

# Secure Biometric Templates from Fingerprint-Face Features

Yagiz Sutcu

yagiz@isis.poly.edu

Qiming Li

qiming.li@ieee.org

Nasir Memon

memon@poly.edu

Polytechnic University, 6 Metrotech Center, Brooklyn, NY 11201, USA

## Abstract

Since biometric data cannot be easily replaced or revoked, it is important that biometric templates used in biometric applications should be constructed and stored in a secure way, such that attackers would not be able to forge biometric data easily even when the templates are compromised. This is a challenging goal since biometric data are “noisy” by nature, and the matching algorithms are often complex, which make it difficult to apply traditional cryptographic techniques, especially when multiple modalities are considered. In this paper, we consider a “fusion” of a minutiae-based fingerprint authentication scheme and an SVD-based face authentication scheme, and show that by employing a recently proposed cryptographic primitive called “secure sketch”, and a known geometric transformation on minutiae, we can make it easier to combine different modalities, and at the same time make it computationally infeasible to forge an “original” combination of fingerprint and face image that passes the authentication. We evaluate the effectiveness of our scheme using real fingerprints and face images from publicly available sources.

## 1. Introduction

A typical biometric authentication system consists of two phases (as depicted in Figure 1). During the *enrollment* phase, a user (say, Alice) scans her biometric data, from which features are extracted and a *template* is created and stored, either in a central database, or on a mobile device. During the *authentication* phase, a user who claims to be Alice would scan her biometric data again, and the same feature extraction algorithm is applied. The result is then compared with the stored template. If they are sufficiently *similar* according to some similarity measure, the matching algorithm outputs a *yes*, which indicates that the user is authentic, or a *no* when the user is not authentic.

Since it is difficult (if possible at all) to replace or revoke biometric data, it is important to keep user biometric data safe when they are used in these authentication systems. In

many systems the original raw biometric data are not stored, but some features are extracted and stored in the system instead. However, in these systems, it is often not clear exactly how difficult it is to forge some biometric data such that similar features can be extracted from them (i.e., to invert the feature extraction function approximately). In fact, an efficient algorithm is recently discovered that can generate a fingerprint from its matching minutiae points [15]. Therefore, storing the biometric features directly as templates would not be secure enough.

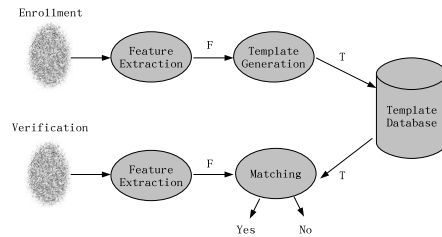


Figure 1. Biometric Authentication System.

There has been intensive study on how to secure the templates such that (1) they can still be used for matching with reasonable performance, and (2) it is hard to forge “original” biometric data that would match a given template. Currently known methods can be roughly categorized into three types: (1) Robust hash functions, where small changes in a biometric sample would yield the same hash value (e.g., [21, 19, 4, 20, 9]); (2) Similarity-preserving hard-to-invert transformations, where similarity of biometric samples would be preserved through the transformation, yet it is difficult to find the original template from a transformed one (e.g., [12, 1, 13, 14]); and (3) Secure sketch based methods, where a recently proposed cryptographic primitive called *secure sketch* is employed, such that given a noisy biometric sample, the original biometric data can be recovered with the help of some additional information (i.e., a sketch), which makes it possible to use biometric data in the same way as passwords. These techniques include [8, 7, 5, 3, 10].

We note, however, that there lacks rigorous security analysis for the first two types of techniques. In particular, it is

not clear exactly how difficult it is to break these schemes once the hash values (or the transformed templates) are compromised, especially when the hash function, transformation algorithm and related keys and parameters are also compromised. Therefore, we follow the third approach and use secure sketch schemes to construct secure biometric templates.

Another important issue in biometric authentication systems is that biometrics are usually of lower entropy compared with modern standard of cryptographic keys. Therefore, it is interesting to investigate *fusion* techniques that combine two or more different biometrics for authentication. These different biometrics can be the result from different sensors, multiple samples of the same biometrics, different feature representations, or multi-modalities. In this paper, we only study the case of multi-modalities.

