

# Implementing FingerCode-Based Identity Matching in the Encrypted Domain

Tiziano Bianchi, Stefano Turchi, Alessandro Piva

Dipartimento di Elettronica e Telecomunicazioni,

Università di Firenze

Via S. Marta 3, I-50139, Firenze, Italy

(tiziano.bianchi, alessandro.piva) @unifi.it, turchi@lci.det.unifi.it

Ruggero Donida Labati, Vincenzo Piuri, Fabio Scotti

Department of Information Technologies,

Università degli Studi di Milano

Via Bramante 65, I-26013, Crema, Italy

(ruggero.donida, vincenzo.piuri, fabio.scotti) @unimi.it

**Abstract**—In this paper, we address the problem of FingerCode-based identity matching using encrypted templates. Instead of the classical approach of combining secure signal processing (SSP) tools to mimic the behavior of some well-known identity matching algorithm, we will investigate the possibility of using a SSP-friendly biometric implementation, i.e., an implementation based on SSP tools. We will propose two alternative strategies for reducing the size of the FingerCode templates, to make them compatible with existing SSP solutions. Experimental results show that feature size reduction has a very limited impact on the accuracy of the biometric system, demonstrating that encrypted domain identity matching can be implemented without sacrificing biometric performance.

**Index Terms**—Biometric-Based Identification, Fingercode, Secure Signal Processing, Signal Processing in the Encrypted Domain

## I. INTRODUCTION

Biometric data, such as fingerprints, irises, face images, are increasingly viewed as one of the most powerful form of identification in security applications. The reason for this success is that biometric traits are universal, unique and irreplaceable: every person has biometric traits, which are usually unique for each individual, and many of the physical features of the owner are assumed to remain constant or change little during the years. If this is a very desirable property in identifying people, on the other hand this also makes biometric traits invaluable to the owners. Their protection from unauthorized use is therefore critical to prevent identity theft and ensure public acceptance of the techniques that are based on them.

Secure signal processing (SSP), also referred to as signal processing in the encrypted domain, is a field of research that has gained considerable interest in recent years [1]. SSP techniques aim at processing signals in a secure or privacy-preserving manner: the paradigm is that the entity in charge of processing a signal should be able to do it without gathering any information about the signal itself. Therefore, SSP techniques seem particularly suited to biometric applications, since they will allow us to perform biometric matching without

disclosing any information regarding the involved biometric traits.

Although SSP techniques have been already applied to specific biometric problems (e.g., face recognition [2], [3]), there are still several limitations to their use in a broader set of biometric applications. On one hand, to achieve the best matching performance we must use specific algorithms for each different biometric trait. On the other hand, such algorithms usually require complex processing tasks that are difficult to implement via SSP. An example is fingerprint matching: in this case the best performance is obtained by minutiae-based algorithms; however, such algorithms require the solution of a point pattern matching problem, for which we have no efficient protocols in the encrypted domain.

In the field of SSP, very efficient solutions can be found when the matching problem can be modeled as a distance computation followed by a comparison with a threshold. Distances between encrypted vectors or between an encrypted vector and a plaintext one can be easily implemented by relying on homomorphic cryptosystems [4]–[6], whereas the “greater than” problem can be efficiently solved either by specific protocols exploiting the homomorphic properties [2], [7] or by using garbled circuits [3], [8]. Hence, a reasonable approach could be that of looking for biometric matching algorithms that can be implemented relying only on these simple building blocks.

For what concerns fingerprint matching, a natural candidate is the FingerCode-based approach [9]. FingerCodes are templates obtained by applying a bank of Gabor filters to a fingerprint image and computing the average absolute deviation of the result over a set of concentric radial sectors. Since the resulting templates are fixed size vectors, identity comparison is simply performed by comparing the Euclidean distance between FingerCodes with a threshold.

In this paper, we will address the problem of implementing a Fingercode-based matching algorithm in the encrypted domain. Our approach will not be that of investigating new cryptographic protocols, rather we will study how to adapt the original FingerCode algorithm so that it can be implemented relying on existing and efficient protocols. We will see that this adaptation requires some simplification of both the FingerCode algorithm and the FingerCode representation. Particular

The work described in this paper has been partially supported by the Italian Ministry of Education, Universities and Research (MIUR) under the project “Priv-Ware” (contract n. 2007JXH7ET). The research at the University of Milan was also supported in part by the EU within the 7FP project “PrimeLife” under grant agreement 216483.

