A KEYPOINT DESCRIPTOR FOR ALIGNMENT-FREE FINGERPRINT MATCHING

Ravi Garg[†] *and Shantanu Rane*[*]

† University of Maryland, College Park, MD.
∗ Mitsubishi Electric Research Laboratories, Cambridge, MA.

ABSTRACT

Secure fingerprint authentication via encrypted-domain processing imposes constraints on the underlying feature extraction method: Firstly, it requires fixed-length feature vectors to be amenable to computing distances or correlations. Secondly, extra information must be stored in the clear so that the fingerprints can be aligned prior to feature extraction and secure comparison. These constraints potentially restrict the flexibility, increase computational complexity, and even reduce the security of the scheme. We desire feature vectors suitable for encrypted-domain matching while being free of the above constraints. To this end, a local neighborhood is defined around certain detected minutiae points, and features are extracted based on relative locations of close minutia points, local ridge texture and local ridge orientation. The locality of the features provides robustness to rotation and translation. Feature vectors are compared using operations that can be performed using secure primitives. The process of computing the matching scores – genuine or impostor – implicitly yields the best alignment without needing to store unencrypted side information at the access control device. The scheme achieves an Equal Error Rate of 1.46% on a proprietary database and 7.86% on the FVC2002 public database.

Index Terms— biometrics, fingerprints, feature extraction

1. MOTIVATION

Fingerprint-based access control involves extracting discriminative features from an enrollment fingerprint and comparing them against corresponding features extracted from a probe or test fingerprint. There is an extensive literature on feature extraction for fingerprint-based authentication and identification systems [1, 7]. Almost all these schemes assume that the enrollment features computed are either stored in the clear, or in the form of ciphertexts that are fully decrypted during authentication. Thus, when the system is provided with features from a test fingerprint, it is assumed that the enrollment features are readily available for comparison at the access control device. This assumption breaks down for biometric template protection systems, specifically those based on secure multiparty computation, wherein enrollment features are stored in encrypted form, and *never decrypted* during the authentication protocol.

Encryption ensures that an adversary does not discover the fingerprint features of the enrolled identities at any stage of the authentication protocol. The challenge, however, is that the device must now perform the comparison of the test features and the enrollment features in the encrypted domain. In the literature, encrypted-domain comparisons are predominantly realized using homomorphic cryptosystems and garbled circuits [2, 3]. Most of the schemes proposed in the literature involving the comparison of encrypted biometric features have two characteristics: Firstly, the feature extraction algorithm produces a fixed-length feature vector, which allows comparison of biometric features to be cast as a problem of determining distances between the enrollment and test feature vectors. Secure and nearly real-time protocols for computing distance metrics such as Euclidean [4], Hamming [2, 3, 5] and Manhattan distances [6] are now available. Secondly, to guarantee accurate biometric matching, alignment must be performed between the enrollment and probe fingerprints, even prior to feature extraction. Fingerprint registration is a well-studied topic (see [7] for a survey), but most techniques do not apply in a straightforward way to encrypted-domain authentication. Many works on encrypted-domain biometric matching either do not treat the alignment problem or implicitly assume that they are provided with aligned biometrics. Others allow the device to store unencrypted alignment parameters, such as the locations of high curvature points on the enrollment fingerprint [8, 9]. These parameters enable the access control device to first align the test fingerprint to the enrollment fingerprint, before extracting test features and comparing them with the enrollment features in the encrypted domain.

The above two characteristics impose constraints on the design of the access control system. Firstly, requiring a fixed-length feature vector reduces the flexibility of the feature matching scheme because it cannot accommodate, for example, feature matching from cropped images or fingerprints that produce very few minutia points. Secondly, though inferring minutiae locations or orientation fields from high curvature points or other unencrypted alignment information is difficult, this information may allow an adversary to guess whether a fingerprint contains core, or tent or delta structures, thereby narrowing his choice of attack vectors. Our goal is to develop a fingerprint feature extraction scheme suitable for encrypted domain processing that does not require the number of extracted probe features to be equal to the number of stored enrollment features, and does not need to explicitly perform alignment — thus removing the need to store any alignment parameters.

