Comparative Study on Texture Features for Fingerprint Recognition: Application to The BioHashing Template Protection Scheme

Rima Belguechi, Adel Hafiane, Estelle Cherrier, Christophe Rosenberger

Comparative Study on Texture Features for Fingerprint Recognition: Application to The BioHashing Template Protection Scheme

Rima Belguechi[a,c], **Adel Hafiane**[b], **Estelle Cherrier**[c], **Christophe Rosenberger**[c]
[a]ESI Alger, 16309, El Harrach, Algeria
[b]INSA CVL, Univ. Orléans, PRISME EA 4229, 88 boulevard Lahitolle, F-18020 Bourges, France
[c]ENSICAEN - UNICAEN - CNRS, GREYC UMR 6072, F-14050 Caen, France

Abstract. In order to ensure privacy constraints in biometrics, new algorithms called template protection schemes have been proposed in the last ten years in the literature. Most of these algorithms require as input a feature having a fixed size. Texture features can be used within this context for fingerprints as the number of minutiae varies in general for different captures. BioHashing is a two authentication factor algorithm that can be used to enhance to ensure privacy while using biometrics. In this work, we compare different recent texture features from the literature within the BioHashing scheme while considering many constraints: efficiency, maximal representation size and constant size description. Experiments are conducted on three fingerprint databases from the FVC competition. Results permit us to conclude on the texture features to be used within this context.

Keywords: Fingerprint, texture analysis, cancelable biometrics, BioHashing, template protection..

* Christophe Rosenberger, christophe.rosenberger@ensicaen.fr

1 Introduction

Biometric recognition, which consists of checking the identity of a person starting from his anatomical or behavioral attributes, offers a possible solution to the problem of authentication in the identity management systems. Several studies expect an increase of the biometric market in relation with the development of the electronic transfers of data, in particular on the Internet (home-banking, ...). The interest for biometrics is explained by the fact that it offers an ergonomic manner to authenticate or to identify a person. Moreover, it constitutes an elegant way to address the problem of non-repudiation posed by the traditional elements of authentication such as passwords or tokens. However, to place biometrics in a prospect for industrial deployment may cause several reserves, and at the same time new privacy and security risks may arise. Fingerprints represent a large proportion of the biometric market. A fingerprint has the appearance of a surface of

alternating parallel ridges and valleys, on the majority of local regions. The minutiae details constitute the most popular representation for a fingerprint (see Figure 1). The minutiae represent local discontinuities and mark the positions where the ridges finish or fork. One problem of minutiae description is that the number of detected minutiae for different captures for the same individual is not constant. Comparison algorithms have to deal with this aspect.

Many attacks of a biometric system are possible such as on the capture device (spoofing attacks), on the communication link or on the software part. One of the most critical attacks against fingerprint recognition systems remains the hacking of the biometric templates stored in the database, which may either be done by the security personnel or by an intruder. In such situations, this raises multiple issues of privacy. For example, personal (biometric) information could be tracked from one application to another by cross-matching between biometric databases, thus compromising privacy (Imagine a case where a fingerprint template stolen from a bank's database may be used to search for a criminal in a fingerprint database or these data may be crossed with health records). Moreover, when the biometric template is compromised, it cannot be canceled nor revoked, unlike a password. Therefore a question like *What can I do if my biometric data has been stolen or misused?* requires urgent attention, not only to reassure users with regards to privacy, but also to prevent abuses and improve accuracy.

Over the last decade, a new innovative research field has emerged, called cancelable biometrics, aimed at finding algorithms dedicated to template protection.[1–7] The resulting systems must fulfill the following important properties:[8] i) non-invertibility, which refers to the difficulty in recovering the original biometric given the secure template; ii) revocability, which refers to the ability of

generating a new template from the same biometric trait and iii) diversity, which is the difficulty in guessing one secure template given another secure template generated from the same biometric data.[9,10] Many template protection schemes have been applied on the fingerprint. For a fair review, the readers can refer to.[11–13] Until now, the available template protection schemes are not yet mature for large-scale deployment. They do not meet properties mentioned above while keeping a high accuracy performance. In fact, the process of protecting fingerprint template is quite complex. Several points have to be raised before designing a compatible solution: Which biometric feature constitutes the template? How the biometric template must be presented? How is the final representation of the protected template? How the template can be re-issued? How alignement has been handled? Is a pre-alignment needed? The objective of this paper is to deal with the biometric template representation before the transformation. This step related to the feature extractor is very important and constitutes a key question. In table 1, we present some relevant fingerprint protection schemes, whose "Feature", "Original representation", "Approach" and "Final representation" are compared.

We note from table 1 that, whatever the type of the initial representation, most of the final representation is in a binary vector form. This facilitates revocability, usually accomplished by a simple permutation of the generated vector (in this case, the permutation vector is randomly generated from a user key). One can also notice that the vector representation offers the advantage of reducing the comparison process to a simple distance operation such as a logical XOR. Most of the complexity of the calculation is related to the feature extraction and the transformation process. While the other transformations are only specific to minutiae template, the last three approaches of table 1 are more general and can be applied to all biometric characteristics. More precisely,

because of its revocability aspect and its ability to guarantee the unlinkability property, we focus in this paper on the BioHashing algorithm.[14,15] The BioHashing method consists in generating a unitary biocode from two data: the biometric one (for example face, palmprint, fingerprint characteristics) and a random number which needs to be stored (for example on a USB key, or more generally on a token), called tokenized random number. This principle is illustrated in Figure 2. First, a biometric template (called FingerCode for the fingerprint characteristic) is extracted from the raw image. The tokenized random number is generally mixed with the biometric template to obtain the binary output called BioCode.

The BioHashing process is rather simple. It consists of projecting the FingerCode on an orthonormal basis defined by the seed value. The dimension of the projection space is at least equal to the FingerCode size. The second step is a quantization step of the projected FingerCode. This transformation is invertible and is a very low time computation process. Figure 3 presents in detail the BioHashing process.

However, as specified in table 1, if we select the BioHashing (as well as the fuzzy commitment or the shielding function) as the protection algorithm target, we have to adapt the classical fingerprint feature representation to be combined with such template protection techniques. These schemes add two main constraints: 1) They require a fixed-length feature vector, which must be ordered, as input. (2) When combining biometric systems with template protection schemes, the biometric features will be compared in a protected (or transformed) domain. Therefore, applying template protection schemes also requires an alignment-free feature representation.

Some researchers are interested in this issue. In,[16] Bringer and Despiegel focus on fuzzy commitment as target protection algorithm. They highlight the difficulty to directly use minutiae with such construction and emphasize the need of binary vector representation for fingerprints. They propose a vector fingerprint representation from minutiae vicinities based on the use of a global representative dataset of vicinities with a fixed size. Xu *et al.*[17] focus on the same protection algorithm and propose a construction of a feature vector via the spectral representation of the minutiae set. In our work, we focus on BioHashing transform. Our objective is to define a FingerCode that must keep a fixed size for different fingerprints. Therefore, minutiae cannot be used directly within this context. We propose the use of texture features to deal with this problem as ridges of a fingerprint look like a texture. In this paper, we address the problem of choosing the most appropriate texture feature within this task. The required properties are the following:

- FingerCode must be discriminant, i.e. it should be able to provide a correct recognition without any transformation and used within the BioHashing context,
- The size of the FingerCode must be as high as possible as the BioCode can be considered as cryptographic key.
- The redundancy of information in the FingerCode must be as low as possible to guarantee the difficulty for an attacker to guess the BioCode.

Other benefits can be derived from a texture-based feature, namely:

- Given a minutiae set, it is possible to reconstruct the original image and to create a fake fingerprint that can fool the system, as detailed in.[18] This is not the case with texture based features, that are more secure.

- The multibiometric fusion is facilitated by the use of such features.
- The integration of minutiae is possible, since we can look for local comparisons between minutiae by describing each one with the texture feature of its neighborhood region.[19]

The paper is organized as follows. In section 2, we present the state of the art on texture analysis in order to identify among the most recent texture features the ones that can be applied on fingerprints. Section 3 is dedicated to a comparative study of selected texture features used within the context of privacy compliant fingerprint recognition. Experimental results are given in section 4, aim at comparing the ability of each texture feature to provide the best recognition efficiency using the BioHashing algorithm. We conclude in section 5 the study and give some perspectives.

