of different bitlengths: 64, 128, 192, and 256. The different sizes of the keys permit to analyze the variability of the intra- and inter-user in the iris templates.

Key words: Biometrics, Iris Template, False Acceptance Rate, False Rejection Rate, Fuzzy extractor.

1 Introduction

One of the most important uses of Biometrics nowadays is to identify and authenticate individuals by means of one or several of their physiological or behavioral features, like fingerprint, voice, hand geometry, iris, retina, signature, etc.

In general, to validate the identity of a user, biometric procedures consists into two phases: The enrollment and the verification phase. In the enrollment phase, the biometric templates are processed and stored in the database. In the verification phase, a new biometric template (called the query template) is extracted from the user ant it is compared with the data already stored. If the comparison matches, the user identity is validated.

The main properties which permit to consider biometric data as good candidates for security applications are the following: They are unique to each user, they are hard to forge, and they have a good source of entropy. Nevertheless, biometric templates present some drawbacks. 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. The ratios used to measure these two subjects are, respectively: False Rejection Rate (*FRR*) and False Acceptance Ratio (*FAR*).

One desirable characteristic that the biometric templates should have is to be revoked or canceled if necessary, as PIN and passwords do. Several approaches, known as biometric template protection schemes ([6]), have been proposed to secure biometric templates and they can be broadly classified into two categories, namely, feature transformation approach and biometric cryptosystem.

In the feature transformation approach a transformation function is applied to the biometric template before storing it in the database. The function used can be invertible or non-invertible. On the other hand, biometric cryptosystems ([10]) need to generate public information, known as helper data, about the biometric template in order to perform the verification phase. These systems can be classified into three models, key release, key binding and key generation.

In a key binding cryptosystem, the biometric template is secured by binding it with a key in a cryptographic framework. The key and the biometric template are stored in the database as a single entity which represents the helper data. This system has the advantage that it is tolerant to intra-user variability, but this tolerance is determined by the error correcting capability. The limitation of this system is that the matching has to be done using error correction schemes and therefore it is necessary the use of sophisticated matchers. Another limitation is the way the helper data are designed. Several template protection technologies can be considered as key binding approaches, for example fuzzy commitment scheme ([5]), fuzzy vault scheme ([4]), etc.

The fuzzy vault scheme is a cryptographic framework that binds a biometric template with a secret key to build a secure sketch of this template. This sketch is the data which are stored because it is computationally hard to retrieve either the template or the key without any knowledge of the user's biometric data.

In this work, an analysis of the security efficiency of the fuzzy extractor scheme for iris templates proposed in [3] is carried out. This is done by means of a complete study about the behavior of the scheme with cryptographic keys of different bitlengths: 64, 128, 192, and 256. The different sizes of the keys will permit to analyze the intra- and inter-user variability of the iris templates.

The rest of this work is organized as follows. In Section 2 a short review of the fuzzy extractor scheme proposed in [3] is presented. The security efficiency analysis of that scheme is carried out in Section 3; and finally, the conclusions of this work are presented in Section 4.

2 Review of a biometric fuzzy extractor for iris templates

In [3] a biometric fuzzy extractor scheme for iris templates was presented. It is based on fuzzy extractor schemes ([2]) and on fuzzy vault schemes ([4]).

As it is well-known, fuzzy extractor's basic aim, according to the definitions of Dodis *et al.* ([2]), is to authenticate a user using her own biometric template, $\mathfrak{B}$, as the key. To do so, it makes use of another process known as secure sketch to allow precise reconstruction of a noisy input. The correctness of the whole procedure depends on the Hamming distance between $\mathfrak{B}$, used in the enrollment phase, and the query template $\bar{\mathfrak{B}}$, used in the verification phase. Moreover, a fuzzy vault scheme binds a biometric template, $\mathfrak{B}$, with a secret (or a key), S, to build a secure sketch of $\mathfrak{B}$ itself.

Among the papers related to key binding, the scheme whose efficiency is being measured in this work ([3]) has some similarities with the schemes proposed by Lee *et al.* ([7]) and Tong *et al.* ([9]). The main differences among them are that [7] generates the IrisCode from a set of iris features by clustering, technique that is not used in our scheme, and [9] uses fingerprints instead of iris templates.

Therefore, some stages have been adjusted to provide a higher level of security and making them fit with iris templates instead of fingerprints. Another improvement that has been done is that we implement in Java the whole scheme to present some conclusions and useful results.