We observe that many existing fusion techniques (e.g., see [6] and references therein) treat different modalities as independent statistical tests, and combine the results of the tests using various methods. For example, one can assign different scores to give different weights to different tests, and set a threshold on the final score. In the simplest form, these tests are combined with logical *and* and *or* operations. However, it is noted in [6] that fusion at the *feature* level is usually difficult. One of the reasons is that different biometrics, especially in the multi-modality case, would have different feature representations and different similarity measures.

In this paper, we investigate how to combine the features extracted from different modalities, and construct secure templates from the combined features. In particular, we consider minutiae-based fingerprint features and singular value decomposition (SVD) based face features.

There has been some work on how to construct secure sketch for minutiae [3] and SVD face features [10, 17]. Furthermore, there has been some new minutiae-based fingerprint features based on geometric transformations, such as transforming minutiae to points on a circle [18]. It is interesting that after this transformation, fingerprint minutiae can be represented in the same way as SVD coefficients (i.e., an ordered list of components), with unified similarity measures (i.e., each component must be within a certain range to be considered as authentic). Therefore, these features can be combined easily after the transformation. Furthermore, we can extend the construction of secure sketch for SVD coefficients to the combined features to create a secure template for matching or authentication. We call such a template *finger-face* template.

To evaluate the performance and security of the proposed scheme, we use real world biometric data from publicly available sources. In particular, the fingerprint minutiae are taken from the ideal tenprint fingerprints in the NIST special database 27 [11]. These minutiae are validated by human

experts according to the NIST manual, hence representing the *true* distribution of minutiae to certain extent. The face images are taken from the Essex Faces94 database for facial recognition research [16]. These images are of reasonably high quality with limited variations, hence can be considered as samples of faces in controlled environment (as can be expected in many biometric authentication scenarios).

We describe the proposed scheme in detail in Section 2, and evaluate the performance and security of the scheme in Section 3. We conclude in Section 4.

## 2. Proposed Scheme

### 2.1. Minutiae-Based Fingerprint Features

Many existing fingerprint authentication systems are based on minutiae, which are feature points extracted from a raw fingerprint image. Now, suppose each set of minutiae  $F$  consists of  $m$  points ( $m$  can be different for different fingerprints). That is,  $F_r = \{(x_1, y_1), \dots, (x_m, y_m)\}$ , where each  $(x_i, y_i)$  is a point in some 2-D space, which is determined by the resolution and orientation of the fingerprint image. We choose a global parameter  $R$  that is not too large such that all minutiae of all the users can be contained in a circle with radius  $R$ . Now, we apply the geometric transformation proposed in [18] on  $F_r$  and obtain a feature vector  $F$  as below.

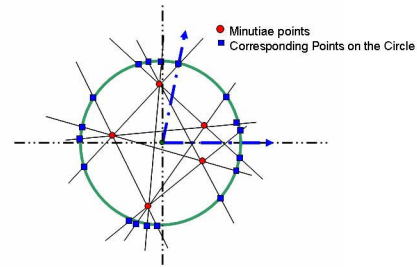


Figure 2. Geometric Transformation of Minutiae [18]

#### TRANSFORM-MINUTIAE

1. Compute the centroid  $C = (x_c, y_c)$  of the minutiae  $F_r$  as the center of mass of all the minutiae. That is,

$$x_c = \frac{1}{m} \sum_{i=1}^m x_i, \text{ and } y_c = \frac{1}{m} \sum_{i=1}^m y_i. \quad (1)$$

2. For every pair of minutiae, if the distance between them is more than a threshold  $T$ , draw a straight line passing through these two points, and mark its intersection with the circle with center  $C$  and radius  $R$ .
3. Partition (quantize) the intersection points obtained in the previous step. In particular, we divide the circle into a list of ordered arcs of  $\Delta$  degrees each. Let the

arcs be  $a_1, \dots, a_k$ , where  $k = 360/\Delta$ . Let the vector  $V = (v_1, \dots, v_k)$  represent the list of number of intersection points on these arcs.