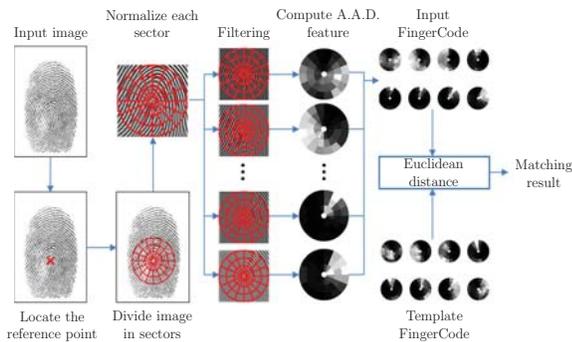


Fig. 1. Scheme of the biometric recognition method based on the template Fingercode.

attention will be devoted to the possibility of reducing the size of the FingerCode and the number of bits used for representing each FingerCode component without affecting the overall performance of the system. An interesting result is that FingerCode matching is still possible using templates with very few bits, which greatly reduces the complexity of an encrypted domain implementation.

The paper is organized as follows. In Section II, FingerCode-based identity comparison is briefly reviewed, while in Section III the proposed modeling of FingerCode distances is introduced, together with appropriate strategies for the reduction of the size of FingerCode features. The effects of feature size reduction are investigated through experimental results in Section IV. Finally, concluding remarks are given in Section V.

## II. FINGERCODE-BASED IDENTIFICATION

The computation of the biometric template in the plain domain is based on a FingerCode method. This method encompasses four main steps:

- 1) determine a reference point;
- 2) tessellate the region of interest (ROI) around the reference point;
- 3) filter the region of interest in eight different directions using a bank of Gabor filters;
- 4) compute the average absolute deviation from the mean of gray values in individual sectors in filtered images to define the feature vector or the Fingercode.

The obtained feature vector is composed by a numerical vector of double precision elements. For example, in [9], the size of this vector ranges from 640 to 896 elements, according to the used fingerprint dataset. Since the FingerCode method is not rotational invariant, during the enroll phase, 9 templates related to different rotations of the original image are computed. The match-score of two templates consists in the minimum Euclidean distance between the 9 enrolled templates and the live template (computed from the fingerprint image captured during the biometric recognition phase). Fig. 1 shows the schema of the method proposed in [9] and the input/output data of the described steps.

Experiments shown that the critical task of this method is the estimation of the reference point. This point must be unique for each image related to the same finger. An incorrect estimation of this point implies a different ROI evaluation, causing an increasing of the identification errors. We proposed different methods to reduce this problem (a discussion related to this argument is presented in [10]).

As shown in Table I, the principal parameters that we considered for the reduction of the number of features of the final template are:

- the number of Gabor filters applied to the image (N. Filters);
- the number of rings used for the tessellation of the ROI (N. Rings);
- the height expressed in pixel of the rings (H. Rings);
- the number of arcs used for the tessellation of the ROI (N. Arcs).

## III. FINGERCODE DISTANCE MODELING

To model the squared distances of genuines and impostors we rely on a simplified model based on the following observation: we can see the measured FingerCode as the composition of an “ideal” component and a noisy component, so that every genuine instance will share the ideal part (as they come from the same finger), while the error will change (as it depends on the image acquisition context). Let  $\tilde{x}$  be the ideal component and  $e$  be the error component. The measured FingerCode  $x$  can therefore be expressed as:

$$x = \tilde{x} + e$$

For the sake of simplicity, momentarily we consider the distances taken without the minimum selection as they were computed between the FingerCode to be authenticated and an enrollment set made of a single FingerCode. The distance  $D_0$  between genuines will have the form:

$$D_0 = \sum_i (e_1(i) - e_2(i))^2$$

while the distance  $D_1$  computed between impostors will have the form:

$$D_1 = \sum_i (\tilde{x}_1(i) - \tilde{x}_2(i) + e_1(i) - e_2(i))^2$$

Under the hypothesis that measurement errors are zero-mean i.i.d. Gaussian variables  $e \sim \mathcal{N}(0, \sigma_e^2)$ , FingerCodes are i.i.d. Gaussian variables ( $\mu_x \gg \sigma_x$ ), and  $x$  and  $e$  are mutually independent, we can see that both  $D_0$  and  $D_1$  are Chi-Square distributed, as follows:

$$D_0 \sim 2\sigma_e^2 \chi_\nu^2$$

$$D_1 \sim 2(\sigma_e^2 + \sigma_x^2) \chi_\nu^2$$

where  $\chi_\nu^2$  is a standard Chi-Square variable with  $\nu$  degrees of freedom, mean  $\nu$  and variance  $2\nu$ .