2 Texture Analysis

Texture analysis provides an important cue as it conveys physical information about the characteristics of objects and surfaces. Several intuitive properties are often associated with texture such as roughness, coarseness, contrast, variation, regularity, randomness, element, shape and arrangement. Texture analysis has been studied for decades in computer vision and image processing field to address several problems such as: scene segmentation, object recognition, object tracking, content-based image retrieval, material inspection, visual effects, etc., in a variety of domains and applications. Many types of texture features have been proposed in the literature.[20] Among the successful techniques, one can refer to the frequential methods such as Gabor filter and local pattern based methods. These techniques have been extensively used in image processing and computer vision showing useful properties for recognition and classification. We address in this paper the use of texture features for fingerprint recognition.[21]

2.1 Texture features

Here, we consider fingerprint as a particular texture upon which we can extract some properties that can be used to enforce the fingerprint description. For that purpose, we list several recent techniques, in order to show their reliability in terms of biometric recognition.

- **Gray Level Cooccurrence Matrix (GLCM)**:[22] is one of the first approach for texture analysis, it uses a joint gray level histogram; where several statistics are extracted from this histogram (i.e moments, contrast, correlation, entropy, etc.)

- **Gabor filters (GABOR)**:[23] the general form results from the multiplication of Gaussian envelope and complex sinusoid functions. The Gabor kernel is band-pass filter with tuneable center, frequency, orientation and bandwidth. Each configuration of these parameters produces a response image. The combination of different values of the parameters yields several responses which can be considered as GABOR features.

- **Local Binary Pattern**:[24] the computation of LBP descriptor is performed in two steps. In the first one, elementary structures called binary patterns, are extracted in the local regions. In the second step, binary patterns are encoded with a histogram which represents their spatial distribution. The basic LBP technique uses a local thresholding over a 3×3 neighborhood associated to each pixel. The central pixel intensity is chosen as the threshold value. A value of 1 is assigned to each neighboring pixels whose intensity is above or equal to the threshold, and 0 for others. The resulting pattern is captured as an 8-bit binary number representing one of 256 distinct known patterns. Then, the histogram is computed for the transformed image and considered as a texture descriptor.

- **Median Binary Pattern (MBP)**:[25] is similar to the LBP but uses the median as the local threshold instead of using the center pixel value. MBP is determined by mapping from the intensity space to a localized binary pattern by thresholding the pixels against their median value within a neighborhood. The histogram of the MBPs is used to measure the distribution of these patterns over the image and forms the texture descriptor.

- **Patch based (PLBP)**:[26] this method uses small regions (patches) around each pixel belonging to a circular neighborhood. The binary code is produced by comparing the values of these patches to produce a single bit value in the code assigned to each pixel. Like LBP, the information is represented by histogram of binary strings. This technique has been proposed with two versions, three patches (TPLBP) and four patches (FPLBP).

- **Local Relational String (LRS)**:[27] is an operator that uses symbolic representation of the local neighborhood. For each pixel, a local string is extracted by comparing the center pixel to its neighbors. The difference between two pixels is modeled by the comparison operator $(<, =, >)$. 4-connected pixels schemes have been used, shaping a string of symbols that represent the relationship between the neighborhood pixels. Since there are three symbols and 4-connected pixels, the maximum number of possible strings is $3^4 = 81$. The LRS descriptor encodes the frequency of the local symbolic strings using the histogram.

- **LBP Histogram Fourier Features (LBPFT)**:[28] uses the phase shift property of Discrete Fourier Transform (DFT) to make the LBP descriptor invariant to the rotation. The LBPFT computes, first, the regular non-invariant LBP histogram, then the DFT is applied to this histogram to handle the global effect of the rotation. As the rotation affects only the phase in the Fourier space, the descriptor contains merely the Fourier magnitude spectrum values of

uniform patterns. The non uniform patterns frequency is also concatenated to the descriptor as an additional feature.

- **Completed LBP (CLBP)**:[29] uses three operators, the first one consists in the signs of local differences between the center pixel and its neighbors, this turns into the original LBP scheme. The second operator uses the magnitude of these differences to encode binary number by thresholding over the local magnitudes, where the threshold value is set to the global mean of all magnitudes in the image. Finally, the third operator produces the local binary numbers by using a global threshold defined as the average of gray levels in the whole image. A joint histogram is used to encode the binary patterns obtained from the three operators.

We show in the next section how these texture features can be used for cancelable biometrics.

2.2 BioCode generation

Texture features are good candidates to compute the FingerCode to be used with the BioHashing algorithm. In general, fingerprint recognition requires many pre-processing over the images such as contrast enhancement, binarization and skeletonization. These pre-processing permit to better detect minutiae. In this work, we identified four scenarios for the computation of the FingerCode with texture features:

- On the original image containing the fingerprint;
- On the binarized image: We used the Ostu's method[30] to define the binarization threshold;
- On a region of interest (ROI) of the original image: This ROI embeds the ridges of the fingerprint. The ROI is determined after applying different operations (normalization

+ erosion + binarization);

– On the ROI of the binarized image: We applied the same method for the binarization than used previously.

Figure 4 shows an illustration of the different computation scenarios on a fingerprint.

We intend to compare different recent texture features described previously considering these scenarios for the BioHashing algorithm.

3 Experimental results

In this section, we present the experimental results we obtained considering different texture features and the benchmark databases. We first define the experimental protocol we used.

3.1 Experimental protocol

For the protocol, we need to define multiple aspects: 1) Texture features used in the comparative study, 2) Fingerprint benchmark databases, 3) Comparison algorithm between two fingerprints and 4) Performance evaluation methodology.

3.1.1 Texture features

The comparative study considers 11 different features from the 8 texture features described in section 2.1 with different parameters. Indeed, by using different stages and orientation in the computation of Gabor features, we can obtain different sizes of the description. In this

paper, we considered 4 sizes (64, 128, 256, 512). All these features have been computed for the four scenarios described in section 2.2.

3.1.2 Benchmark databases

In this study, we used three databases, each one is composed of 800 images from 100 individuals with 8 samples from each user:

- FVC2002 benchmark database DB2: the image resolution is 296×560 pixels with an optical sensor "FX2000" by Biometrika ;
- FVC2004 benchmark database DB1: the image resolution is 640×480 pixels with an optical Sensor "V300" by CrossMatch ;
- FVC2004 benchmark database DB3: the image resolution is 300×480 pixels with a thermal sweeping Sensor "FingerChip FCD4B14CB" by Atmel.

Figure 5 presents one image from each database. We can observe that fingerprints are quite different and representative of what we could process for fingerprint recognition.

3.1.3 Comparison algorithm

Given the biometric reference of an individual, we realize a comparison trial. We have performed two tests:

- Recognition efficiency using FingerCodes: we quantify the accuracy of the texture features (without applying the BioHashing algorithm). We use the Minkowski distance as a comparison metric;

– Recognition efficiency using BioCodes: in this case, the comparison metric is the Hamming distance on BioCodes (after transformation of the FingerCode with the BioHashing algorithm).

The comparison methods are far from being the best solution compared to the state of the art, but we are mainly interested in a relative comparison of the different texture features.

3.1.4 Performance evaluation

We need to define some evaluation metrics in order to compare texture features when used on fingerprints before and after transformation with the BioHashing algorithm.

First, we consider the size of the texture feature as it is related to the size of the BioCode. The FingerCode size must be as high as possible as the size of the BioCode is smaller. Second, we intend to estimate the statistical redundancy of texture features. We use the Principal Component Analysis (PCA) method[31] consisting in projecting data on an orthonormal basis by suppressing redundant information. This is a well known tool in pattern recognition. In this study, we use 99.9% as threshold and we measure the dimension after reduction. The higher it is, the better the texture feature is (less redundant information).

During the experiments, we use a single enrollment process meaning that one sample is used to generate the biometric reference of an individual and other samples for testing the accuracy of the recognition system. As we have 8 samples for the 100 users of each database, We can use $7 \times 100 = 700$ comparison scores (called intra-class) with samples of the same

user and $7 \times 99 \times 100 = 69300$ comparison scores (called inter-class) with other users. As the number of intra-class and inter-class scores are unbalanced, we apply an iterative process for performance evaluation. A fixed number N of scores are randomly selected among the 700 intra-class ones and the 69300 inter-class ones. We compute the considered evaluation criterion C for each random selection of the scores. By averaging its value for each random selection, we obtain a more precise estimate of the criterion. A confidence interval at level 95% can be computed for the evaluation criterion C to be in $[E[C] \pm \epsilon]$ where $E[C]$ is the average value of the criterion and ϵ is computed as following:

$$\epsilon = 1.96 \times \frac{\sigma}{\sqrt{k}} \tag{1}$$

where k is the number of random selections of intra-class and inter-class scores and σ is the standard deviation of the criterion C for each random selection. A low value of ϵ indicates a good approximation of the evaluation criterion. In our experiments, we set $N = 500$ and $k = 100$ ($N \in [0\ 700]$ in our case and k can be as high as possible but has an impact on the computation time).

