A Study on the Convergence of FingerHashing and a Secured Biometric System

Rima Belguechi[1], Christophe Rosenberger[2],

[1] LCSI Laboratory, National School of Computer Science, ESI, Algeria
r_belguechi@esi.dz
[2] GREYC Laboratory, ENSICAEN - University of Caen - CNRS, France
christophe.rosenberger@greyc.ensicaen.fr

Abstract. Because biometrics-based authentication offers several advantages face to other authentication methods, it is important that such systems be designed to withstand attacks. Reliability and privacy for the public acceptance of the system are also important factors. BioHashing is presented as a new technique to moderate the impact of susceptible threats. The acceptance of this approach depends on whether it has low error rates and is tamper proof. We study so in this paper, the relevant advances in this area being more focused on fingerprint modality due to its widespread usage. We also consider a FingerHashing smartcard-based implementation and try to emphasize how this system can meet a secured biometric system.

Keywords: Authentication, Biometry, FingerHashing, Security.

1 Introduction

User authentication is a great challenge for security reasons. Integrity of data and transactions in various applications relies on verifying the participants' identities. A reliable personal authentication is critical in many daily operations. For example, physical access control and computer privileges are becoming ever more important to prevent their abuse.

The three basic forms of user authentication that can be used independently or in combination with others, are *knowledge based* which rely on a secret such as a password held by the user, *token based* that rely on possession of a 'token' (such as a physical key or a smartcard) and *biometric based* that uses unique characteristics of individuals (such as fingerprints or voice prints). While knowledge can be forgotten or duplicated, tokens stolen or lost, biometrics does not suffer from these deficiencies, and can provide the security of long passwords without sacrificing the ease of memorizing short ones [1]. In addition, biometric authentication is not easy to transfer or share; it is a powerful weapon against repudiation. Biometric authentication necessitates two phases: *enrollment* and *authentication* (or *verification*). Enrollment involves measuring an individual's biometric data to construct a *template* for storage. A template is a small file containing distinguishing features of the user derived from his/her biometric data. Authentication involves a measurement of the same data and comparison with the stored template. Even though automated biometrics can help alleviate the problems associated with the existing methods of user authentication, hackers will still find there are weak points in the system.

Password systems are prone to brute force dictionary attacks. Biometric systems, on the other hand, require substantially more effort for mounting such an attack. Yet, there are several new possible types of attacks. In this paper, we highlight the main weaknesses related to biometrics and try to emphasize some existing solutions rising above these limitations. The goal is to outline the *BioHashing* technique as a promising practical solution, being more focused on fingerprint modality in one hand. In the second hand, we try to study the impact of using a trial factor authentication composed of: smartcard, biometry and tokenized-random number.

The paper is organized as follows: In Section 2, we discuss the security elements in a biometric-based system. In Section 3, we present the BioHashing technique as a dual factor authentication. In Sections 4 and 5, we detail the research issues in FingerHashing. In section 6, we study the security relevance of a FingerHashing system embedded in a smartcard.

2 Biometric authentication

Biometric-based authentication has many usability advantages over traditional systems; however, suffering from some inherent security threats as it is underlined here. Biometric authentication, in terms of pattern recognition system, is exposed to brute-force attacks in each level of the complete process (sensor, feature extractor, template matcher). These attacks such as fake biometric signal are discussed in [2].

A problem with biometric authentication systems arises when the data associated with a biometric feature has been compromised. For authentication systems based on physical tokens such as keys and badges, a compromised token can be easily canceled and the user can be assigned a new token. Similarly, user IDs and passwords can be changed as often as required. If the biometric data is compromised, the replacement is impossible. In order to alleviate this problem, Ratha [1] introduces the concept of *cancellable biometrics*.

Deploying biometrics in a mass market, like credit card authorization or bank ATM access, raises additional concerns beyond the security of the transactions. One such concern is the public perception of a possible invasion of privacy. If an attacker can intercept a person’s biometric data, then the attacker might use it to masquerade as the person, or perhaps simple to monitor his/her private activities.