4. Assume that there is a random  $k$  by  $k$  matrix  $M_r$  (called *randomization matrix*) associated with each user. Compute  $VM_r = (z_{k+1}, \dots, z_n)$ .
5. The final feature is the vector  $F = (z_1, \dots, z_k)$ .

Figure 2 illustrates a simple example with 5 minutiae points and  $\Delta = 90$ . Let the feature for the  $i$ -th user is  $F_i^{(1)} = (z_1, \dots, z_k)$ . Given another fingerprint, the same extraction algorithm is followed. Let the result be  $F' = (w_1, \dots, w_k)$ . The fingerprint is considered as authentic if for each  $1 \leq j \leq k$ ,  $z_j - \delta_{i,j} \leq w_j \leq z_j + \delta_{i,j}$ , where  $\delta_{i,j}$  is associated with the  $j$ -th component of the  $i$ -th user.

We note that this transformation is not rotation-invariant. It is possible to use additional information to align two minutiae (e.g., using cores and deltas), or we can simply try a few different rotations for matching. In this paper we assume that the fingerprints are already aligned.

We also note that the purpose of the randomization is to reduce the penalty caused by outliers, which is similar to that in [2]. However, their technique cannot be easily applied since we do not have the original data (not even in the encrypted form) during authentication.

## 2.2. SVD-Based Face Features

We employ the feature extraction scheme in [17]. In particular, during enrollment, given a face image, we follow the following steps to generate a feature vector of size  $n - k$ .

### EXTRACT-FACE-FEATURE

1. Compute the singular value decomposition (SVD) of the given face image, and take the  $n - k$  most significant components. Let the result be the vector  $F_s = (v_{k+1}, \dots, v_n)$ .
2. Similar to the fingerprint features, assume that there is another random  $n - k$  by  $n - k$  randomization matrix  $M_s$  associated with each user. Compute  $F_s M_s = (u_{k+1}, \dots, u_n)$ .
3. Quantize the above feature vector. In particular, we choose a scalar quantizer with step size  $\lambda_j$  for the component  $u_j$ , and each  $\lambda_i$  is selected according to the variation of  $u_j$  over the entire population. Let the final discrete feature for the  $i$ -th user be  $F_i^{(2)} = (z_{k+1}, \dots, z_n)$ .

During authentication, given another face image, we can extract its feature in the same way and get another feature vector  $F' = (w_{k+1}, \dots, w_n)$ . This vector is considered as

similar to  $F_i^{(2)}$  if and only if  $z_j - \delta_{i,j} \leq w_j \leq z_j + \delta_{i,j}$  for all  $k+1 \leq j \leq n$ , where  $\delta_{i,j}$  is some parameter for the  $j$ -th component of the  $i$ -th user.

## 2.3. Combined Feature and Secure Sketch

After we generate the features from both minutiae and the face image of the same person (say, the  $i$ -th user), we can combine the two vectors  $F_i^{(1)}$  and  $F_i^{(2)}$  into one, with total number of  $n$  component. That is, the feature vector for the  $i$ -th user is  $F_i = (z_1, \dots, z_n)$ , where another feature vector  $F' = (w_1, \dots, w_n)$  is considered as similar to  $F_i$  if and only if  $z_j - \delta_{i,j} \leq w_j \leq z_j + \delta_{i,j}$  for all  $1 \leq j \leq n$ .

Next, we follow the scheme in [10] and [17] to generate a sketch for the combined feature. In particular, given a feature vector  $F_i = (z_{i,1}, \dots, z_{i,n})$  for the  $i$ -th user, we follow the following steps for each component  $z_j$  for each  $1 \leq j \leq n$ .

### SKETCH GENERATION

1. Construct a 1-D codebook  $\mathcal{C}_{i,j}$ , where each codeword is in the form of  $c = k(2\delta_{i,j} + 1)$  for some integer  $k$ .
2. Find the codeword  $c_{i,j} \in \mathcal{C}_{i,j}$  that is nearest to  $z_{i,j}$ .
3. Compute  $p_{i,j} = z_j - c_{i,j}$ .