Based on the result of the comparison algorithm, we have to decide if the identity of the user is verified or not (authentication scheme). We use a threshold for that. As we know the expected result, we can compute two metrics:

- False non-match rate (FNMR): proportion of the completed biometric mated comparison trials that result in a false non-match;

– False match rate (FMR): proportion of the completed biometric non-mated comparison trials that result in a false match.

These two metrics vary depending on the value of the threshold. A DET (Detection Error Tradeoff) curve represents the evolution of the couple (FMR,FNMR) for different values of the threshold. This curve describes the global behavior of a biometric system. In our case, it gives some indications on the accuracy of texture features for different values of the decision threshold. The Equal Error Rate (EER) corresponds to the point on the DET curve where the FMR value equals the FNMR one. The performance of biometric systems is often characterized by the EER value (even if it is not an interesting operational functioning point).

3.2 Results

Table 2 gives the size of the representation provided by each descriptor on the FVC2002 DB2. Statistical redundancy is estimated by using the principal component analysis method (PCA). The dimension of the space while keeping 99.9% of the information from descriptors computed on fingerprints data (800 samples) allows us to estimate the redundancy. As for example, after applying the PCA, the size of the LBP features (initial dimension 256) is reduced to 38, so the computed redundancy is computed as $(256 - 38)/256 = 85\%$. As one can see from this table, nearly all descriptors have a high redundancy. Gabor parameters have a lower redundancy compared to others and they have the advantage to be adjustable in term of size. Considering only these aspects, Gabor512 is a good choice for representing fingerprints (results are similar on other databases).

Table 3 gives the obtained EER value for each texture descriptor and computation scenarios when using the FingerCode on the FVC2002 DB2 database as illustration. If we only consider the state of the art, results are poor but we are interested here by the relative comparison of texture features. As we can see, the confidence interval is very small (near $2.10^{-}3$) meaning we have a good estimate of the EER value for each case. In the following, we do not show anymore the confidence interval associated to an evaluation criterion for clarity reasons even it still remains very small. For most features, better results are obtained on binary images. Using a region of interest sometimes decreases the performance of the recognition for some features like LBP, TPLBP, LRS or MBP. Gabor parameters permit to obtain the best results. Using the binary image instead of the original one permits a huge gain of 9% of the EER value. The best computation scenario is the region of interest on the binary image.

Figures 6, 7, 8 present the DET curves of the different biometric systems when using the different texture features as FingerCode (i.e. without any protection) for the different scenarios for each database. We can see clearly that the relative performance of texture features are similar for each database. We can also see the relative difficulty of each benchmark database.

Table 4 presents for each database the most efficient texture feature and the associated EER values. Even if the obtained efficiency is different for each database (the second one is more complex as for example), the best feature is always Gabor whatever the used computation scenario. As a conclusion on fingerprint recognition with texture features, we can say that Gabor is the most appropriate choice.

In the following, we study the efficiency of texture feature for fingerprint recognition using the BioHashing algorithm (i.e. using BioCodes). Figures 9, 10, 11 present the DET curves of the different biometric systems when using the different texture features after applying the BioHashing algorithm (i.e. BioCodes) for the different computation scenarios for each database. We can notice that many DET curves are not or partially present, this is due to a perfect performance (i.e. with an area under the curve near 0%).

Table 5 presents for the FVC2002 DB2 database (that can be considered of medium complexity) the EER values of the different texture features when used for computing the BioCodes with the algorithm given in Figure 3. We can see clearly that the EER values are very low ; it is well known that the BioHashing improves the performance (because the seed value defining the projection can be seen as an *a priori* information). We can notice that 5 features among LBP, CLBP, LBPFT, GABOR512 and MBP permit to obtain a perfect recognition for all computation scenarios. Others have very low error rates for few scenarios.

It is possible to compare BioCodes from different databases. We made another experiment to illustrate this point. We concatenated the three databases to obtain a chimeric one composed of 300 individuals with 8 samples. We computed the EER value when we use the BioCode from the Gabor features (256 bits) on the ROI binary image. It equals 4 10≏4 showing the reliability of obtained results.

The previous case assumes the attacker does not know the secret that can be seen as one of the two-factor authentication scheme. We made another experiment consisting in simulating the stolen token attack. If this case, we suppose the attacker knows the secret of the user he/she wants to impersonate. In this case, impostors scores are computed from BioCodes generated with impostors samples and the secret of the individual to impersonate. We estimate the EER value using the BioCode. Table 6 presents results on the FVC2002 DB2 database for the four computation scenarios. We obtained similar results to Table 3 and even slightly better (this is normal). Conclusions on efficiency are identical concerning the robustness to attack (especially considering the stolen token one that is the most problematic for the BioHashing algorithm). Gabor features present the best behavior on fingerprints.

Table 7 presents a comparison of the studied texture features with some recent papers (most of them are using the BioHashing algorithm or a variant) on the FVC2002 DB2 database. Other papers improve the robustness to the stolen token attack that is not our purpose in this paper. Results obtained by using the best features are not very far from them and they are computed on the global images or on a ROI.

Figure 12 shows the comparison of the performance when using GABOR Features of size 256 and 512 as Fingercode and BioCode in the context of the stolen token attack. This shows a slight improvement even in the context of an attack.

4 Conclusion and perspectives

In this paper, we studied different texture features from the state of the art for fingerprint recognition. We are particularly interested in the cancelable biometrics approach consisting in projecting the texture feature in a secret place. We tested also four computation scenarios (original image, binary image, region of interest on the original image, region of interest on the binary image). We considered 4 criteria: dimension of the representation of the texture feature, redundancy, performance before and after transformation with the BioHashing algorithm. For all these criteria, Gabor features reveal themselves as the best candidate for this issue. For a better performance and robustness, a dimension at least equal to 256 or 512 should be chosen.

Perspectives of this study concern the study of the robustness of the BioHashing algorithm for the different texture features we studied in this paper. We intend to quantify to what extent some attacks are possible for the different texture features.

References

1 A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *ACM conference on Computer and communication security*, 28–36 (1999).

2 A. Nagar, K. Nandakumar, and A. K. Jain, "Biometric template transformation: A security analysis," *Proceedings of SPIE, Electronic Imaging, Media Forensics and Security XII* (2010).

3 L. Nanni, L. S. Brahnam, and A. Lumini, "Biohashing applied to orientation-based

minutia descriptor for secure fingerprint authentication system," *Electronics letters* **47**(15), 851–853 (2011).

4 A. Jin, A. Teoh, T. Ong, and C. Tee, "Fingerprint template protection with minutiae-based bit-string for security and privacy preserving," *Expert systems with applications* **39**, 6157–6167 (2012).

5 S. Wang and J. Hu, "Alignment-free cancellable fingerprint template design: a densely infinite-to-one mapping (ditom) approach," *Pattern recognition* **45**, 4129–4137 (2012).

6 B. Topcu, H. Erdogan, C. Karabat, and B. Yanikoglu, "Biohashing with fingerprint spectral minutiae," in *IEEE International Conference of the Biometrics Special Interest Group (BIOSIG)*, (2013).

7 S. Brahnam, L. Nanni, and A. Lumini, "A secure multimatcher system for fingerprint verification," in *Proceedings for the Northeast Region Decision Sciences Institute*, (2013).

8 J. Breebaart, C. Busch, J. Grave, and E. Kindt, "A reference architecture for biometric template protection based on pseudo identities," in *IEEE I nternational Conference BIOSIG*, (2008).

9 W. Cheirer and T. Boult, "Cracking fuzzy vaults and biometric encryption," in *Proceedings of Biometrics Symposium*, 1–6 (2007).

10 X. Zhou, S. Wolthusen, C. Busch, and A. Kuijper, "Feature correlation attack on biometric privacy protection schemes," in *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 1061–1065 (2009).

11 A. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," in *EURASIP J. Adv. Signal Process 2008*, (2008).

12 C. Rathgeb and A. UHL, "A survey on biometric cryptosystems and cancelable biometric," *EURASIP Journal on Information Security* **3**, 0–0 (2011).

