Biohashing for securing fingerprint minutiae templates

Rima Belguechi, Christophe Rosenberger, Samy Ait Aoudia

BioHashing for securing fingerprint minutiae templates

Rima Belguechi[1], Christophe Rosenberger[2], Samy Ait Aoudia[1]
[1]National School of Computer Science, ESI, Algeria
[2]GREYC Laboratory, ENSICAEN – University de Caen Basse Normandie – CNRS, France

Abstract

The storage of fingerprints is an important issue as this biometric modality is more and more deployed for real applications. The a prori impossibility to revoke a biometric template (like a password) in case of theft, is a major concern for privacy reasons. We propose in this paper a new method to secure fingerprint minutiae templates by storing a biocode while keeping good recognition results. We show the efficiency of the method in comparison to some published methods for different scenarios.

1. Introduction

Compared to typical credentials based on knowledge or possession, morphological modality presents many advantages including ease-of-use and stronger non-repudiation properties. Despite of its inherent qualities, in practice, there are some obstacles in a wide adoption of biometrics [1,2]. One of critical threat in biometric systems is the theft of biometric data (a biometric template is considered stolen when either an attacker captures the stored template or creates a physical spoof [3]). Further, a hacker can cross link the stolen templates with other biometric databases, allowing him to track the activities of an enrolled person, thereby compromising his privacy.

Unlike passwords, when the biometric template is compromised, it cannot be cancelled or revoked. It remains stolen for life! To address this urgent issue, template protection schemes [4] are presented as prominent solutions, but unfortunately not yet mature for large scale deployment.

Considering the fingerprint modality which is major biometric one, many technologies integrating biometrics with cryptography have been proposed. Fuzzy vault approach consisting of binding a key to the template is the most used one to secure fingerprints [5]. In parallel, techniques inspired from password salting mainly known as Biohashing; and, non-invertible transforms where the original biometric is transformed using a one-way function are also attractive for their revocability or anonymity properties. To have a good review of these techniques, a rigorous analysis is needed. The main criteria to be considered when dealing with a protection scheme are: i) *performance*, ii) *non-invertibility* and iii) *cancelability* or *diversity*. The most robust implementation of Fuzzy vault was done in [5] but without any mention of cancelability scenario. Ratha et al. [6] proposed three different one-way transformations (cartesian, polar and functional). However, the administration of revocability is not easy and the performance is largely decreased compared to baseline system. Farooq et al. [7] proposed a revocable linear fingerprint template which does not decrease the baseline performance but security becomes questionable. Teoh et al. [8] performs FingerHashing to WFMT (Wavelet Fourier Mellin Transform) feature of fingerprint. It consists of iterative inner product upon WFMT and a random base generated from a user-specific key (salt). The 0-EER can be achieved but if this key is stolen, the EER may be much higher than the plain system. Recently, in [9] authors show on face biometric how can BioHashing be immunized from performance degradation. Thereby, BioHashing presents good revocability properties. Because of its inability to deal an unordered set of points, FingerHashing was always applied to texture features which requires a reliable registration point (core) instead minutiae even reputed more robust (i.e. EER= 1.6% vs EER=12% on the same database). This prevents compatibility with existing databases and commercial fingerprint sensors. This paper presents new method for protecting minutiae templates with BioHashing process in order to satisfy criteria of privacy and revocability without lose of verification performance.

This paper is organized as follows: Section 2 describes in detail the proposed minutiae-based FingerHashing algorithm for template protection. In Section 3, we present and discuss experimental results. We finally, draw conclusion and discuss future perspectives in section 4.

2. Minutiae-based fingerhashing

As mentioned before, BioHashing process was exclusively applied to texture features of fingerprint. Mainly, these features are extracted using FingerCode[10] in region of interest around the core point. In order to overcome the dependence on reference point and to increase the robustness of recognition, the idea is to represent each minutia by its FingerCode and to protect each FingerCode by the BioHashing process (see Fig.2.). The steps of the proposal system are:

- **Feature computation**

1. Extract minutiae template from the raw image.
2. Compute for each minutia its FingerCode. The result will be called MinuCode.
3. Process BioHashing to each MinuCode.