The two phases associated to this scheme are the following:

2.1 Enrollment phase

In the enrollment phase, from the user's iris template, $\mathfrak{B}$, and a key/secret, S, chosen by herself, the scheme produces two sets, H and $\varDelta$, as public helper data, which are stored in the database. The different stages of this phase are:

1. The key S is represented in a *base* (10, 16, 256, 512, etc.). The digits of S in that base are considered as the coefficients of a polynomial $p(x)$ of degree d. That is, if $S = \{s_0, s_1, \ldots, s_d\}$, then $p(x) = s_0 + s_1x + s_2x^2 + \ldots + s_dx^d$.
2. Next, n random integer numbers, $x_i \in \mathbb{Z}$, are generated in order to compute n pairs of points, (x_i, y_i), verifying $p(x)$, i.e., $y_i = p(x_i)$, $0 \leq i \leq n-1$. The parameter n determines the level of fuzziness of the system, so n must be much greater than d, $(n \gg d)$.
3. The points are encoded by using a Reed-Solomon code into n codewords determining a set $C = \{c_0, c_1, \ldots, c_{n-1}\}$. This codification is done to avoid somehow the intra-user variability thanks to the error-correction properties of the Reed-Solomon codes.
4. A hash function, $\mathfrak{h}$, is applied to the elements of C to obtain a new set $H = \{\mathfrak{h}(c_0), \mathfrak{h}(c_1), \ldots, \mathfrak{h}(c_{n-1})\}$.
5. The iris template of each user is divided into n parts, as many as points were calculated: $\mathfrak{B} = b_0 \parallel b_1 \parallel \ldots \parallel b_{n-1}$.
6. Then, from the values b_i and c_i, $0 \leq i \leq n-1$, the elements of the set $\varDelta = \{\delta_0, \delta_1, \ldots, \delta_{n-1}\}$ are calculated, where $\delta_i = c_i - b_i, 0 \leq i \leq n-1$.

Finally, once the helper data (H and $\varDelta$) are determined, they are stored in the database. Moreover, the control parameters are made public.

2.2 Verification phase

The first task of this phase is to obtain the control parameters previously stored in the enrollment phase. Then, the following stages are performed:

1. The query iris template, $\bar{\mathfrak{B}}$, is divided into n parts, as it was done in the enrollment phase: $\bar{\mathfrak{B}} = \bar{b}_0 \parallel \bar{b}_1 \parallel \ldots \parallel \bar{b}_{n-1}$.
2. Next, from the sets $\varDelta$ and $\bar{\mathfrak{B}}$, a new set is computed: $\bar{C} = \{\bar{c}_0, \bar{c}_1, \ldots, \bar{c}_{n-1}\}$, where $\bar{c}_i = \delta_i + \bar{b}_i$,.
   Note that each value $\bar{c}_i$ is supposed to be similar to the corresponding value $c_i \in C$, but with some differences due to the intra-user variability.
3. The same hash function, $\mathfrak{h}$, is applied to the elements of the $\bar{C}$, and the result is compared with the elements of the set H.
   In this comparison, at least $d+1$ coincidences between H and $\mathfrak{h}(\bar{C})$ are necessary to continue with the process, due to the fact that Lagrange interpolation method is used to rebuild the polynomial $p(x)$ of degree d. This comparison shows the importance of the parameter n, because it will determine the rate of errors admitted in the system due to the intra-user variability.
4. The coincident values are decoded by means of the Reed-Solomon code and $d+1$ points, (x_i, y_i), at least, are obtained.
5. By using that points and the Lagrange interpolation method, $p(x)$ is rebuilt.
6. Finally, from the coefficients of $p(x)$, the secret S is determined and retrieved to the user.

3 Security efficiency analysis

In order to determine the security efficiency of the fuzzy extractor scheme presented above, the False Rejection Rate and the False Acceptance Rate of this biometric system will be computed by using different sizes of the secret S.

A fix value of 192 bits for S was considered in [3]. In the present analysis different bitlengths of S, denoted by $|S|$, will be used to make a comparison between all of them and to determine the security efficiency of the system. The different bitlengths selected are 64, 128, 192, and 256 because they are the standard sizes for cryptographic symmetric keys used nowadays ([8]). This analysis is relevant because if a base to represent the secret S is fixed, the value of the degree d of $p(x)$ depends on the size (bitlength) of S, $|S|$.