Another risk is related to the database of stored templates which may be tampered. The data might be distributed over several servers. Here, the attacker could try to modify some templates in the database, which could result either in authorizing a fraudulent individual or denying service to the persons associated with the corrupted template.

Performance evaluation of biometric-based authentication systems is another important issue. Authentication session compares a live biometric sample provided by the user with the user’s reference template generated by the system during the enrollment procedure. This biometric matching determines the degree of similarity between the live submitted biometric sample and the reference template. The result of this comparison is a number known as a match score, which, in most systems is compared against a tolerance threshold. Let's denote the stored template P' and the acquired one by *P*. In terms of hypothesis testing, we have:

H_0 : *P = P'*, the person is genuine.

H_1 : $P \neq P'$, the person is an impostor.

A similarity measure s = *Sim (P, P')* is often defined and *H0* is decided if s $\geq$*Th (Th* is a the biometric decision threshold*)* and *H1* is decided if s < *Th.* Deciding *H0* when *H1* is true gives a false acceptation; deciding *H1* when *H0* is true results in a false rejection. False Accept Rates (*FAR*) and False Reject Rates (*FRR*) are important intrinsic characteristics of a matcher. The choice of value for the tolerance threshold therefore involves a trade-off between the two types of error and determines the security and convenience of a biometrics-based authentication system. In practice, it is almost impossible to obtain both zero FAR and FRR errors, so realization of relatively low FAR, i.e. acceptance of impostors, will yield relatively high FRR, i.e. rejection of genuine and otherwise. In [7], the impact of denial of access in biometric systems is pointed. Another index of performance is equal error rate (EER) defined as the point where FAR and FRR are equal. A perfect system would have a zero EER value.

There is a substantial research going on to find solutions/alternatives to the problems mentioned above:

2.1 Enhance biometric privacy

The most straightforward way to secure the biometric template is to put it on a smartcard. In 1998, Davida et al. [3] were among the first to suggest biometric based authentication systems which do not require the incorporation of an on-line database for the security infrastructure. An off-line biometric system is achieved by incorporating a biometric template on a storage device/token (smartcard). Presently, there are quite a number of literatures that reported the integration of biometrics into the smartcard [4–5]. However, the only effort being applied in this line is to store the user's template inside a smartcard, protected with Administrators Keys, and extracted from the card by the terminal to perform the verification. Some are allowed to verify themselves in the card, but with performance downside [5].

Assuming that such tokens are tamper resistant is not always true. In general, there are two main classes of physical attacks against smartcards: non-invasive and invasive attacks [8]. So, it is possible that the template can be gleaned from a stolen card.

Encrypting the template prior to storage can make template compromise harder. But, due to the intra-user variability over multiple acquisitions of the same biometric trait, one cannot store a biometric template in an encrypted form and then perform matching in the encrypted domain.

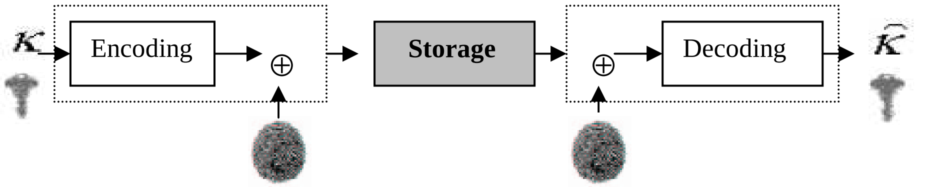
Today, we observe that the wide range of techniques to protect the privacy of the generated template can be widely divided into two categories:

i – Biometric cryptosystems

ii – Discretisation or feature transformation.

In *biometric cryptosystems* a helper data as a secret key k is combined with the template to lock the biometric set. Here, error correcting codes were designed as an alternative to deal with the problem of changed data between two different scans of the same biometric data. Figure 1 illustrates a possible crypto-system scheme:

Figure 1. Example of template protection by cryptosystem technique.