In Fig. 2 is shown the outcome of the Chi-Square fitting test with the actual distances between genuines and impostors:

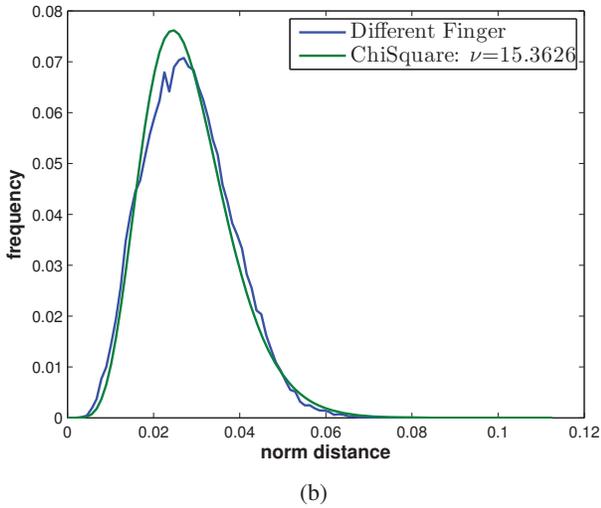
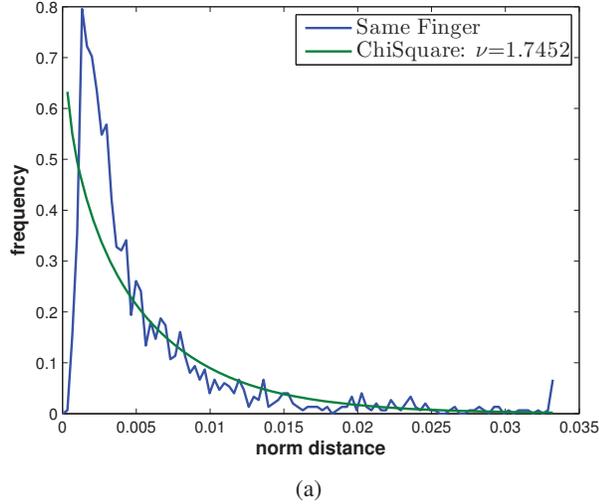


Fig. 2. Distribution and Chi-Square fitting of the distances between FingerCodes, using a single template per enrollment: (a) genuines; (b) impostors.

such distances are computed over the data set described in Section IV, using FingerCode vectors of 640 features. As to  $D_0$ , the fitting with a Chi-Square distribution is quite poor, probably due to the fact that measurement errors on the same finger are correlated and exhibit non-Gaussian statistics. On the other hand,  $D_1$  is well modeled by a Chi-Square distribution. Interestingly, the values of the degrees of freedom of the Chi-Square functions ( $\nu$ ) are far under the number of FingerCode features. This fact suggests that the length of FingerCode vectors can be considerably reduced to make it suitable for encrypted domain computing, without losing FingerCode discriminating properties.

A simple strategy to exploit the correlation of FingerCode features could be to decimate the FingerCode representation. A possible approach is to reduce the number of sectors of the tessellation to yield shorter vectors. Even if this strategy does not guarantee to extract truly independent features, it has the advantage of being applicable to every dataset.

An alternative approach is that of looking for the most compact representation. Taking into account the reduced number of degrees of freedom, we can model a FingerCode as follows

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (1)$$

where  $\mathbf{s}$  is a  $M \times 1$  vector of i.i.d. Gaussian variables and  $\mathbf{A}$  is a  $N \times M$  projection matrix such that  $\mathbf{A}^T \mathbf{A} = \mathbf{I}_M$ , with  $M \ll N$ ,  $N$  being the length of FingerCode vector  $\mathbf{x}$ . Considering the difference of mutually independent FingerCodes, this will be given as

$$\mathbf{x}_1 - \mathbf{x}_2 = \mathbf{A}(\mathbf{s}_1 - \mathbf{s}_2) = \mathbf{A}\mathbf{d} \quad (2)$$

i.e., the squared distance between mutually independent FingerCodes is  $D_1 = (\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2) = \mathbf{d}^T \mathbf{A}^T \mathbf{A} \mathbf{d} = \mathbf{d}^T \mathbf{d}$ , which is distributed as a Chi-Square with  $\nu = M$  degrees of freedom.

From the above model, it is evident that a set of reduced FingerCodes obtained as  $\mathbf{s} = \mathbf{A}^T \mathbf{x}$  will retain the same discriminating capabilities as the original FingerCodes. If we concentrate on a specific dataset, the projection matrix  $\mathbf{A}$  can be obtained via principal component analysis (PCA). First of all, we estimate the covariance matrix from the data set, i.e.,

$$\mathbf{C}_x = \frac{1}{L} \sum_j \mathbf{x}_j \mathbf{x}_j^T - \bar{\mathbf{x}} \bar{\mathbf{x}}^T \quad (3)$$

where  $L$  is the size of the dataset and  $\bar{\mathbf{x}} = \frac{1}{L} \sum_j \mathbf{x}_j$ . Then, we compute the eigendecomposition of the covariance matrix as  $\mathbf{C}_x = \mathbf{V} \mathbf{\Sigma} \mathbf{V}^T$ . If the data exactly follow the model in (1), then only the first  $M$  eigenvalues in  $\mathbf{\Sigma}$  will be different from zero. In practice this will not be always true, however the projection matrix can be computed as the  $M$  eigenvectors corresponding to the  $M$  eigenvalues having higher magnitude.