- **Feature matching**

1. Correct rotation deformation; note that fingercode is tolerant to translation.
2. Process Biohashing to the set of fresh MinuCodes.
3. Perform the local matching algorithm between the two template maps.

2.1. Feature computation

We use the same process as reported in [11] for minutiae extraction. The following algorithm generates MinuCodes with slight differences between the original approach [12]:

- **For each minutia *m* do**

- Valid the region of interest ROI surrounding m: this ROI is determined by a circular tessellation using B bands of b width. Each band is devised in 16 sectors of the same angle (22,5°). This ROI is valid if it is in the boundary of the image and each sector S represents an alternation of ridges and valleys. We express this alternation by the energy E of Fourier spectrum so, if $E>T_r$ then S is valid (T_r is a global Otsu threshold).
- Filter the ROI in eight different directions using a bank of Gabor filters. Contrary to the original method, we don't need to normalize the ROI since we work on a binary image with no contrast (Fig.1.).
- Let $\mathrm{Im}_{i\theta}$ be the θ-direction filtered image for sector S_i, $i=\{1..Bx16\}$. The feature vector or the MinuCode is $F=(f_1,...f_u)$ / u=Bx16x8(directions)

with $\forall f_{i\theta}\in F, f_{i\theta}=\frac{1}{n_i}\sum_{n_i}\left|\mathrm{Im}_{i\theta}(x,y)-p_{i\theta}\right|$, n_i is the number of pixels in S_i and $p_{i\theta}$ is their mean.

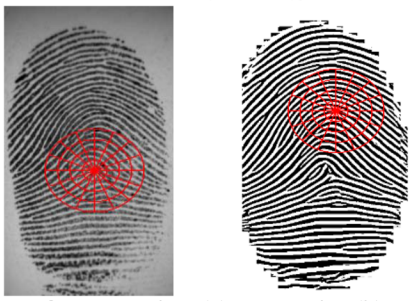


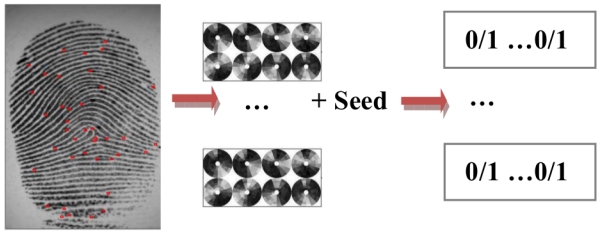
Fig.1. Reference point: (a) core point (b) minutia point

- **For each MinuCode *mc* process BioHashing as:**

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.
2. Apply the Gram-Schmidt process to transform the basis Γ into an orthonormal set of matrices $r_{\perp i}$, $i=1..v$ and $v\le u$.
3. Compute the inner product between the biometric feature f and $r_{\perp i}$ $\langle f|r_{\perp i}\rangle$, $i=1..v$.
4. Compute a v-bits BioHash denoted b $(b\in 2^v)$,

$$b_i=\begin{cases}0 \text{ if } \langle f|r_{\perp i}\rangle\le\tau\\ 1 \text{ if } \langle f|r_{\perp i}\rangle>\tau\end{cases}, \tau \text{ is a preset threshold.}$$

The resulting bitstring **b,** named *BioCode* represents the feature of each minutia.



(a) minutiae extraction (b) MinuCodes (c)BioCodes

Fig.2. Minutiae template protection by BioHashing Only BioCodes will be stored for matching.

2.2. Feature matching

Before matching, we first need to correct rotation deformation between template and input fingerprints. For this purpose, we compute the reference orientation of each image as presented in [13]. Correction consists

of rotating input image by the difference between the two reference orientations. Results obtained prove that when parameters are well tuned, the algorithm is very robust. Figure bellow corrects rotation when compared to the image (a) in Fig.2.