Davida et al. [3] presented an authentication algorithm based on Hamming error-correcting codes, the error correcting digits and some other verifying data are stored. This approach was applied to iris scans. The amount of correction required serves as a measure of the authentication success. The author’s assumption that only 10% bits of iris code can change among different presentations of the iris of a person is too restrictive. In fact, the disagreement of the inter-personal iris codes is usually 40%-60%. In [9], the authors use a concatenation of Hadamard and Reed-Solomon codes that can correct out of 32% of errors, results become far well. Monrose et al. [10] map keystroke derived attributes into binary string using the Shamir thresholding technique. Shamir thresholding and Hamming coding are considered equivalent. The same process was applied to voice [11]. They achieve a FRR of 48% for the keystroke and 20% for the voice.

Thereby, while error correcting codes are considered suitable for iris recognition, dealing with fingerprint is more difficult. A fussy vault scheme was presented by Juels and Sudan [12]. This approach is more compatible with partial and reordered data like fingerprint minutiae. It uses the polynomial interpolation to lock the template set. The application of fuzzy vault to fingerprint identification appeared in the work of Clancy et al. [13]. The shortcoming of the fuzzy scheme is the high FRR which is near 30%.

In *discretisation techniques*, the goal is to transform the continuous biometric data x with an error tolerant function $H(x)$ to obtain a discrete bitstring code. Such approaches are *direct hashing* to store a hash of the biometric data rather than the biometric itself. Biometric hashes are largely described in [14]. However, these attempts suffer from an excessive FRR (usually over 20%). In [8], the authors present a symmetric hash function for fingerprint. This algorithm performs good results (EER=3%) but it has still a lowest accuracy than the baseline system (EER=1,7%).

In [15] , Goh and Ngo introduce the tokenized biometric discretization. By combining the high uncertainty and low entropy biometric data with user specific random data, the inherent entropy of the resulting template is increased. Another advantage of combining tokenized pseudo-random is to obtain a cancellable biometric data. To re-issue the user identity, we need to provide a specific new token. They denote this model as *BioHashing*. This is beginning to approach the parameters needed for a practical system. The next session exposes in more details the realistic model.

2.2 Enhance biometric performance

The biometric data acquired from an individual during his verification may be very different from the data used to generate the template during the enrollment, thereby affecting the matching process. This variation is typically caused by a user who is incorrectly interacting with the sensor, or when sensor characteristics are modified.

Multimodal biometrics can increase the system performance. Despite that, multimodal biometrics is not a solution for the privacy invasion problem. Moreover, the use of multiple biometric measurement devices will certainly impose significant additional costs, more complex user–machine interfaces and additional management complexity.

In the next section, we present the BioHashing method as a solution which may tackle in one way much of these presented biometric weak links.

3 An overview of BioHashing

3.1 Principles

In general, the process of BioHashing (see Figure 1) has two stages. In the first stage, certain features $\left(f_1, f_2,..., f_n\right)$ are derived from the raw biometric signal β. In the second stage, features are mapped to a binary descriptor $b \in \{0,1\}^m$, where m is the length of the bitstring code. The extraction process includes signal acquisition, pre-processing and feature extraction. Different biometric signals exploit different techniques in the first process but the focus of our analysis is discretization, the secret of BioHashing, consisting of four steps [15]:

1) Generate a set of pseudo-random vectors Γ. In practice, random number sequence r could be generated from a physical device, i.e. an USB token or a smartcard through a random number generator. The seed is different among different users. For test, random bit/number algorithms are publicly available such as ad hoc scheme.
2) Apply the Gram-Schmidt process to transform the basis Γ into an orthonormal set of matrices $r_{\perp i}$ $i = 1..m$.
3) Compute the inner product between the biometric feature f and $r_{\perp i}$ ($\langle f | r_{\perp i} \rangle$), $i = 1..m$. This projection results in an error tolerant representation.
4) Compute a m-bits BioHash denoted b ($b \in 2^m$), $b_i = \begin{cases} 0 \; if & \langle f | r_{\perp i} \rangle \leq \tau \\ 1 \; if & \langle f | r_{\perp i} \rangle > \tau \end{cases}$, where τ is a preset threshold.