We propose a new descriptor containing information about minutiae structure, ridge texture, and ridge orientation from a local neighborhood around key minutiae points. Due to the localized nature of the feature extraction mechanism, the proposed approach does not require any pre-alignment step. A fixed-length feature vector is derived *for each keypoint*, though enrollment and probe fingerprints can have a *different number of key points*. This compromise allows matching of local features using existing secure computation methods, a procedure which *implicitly performs alignment during matching*. We have recently became aware of an alignment-free matching scheme that employs similar ideas as our work, especially the idea of constructing local descriptors that can be matched using a similarity matrix consisting of pairwise descriptor distances [10]. This scheme has a slightly different construction of minutiae-based

This work was carried out when R. Garg was an intern at Mitsubishi Electric Research Laboratories.

features from the ones covered here, moreover, it does not employ texture and orientation-based features or their fusion into a final matching score. The remainder of this paper is organized as follows: The proposed keypoint descriptor is described in the Section 2. Section 3, describes the performance of the proposed method on two databases. Section 4 briefly summarizes the key features of the alignment-free approach and concludes the paper.

2. A KEYPOINT-BASED FINGERPRINT DESCRIPTOR

In this work, a keypoint is defined as a minutiae point that contains at least n minutiae as its neighbors within a circle of radius R. This notion is made precise below. Our goal is to compute a fixed-length descriptor for each keypoint that represents the fingerprint structure in the local neighborhood of that keypoint. The descriptor is composed of minutiae-based, texture-based and orientation-based features obtained from the local neighborhood of each keypoint.

Let U_k be the number of minutiae points in the k^{th} fingerprint image. Typically, U_k varies across impressions obtained from different fingers. Further, it also varies across impressions obtained from the same finger owing to measurement noise. Represent the i^{th} minutia point in the k^{th} fingerprint image as $m_i^k = \{x_i^k, y_i^k, \theta_i^k\}$ using the X-coordinate, Y-coordinate, and Θ-coordinate of the minutia point. Let the set of neighbors of m_i^k be given by

$$\mathcal{D}(m_i^k) = \{m_j^k : d(m_i^k, m_j^k) < R, j = 1, \ldots, U_k, j \neq i\}$$
$$\text{where } d(m_i^k, m_j^k) = \sqrt{(x_i^k - x_j^k)^2 + (y_i^k - y_j^k)^2}$$

where R is a predefined radius. Now, the n-nearest neighbor set of m_i^k is the subset of $\mathcal{D}(m_i^k)$ containing n elements closest in Euclidean distance to m_i^k. Denote the n-nearest neighbor set of m_i^k by $\mathcal{N}(m_i^k) = \{\bar{m}_1^k, \ldots, \bar{m}_n^k\}$, where $\bar{m}_l^k = \{\bar{x}_l^k, \bar{y}_l^k, \bar{\theta}_l^k\}, l = 1, 2, \ldots, n$ is the l^{th} nearest neighbor of m_i^k. Any minutia location that has a n-nearest neighbor set is considered as a keypoint.

2.1. Minutiae-Based Features

For each keypoint m_i^k, the proposal is to extract features that can uniquely represent the local geometric structure of $\mathcal{N}(m_i^k)$ relative to m_i^k . For this purpose, denote the minutia-based feature vector around m_i^k as

$$\mathbf{m}_i^k = [d_{i1}^k, \ldots, d_{in}^k, \theta_{i1}^k, \ldots, \theta_{in}^k, \phi_{i1}^k, \ldots, \phi_{in}^k]$$