13 M. Ferrara, D. Maltoni, and R. Cappelli, "Noninvertible minutia cylinder-code representation," *IEEE Transactions on Information Forensics and Security* **7**(6), 1727–1737 (2012).

14 A. Teoh, D. Ngo, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern recognition* **40** (2004).

15 R. Belguechi, C. Rosenberger, and S. Aoudia, "Biohashing for securing minutiae template," in *Proceedings of the 20th International Conference on Pattern Recognition*, 1168–1171, (Washington, DC, USA) (2010).

16 J.Bringer and V.Despiegel, "Binary feature vector fingerprint representation from minutiae vicinities," in *4th IEEE International conference on biometrics compendium*, (2010).

17 H.Xu and N. Veldhuis, "Fingerprint verification using spectral minutiae representations," *IEEE Trans on information forensics and security* **4**, 0–0 (2009).

18 R. Cappelli, D. Maio, A. Lumini, and D. Maltoni, "Fingerprint image reconstruction from standard templates," *IEEE Transactions On Pattern Analysis And Machine Intelligence* **29**, 1489–1503 (2007).

19 R. Belguechi, E. Cherrier, C. Rosenberger, and S. Ait-Aoudia, "Operational bio-hash to preserve privacy of fingerprint minutiae templates," *IET biometrics* **2**(2), 76–84 (2013).

20 M. Petrou and P. G. Sevilla, *Image processing: dealing with texture*, Wiley (2006).

21 C. Soutar, A. Stoianov, and G. Tomko, "Hybrid optical-digital method and apparatus for fingerprint verification," (1998).

22 R. Haralick, K. Shanmungan, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems Man and Cybernetics* **6** (1979).

23 D. Gabor, "Theory of communications," *J. Inst. Elect. Eng* **93**, 429457 (1946).

24 T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns.," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(7), 971–987 (2002).

25 A. Hafiane, G. Seetharaman, K. Palaniappan, and B. Zavidovique, "Rotationally invariant hashing of median binary patterns for texture classification.," in *ICIAR*, *Lect Notes Comput Sci* **5112/2008**, 619–629, Springer (2008).

26 L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Real-Life Images workshop at the European Conference on Computer Vision (ECCV)*, (2008).

27 A. Hafiane and B. Zavidovique, "Local relational string and mutual matching for image retrieval," *Inf. Process. Manage.* **44**(3), 1201–1213 (2008).

28 T. Ahonen, J. Matas, C. He, and M. Pietikäinen, "Rotation invariant image description with local binary pattern histogram fourier features," in *Proceedings of the 16th SCIA*, 61–70 (2009).

29 Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *Trans. Img. Proc.* **19**, 1657–1663 (2010).

30 N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems Man and Cybernetics* **9**, 62–66 (1979).

31 K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine* **11**(2), 559572 (1901).

32 N. Ratha, S. Chikkerur, J. Connell, and R. Bolle, "Generating cancelable fingerprint templates," *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(4), 561–572 (2007).

33 B. Yang and C. Busch, "Parameterized geometric alignment for minutiae-based fingerprint template protection," in *3rd IEEE BTAS*, 340–345 (2009).

34 A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *6th ACM Conf. Computer and Comm. Security*, 28–36 (1999).

35 J. Linnartz and P. Tuyls, "New shielding functions to enhance privacy and prevent misuse of biometric templates," in *4th International Conference on Audio- and Video-Based Biometric Person Authentication*, (2003).

Adel Hafiane is assistant professor at INSA CVL (National school of engineering). He works in the PRISME research laboratory. His research activities concern texture analysis and computer vision.

Estelle Cherrier is assistant professor at ENSICAEN (National school of engineering). She works in the GREYC research laboratory within the "E-payment & Biometrics" research unit. Het research activities concern biometrics.

Christophe Rosenberger is full professor at ENSICAEN (France). He obtained his Phd from the university of Rennes I in 1999. Since 2007, He belongs to the GREYC laboratory in the E-payment & Biometrics research unit. His research interests concern biometric systems and computer security.

List of Figures

List of Tables

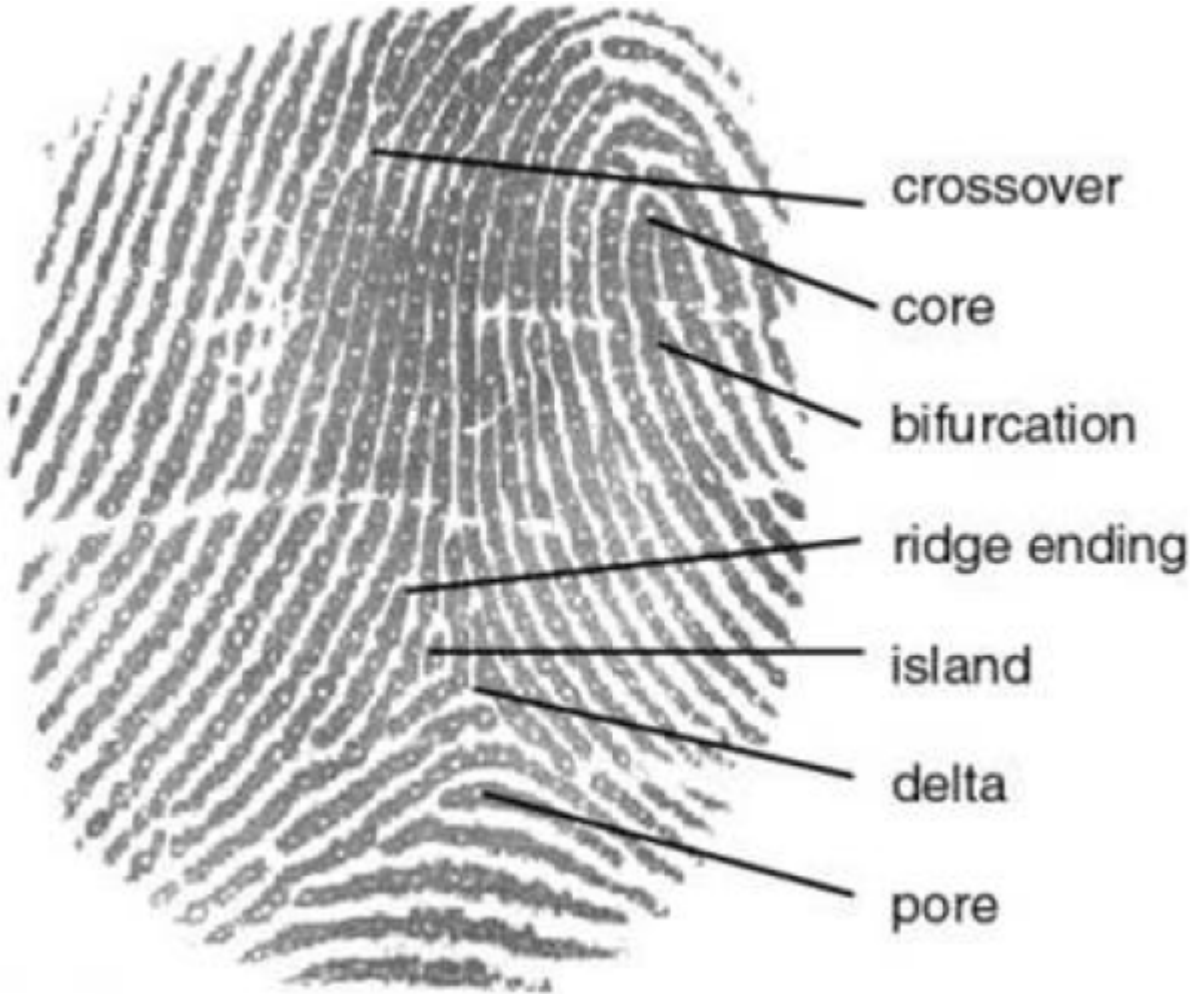


Fig 1 Different representations of a fingerprint: 1) ridge, 2) minutiae and 3) pores (source International Biometric Group)