The resulting bitstring **b** named *BioHash code* is compared by the Hamming distance for a matching score. The security of the process is assured if the BioHash code is non invertible.

3.2 Performance evaluation

We will depict the performance of BioHashing by exploiting the main dedicated works (Table 1). The performance of a biometric system is commonly described by its false acceptance rate (FAR) and false rejection rate (FRR). Another index of performance is equal error rate (EER) defined as the point where FAR and FRR are equal. A perfect system would have zero EER. The table 1 resumes the main results of the approaches that have since been developed. We can conduct the following remarks:

- BioHashing performance does not rely on specific biometrics ;
- Zero equal error rate can be achieved ;
- Clean separation between impostor and genuine distribution ;
- Even if the feature extractor is low, performance is accurate ;
- Privacy is granted.

Hence, all seems to be perfect, until in [19], the authors put in evidence the anomalies of the BioHashing approach and conclude that the claim of having achieved a zero EER is based upon the impractical hidden assumption of no stealing of the Hash key. Moreover, they proved that in a more realistic scenario where an impostor steals the Hash key the results are worse than when using the biometric alone. So, today, the challenge is to overcome this drawback. We focus in the next section on FingerHashing that uses the fingerprints of an individual as biometric modality.

Table 1. Summary of BioHashing main implementations

Biometric modality	BioHashing EER	Baseline system EER	Reference
Face	0%	> 10%	[15]
Palmprint	0%	2,015%	[17]
Fingerprint	0%	5,66%	[16]
Iris	0%	3,20%	[18]

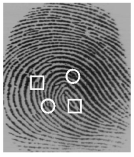
4 FingerHashing

FingerHashing can be decomposed into two components: (i) feature extraction and (ii) discretisation steps.

i. Feature extraction

Various approaches of automatic fingerprint matching have been proposed in the literature. Fingerprint matching techniques may be classified as being minutiae-based, correlation-based or image-based. Most of the existing systems are based on minutiae features (ridge bifurcation and ending; see Figure 2). Such systems first detect the minutiae in a fingerprint image and then use sophisticated alignment techniques to match two minutiae sets.

Figure 2. Example of fingerprint minutiae, ridge endings (□) and ridge bifurcations (○).



In correlation-based approaches, the template and the query fingerprint image are spatially correlated to establish the degree of similarity. In image-based approaches, the features are directly extracted from the raw image. Moreover, image based methods may be the only viable choice, for instance, when image quality is too low to allow reliable minutiae extraction.

Fingerprint matching is affected by the non-linear distortion introduced during the image acquisition due to the elastic nature of the skin. The non-linear deformation causes fingerprint features such as minutiae points to be distorted in a complex manner: Consider an image $f_2(x,y)$ that is a rotated, scaled translated replica of $f_1(x,y)$:

$$f_2 = f_1\left(\sigma(x\cos\alpha + y\sin\alpha) - x_0, \sigma(-x\sin\alpha + y\cos\alpha) - y_0\right) \qquad (1)$$

where α is the rotation angle, σ the uniform scale factor, and x_0 and y_0 are translational offsets.

For a reliable matching, this non-linear deformation must be accounted.

In [20], the authors proposed a novel representation scheme that captures global and local features of a fingerprint in a compact fixed-length feature vector denoted as *FingerCode*. This technique uses texture features available in a fingerprint to compute the feature vector by Gabor Filters. Their scheme for generic representation of oriented texture relies on extracting a core point in the fingerprint. The decision is made using Euclidean distance between FingerCodes. In [22], the authors made a focus on this comparison step and replace the Euclidean distance by a more robust classification technique.