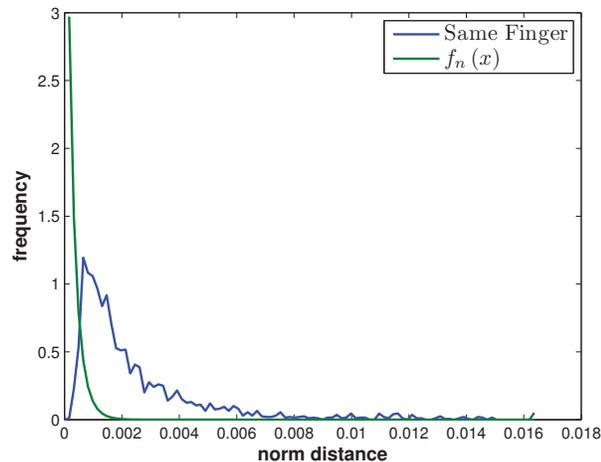
Now it is possible to approach the real model, introducing the minimum distance selection. We remind that we use an enrollment set composed by nine different FingerCodes, each corresponding to the fingerprint image rotated by an angle between  $-45$  and  $45$  degrees.

It is possible to obtain the distribution  $f_n(x)$  of the distances taken as the minimum of a set of  $n$  elements from the distribution  $f(x)$  of the 1:1 distances, under the hypothesis that the  $n$  observations are independent, with the following relation:

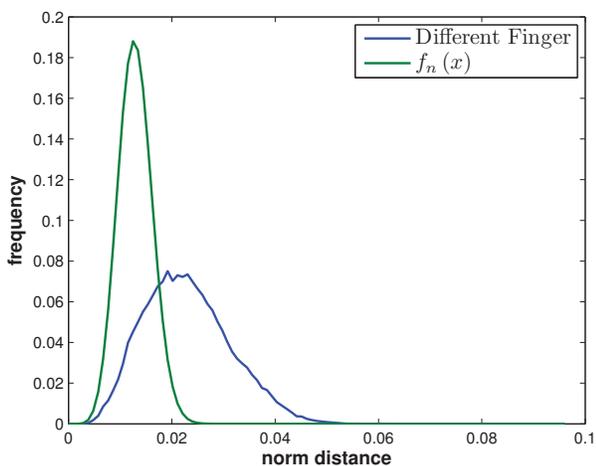
$$f_n(x) = n f(x) [1 - F(x)]^{n-1}$$

where  $F(x)$  is the cumulative distribution function of the 1:1 distances.

In Fig. 3 is shown the outcome of the test. We can see that the fitting quality is very poor, probably because the nine distances between the FingerCode to be authenticated and the FingerCodes of the enrollment set are not independent at all. In fact, the FingerCodes that compose the enrolled template of an individual represent the same image rotated. For this reason, it is plausible that functions of the Euclidean distances between these data are dependent.



(a)



(b)

Fig. 3. Distribution and Chi-Square fitting of the distances between FingerCodes, using nine templates per enrollment: (a) genuines; (b) impostors.

#### A. Quantization Effects

Since our aim is to simplify the FingerCode representation to make it suitable for encrypted domain computations, we try to introduce the quantization of the feature values in the model. Let  $x$  be a FingerCode and let  $\rho(\cdot)$  be a quantization function. The quantized FingerCode is denoted by  $x_q = \rho(x)$  and  $x = x_q + q$ , where  $q$  is the quantization error introduced by the quantization process. According to the previous representation, a FingerCode is made by an ideal component, an acquisition noise component, and a quantization noise component:  $x_q = \tilde{x} + e + q$ . We can compute the distances  $D_0$  between genuines and find

$$D_0 = \sum_{i=1}^N (\tilde{x}_1(i) + e_1(i) + q_1(i) - (\tilde{x}_2(i) + e_2(i) + q_2(i)))^2$$

that can be expressed in short as follows:

$$D_0 = \sum_{i=1}^N (\Delta_e(i) + \Delta_q(i))^2$$

where  $N$  is the number of elements in a FingerCode, and  $\Delta_e(i)$ ,  $\Delta_q(i)$  are the differences between acquisition errors and quantization errors, respectively. Unluckily, the quantization errors  $e$  and the FingerCode  $x$  are not independent and the model becomes very complicated.