Technique	Feature	Original rep.	Approach	Final rep.
Transformation "key"[32]	Minutiae	Point set	Random projection and binarization	Binary vector
Yang & Busch[33]	Minutiae	Point set	minutiae vicinities	Binary vector
Bringer & Despiegel[16]	Minutiae	Point set	minutiae vicinities	Binary vector
Nagar *et al.*[2]	Minutiae	Point set	Local aggregates	Binary vector
Jin *et al.*[4]	Minutiae	Point set	Transformation minutiae points	Minutiae set
Wang and Hu[5]	Minutiae	Point set	Pair-minutiae quantization and mapping	Binary vector
Ferrara *et al.*[13]	Minutiae	Point set	Cylinder representation	Binary vector
Fuzzy commitment[34]	Fixed length vector	Binary vector	Error correction code scheme	Binary vector
Shielding function[35]	Image-based	Real vector	Secret extraction from common randomness	Binary vector
BioHashing[14]	Image-based	Real vector	Random projection and binarization	Binary vector

Table 1 Comparison of fingerprint template protection algorithms

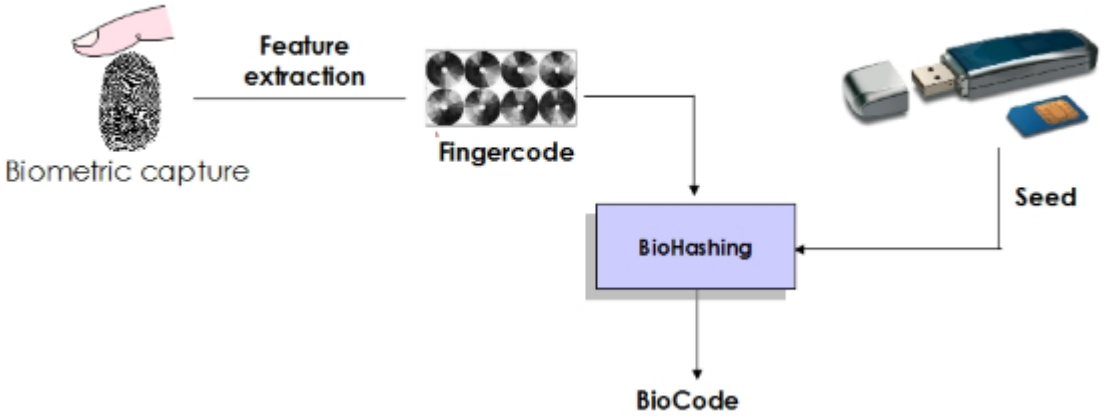


Fig 2 General principle of the BioCode generation

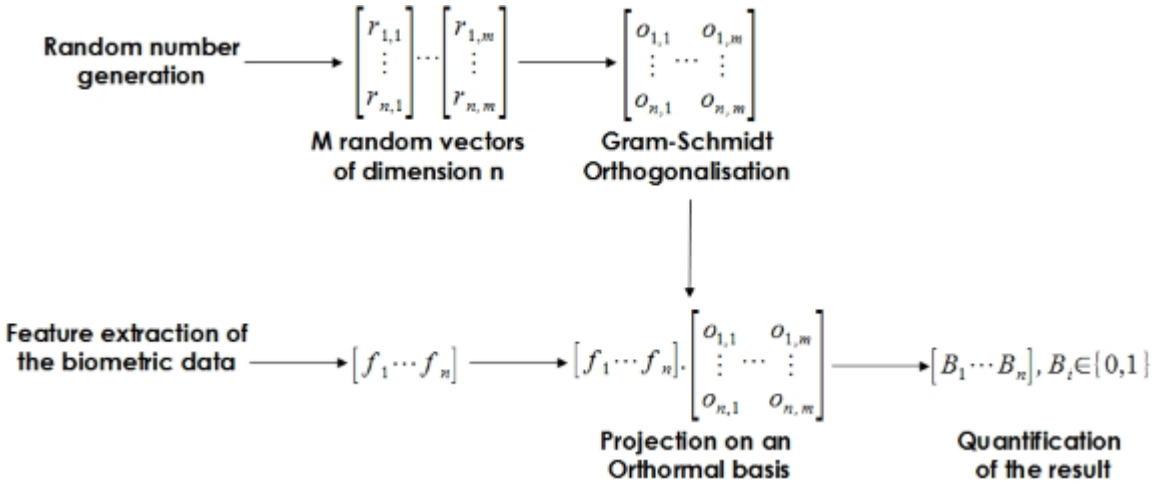
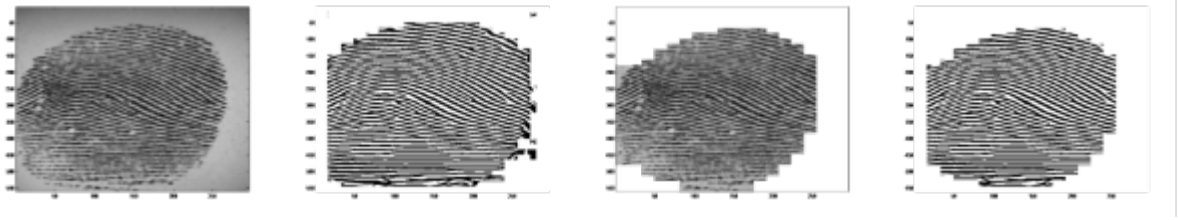


Fig 3 BioHashing description



$original\ image$ $binarized\ image$ $ROI\ original\ image$ $ROI\ binarized\ image$

Fig 4 Example of pre-processing used for the computation of the FingerCode with a texture feature.

(a) (b) (c)

Fig 5 One fingerprint example from each database: (a) FVC2002 DB2, (b) FVC2004 DB1, (c) FVC2004 DB3

Texture features	Dimension	Redundancy
LBP	256	85%
TPLBP	7280	89%
CLBP	512	92%
GLCM	8	50%
LBPFT	152	75%
GABOR64	64	**9%**
GABOR128	128	23%
GABOR256	256	38%
GABOR512	512	67%
LRS	81	79%
MBP	256	88%

Table 2 Texture features: Dimension and redundancy

Features	Original image	Binary image	ROI original image	ROI binary image
LBP	32.0 $\pm$ 0.002	33.8 $\pm$ 0.0027	40.6 $\pm$ 0.003	35.6 $\pm$ 0.003
TPLBP	43.7 $\pm$ 0.0033	40.0 $\pm$ 0.0033	45.6 $\pm$ 0.0031	42.5 $\pm$ 0.003
CLBP	38.4 $\pm$ 0.0026	33.7 $\pm$ 0.0029	40.0 $\pm$ 0.0029	38.8 $\pm$ 0.0028
GLCM	46.3 $\pm$ 0.0032	38.1 $\pm$ 0.0027	43.3 $\pm$ 0.0035	38.7 $\pm$ 0.0027
LBPFT	37.9 $\pm$ 0.003	36.1 $\pm$ 0.0029	41.2 $\pm$ 0.0034	37.0 $\pm$ 0.0029
GABOR64	40.5 $\pm$ 0.0033	41.2 $\pm$ 0.0031	43.1 $\pm$ 0.0031	40.0 $\pm$ 0.0028
GABOR128	35 $\pm$ 0.0027	28.0 $\pm$ 0.0027	35.2 $\pm$ 0.0028	27.2 $\pm$ 0.0029
GABOR256	**32.1 $\pm$ 0.0035**	**22.9 $\pm$ 0.0028**	**32.0 $\pm$ 0.003**	21.8 $\pm$ 0.0027
GABOR512	32.2 $\pm$ 0.0029	22.9 $\pm$ 0.0026	32.5 $\pm$ 0.0029	**21.6 $\pm$ 0.0023**
LRS	36.4 $\pm$ 0.0027	35.1 $\pm$ 0.003	39.8 $\pm$ 0.0031	35.1 $\pm$ 0.0028
MBP	32.1 $\pm$ 0.0026	33.8 $\pm$ 0.0031	40.0 $\pm$ 0.0027	35.5 $\pm$ 0.0032

Table 3 FingerCode efficiency for the different scenarios (EER in %) for the FVC2002 DB2 database and the associated confidence interval (at level 95%).