In [21], the authors proposed an integrated Wavelet and Fourier–Mellin transformed (WFMT) feature. The wavelet transform preserves the local edges and noise reduction in the low-frequency domain and FMT is translation invariant and represents rotation and scaling as translations along the corresponding axes in the parameter space. Because these presented techniques are invariant to non-linear deformation contrary to minutiae features, furthermore, they extract a feature vector of a fixed length (FingerCode = 640 real values), all the feature extractor used in the FingerHashing are image-based method.

ii. Discretisation

This step has been described in Section 3.1. The main contributions on FingerHashing from the state of the art are detailed in Table 2. We denote M1 the FingerHashing performed from the WFMT feature vector. In M2, M2+ and M3, this feature is considered as FingerCode. In M3, the FingerCode is concatenated with the DCT (Discret Cosinus Transform) of face features while in M4, this FingerCode is concatenated with the Reed-Solomon code. Biometric matching column is related to the comparison method when using the baseline biometric method alone. BioHash matching is realized when comparing bio codes. The rest of parameters are resumed in the table, they consist of m the length of BioHash code, τ the binarization threshold and N the normalisation prefix (when the feature vector is normalized), with some particularity on M2+. In this case, a set of $k \times p$ codes is generated.

Table 2. Summary of the principle contributions in FingerHashing in the state of the art.

	Feature vector	**Biometric matching**	**m**	τ	**N**	**BioHash matching**	**Ref**
M1	WFMT	ED	80	0.13	No	HD	[16]
M2	FingerCode	PWC	100	0	Yes	HD	[22]
M2+	FingerCode	PWC	Generate k spaces	Varying τ in p steps	Yes	Totalling the $k \times p$ scores	[22]
M3	FingerCode	SC	100	0	Yes	HD	[19]
M3+	FingerCode|DCT face features	SC	200	0	Yes	HD	[19]
M4	FingerCode|RS code	ED	180	0	No	HD	[24]

Acronyms: *ED* : Euclidean Distance , *HD* : Hamming Distance, *PWC* : Parzen Window Classifier , *SC* : Specific Classifier.

5 Comparative study

We intend to measure the efficiency of the previous methods mentioned above as reported in the literature. The comparison is achieved on images taken from FVC2002 [23]. FVC2002 provided four fingerprint databases: DB1, DB2, DB3 and DB4, three of these databases are acquired by various sensors, low cost and high quality optical and capacitive whereas the fourth contains synthetically generated images. In this paper, we selected DB2 as the experimental benchmark. DB2 contains eight impressions of 100 different fingers, hence 800 images in total. However, the comparison only can be done if both fingerprint images contain their respective core points, but two of eight impressions for each finger have no core point due to the exaggerate displacement. In experiments, these two impressions were excluded resulting in 600 images. The performance is evaluated in term of EER.

Table 3 shows the results in the case *Bio* where the sole biometric data is used, *Best* when FingerHashing is performed in the best hypothesis while never an impostor steals the key and *Worst* when always an impostor steals the key. As a conclusion, this comparison shows that:

- The FingerHashing outperforms dramatically the base biometric in the best cases for all methods ;
- In the worst case, the sole biometric is always better. Note that M1 has not been tested under this hypothesis ;
- Tuning the correct range interval for τ is a critical operation (M2+) ;
- The length of the BioHash code is a critical point. By increasing this space (M2+), the performance becomes better. Augmenting the feature vector is another alternative to enhance this length, as in (M3+) done by sequencing face and fingerprint vectors ;

– Normalisation of the feature vector is recommended ;
– (M2+), (M3+) offer a good trade-off between best and worst cases ;
– The invariance of these methods is proven by reliable detection of the core point;
– A study has to be done in order to understand why the error correcting codes provide very bad results. They perform very well in the best case ;
– The BioHash codes are always compared using the Hamming distance. The problem can also be seen as a two class pattern recognition problem ;

Table 3. Results obtained from FingerHashing-based methods in term of EER

	Bio	Best	Worst
M1	5,3%	**0%**	-
M2	5,2%	1%	15,5%
M2+	**5,2%**	**0,2%**	**7,5%**
M3	2,5%	1,5%	10,9%
M3+	**4,9%**	**0,7%**	**2,5%**
M4	11,7%	0,1%	50%