To overcome these problems, we decided to deal with the data from an empirical point of view. We investigated the reduction of the FingerCode features with two different approaches. The first approach is based on the decimation of the tessellation of the region of interest and we will name it tessellation reduction. The second approach is based on the application of PCA to the chosen dataset. It is important to notice that while the first method can be applied to every database, i.e., it is not data-dependent, the PCA approach implies an eigenvalue recomputation for every different database. The effect of quantization on the obtained datasets will be then evaluated by investigating the statistical properties of the distances, i.e., mean and standard deviation when the FingerCode features are quantized using different number of bits, and by analyzing the matching performance of the quantized FingerCodes.

#### IV. EXPERIMENTAL RESULTS

In this section it will be shown the behavior of data statistic parameters depending on the number of quantization bits and FingerCode features used. Finally we will focus upon Equal Error Rate (EER) and receiver operating characteristic (ROC) curves to describe the performance achieved by FingerCode matching with tessellation reduction and PCA.

Results are based on a database of 408 fingerprint samples captured by a Cross Match Verifier 300 scanner at 500 dpi [11]. There are 8 samples for each finger, resulting in a total of 51 different fingers. For each sample, we generate nine FingerCode templates corresponding to a set of nine rotations from  $-45^\circ$  to  $45^\circ$  and we consider the template corresponding to  $0^\circ$  as the reference FingerCode. The statistic of the distances of genuines are evaluated by computing the distance between each reference FingerCode and the templates of the other 7 acquisitions of the same finger, resulting in  $7 * 51 = 357$  distance values. The statistic of the distances of impostors are evaluated by computing the distance between each reference FingerCode and the templates of the other 50 fingers, resulting in  $8 * 50 * 51 = 20400$  distance values.

When using tessellation reduction, we generate 8 sets of FingerCodes with length ranging from 640 features to 8 features. The choice of FingerCode parameters for each set is detailed in Table I. When using PCA, we consider reduced representations with length ranging from 64 features to 4 features, obtained by applying PCA to the original FingerCode set having length 640 features.

In Fig. 4-(a) it is shown the mean of the distances between genuines using tessellation reduction: we can see that the mean

TABLE I  
PARAMETERS OF FINGERCODES OBTAINED BY TESSELLATION  
REDUCTION.

N. Features	N.Filters	N. Rings	H. Rings (pixel)	N. Arcs
640	8	5	20	16
384	8	4	25	12
192	8	3	20	8
96	4	3	33	8
48	4	3	33	4
32	4	2	50	4
16	4	2	50	2
8	2	2	50	2

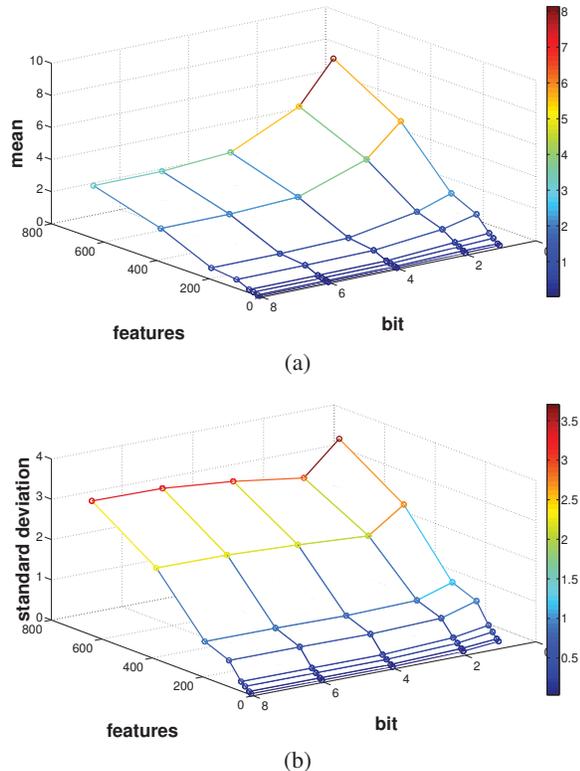


Fig. 4. Mean (a) and standard deviation (b) of the distances between genuines when using tessellation reduction.

grows slightly with the number of features, while it starts to increase with strong quantizations, especially using long FingerCodes. The standard deviation parameter (Fig. 4-(b)) is less sensitive to the quantization strength, and it seems to increase only depending on the number of features. The trend of mean and standard deviation of the distances between impostors is similar (Fig. 5). It is possible to explain the behavior of the distances between impostors considering that the contribution of quantization noise is not enough to change the original statistic.