where each feature element is given by

$$d_{ij}^k = (1/R)\, d(m_i^k, \bar{m}_j^k)$$
$$\theta_{ij}^k = \begin{cases} \frac{\bar{\theta}_j^k - \theta_i^k}{2\pi} & \text{if } \bar{\theta}_j^k \geq \theta_i^k \\ \frac{2\pi - (\theta_i^k - \bar{\theta}_j^k)}{2\pi} & \text{if } \bar{\theta}_j^k < \theta_i^k \end{cases}$$
$$\psi_{ij}^k = \begin{cases} \frac{\bar{\psi}_j^k - \theta_i^k}{2\pi} & \text{if } \bar{\psi}_j^k \geq \theta_i^k \\ \frac{2\pi - (\theta_i^k - \bar{\psi}_j^k)}{2\pi} & \text{if } \bar{\psi}_j^k < \theta_i^k \end{cases}$$

where $\bar{\psi}_j^k$ measures the radial placement of the nearest neighbor minutae around m_i^k according to:

$$\beta = \tan^{-1}\left(\frac{|\bar{y}_j^k - y_i^k|}{|\bar{x}_j^k - x_i^k|}\right)$$

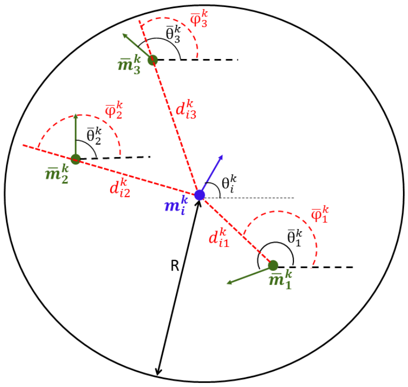


Fig. 1. Example of a minutiae structure-based keypoint descriptor.

$$\bar{\psi}_j^k = \begin{cases} \beta & \text{if } \bar{y}_j^k > y_i^k, \bar{x}_j^k > x_i^k \\ \pi - \beta & \text{if } \bar{y}_j^k > y_i^k, \bar{x}_j^k < x_i^k \\ \pi + \beta & \text{if } \bar{y}_j^k < y_i^k, \bar{x}_j^k < x_i^k \\ 2\pi - \beta & \text{if } \bar{y}_j^k < y_i^k, \bar{x}_j^k > x_i^k \end{cases}$$

Defining the feature vectors as above ensures that each feature value is in $[0, 1]$ range, and that keypoint features are invariant to rotation and translation with respect to the keypoint. A diagrammatic representation of the minutiae-based features around a keypoint m_i^k is given in Fig. 1 for the case of $n = 3$. The minutia-based features represent the relative minutiae distances, relative minutiae orientations and the relative radial minutiae placement around the keypoint, thus capturing the local structure in the neighborhood of the keypoint m_i^k. Unfortunately, in some fingerprint images, the number of keypoints may be small and this adversely affects the matching performance. To mitigate the effect of such fingerprints, we also employ ridge texture and orientation-based features as explained below.

2.2. Ridge Texture-Based Features

To extract the ridge texture information around the location of keypoint m_i^k, a square patch of dimension R_1 around m_i^k is passed through directional Gabor filters with dominant angle $\frac{\pi * p}{P}, p = \{0, 1, \ldots, P-1\}$ relative to θ_i^k. From the output of each filter, we calculate the standard deviation of the output image patch as a feature, similar to the method used in [11, 12]. The feature obtained can thus be represented as:

$$\mathbf{t}_i^k = \frac{1}{t_{\max}}[t_0, t_1, \ldots, t_{P-1}]$$

where $t_p, p = \{0, 1, \ldots, P-1\}$ denotes standard deviation of the image patch obtained as the output of the directional Gabor filters at an angle of $\frac{\pi p}{P}$. The normalizing constant $t_{\max}$ is the maximum value of the standard deviation computed for all image patches around all key points in the training set, and it is used to restrict each feature vector into the range $[0, 1]$. An example of the texture-based feature extraction for $P = 4$ is shown in Fig. 2. The filtered images show the dominant ridge orientation in the range of corresponding angle of the directional Gabor filters. The square patch represented in red around the keypoint represents square patch of dimension R_1 in which the standard deviation is computed.