6 Proposal of FingerHashing authentication system in smartcard

From this previous study, we believe that as yet there no “best” approach for biometric system security. The application scenario and requirements play a major role in the selection of the biometric technique scheme. For instance, dealing with fingerhashing as promising overall solution, we will try to propose a secured architecture system. As it happens, biometric researchers pointed Biohashing perils (see worst case, table3) in a situation where imposters gain access, at the same time to the biometric template as the randomized token. So, if we assume that both the BioCode and the token will not be compromised simultaneously; BioHashing can be a sufficient secure scheme. For this purpose, we will use a smartcard as a secured way for storing the biocode template and for making the match-on-card operation. The validity of data in smartcard will be guaranteed by a certificate authority (CA). And we will try to emphasize how the system could counter threats to security.

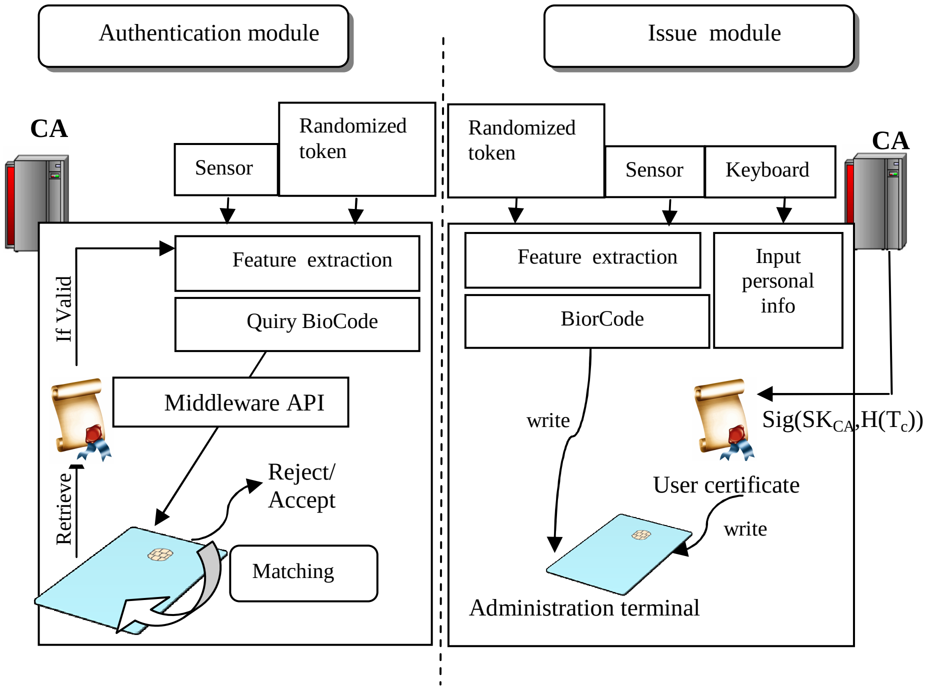
6.1 Development of the prototype system

The system infrastructure (see Figure 2) will be composed by two modules:

- Issue module
    - Administration terminal: collect personal data, acquire biometry, extract the fingerprint template which is the fingercode as done in [24] and generate the biocode.
    - Certificate authority: Issues X.509 certificates T_C that are signed by the authority private key SK_{CA}.
    - JavaCard to embed the biocode and the user certificate.
- Authentication module

- Client interface: The certificate validity is verified using the authority public key PK_{CA}. The matching is processed in the smartcard using hamming distance.

Figure 2. Authentication infrastructure system



6.2 Security analysis

For the sake of convenience, we use the notation *{I,B,T}* to represent user credential, a user smartcard *I*, its associated biometric signal *B* and its randomized token *T*. A registered user X is enrolled by the information $\{I_X,B_{XD},T_X\}$. Suppose user X provides his/her credential $\{I_X,B_{XV},T_X\}$ at the time of verification. Even though B_{XD} and B_{XV} are from the same person, because of various noises, they are not identical. We represent an imposter by Y pseudo. The fish-bone model in figure 3 modelises this biometric system security. We assume that either token or smartcard can be forgotten, lost, stolen, duplicated or shared; otherwise the need of biometry will be meaningless. We summarise the biometric system failure by two main kind : (i) denial of service, (ii) intrusion.