The final sketch for the  $i$ -th user is  $P_i = (p_{i,1}, \dots, p_{i,n})$ . Note that for each  $p_{i,j}$ , we have  $-\delta_{i,j} \leq p_{i,j} \leq \delta_{i,j}$ , or  $|p_{i,j}| = \log(2\delta_{i,j} + 1)$ .

## 2.4. Finger-Face Template and Verification

The final finger-face template for the  $i$ -th user consists of the sketch  $P_i$ , the randomization matrix  $M_i$  for this user, the codebooks  $\mathcal{C}_{i,j}$  for all  $1 \leq j \leq n$  (or equivalently, the threshold values  $\delta_{i,j}$ ), and  $\mathcal{H}(F_i)$ , where  $\mathcal{H}$  is a cryptographically secure one-way function. Global parameters required for verification include  $R$ ,  $\Delta$  and  $T$  used in minutiae-based feature, and the one-way function  $\mathcal{H}$ .

Given new biometric samples of a fingerprint and a face image from a user claimed to be the  $i$ -th user, the same feature extraction algorithms as in Section 2.1 and 2.2 are followed, and some feature vector  $F'$  is extracted. If  $F'$  is similar to  $F_i$ , then by using the sketch  $P_i$  and codebook  $\mathcal{C}_{i,j}$ , we can recover the original feature vector  $F_i$ , and compute  $\mathcal{H}(F_i)$ , which is then matched against the stored value  $\mathcal{H}(F_i)$ .

## 2.5. Security Measure

In practice, the templates can be encoded and stored separately for additional security. For example, one can employ the multi-factor scheme as in [17] and store only some encoded information on a smartcard and some other encoded information in a central server (or another device the user

needs to access), such that if only one of the factors (smart-card or device/server) is compromised, no information is leaked, and we can still obtain certain level of computational security if both factors are compromised. Nevertheless, in this paper, we focus on the case where all the template information is captured by the attacker.

If an attacker (say, Bob) obtains all the template data of a user, he would try to *invert* the template and find some forged biometric data that would generate the same value  $\mathcal{H}(F_i)$  during authentication. Since the process of sketch generation and recovery is easily invertible, the main difficulty for the attacker would be to invert the one-way function  $\mathcal{H}$  with additional information (i.e., other parts of the templates). Hence, we require  $\mathcal{H}$  to be hard to invert even with partial information about the pre-image.

In addition, the security depends on how difficult it is to guess the original feature vector given that some amount of information is leaked by the compromised sketch. Other parts of the template may also reveal some information about the original feature vector. However, we note that the global parameters (i.e.,  $R$  and  $\Delta$ ) reveals very little information, and the codebooks (or the  $\delta_{i,j}$ 's) do not reveal more information than the sketch itself<sup>1</sup>.

To measure the information leakage, we follow the formal approach in [5, 10], and examine the *entropy loss*. First, we define the min-entropy  $\mathbf{H}_\infty(A)$  of a discrete random variable  $A$  as  $\mathbf{H}_\infty(A) = -\log(\max_a \Pr[A = a])$ . For two discrete random variables  $A$  and  $B$ , the average min-entropy of  $A$  given  $B$  is defined as  $\tilde{\mathbf{H}}_\infty(A | B) = -\log(\mathbb{E}_{b \leftarrow B}[2^{-\mathbf{H}_\infty(A|B=b)}])$ . Now, for the feature vector  $F_i$ , the entropy loss of the sketch  $P_i$  is defined as  $\mathcal{L} = \mathbf{H}_\infty(F_i) - \tilde{\mathbf{H}}_\infty(F_i | P_i)$ . As noted in [5], the entropy loss can be conveniently bounded by the size of the sketch. Hence, in our scheme, the entropy loss is bounded by

$$\begin{aligned} \mathcal{L} &= \frac{1}{N} \sum_{i=1}^N \left( \mathbf{H}_\infty(F_i) - \tilde{\mathbf{H}}_\infty(F_i | P_i) \right) \\ &\leq \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^n |p_{i,j}| \end{aligned} \quad (2)$$

where  $N$  is the total number of users.