2.3. Ridge Orientation-Based Features

To extract the orientation features, we use the method proposed in [13] which samples the ridge orientations in a regular radial sam-

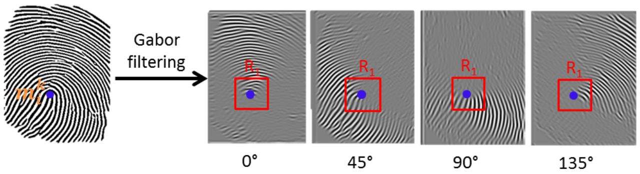


Fig. 2. Example of a texture-based keypoint descriptor.

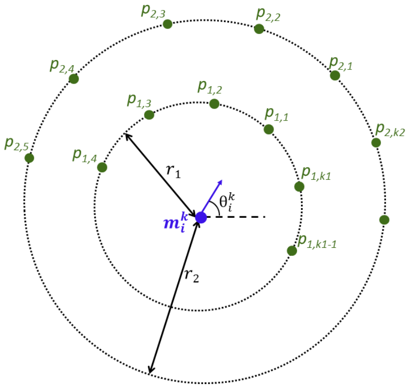


Fig. 3. Example of a orientation-based keypoint descriptor.

pling grid around the keypoint. As shown in Fig. 3, the sampling grid around a given minutiae m_i^k consists of L concentric circles of radii $r_l, 1 \leq l \leq L$, with each circle having K_l sampling points $p_{l,m}$ equally distributed along its circumference. To make these features invariant to rotation around the keypoint, the orientations at the sampling points $\frac{2\pi r_l m}{K_l}, (0 \leq m \leq K_l - 1)$ are taken relative to the orientation θ_i^k of minutiae m_i^k. Denoting by $\theta_{l,m}^k$, the local ridge orientation estimated at $p_{l,m}$, the descriptor is given by

$$\mathbf{o}_i^k = \left\{ \left\{ \frac{\min(|2\pi - (\theta_{l,m}^k - \theta_i^k)|, |\theta_{l,m}^k - \theta_i^k|)}{\pi} \right\}_{m=1}^{K_l} \right\}_{l=1}^{L}$$

Division by π ensures that each feature element is in the range $[0, 1]$.

After deriving the minutiae-based, texture-based, and orientation-based descriptors around each key-point, the descriptor for the k^{th} fingerprint image is obtained by concatenating the descriptor for each key-point as follows:

$$\mathbf{M}^k = \begin{bmatrix} \mathbf{m}_1^k \\ \vdots \\ \mathbf{m}_U^k \end{bmatrix} \quad \mathbf{T}^k = \begin{bmatrix} \mathbf{t}^k \\ \vdots \\ \mathbf{t}_U^k \end{bmatrix} \quad \mathbf{O}^k = \begin{bmatrix} \mathbf{o}_1^k \\ \vdots \\ \mathbf{o}_U^k \end{bmatrix}$$

For a query fingerprint, feature extraction is performed exactly as described above, and matching scores between the query and the enrollment fingerprints are obtained as described below.

2.4. Fingerprint Matching

The features extracted from the given query image are matched with the claimed enrollment features in the database, using a keypoint-wise pattern matching technique. The algorithm described below returns a matching score that represents the similarity between the two fingerprints. For conciseness, we describe the matching algorithm using only minutiae-based features; the same algorithm is used for comparing the texture descriptor and orientation descriptors of the enrollment and probe fingerprints.