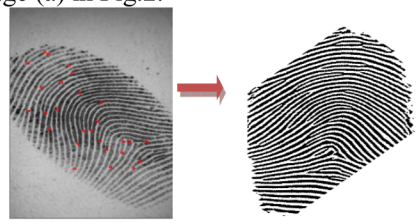


Fig.3. Example of rotation correction

Let $T = \{t_1, \ldots t_m\}$ and $P = \{p_1, \ldots p_n\}$ be the BioCodes lists extracted from the template and input fingerprints. The objective is to find a point p_j in P that exclusively corresponds to each point t_i in T if it exists. Due to combination of factors as intra user variation or non-linear distortion, reliable minutiae matching algorithm is still a challenging problem. Broadly, algorithms can be classified as Global matching or local matching. When compared to global approach, local matching algorithms are more robust to non-linear distortion and partial overlaps. So we have implemented here a local matching algorithm as follows motivated by the work done in [15]:

- We define a local neighborhood of minutiae m_i by the set $\{m_1, \ldots, m_k\}$ of K-nearest neighbors of m_i in term of euclidian distance.

The algorithm now consists of two phases:

- **Phase1:** it consists of the selection of the best matched pair ($root_1$,root2)/$root_1 \in T$ and $root_2 \in P$ by using the following cost estimation technique:

min = initial value; $root_1$ = -1; $root_2$=-1;
for i=1 to m
for j=1 to n
dist = D(t_i,p_j);
if (dist < min)
{min = dist; $root_1$ = i; $root_2$ = j;}

$D(t_i,p_j)$ is the hamming distance between biocodes of minutia t_i and of minutia p_j.

- **Phase2:** consider $root_1$ and $root_2$ first nodes to explore in T and I resp. Now, we have to match k-neighbors of $root_1$ with k-neighbors of $root_2$. Each matched pair will be pushed in a queue. This best candidate selection scheme will now be recursively repeated until the queue becomes empty (at each time the pair ($root_1$, root2) is popped from the queue head). To match two neighborhoods, we use a dynamic programming technique with a cost function equal to Hamming distance between biocodes. Finally, the matching score is computed by the following formula:

$$score = \frac{nb\ mathed\ pair}{Minimum(m,n)}$$ m,n size of T, P resp.

The figure below resumes process for the two previous fingerprint images of the FVC2002 DB2 benchmark [14]:

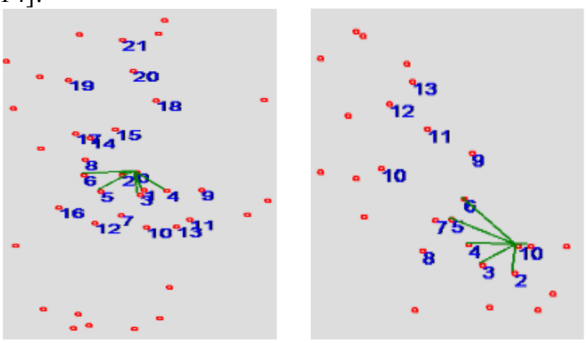


Fig.4. Numbering of minutiae with valid ROI and first neighborhood to be matched in each image ($root_1$=0,$root_2$=1).

Some remarks have to be done about Fig.4. :
- Not all minutiae are considered because the constraint of having valid ROI (we take B=3).
- The choice of k which is the number of neighbors is very important to avoid the local minimum problem, k=6 achieves a good compromise.
- The phase1 is sensitive to false minutiae. The extractor may be robust chiefly in the image center.

3. Experimental analysis

To evaluate the performance of the proposed system, we used a public-domain fingerprint database, namely, the FVC2002-DB2. This database [14] consists of 800 images of 100 fingers with 8 impressions per finger obtained using an optical sensor. The size of the images in this database is 560×296, the resolution of the sensor is 569 dpi and the images are generally of variable quality. We expect to compare our method with works done in [10]. Here, 2 out of 8 impressions for each finger in FVC2002 have an exaggerate displacement in core point, these two impressions were excluded, and hence, there are only 6 impressions per finger yielding 600 images in total. We put B=3 (number of bands) involving u=384 (MinuCode size) and v=384 (BioCode size).

3.1. Verification performance

The first experiment, where the key is never lost, is to find the baseline accuracy of the system (tab.1). We assigned each individual a randomized-token (simulation). One of six prints was enrolled in the database. We store BioCodes of minutiae template and neighborhood. We evaluate method in term of *FAR*

(False Acceptane Rate), *FRR* (False Rejection Rate) and EER (*EER = (FAR+FRR)/2*). We call M1, the proposed model when the matching is done with MinuCodes (without any protection), MP1 when we match BioCodes. F1 when we use FingerCode of the core point, FP1 when we protect FingerCode. O1 the original approach of FingerCode [12] and RP1 the protected model in [10].