Finally, we need to determine the min-entropy of the original feature vector. Since there is no known model for biometrics, it would be infeasible to determine the min-entropy of the original data experimentally, due to the large feature space. Therefore, we measure the min-entropy approximately by assuming that each component is independent, measuring the min-entropy of each component as induced by the data set, and computing the sum of the min-

<sup>1</sup>Basically  $\delta_{i,j}$ 's can be implied by the number of bits required to encode the sketch, which should be specified by the description of the sketch anyway.

entropy of all components. The same approach is also used in [17].

### 3. Evaluation

#### 3.1. Data Sets

**Fingerprints and Minutiae Points** In this paper, we use the fingerprints and minutiae points from the NIST Special Database 27 [11]. This database contains 258 pairs of matching fingerprints, among which 152 are chosen for the experiments, which matches the number of subjects in the face database. Each pair consists of a *tenprint* fingerprint, which is taken in a controlled environment and is of high quality, and a matching *latent* fingerprint of lower quality, taken from crime scenes. Each pair of fingerprints is also associated with four sets of minutiae. Two of them are the minutiae extracted from the tenprint and latent fingerprints respectively, and the other two are the matching minutiae of the first two sets of minutiae. All the minutiae are validated (and some of them are marked) by human experts, hence false-minutiae are unlikely.

In this paper, we use the minutiae extracted from the tenprint fingerprints as the original fingerprints. Although the tenprint fingerprints and minutiae are of high quality, there is only one matching latent fingerprint for each tenprint fingerprint, and the latent fingerprints are mostly partial, since they are lifted from crime scenes. Therefore, we choose to apply synthesized noise on the tenprint minutiae sets to generate training and test data. In particular, we consider the combination of two types of noise: (1) White noise, which moves each minutia up to  $\lambda_w$  pixels, and (2) replacement noise, which removes some amount of the minutiae and injects some new ones.

The white noise is synthesized according to the difference between the matched tenprint minutiae and matched latent minutiae in the database. In particular, we examine the matched minutiae, and find the histogram of the distances, and generate the white noise according to the histogram. The replacement noise is synthesized according to the difference between ideal latent minutiae and matched latent minutiae. That is, we record the percentage of the minutiae that are in the ideal latent minutiae but not in the matched latent minutiae for every pair of fingerprints with quality labeled as *good* (there are 88 such pairs), and estimate the distribution<sup>2</sup>. In the experiments, we observe that the distribution of the percentile is very close to a Normal distribution. Hence we use a Normal distribution with zero mean and variance of 0.09 (i.e., 9%) for the replacement

<sup>2</sup>It should be noted that the NIST database contains a small number of pairs where the matched latent minutiae is not a subset of the ideal latent minutiae, which is probably due to some manual adjusting for the matching. In our experiments, we exclude those pairs from the data set when estimating the noise.

noise.

Finally, we employ the method in [18] and transform each of the minutiae set into points on a circle around it with radius  $R = 4000$  and threshold  $T = 100$ . We choose  $R$  to be approximately double the largest distance between any two minutiae in the dataset, and our result is not sensitive to the actual value of  $R$ . The transformed points are then quantized and randomized (Section 2.1). In total, 12 training and 8 testing minutiae sets are generated by adding the synthesized noise to the ideal tenprint minutiae.

**Face Images** The face images are taken from the Essex Faces94 database [16] for face recognition research. There are 20 still photos of each of the 152 subjects, where the variation is relatively small compared to other publicly available face image databases. This is reasonable in many authentication scenarios where the photos are taken in a controlled environment and certain level of user cooperation can be expected.

We divide the 20 images for each subject into two sets: 12 of them are used for training and estimate the parameters for the authentication scheme, and the remaining 8 images are used for testing. In particular, for each subject we compute the SVD for all the training images, take the first 20 components (as suggested in [17]), randomize them using a randomization matrix associated with this identity, and quantize the results to get the final feature samples. From those samples, similar to the fingerprint features, for each of the component we find the mean, the minimum and maximum values, and use these values to form the template.