Following the notations from the previous section, denote by $\mathbf{M}^k$ and $\mathbf{M}'$ the keypoint minutiae descriptor of the k^{th} enrollment and the query fingerprints, respectively. Let there be U_k minutiae key-points in $\mathbf{M}^k$ and U' key-points in $\mathbf{M}'$. The matching method estimates one-to-one correspondences between the key-points in the probe and enrollment fingerprints. To accomplish this, it is necessary to have a measure of similarity between the minutiae-based descriptors $\mathbf{M}^k = \{\mathbf{m}_i^k\}_{i=1}^{U_k}$ and $\mathbf{M}' = \{\mathbf{m}'_j\}_{j=1}^{U'}$. We define a $U_k \times U'$ similarity matrix whose elements are given by $S(i,j) = \|\mathbf{m}_i^k - \mathbf{m}'_j\|_1$. Indices (i,j) of elements having a small value in the matrix $\mathbf{S}$ are indicative of a potential correspondence between keypoints. We consider that keypoint i in $\mathbf{M}^k$ and keypoint j in $\mathbf{M}'$ match if the corresponding descriptors are more similar to each other than to descriptors from any other keypoint pairs in $\mathbf{M}^k$ and $\mathbf{M}'$. This is formulated into the following measure of correspondence:

$$P_m(i,j) = \frac{S(i,j)}{\sum_{j'=1}^{U'} S(i,j') + \sum_{i'=1}^{U_k} S(i',j) - 2S(i,j)}$$

After the matrix $\mathbf{P_m}$ is computed, the coordinates of the minimum value in $\mathbf{P_m}$ indicate a match between a keypoint in the enrollment fingerprint and a key point in the probe fingerprint. The corresponding column and row are removed and the process is repeated until all keypoint matches are identified. Finally, the number of matching points whose correspondence value is less than a certain threshold τ_m is used to compute the matching score as

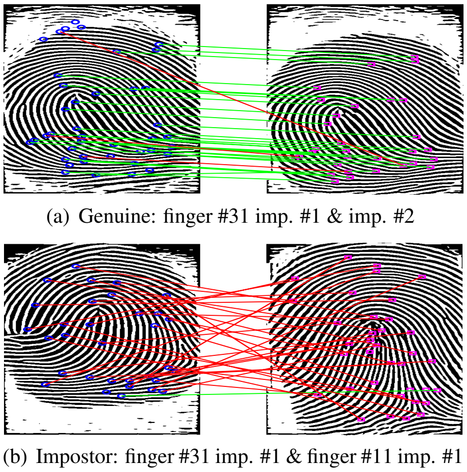
$$\mathcal{S}_m = \frac{M_{\text{matched}}}{\min(U_k, U')}$$

where M_{matched} is the number of keypoint correspondences (i,j) for which $P_m(i,j) \leq \tau_m$. Similar score calculation is performed for texture-based and orientation-based descriptors to obtain scores $\mathcal{S}_t$ and $\mathcal{S}_o$, respectively. The value of the threshold τ_m, and corresponding thresholds τ_t and τ_o for texture and orientation based descriptors can be derived empirically. The scores can now be used to compute the receiver operating characteristic (ROC) curves, either individually, or in combinations of two or three feature types, for e.g., when all three scores are to be used, the fused score is the linear combination, $\mathcal{S} = w_m\mathcal{S}_m + w_t\mathcal{S}_t + w_o\mathcal{S}_o$, where the weights satisfy $w_m + w_t + w_o = 1$. The values of w_m, w_t, w_o are obtained by running an exhaustive search to minimize the equal error rate (EER) on a training subset of the dataset, as will be described in Section 3.

3. EXPERIMENTAL RESULTS

The proposed scheme is evaluated separately on two databases, a proprietary database labeled “DTL” and the public-domain FVC2002 database [14], each of which consist of 100 fingerprints with 8 impressions per fingerprint. To learn the weights w_m, w_t and w_o for score level fusion, impression #1 of each finger is matched with all other impressions of that finger, giving 700 genuine scores. Also, impression #1 of each finger is matched with impression #2 of all other fingers to obtain 9900 impostor scores. Exhaustive search is used to obtain the values of w_m, w_t, w_o that minimize the Equal Error Rate (EER) for the scheme employing the fused matching score.