	FAR	FRR	**EER**
Template without protection			
M1(proposed)	7.76%	8.81%	**8,28%**
F1	8.10%	9.85%	**8,98%**
O1	-	-	**12%**
Template with protection			
MP1(proposed)	0%	5.12%	**2,56%**
FP1	0%	7.98%	**3,99%**
RP1	-	-	**1,5%**

Tab.1. Results in never key lost scenario

We remark that without any protection scheme; M1 outperforms F1 which in turn is better than the original approach O1. With protection, MP1 is less than FP1 or RP1 because of the non overlap minutiae region since we just consider valid ROI.

The second experiment is the stolen token scenario (tab.2). This scenario is considered as the most critical one. We obtain following results:

	FAR	FRR	**EER**
MP1 (proposed)	9.63%	5.12%	**7,38%**
FP1	19.70%	7.98%	**13,84%**
RP1	-	-	**10,90%**

Tab.2. Results in always key lost scenario

Here, MP1 is the best one. The FRR is enhanced because the use of key increases the similarity in the intra class case.

3.2. Cancelability (diversity)

In the case of lost token or eavesdropping on database, we should be able to cancel the template and assuring diversity which means the difficulty in guessing one secure template given another secure template. For testing this, we assign each individual with *n* different keys and make comparison between templates. We always find matching score equal to 0% which means that templates are sufficiently distant.

3.3. Security

The proof of the non-invertibility property of BioHash have be done in [9]. So here, we just consider a brute force attack when the impostor does not have any knowledge of genuine BioCode or token. The complexity to guess the BioCode is at minimum equal to 384 bits (because we have considered only one minutia) so this is sufficiently hard to compute.

4. Conclusion

The novelty of the proposed method is to protect minutiae templates with BioHashing. The use of minutiae is much conform to existing databases. BioHashing, as our tests confirm it, is strongly cancelable (score=0) and it is mathematically proven to be non-invertible. In worst case, when the token is stolen, we have enhanced results compared to some published methods but we still believe that this is insufficient. In the near future, we expect to improve results by considering all minutiae to overcome the non overlap problem and by dynamically estimate the length of BioCode from the parameter B. We have also to enhance the CPU time; it is 30.92s in a 32bits PC.

References

[1] N. K. Ratha, J.H. Connell, and R.M. Bolle. An analysis of minutiae matching strength. *Conf on AVBPA*, 2001.

[2] B. Scheneier. Thu uses and abuses of biometrics. *Comm. ACM*, 42(8):136-136, 1999.

[3] R. Cappelli, A. Lumini, D. Maio, D. Maltoni. Fingerprint image reconstruction from standard templates. *IEEE Trans. PAMI,* 2007.

[4] A.K. Jain, K. Nandakumar, A. Nagar. Biometric template security. *EURASIP Journal*, 2008.

[5] K. Nandakumar, A.K. Jain, S. Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Trans. Inform. Forensics Security*, 2007.

[6] N. K. Ratha, S. Chikkerur, J. Connell, and R. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on PAMI*, 29(4):561.572, 2007.

[7] F. Farooq, R. Bolle, T. Jea, N. Ratha. Anonymous and revocable fingerprint recognition. *IEEE CVPR*, 2007.

[8] A. Teoh, D. Ngo, A. Goh. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition,* 2004.

[9] A. Teoh, Y. Kuan, S. Lee. Cancellable biometrics and annotations on BioHash. *Pattern recognition*, 2007.

[10] L. Nanni, A. Lumini. Empirical tests on biohashing. *Neurocomputing*, 2006.

[11] R.Belguechi, C.Rosenberger. A minutiae level fusion for AFIS systems. *EUSIPCO*, 2009

[12] A.Jain, S.Prabhakar, L.Hong, S.Pankanti. Filterbank based fingerprint matching. *Trans Image Process*, 2000.

[13] M. Liu, X. Jiang, A. Kot. Fingerprint reference-point detection. *AURASIP Journal*, 2005.

[14] FVC 2002, http://bias.csr.unibo.it/fvc2002/

[15] S. Chikkerur. *Online fingerprint verification system*. M.S Thesis, Univ of New York , 2005.