### 3.2. Performance Analysis

The performance of the scheme is measured by the false accept rate (FAR) and false reject rate (FRR). In this paper, since we use a rather simple method to combine the features, a user is considered as authentic if and only if both the face and fingerprint features match the template. Hence, our scheme is equivalent to a fusion scheme that takes the logical and of the two tests in terms of performance. For simplicity, we conduct experiments on the face and fingerprint data independently, and combine the results to obtain the performance of the resulting system.

For each modality, we have a template and a few test samples for each user. We define FAR as the ratio of the number of test samples that do not belong to a user yet are considered as authentic by the scheme over the total number of test samples. FRR is defined in a similar way.

Figure 3 illustrates the ROC curves of the fingerprint and face features. The ROC curves are obtained by varying the detection parameters. The parameters  $z_{i,j}$  and  $\delta_{i,j}$  (for  $1 \leq j \leq k$ ) are determined from the training data. In particular,  $z_{i,j}$ 's are taken as the mean of the corresponding

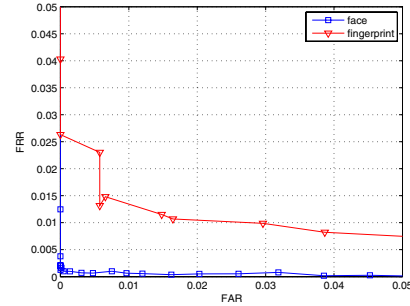


Figure 3. ROC Curve for Fingerprint and Face Features.

components, and  $\delta_{i,j}$ 's are based on the minimum and maximum values of the corresponding components. By varying the values of  $\delta_{i,j}$ 's, we can obtain the operation characteristics at different points, which are then used to plot the ROC curves. In these experiments, we use  $\Delta = 18$ , which means that there are  $360/\Delta = 20$  components in the feature vector for a fingerprint. Similar ROC curve can be obtained for the face features, which is similar to that in [17].

Since we assume that fingerprints and face features are independent, overall performance of the combined system can be obtained by examining relevant points from the ROC's of the individual modalities. For example, we can pick one point from each ROC curve in Figure 3 such that the FRR value for fingerprints is about 1.3% and that for face features is about 0.10%. In this case, the overall FRR in this case would be about 1.4%. Also, in this case the FAR for the fingerprint features is about 0.58%, but the FAR for the face features is 0.05%, hence the overall FAR is  $2.9 \times 10^{-6}$ . We can further bring down the FRR with a little trade-off from FAR. For example, we can choose a point on the fingerprint ROC curve such that the FRR is 0.82% with FAR 3.9%, and the same point for face biometrics (with FRR 0.10% and FAR 0.05%). In this case, the overall FRR would be 0.92% and the overall FAR would be  $1.9 \times 10^{-5}$ .

### 3.3. Security Analysis

We employ the same method as in [17] to evaluate the security of the scheme. In particular, we first estimate the min-entropy of the original combined feature, and then compute the bound on the information leakage (i.e., entropy loss) from the size of the sketch. The difference between these two quantities would be the strength of the scheme, assuming that the hash function is secure.

To estimate the min-entropy in high dimensions, we assume that the individual components of the feature before randomization are independent, and compute the min-entropy induced by the data sets. On the other hand, the entropy loss is simply taken as the size of the sketch, averaged over all users in the system.

For the data sets used in our experiments, our estimation

of the min-entropy of the face features is 85 bits, and that of the fingerprints is 58 bits. The size of the sketch for the face features is 49 bits in average, and that for fingerprints is 55 bits. Therefore, the estimated lower bound of the strength of the scheme is about 39 bits. Note that the entropy loss is the upper bound of the actual information leakage. Hence the exact security of the system could be better.

## 4. Conclusions

In this paper, we study the fusion of fingerprint and face biometrics in the feature level, and the construction of secure templates that would not give attackers much advantage even when they are compromised.

We employ a known geometric transformation for fingerprint minutiae to transform the minutiae sets to feature vectors of fixed lengths. We also use a known face feature extraction algorithm that makes use of SVD values of the face images, which is also of fixed lengths. In this way, the features for both modalities would have the same representations that makes fusion at feature level much easier.