For matching in the testing phase, each impression of each finger is compared with all other impressions for the same user, giving 2800 genuine scores. For impostor matching, impression #1 of each user is compared with impression #1 of all other users, giving 9900



(a) Genuine: finger #31 imp. #1 & imp. #2

(b) Impostor: finger #31 imp. #1 & finger #11 imp. #1

Fig. 4. Correspondences obtained by finding smallest entries of $\mathbf{P_m}$. Green lines denote key-point matches retained after thresholding by τ_m, Red lines are matches rejected after thresholding by τ_m.

impostor scores. The values of the parameters described in Section 2 are: $n = 2$, $R = 90$, $P = 8$, $L = 3$, $r_1 = 27$, $r_2 = 45$, $r_3 = 63$, $K_1 = 10$, $K_2 = 16$, $K_3 = 22$, $R_1 = 50$, $\tau_m = 0.102$, $\tau_o = 0.155$, and $\tau_t = 0.075$ for both databases. These values are determined empirically, and may change for different biometric sensors.

An example of matching using the proposed keypoint descriptor is given in Fig. 4. The genuine match produces a large number of nearest neighbor correspondences in which the score is less than the threshold τ_m, while the impostor match produces very few such correspondences. The ROC curves obtained after using the proposed algorithm on the DTL and FVC2002 databases are shown in Fig. 5. Among minutiae-based, texture-based and orientation-based features, the former provide the best performance, confirming that minutiae are the most discriminate features in a fingerprint. Combining the features via score level fusion as described in Section 2 improves the performance. Table 1 lists the EER of the proposed approach on the two databases for training and test cases described above. The results depend on the homogeneity of the data set, or more precisely, on the differences between the training and test data sets. In particular, for the DTL dataset, which is more homogeneous, the EER is lower for the test set than the training set. However, for the FVC dataset, which has more variation among a user's fingerprint impressions, this situation is reversed. To confirm that this is not an accident, we also determined the best performance when weights are optimized on the test set, as shown in Table 1. Evidently, fusion of scores from different descriptors reduces the EER, which is lowest when all three descriptors are combined. We conjecture that the EER for FVC2002 is worse because it contains noisier images, has higher variability across the images, and suffers from missing minutiae across genuine fingerprint pairs of many users.

Descriptor type	DTL (train)	DTL (test)	DTL (best)	FVC (train)	FVC (test)	FVC (best)
M	4.67	2.43	2.43	7.78	10.74	10.74
T	9.81	7.99	7.99	21.70	28.56	28.56
O	8.44	5.78	5.78	9.57	12.83	12.83
M+T	2.66	1.77	1.61	7.24	11.74	10.74
M+O	2.65	1.71	1.70	4.84	8.03	7.53
O+T	6.39	4.93	4.45	9.21	11.97	11.72
M+T+O	1.75	1.46	1.32	4.23	7.86	7.53

Table 1. EER for minutiae (M), texture (T), and orientation (O) based descriptors. Below, "train" refers to fusion weight estimation on training data, "test" refers to using trained weights on test data, "best" refers to the oracle case, i.e., weight estimation on test data.