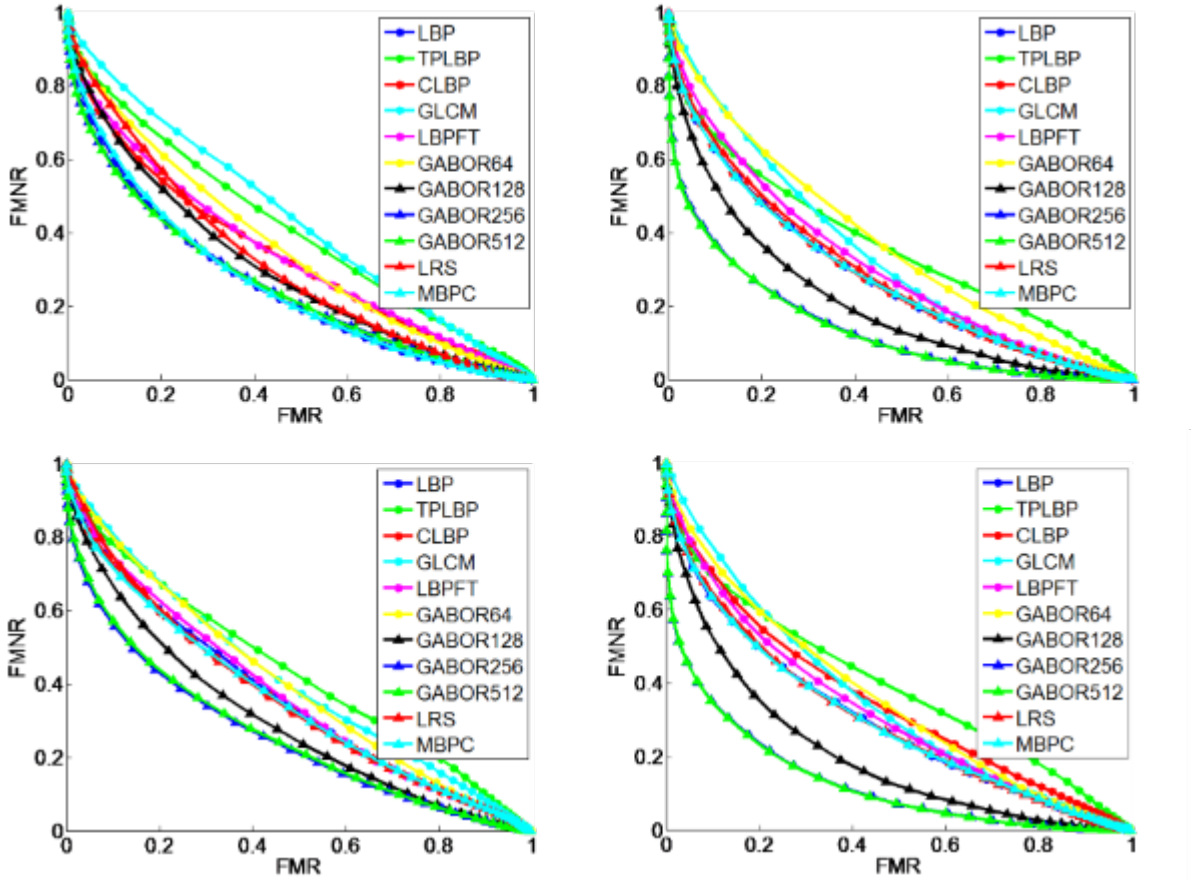


Fig 6 FingerCode DET curve (database 1): original image (left 1st row), region of interest (right 1st row), binary image (left 2nd row) and region of interest of the binary image (right 2nd row)

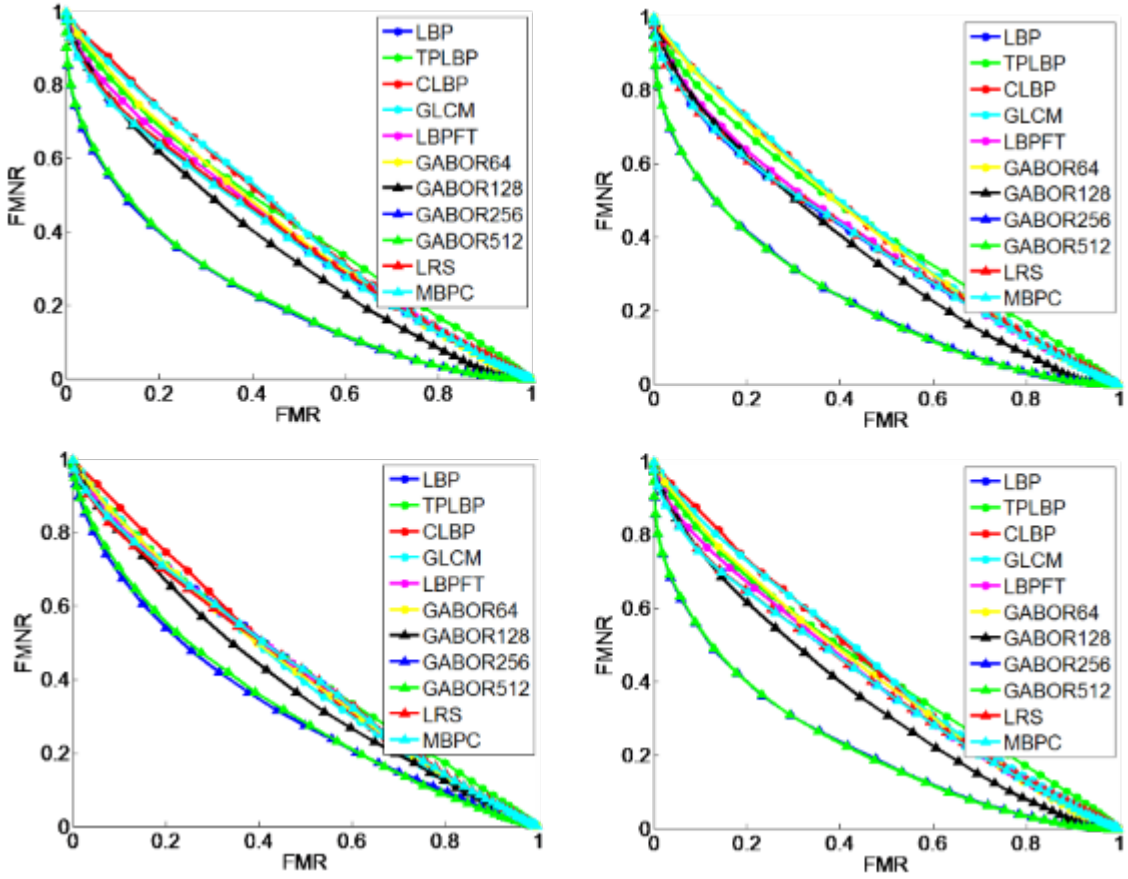


Fig 7 FingerCode DET curve (database 2): original image (left 1st row), region of interest (right 1st row), binary image (left 2nd row) and region of interest of the binary image (right 2nd row)