Case1: This is a normal case when a genuine user X will use the client interface. From the matching score, there are two possible responses: “correct acceptance” or “false rejection” which depends on the intrinsic performance of the biometric system. Our biometric system meets globally the same performance as M4 in table3, so the denial of service risk approximates 0%.

*Case2:*This case occurs when an impostor tries to use a counterfeited card. Or, the authenticity of cards are controlled by a PKI infrastructure, so the success of a counterfeiting attack depends on the secrecy of the CA secret key which is isolated on the administration terminal from any suspicious network liaison. Concerning the certificate, it is clear that in the infrastructure proposed the user certificate can traverse some communication channels. By this mean, it becomes prone to replay or man in the middle attacks. Note that for the sake of simplicity, we have only considered social engineering problems as card stolen or shared. For the network attacks, we not make a focus on possible solutions as Challenge/Response or time stamp mechanisms. Consequently, the intrusion risk depends on the validity of the PKI infrastructure. We can assume that it is about 0%.

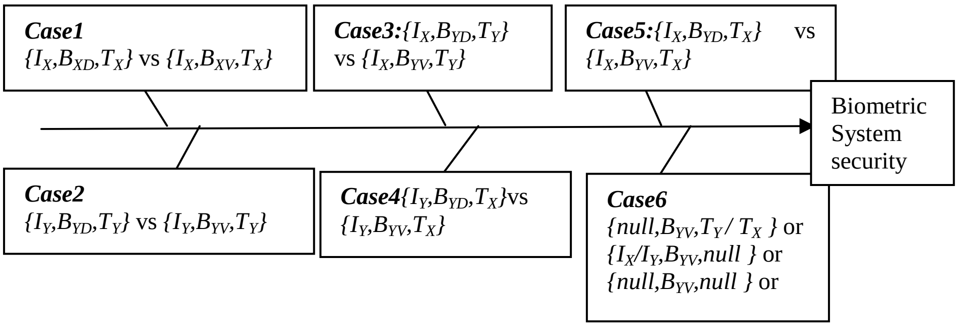
Case3: Here, the card of the user X has been shared, stolen or duplicated by the impostor . Since, he has not the randomized token of X; the risk of intrusion is of 0%.

Case4: the impostor has the token of the user X but not his card. The intrusion risk is at 0%.

Case5: This case is the worst case, when an impostor has simultaneously the card and the token of the user X. The intrusion risk depends on the intrinsic performance of the biometric system which approximates, here, 18%.

*Case6:*If the impostor, don’t present at the interface any one of the authenticator card or token, the system will automatically refuses his transaction.

Figure 3. Fish-bone model for enumerating security threats of the proposal system.



Case1 $\{I_X,B_{XD},T_X\}$ vs $\{I_X,B_{XV},T_X\}$
Case3:$\{I_X,B_{YD},T_Y\}$ vs $\{I_X,B_{YV},T_Y\}$
Case5:$\{I_X,B_{YD},T_X\}$ vs $\{I_X,B_{YV},T_X\}$
Biometric System security
Case2 $\{I_Y,B_{YD},T_Y\}$ vs $\{I_Y,B_{YV},T_Y\}$
Case4$\{I_Y,B_{YD},T_X\}$vs $\{I_Y,B_{YV},T_X\}$
Case6 $\{null,B_{YV},T_Y/ T_X\}$ or $\{I_X/I_Y,B_{YV},null\}$ or $\{null,B_{YV},null\}$ or

7 Conclusion

Any system, including a biometric one, is vulnerable when attacked by determined hackers. We have highlighted the research advances to enhance such systems performance. We focused our attention on BioHashing which is a recent technique that can address simultaneously the invasion of privacy issue and the denial of access problem. We have proposed a FingerHashing-based authentication system which can converge to a secured system unless in case of the worst scenario when both registered token and card are in a possession of an impostor. In this case, we have seen that the weakness of the approach is related to the length of the biohash code and the decision making which is always done by hamming distance. In future work, we intend to combine all these remarks for augmenting the security degree of a FingerHashing authentication system.