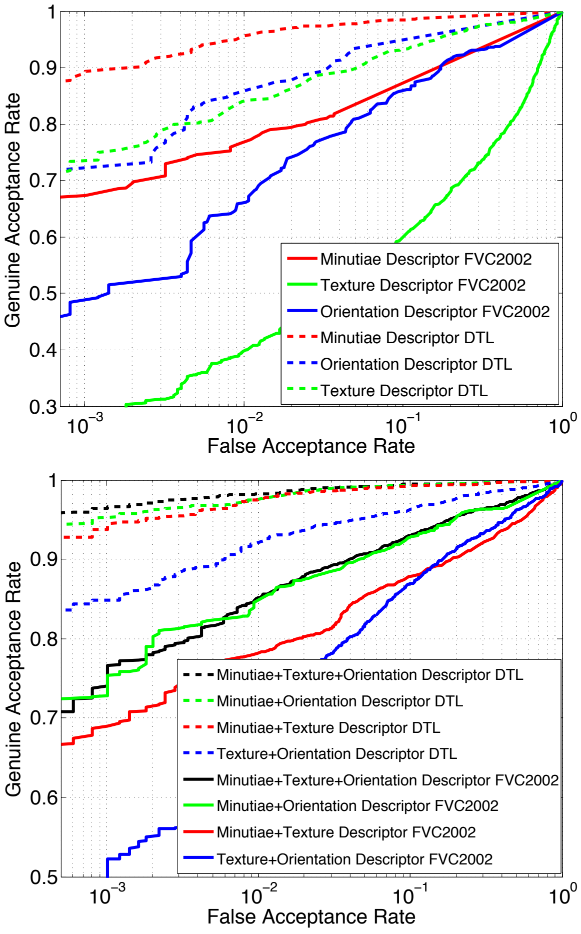


Fig. 5. ROC of individual features (above) and combination of different features (below) for the proposed descriptor (testing phase).

4. SUMMARY

The proposed algorithm uses minutiae-based, texture-based, and orientation-based descriptors in the local neighborhood of a minutia keypoint. This is not constrained to be a fixed-length feature representation since different fingerprint impressions can contain a different number of keypoints. However, individual descriptors do have a fixed length, allowing pairwise distance computation and determination of keypoint correspondences between pairs of fingerprints. The mathematical operations required to obtain these correspondences can be performed in the encrypted domain. The design of efficient and secure protocols for this purpose is a part of our ongoing work. Significantly, no alignment is performed prior to feature extraction; alignment happens *implicitly* as a byproduct of obtaining the matching scores. The obtained EER is competitive with schemes that store explicit alignment information at the access control device [9].

5. REFERENCES

[1] N. Ratha and R. Bolle, *Automatic fingerprint recognition systems*, Springer, 2003.

[2] M.Barni, T.Bianchi, D.Catalano, M.Di Raimondo, R.Donida Labati, P.Failla, D.Fiore, R.Lazzeretti, V.Piuri, F.Scotti, and A.Piva, "Privacy-preserving fingercode authentication," in *ACM Workshop on Multimedia and Security (MMSEC)*, Rome, Italy, Sept 2010.

[3] M. Blanton and P. Gasti, "Secure and efficient protocols for iris and fingerprint identification," in *European Symposium on Research in Computer Security (ESORICS)*, Leuven, Belgium, September 2011.

[4] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, R.L. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *International Symposium on Privacy Enhancing Technologies (PET)*, Seattle, WA, August 2009, pp. 235–253.

[5] J. Bringer, H. Chabanne, M. Izabachène, D. Pointcheval, Q. Tang, and S. Zimmer, "An application of the goldwasser-micali cryptosystem to biometric authentication," in *Australasian Conference on Information Security and Privacy (ACISP)*, 2007, pp. 96–106.

[6] S. Rane, W. Sun, and A. Vetro, "Privacy-Preserving Approximation of L1 Distance for Multimedia Applications," in *IEEE International Conference on Multimedia and Expo (ICME)*, Singapore, July 2010, pp. 492–497.

[7] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*, springer, 2009.

[8] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 4, pp. 744–757, December 2007.

[9] A. Nagar, S. Rane, and A. Vetro, "Privacy and security of features extracted from minutiae aggregates," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Mar. 2010, pp. 1826 –1829.

[10] A. Kisel, A. Kochetkov, and J. Kranauskas, "Fingerprint minutiae matching without global alignment using local structures," *Informatica*, vol. 19, no. 1, pp. 31–44, 2008.

[11] L. Hong, *Automatic personal identification using fingerprints*, Michigan State University, 1998.

[12] A. Ross, A. Jain, and J. Reisman, "A hybrid fingerprint matcher," *Pattern Recognition*, vol. 36, no. 7, pp. 1661–1673, 2003.

[13] M. Tico and P. Kuosmanen, "Fingerprint matching using an orientation-based minutia descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 1009 – 1014, Aug. 2003.

[14] "Fingerprint Verification Competition, Database 2 Set A," http://bias.csr.unibo.it/fvc2002/download.asp, 2002.