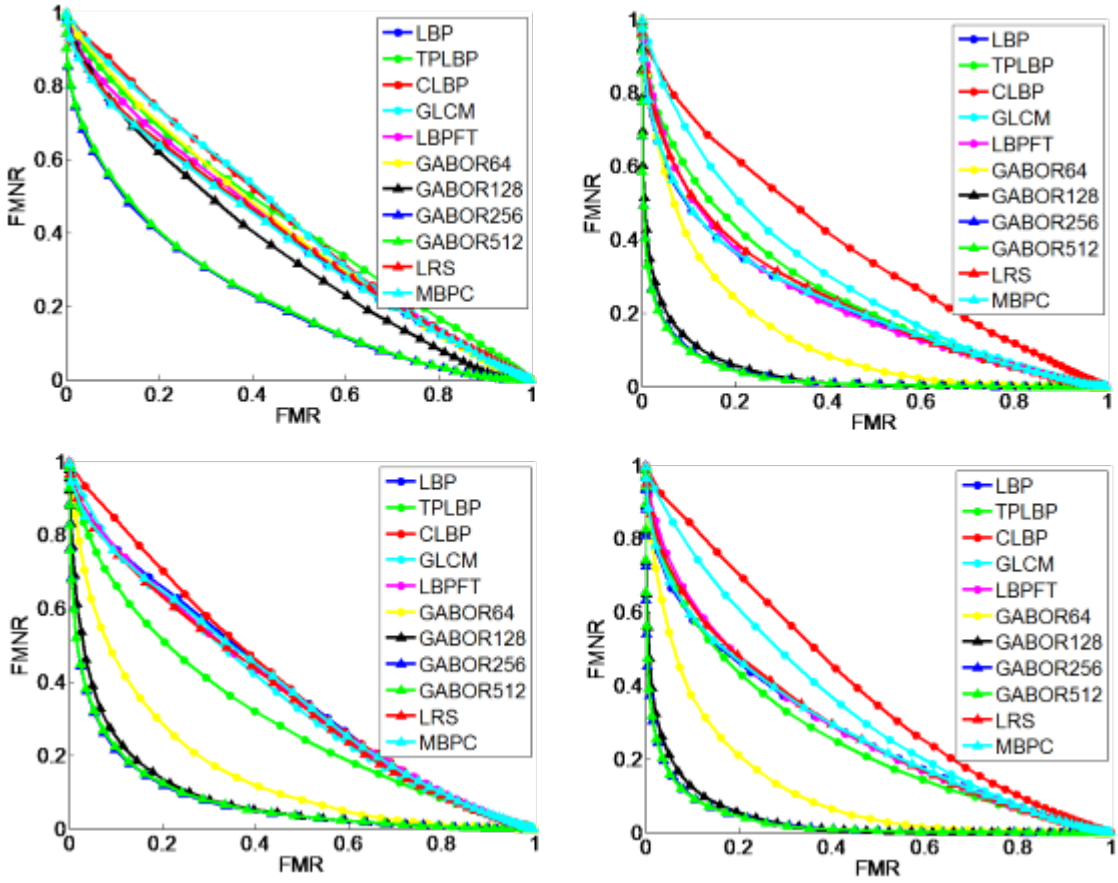


Fig 8 FingerCode DET curve (database 3): original image (left 1st row), region of interest (right 1st row), binary image (left 2nd row) and region of interest of the binary image (right 2nd row)

Databases	Original image	Binary image	ROI original image	ROI binary image
FVC2002 DB2	Gabor256	Gabor256	Gabor256	Gabor512
	32.1	22.7	32	**21.7**
FVC2004 DB1	Gabor256	Gabor512	Gabor256	Gabor256
	29.9	30.5	36.9	**30.1**
FVC2002 DB3	Gabor512	Gabor512	Gabor512	Gabor512
	9.5	9.9	15.1	**9.2**

Table 4 FingerCode efficiency for the three databases (EER in %).

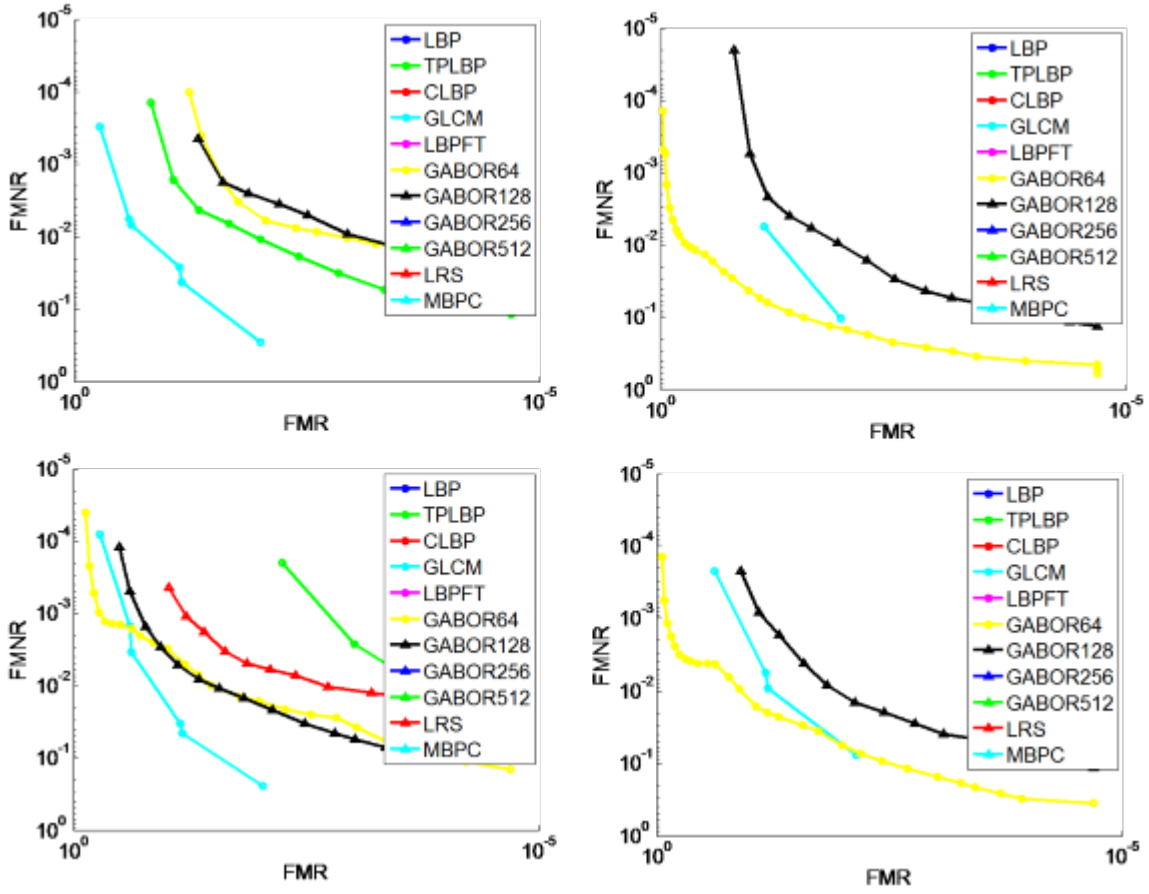


Fig 9 BioCode DET curve on database 1: original image (left 1st row), region of interest (right 1st row), binary image (left 2nd row) and region of interest of the binary image (right 2nd row)

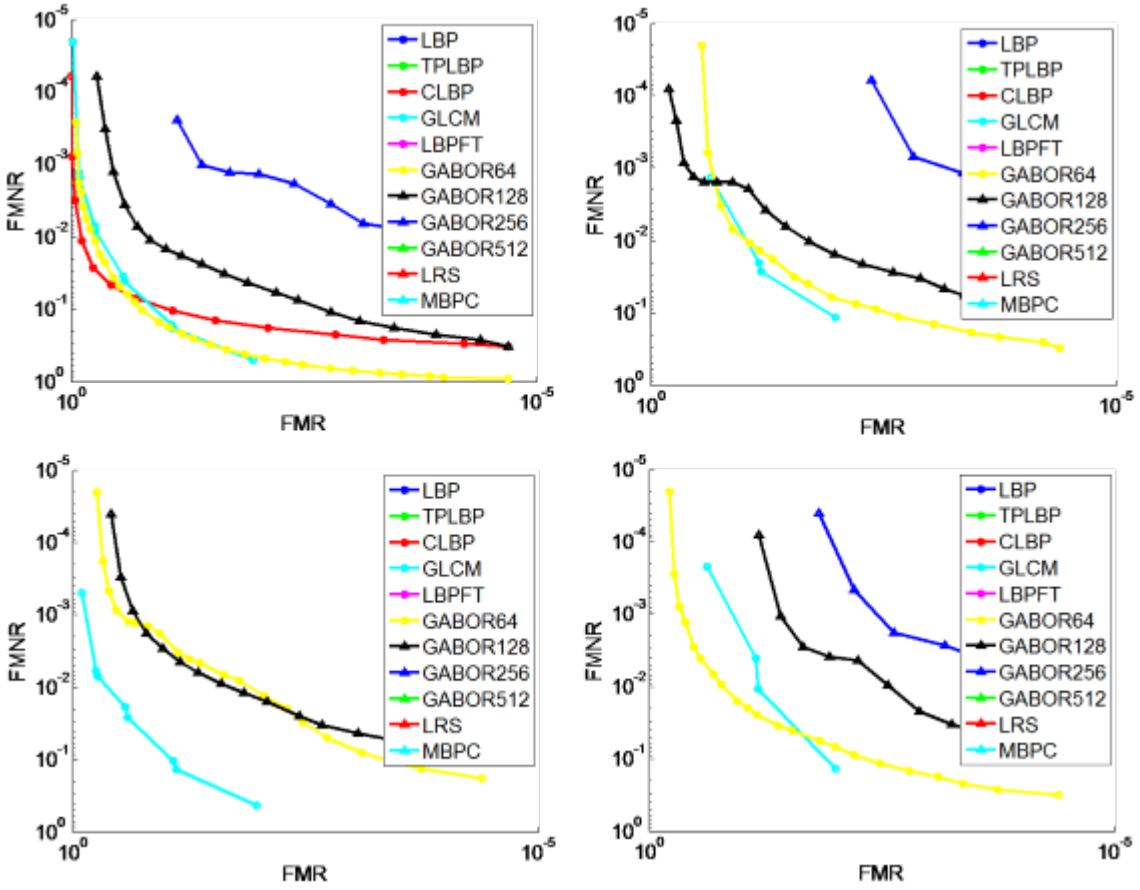


Fig 10 BioCode DET curve on database 2: original image (left 1st row), region of interest (right 1st row), binary image (left 2nd row) and region of interest of the binary image (right 2nd row)