Now, we focus on the same parameters estimated on the dataset treated with a PCA approach. The parameters behavior is almost the same for genuines and impostors, and it shows a very weak dependence on the number of features, while a strong increase of the distances is observed with few quantization bits, especially for longer FingerCodes (Figs. 6-7). A

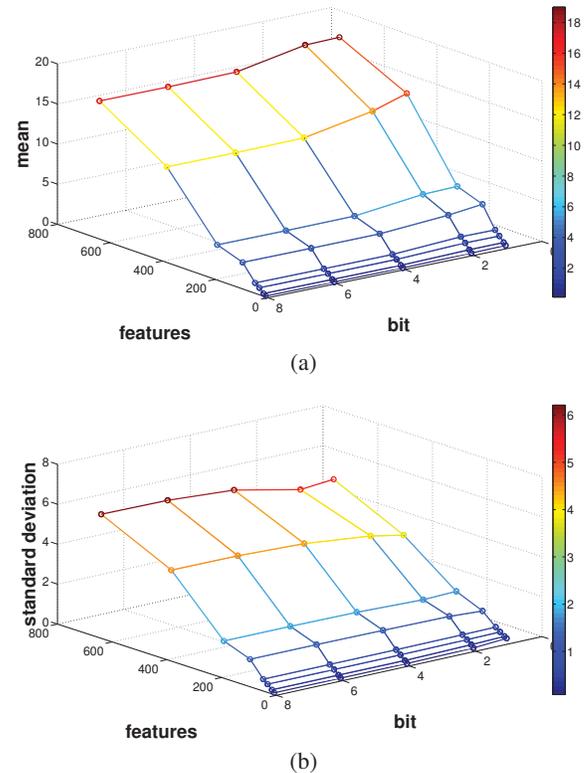


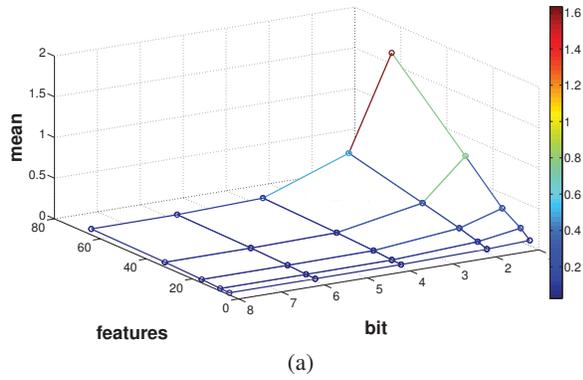
Fig. 5. Mean (a) and standard deviation (b) of the distances between impostors when using tessellation reduction.

reason for the weak dependence on the number of features is that PCA approach allows the selection of the most distinctive values of the FingerCode to represent the vector itself.

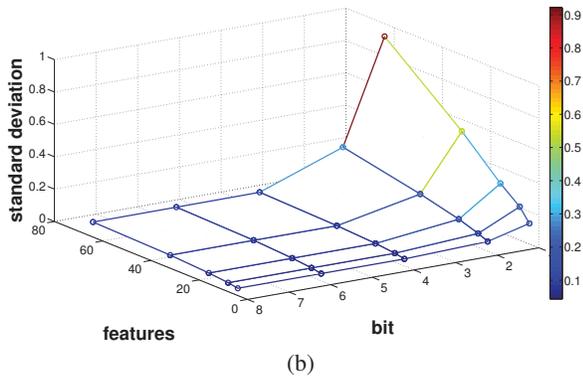
Now we pay attention to the biometric system performance. The ROC curves are indexes of the system quality and they are created plotting the False Non-Match Rate (FNMR) against the False match Rate (FMR) as the discrimination threshold varies. The more the system is accurate, the more the ROC curve is close to the axis. Comparing the ROC curves for databases with 640 (Fig. 8-(a)) and 96 features (Fig. 8-(b)) produced with tessellation reduction we can see that the performances are very similar, even using a strong quantization. To complete the assessment of the system performance we look at the Equal Error Rate (EER). Fig. 9-(a) shows the trend of the EER as the number of bits and features varies. It is possible to set two imaginary boundaries corresponding to a quantization of 2 bit and a FingerCode length of 96 features, where the performance is approximately unaffected by the FingerCode simplification.

Fig. 8-(c) shows the ROC curves of a database made of FingerCodes 8 features long, computed via PCA. We can see that the performance achieved is quite good up to a quantization of 4 bits, while increasing the quantization strength it starts to get worse quickly.

Finally, we focus on the EER trend. Fig. 9-(b) shows that even in this case, it is possible to set two imaginary boundaries for a quantization of 4 bits and a FingerCode length of 8

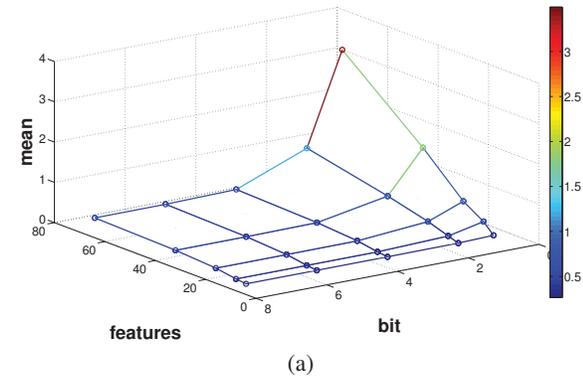


(a)