In this way, when the comparison is carried out in the verification phase, as the value of d is different, a different number of coincidences between H and $\mathfrak{h}(\bar{C})$ are necessary to validate the user's iris template. This fact can be seen as an advantage but at the same time as a drawback, because it can be "easier" to recognize a known user but it does the same to a possible attacker.

The results of this analysis have been obtained by using the same 25 users as those ones used in [3]. These data have been taken from the CASIA database of iris images. Each one of these users have 7 different images of their irises and the corresponding templates of all these $25 \cdot 7 = 175$ images have been extracted by using the algorithm designed by Diez Laiz ([1]).

The parameters used in this analysis are the same (or similar for the Reed-Solomon codes) than those used in [3] in order to do a trustworthy comparison. The only different parameter is the degree of $p(x)$, d, as it depends on the bitlength of S. In fact, the values considered are: The base used for S is 512; the hash function is $\mathfrak{h} =$ SHA-512; and the fuzziness parameter is $n = 384$.

The values for d as function of the bitlength of S are shown in Table 1.

Table 1. Values of d depending on the bitlength of S.

bitlength of S: $\|S\|$	64	128	192	256
Value of d	8	14	21	28

3.1 Intra-user variability: FRR

In this analysis each one of the 7 templates of the 25 users is compared with the rest of the templates of the same user. In this way, it is analyzed whether the user is recognized or not, and the False Rejection Rate is determined. The number of comparisons done for each user is $\binom{7}{2} = 21$.

Tables 2, 3, 4, and 5 shows the comparisons obtained with $d+1$ coincidences, at least, for each of the 25 users compared to herself, and for each value of d.

Table 2. Number of comparisons with, at least, $d + 1 = 8$ coincidences for $|S| = 64$.

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

Table 3. Number of comparisons with, at least, $d + 1 = 15$ coincidences for $|S| = 128$.

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

Table 4. Number of comparisons with, at least, $d + 1 = 22$ coincidences for $|S| = 192$.

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

Table 5. Number of comparisons with, at least, $d+1=29$ coincidences for $|S|=256$.

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	User 12	User 13	User 14	User 15	User 16	User 17	User 18	User 19	User 20	User 21	User 22	User 23	User 24	User 25
$> d = 28$	21	17	21	19	16	17	15	10	17	13	11	12	11	15	18	16	21	15	13	13	13	19	15	15	18

Considering that the total number of comparisons is $21 \cdot 25 = 525$, Table 6 shows the values of Genuine Acceptance Rate (GAR) and False Rejection Rate ($FRR = 1 - GAR$) for each value of $|S|$.

Table 6. Values of GAR and FRR for each value of $|S|$.

bitlength of S: $\lvert S\rvert$	64	128	192	256
GAR	98.7%	97.3%	88.9%	74.5%
FRR	1.3%	2.7%	11.1%	25.5%

3.2 Inter-user variability: FAR

In this analysis, the templates of a given user are compared with all the templates of the rest of users. In this way a measure of the similarities among them is obtained.

This analysis is divided in two parts. In both parts of the analysis instead of using the 7 templates of each user, only one template is randomly chosen. In the first part, the templates considered are compared with the whole database (Templates vs. Database), and in the second part, the comparison is done only between the 25 chosen templates themselves (Templates vs. Templates). Then, two values for the False Acceptance Rate are obtained, FAR_1 and FAR_2, respectively.

Analysis of Templates vs. Database

In the first part of this analysis the 25 chosen templates are compared with the whole database formed by the 24 other users. The number of coincidences obtained for each value of bitlength are shown in Tables 7, 8, and 9 (for the value $|S| = 256$, there are only 3 coincidences, in the users 15, 18 and 22, so the table for this case is not shown).

Finally, taking into account all the values obtained, the False Acceptance Rate, FAR_1, for each bitlength, $|S|$, was computed, as it is shown in Table 10.

Analysis of Templates vs. Templates

In this part, the comparison is done only between the 25 chosen templates themselves, so in total there are 300 comparisons.

Table 11 shows the number of comparisons with, at least, $d+1$ coincidences and the corresponding value for the False Acceptance Rate, FAR_2.

Table 7. Number of comparisons with, at least, $d + 1 = 8$ coincidences for $|S| = 64$.