References

1. Ratha, N.K., Connell, J.H., Bolle, R.: Enhancing Security and Privacy in Biometrics-Based Authentication System. IBM Systems J., vol. 40, no. 3, pp. 614-634 (2001).
2. Bolle, R.M. ,Connel, J.H., Ratha, N.K.: Biometric Perils and Patches. Pattern Recognition 35, pp. 2727–2738 (2002).
3. Davida, G.I., Frankel, Y. and Matt, B.: On Enabling Secure Applications through Off-Line Biometric Identification. Proc Security and Privac (1998).
4. Isobe, Y., Seto, Y., Kataoka, M. :Development of personal authentication system using fingerprint with digital signature technologies. Proc. System Sciences (2001).
5. Sanchez-Reillo, R. : Including biometric authentication in a smart card operating system. AVBPA, 2001.
6. Rila, L. : Denial of access in biometrics-based authentication systems. Infrasec (2002).
7. Kuhn, M., Anderson, R. :Tamper resistance: A cautionary note". Workshop on Electronic Commerce (1996).
8. Tulyakov, S., Chavan, V.S. and Govindaraju, V.: Symmetric Hash Functions for Fingerprint Minutiae. Proc. in Workshop Pattern Recognitione, pp. 30-38 (2005).
9. Hao, F., Anderson, R., Daugman, J. : Combining cryptography with biometrics effectively. Trans on computers, pp. 1081-1088 (2006).
10. Monrose, F., Reiter, M.K. and Wetsel, S.: Password hardening based on key stroke dynamics. Conference on Computer and Communications Security (1999).
11. Monrose,F.: Crypptugruphic Key Generation from Voice. IEEE Symp QI-I Security & Privacy (2001).
12. Juels, A., and Sudan, M. : A fuzzy vault scheme, Proc. IEEE Int. Symp. Information Theory (2002).
13. Clancy, T.C., Kiyavash, N., Lin, D.J.: Secure smartcard-based fingerprint authentication. Proc. Multimedia, Biometrics Methods and Applications (2003).
14. Pawan, K.J., Siyal, M.Y.: Novel biometric digital signature for Internet based applications, Information Management and Computer Security. pp. 205–212 (2001).
15. Goh, A., Ngo, D.C.L.: Computation of cryptographic keys from face biometrics. Information Processing (2003).
16. Teoh, A., Ngo, D., Goh, A.,: BioHashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern Recognition (2004).
17. Connie, T., Teoh, A., Goh, M., Ngo, D. : PalmHashing: a novel approach for dual-factor authentication. Pattern Anal. Appl, pp. 255–268 (2004).
18. Nanni, L., Lumini, A. : Empirical tests on BioHashing. Neurocomputing (2006).
19. Kong, B., Cheung, K., Zhang, D., Kamel, M., You, J. :An analysis of BioHashing and its variants, Pattern Recognition , pp. 1359–1368 (2006).
20. Jain, A.K., Prabhakar, S., Hong, L., Pankanti, S.: Filterbank-based fingerprint matching. IEEE Trans. Image Process, Vol 5, pp. 846–859 (2000).
21. Teoh, A., Ngo, D., Song, O.T. :An efficient fingerprint verification system using IWFM Invariant Transform. Image and Vision Computing Journal, Vol 22 (2004).
22. Maio, D., Nanni, L. :An efficient fingerprint verification system using integrated Gabor filters and Parzen Window Classifier. Neurocomputing, pp. 208–216 (2005).
23. http://bias.csr.unibo.it/fvc2002/
24. Song, O., Jin, T., Ngo, L.: Application Key Release Scheme from Biometrics. Network Security J, (2008).