Although our fusion technique is relatively simple, which is equivalent to a logical and of independent tests of fingerprint and face biometrics, it nevertheless shows possibilities for much more complicated operations that can be performed over the combined biometric features before doing classification or authentication. Our construction of secure templates is based on known secure sketch schemes, such that it allows rigorous yet easy security analysis based on the size of the sketch.

A major open problem is the precise measure of min-entropy of the original features, due to the high dimensional space and limited data. We assume that each component of the original data is independent and estimate the min-entropy by analyzing each component experimentally. Another important open problem is how to determine the exact information leakage due to the sketch.

## References

- [1] R. Ang, R. Safavi-Naini, and L. McAven. Cancelable key-based fingerprint templates. In *ACISP*, volume 3574 of *LNCS*, pages 242–252, 2005. **1**
- [2] T. Boulton. Robust distance measures for face-recognition supporting revocable biometric tokens. In *IEEE, 7th Intl. Conf. on Automatic Face and Gesture Recognition*, pages 560–566, 2006. **3**
- [3] E.-C. Chang and Q. Li. Hiding secret points amidst chaff. In *Eurocrypt*, volume 4004 of *LNCS*, pages 59–72, St. Petersburg, Russia, May 2006. Springer Verlag. **1, 2**
- [4] T. Connie, A. Teoh, M. Goh, and D. Ngo. Palmhashing: a novel approach for cancelable biometrics. *Information Processing Letters*, 93:614–634, 2005. **1**
- [5] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Eurocrypt*, volume 3027 of *LNCS*, pages 523–540. Springer-Verlag, 2004. **1, 4**
- [6] A. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *Pattern Recognition*, 38(12):2270–2285, 2005. **2**
- [7] A. Juels and M. Sudan. A fuzzy vault scheme. In *IEEE Intl. Symp. on Information Theory*, 2002. **1**
- [8] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proc. ACM Conf. on Computer and Communications Security*, pages 28–36, 1999. **1**
- [9] T. Kevenaar, G. Schrijen, M. V. der Veen, A. Akkermans, and F. Zuo. Face recognition with renewable and privacy preserving binary templates. *Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pages 21–26, 2005. **1**
- [10] Q. Li, Y. Sutcu, and N. Memon. Secure sketch for biometric templates. In *Asiacrypt*, Shanghai, China, December 2006. **1, 2, 3, 4**
- [11] NIST Special Database 27: Fingerprint minutiae from latent and matching tenprint images. <http://www.nist.gov/srd/niststd27.htm>. **2, 4**
- [12] N. Ratha, J. Connell, and R. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001. **1**
- [13] N. Ratha, J. Connell, R. Bolle, and S. Chikkerur. Cancelable biometrics: A case study in fingerprints. In *Intl. Conf. on Pattern Recognition*, pages 370–373, 2006. **1**
- [14] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):561–572, 2007. **1**
- [15] A. Ross, J. Shah, and A. K. Jain. From template to image: Reconstructing fingerprints from minutiae points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):544–560, 2007. **1**
- [16] L. Spacek. The Essex Faces94 database. <http://cswww.essex.ac.uk/mv/allfaces/>. **2, 5**
- [17] Y. Sutcu, Q. Li, and N. Memon. Protecting biometric templates with sketch: Theory and practice. *IEEE Transactions on Information Forensics and Security*, 2007. To appear. **2, 3, 4, 5**
- [18] Y. Sutcu, H. T. Sencar, and N. Memon. A geometric transformation to protect minutiae-based fingerprint templates. In *SPIE International Defense and Security Symposium*, 2007. **2, 5**
- [19] Y. Sutcu, T. Sencar, and N. Memon. A secure biometric authentication scheme based on robust hashing. In *ACM MMSEC Workshop*, 2005. **1**
- [20] A. Teoh, D. Ngo, and A. Goh. Personalised cryptographic key generation based on facehashing. *Computers and Security*, 23:606–614, 2004. **1**
- [21] C. Vielhauer, R. Steinmetz, and A. Mayerhoefer. Biometric hash based on statistical features of online signatures. *IEEE International Conference on Pattern Recognition (ICPR)*, 2002. **1**