	Tpl. 1	Tpl. 2	Tpl. 3	Tpl. 4	Tpl. 5	Tpl. 6	Tpl. 7	Tpl. 8	Tpl. 9	Tpl. 10	Tpl. 11	Tpl. 12	Tpl. 13	Tpl. 14	Tpl. 15	Tpl. 16	Tpl. 17	Tpl. 18	Tpl. 19	Tpl. 20	Tpl. 21	Tpl. 22	Tpl. 23	Tpl. 24	Tpl. 25		
$> d = 7$	120	144	107	57	109	118	150	11	90	102	88	110	134	115	143	159	146	136	90	104	121	107	147	130	109		

Table 8. Number of comparisons with, at least, $d + 1 = 15$ coincidences for $|S| = 128$.

	Tpl. 1	Tpl. 2	Tpl. 3	Tpl. 4	Tpl. 5	Tpl. 6	Tpl. 7	Tpl. 8	Tpl. 9	Tpl. 10	Tpl. 11	Tpl. 12	Tpl. 13	Tpl. 14	Tpl. 15	Tpl. 16	Tpl. 17	Tpl. 18	Tpl. 19	Tpl. 20	Tpl. 21	Tpl. 22	Tpl. 23	Tpl. 24	Tpl. 25		
$> d = 14$	18	44	41	3	11	17	42	0	5	18	3	25	35	35	52	65	55	59	13	16	10	37	37	33	31		

Table 9. Number of comparisons with, at least, $d + 1 = 22$ coincidences for $|S| = 192$.

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

Table 10. Values of FAR_1 for each bitlength of S.

bitlength of S	64	128	192	256
FAR_1	67.78%	16.79%	1.35%	0.07%

Table 11. Values of FAR_2 for each bitlength of S.

bitlength of S	64	128	192	256
$> d$ coincidences	179	29	2	0
FAR_2	59.67%	9.67%	0.67%	0%

4 Conclusions and Future work

In this work, an analysis of the security efficiency of a fuzzy extractor scheme for iris templates is presented. The main conclusions are the following:

1. Referring to the global efficiency of the scheme, it can be stated that the lower the bitlength of S is, the easier to recognize a known user. Therefore the lower the percentage of False Rejection Rate is, which is a good improvement.

2. Nevertheless, at the same time, the lower the bitlength of S is, the easier to recognize an attacker as a known user, as the values of FAR show.
3. As the False Acceptance Rate is a security relevant measure while the False Rejection Rate is more a comfort criteria, a commitment between these two values has to be taken, but giving more importance always to the FAR.
4. Thus, from the four different values of $|S|$ analyzed, the best one in relation to the intra-user variability is $|S| = 256$ ($FAR_1 = 0.07\%$ and $FAR_2 = 0\%$); whereas the best value in relation to the intra-user variability is $|S| = 64$ ($FRR = 1.3\%$). So, it cannot be stated in a definitive way what value of $|S|$ is the best. That value will depend on the security requirements of the application where this scheme will be used. Anyway, the most balanced solution from the values of FAR and FRR could be $|S| = 192$.

From the previous conclusions, it would be of interest to improve the implementation of the scheme in order to perform the experiments faster and to use bigger values for $|S|$, for example, 384, 512, etc. Moreover, it is important to improve the extraction algorithms for iris templates in order to reduce the inter-variability and increase the intra-variability of users.

Acknowledgment. This work has been supported by Ministerio de Industria, Turismo y Comercio (Spain) in collaboration with Telefónica I+D (Project SEGUR@) with reference CENIT-2007 2004.

References

1. E. Diez Laiz, Master Thesis, Universidad Politécnica de Madrid, to appear.
2. 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.
3. F. Hernández Álvarez, L. Hernández Encinas, and C. Sánchez Ávila, Biometric Fuzzy Extractor Scheme for Iris Templates, *Proc. of The 2009 World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLDCOMP'09)*, to appear.
4. A. Juels and M. Sudan, A fuzzy vault scheme, *Designs, Codes and Cryptography*, **38**, 2 (2006), 237–257.
5. A. Juels and M. Wattenberg, A fuzzy commitment scheme, *Proc. of the 6th ACM conference on Computer and Communications Security*, (1999), 28–36.
6. A.K. Jain, K. Nandakumar, and A. Nagar, Biometric Template Security, *Journal on Advances in Signal Processing* **8**, 2 (2008), 17 pp.
7. 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.
8. A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, 1997.
9. 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.
10. U. Uludag, S. Pankanti, S. Prabhakar, and A.K. Jain, Biometric Cryptosystems: Issues and Challenges, *Proc. of the IEEE*, **92**, 6 (2004), 948–960.