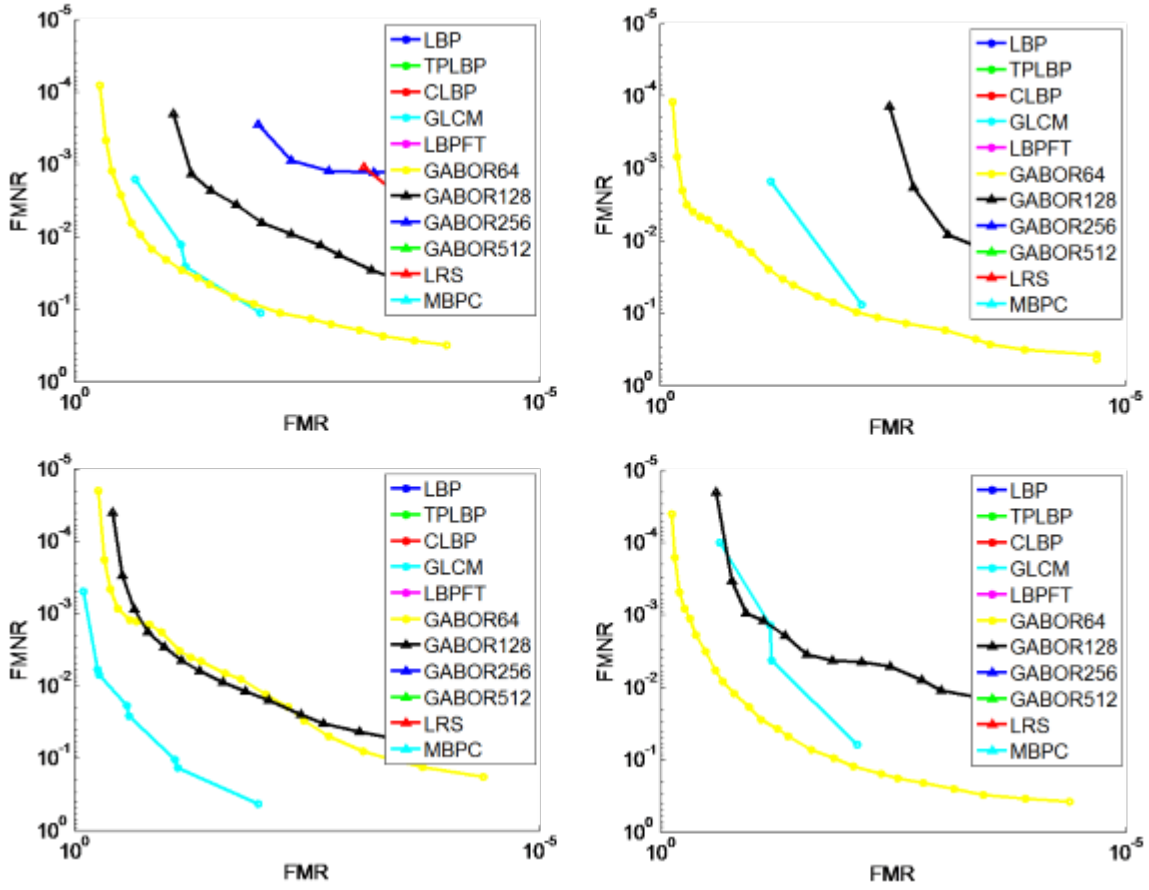


Fig 11 BioCode DET curve on database 3: original image (left 1st row), region of interest (right 1st row), binary image (left 2nd row) and region of interest of the binary image (right 2nd row)

Texture features	Original image	Binary image	ROI original image	ROI binary image
LBP	0	0	0	0
TPLBP	8.10^{-3}	0	9.10^{-3}	0
CLBP	0	0	0	0
GLCM	5.10^{-2}	4.10^{-2}	5.10^{-2}	3.10^{-2}
LBPFT	0	0	0	0
GABOR64	6.10^{-3}	6.10^{-2}	10^{-2}	2.10^{-2}
GABOR128	4.10^{-3}	9.10^{-3}	10^{-2}	10^{-2}
GABOR256	0	0	10^{-4}	0
GABOR512	0	0	0	0
LRS	3.10^{-4}	0	6.10^{-3}	0
MBP	0	0	0	0

Table 5 BioCode efficiency for the different scenarios (EER in %) for the FVC2002 DB2 database.

Texture features	Original image	Binary image	ROI original image	ROI binary image
LBP	28.2	42.8	35.5	41
TPLBP	42.6	42.3	44.8	44.8
CLBP	32.3	30	38.3	37.5
GLCM	50	48.6	50	48.5
LBPFT	39.9	37.9	42	37
GABOR64	41.3	39.6	45	35.1
GABOR128	28.7	28	29.5	25
GABOR256	**22.2**	**18.3**	**24**	18
GABOR512	25.1	18.3	26.2	**17**
LRS	34.7	37.6	37.1	36
MBP	30.7	39.6	35.1	37.7

Table 6 BioCode efficiency for the different scenarios (EER in %) for the FVC2002 DB2 database under the stolen token attack.

Methods	Best case (without attack)	Worst case (stolen token attack)
BioHashing with Gabor256	0	17
Topcu et al.[6]	0	13.1
Brahnam et al.[7]	0.45	5.68
Nanni et al.[3]	1.81	3.45

Table 7 Comparison of different methods from the state of the art on the FVC2002 DB2 database (EER in %).

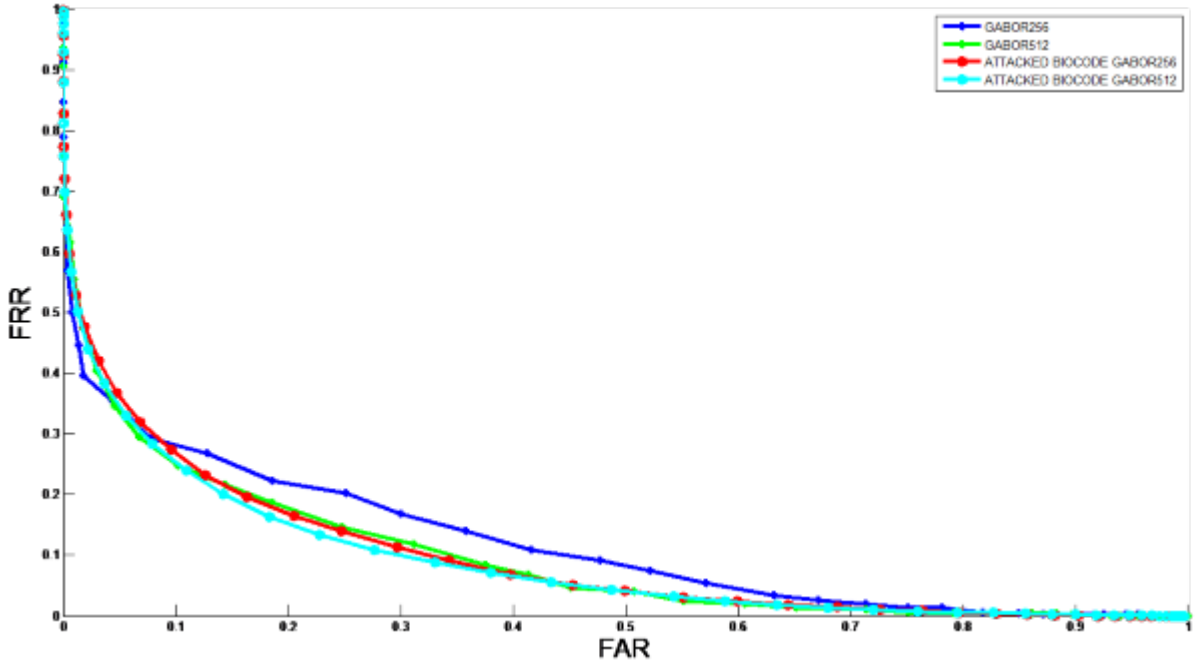


Fig 12 Comparison of ROC curves when using GABOR Features of size 256 and 512 as Fingercode and BioCode in the context of the stolen token attack.