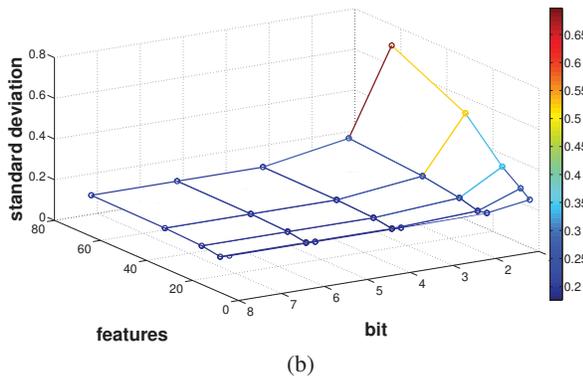


(b)

Fig. 6. Mean (a) and standard deviation (b) of the distances between genuines when using PCA.

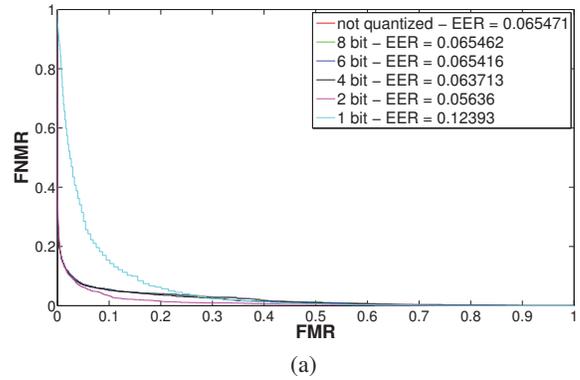


(a)

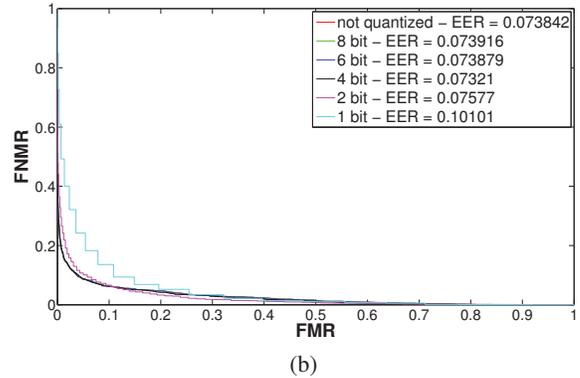


(b)

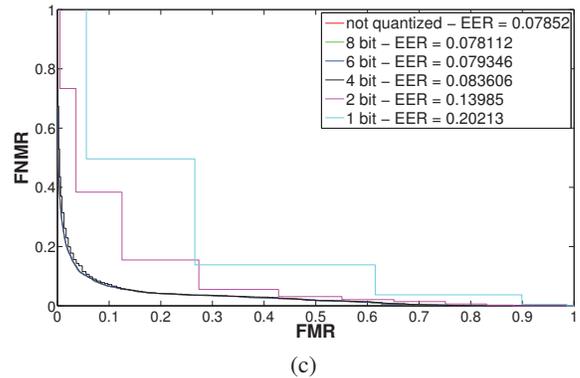
Fig. 7. Mean (a) and standard deviation (b) of the distances between impostors when using PCA.



(a)



(b)



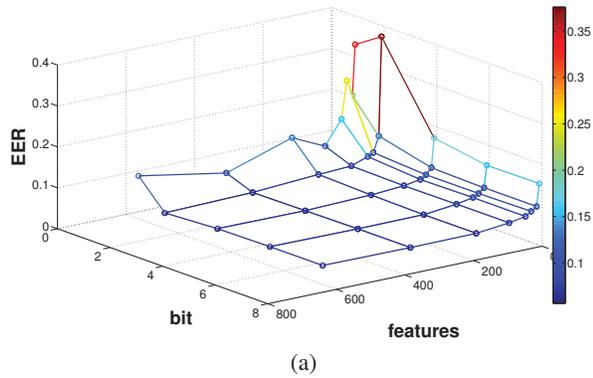
(c)

Fig. 8. ROC curves of the FingeCode system using different quantization steps: (a) 640 features (original configuration); (b) 96 features with tessellation reduction; (c) 8 features with PCA.

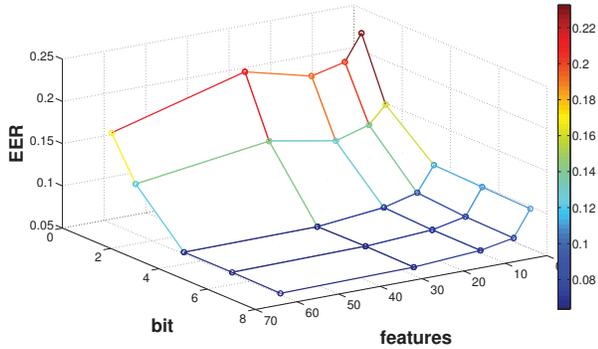
features to define a region where the EER remains acceptable.

## V. DISCUSSION AND CONCLUSIONS

By looking at the results shown in the previous section, it is evident that the size of FingerCode templates can be significantly reduced without affecting the performance of the biometric system. To evaluate the effects of this reduction of the template size on the complexity of an encrypted domain implementation, we can refer to the data provided in [10], which describes a complete implementation of the encrypted domain FingerCode matching protocol based on homomorphic encryption and garbled circuits. The computational time and the bandwidth used by the SSP protocol proposed in [10]



(a)



(b)

Fig. 9. Equal error rate (EER) achieved by the different feature reduction strategies in the various configurations: (a) tessellation reduction; (b) PCA.

TABLE II  
COMPUTATIONAL TIME OF THE SSP PROTOCOL IN [10] WITH A DATABASE OF 100 ENROLLED ENTRIES (900 FEATURE VECTORS). SECURITY PARAMETER: 80 BITS

Features	Quantization	Time (s)
96	2	37.43
96	4	45.58
192	2	44.43
192	4	53.66
640	8	114

are reported in Tables II and III, respectively, using different FingerCode configurations. As we can see, a FingerCode configuration using 96 features obtained through tessellation reduction and quantized with 2 bits employs approximately the 33% of the computational time and the 40% of the bandwidth required by the original configuration. Since the proposed configuration achieves a biometric performance very close to that of the original configuration, the proposed reduction of the template size can considerably ease the implementation of encrypted domain biometric matching.

A further complexity reduction can be expected using the PCA strategy. However, in this case we can expect the optimal projection matrix to be data dependent. Hence, an encrypted domain implementation will require either to make this projection matrix public, which may leak information on the identities contained in the database, or to perform PCA in the encrypted domain, which will increase the overall complexity

TABLE III  
BANDWIDTH (BITS) OF THE SSP PROTOCOL IN [10] WITH A DATABASE OF 100 ENROLLED ENTRIES (900 FEATURE VECTORS). SECURITY PARAMETER: 80 BITS

Features	Quantization	B (bits)
96	2	1,610,000
96	4	1,912,400
192	2	1,691,670
192	4	1,994,070
640	8	4,394,200

of the protocol. Further research will be devoted to assessing the actual pros and cons of the PCA strategy when applied in a realistic scenario.

## REFERENCES

- [1] Z. Erkin, A. Piva, S. Katzenbeisser, R. L. Lagendijk, J. Shokrollahi, G. Neven, and M. Barni, "Protection and retrieval of encrypted multimedia content: When cryptography meets signal processing," *EURASIP Journal on Information Security*, vol. 2007, Article ID 78943, 20 pages, 2007.
- [2] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *PETS '09: Proceedings of the 9th International Symposium on Privacy Enhancing Technologies*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 235–253.
- [3] A. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *ICISC '09: Proceedings of the 12th Annual International Conference on Information Security and Cryptology*, ser. LNCS, vol. 5984. Springer-Verlag, December 2–4, 2009, pp. 235–253.
- [4] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Lecture Notes in Computer Science*, vol. 1592. Springer-Verlag, 1999, pp. 223–238.
- [5] T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discrete Fourier transform in the encrypted domain," *IEEE Transaction on Information Forensics and Security*, vol. 4, no. 1, pp. 86–97, March 2009.
- [6] —, "Composite signal representation for fast and storage-efficient processing of encrypted signals," *IEEE Transaction on Information Forensics and Security*, vol. 5, no. 1, pp. 180–187, March 2010.
- [7] I. Damgård, M. Geisler, and M. Krøjgaard, "Efficient and secure comparison for on-line auctions," in *ACISP*, ser. Lecture Notes in Computer Science, J. Pieprzyk, H. Ghodosi, and E. Dawson, Eds., vol. 4586. Springer, 2007, pp. 416–430.
- [8] B. Pinkas, T. Schneider, N. Smart, and S. Williams, "Secure two-party computation is practical," in *ASIACRYPT*, ser. Lecture Notes in Computer Science, M. Matsui, Ed., vol. 5912. Springer, 2009, pp. 250–267.
- [9] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, May 2000.
- [10] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. Donida Labati, P. Failla, D. Fiore, R. Lazzarotti, V. Piuri, A. Piva, and F. Scotti, "A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates," in *IEEE Fourth International Conference On Biometrics: Theory, Applications And Systems (BTAS 2010)*, September 2010.
- [11] "Sample fingerprint databases," VeriFinger download page, 2006, available at <http://www.neurotechnology.com/download.html>.