To Appear in Security and Privacy in Biometrics, P. Campisi (ed.), Springer, 2012

Fingerprint Template Protection: From Theory to Practice

Anil K. Jain, Karthik Nandakumar and Abhishek Nagar

Abstract One of the potential vulnerabilities in a biometric system is the leakage of biometric template information, which may lead to serious security and privacy threats. Most of the available template protection techniques fail to meet all the desired requirements of a practical biometric system like revocability, security, privacy, and high matching accuracy. In particular, protecting the fingerprint templates has been a difficult problem due to large intra-user variations (e.g., rotation, translation, nonlinear deformation, and partial prints). There are two fundamental challenges in any fingerprint template protection scheme. First, we need to select an appropriate representation scheme that captures most of the discriminatory information, but is sufficiently invariant to changes in finger placement and can be secured using available template protection algorithms. Secondly, we need to automatically align or register the fingerprints obtained during enrollment and matching without using any information that could reveal the features, which uniquely characterize a fingerprint. This chapter analyzes how these two challenges are being addressed in practice and how the design choices affect the trade-off between the security and matching accuracy. Though much progress has been made over the last decade, we believe that fingerprint template protection algorithms are still not sufficiently robust to be incorporated into practical fingerprint recognition systems.

Anil K. Jain
Department of Computer Science & Engineering, Michigan State University, East Lansing, MI 48824, USA and Department of Brain & Cognitive Engineering, Korea University, Seoul. e-mail: jain@cse.msu.edu,

Karthik Nandakumar
Institute for Infocomm Research, A*STAR, Fusionopolis, Singapore. e-mail: knandakumar@i2r.a-star.edu.sg,

Abhishek Nagar
Department of Computer Science & Engineering, Michigan State University, East Lansing, MI 48824, USA. e-mail: nagarabh@cse.msu.edu

1 Introduction

The primary purpose of using a biometric system is to provide non-repudiable authentication. Authentication implies that (i) only legitimate or authorized users are able to access the physical or logical resources protected by the biometric system and (ii) impostors are prevented from accessing the protected resources. Non-repudiation ensures that an individual who accesses a certain resource cannot later deny using it. From the perspective of the users, there are two main requirements that a biometric system must meet. Firstly, the legitimate users must have timely and reliable access to the protected resource/service. Secondly, the biometric system and the personal data stored in it must be used only for the intended functionality, which is to control access to a specific resource and not for other unintended purposes. However, attacks by adversaries may prevent the biometric system from satisfying the above functionalities and requirements.

While a biometric system can be compromised in a number of ways, one of the potentially damaging attacks is the leakage of biometric template information. The leakage of this template information to unauthorized individuals constitutes a serious security and privacy threat due to the following two reasons:

1. **Intrusion attack**: If an attacker can hack into a biometric database, he can easily obtain the stored biometric information of a user. This information can be used to gain unauthorized access to the system by either reverse engineering the template to create a physical spoof or replaying the stolen template. For example, it has been shown that fingerprint images can be reconstructed from minutiae templates (see Figure 1), which may in turn be used to construct a spoof [44, 7, 18].
2. **Function creep**: An adversary can exploit the biometric template information for unintended purposes (e.g., covertly track a user across different applications by cross-matching the templates from the associated databases), compromising user privacy.

Fig. 1 Reconstruction of a fingerprint image from the minutiae template [18].

Due to these reasons, biometric templates (or the raw biometric images) should not be stored in plaintext form and fool-proof techniques are required to securely store the templates such that both the security of the application and the users' privacy are not compromised by adversary attacks. The fundamental challenge in designing a biometric template protection scheme is to overcome the large intra-user variability among multiple acquisitions of the same biometric trait (see Figure 2).

Fig. 2 Illustration of fingerprint intra-class variability. Two different impressions of the same finger with differences in the number and location of minutiae are shown. Among the 33 and 26 minutiae in the left and right images, respectively, only 16 minutiae match and some of these matches are marked.

1.1 Biometric Template Security Requirements

A biometric template protection scheme should have the following three properties.

1. **Cryptographic security**: Given a secure template, it must be computationally difficult to find a biometric feature set (commonly known as a *pre-image*) that will match with the secure template. This pre-image resistant property defends against the possibility of an attacker intruding into the biometric system under consideration by replaying the pre-image.

   The concept of pre-image resistance is also related to *one-way* or *non-invertible* mathematical functions. A function $\mathbf{f}$ is referred to as a one-way function if it is "easy to compute" (in polynomial time) but "hard to invert" (given $\mathbf{f}(\mathbf{x})$, the probability of finding $\mathbf{x}$ in polynomial-time is small). A non-invertible template protection scheme implies that it will be computationally hard to obtain the original biometric features from the secure template. This prevents an adversary from creating a physical spoof of the biometric trait and intruding an-

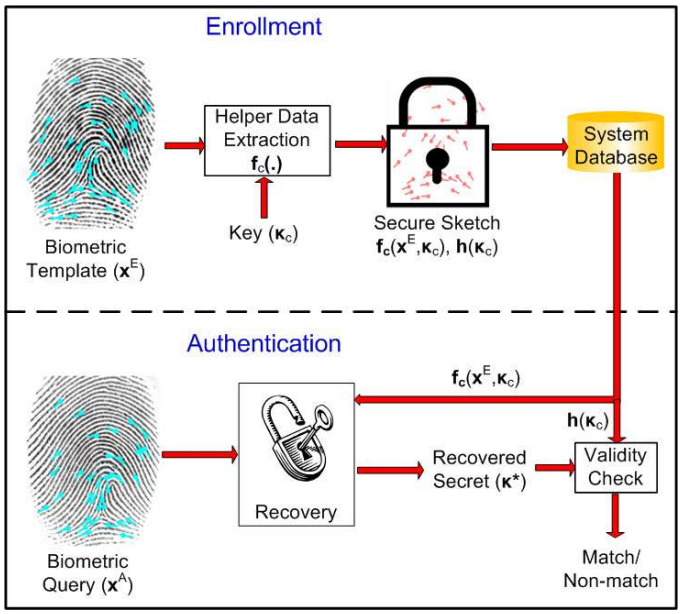
other biometric system that makes use of the same biometric trait. Thus, a secure template must be pre-image resistant and non-invertible.
2. **Performance**: The biometric template protection scheme should not degrade the recognition performance (False Match Rate (FMR) and False Non-Match Rate (FNMR)) of the biometric system.
3. **Revocability**: It is desirable to have a template protection scheme that can generate multiple secure templates from the same biometric data. These multiple secure templates must be such that even if an adversary obtains two or more of them, it must be computationally hard to: (i) identify that they are derived from the same biometric data, and (ii) obtain the original biometric features of the user. This revocability or cancelability property ensures that cross-matching across biometric databases is not possible, thereby preserving the user's privacy. Revocability also makes it straightforward to discard a compromised template and reissue a new one based on the same biometric data.

Ideally, the template protection scheme should satisfy all the three requirements simultaneously. However, it is quite a challenge to design such a technique. The simplest way to secure biometric templates is to encrypt them using standard cryptographic techniques like RSA and AES. This is the methodology deployed in most of the existing commercial biometric systems. However, it must be emphasized that multiple acquisitions of the same biometric trait do not result in the same feature set. Typically, standard encryption functions are not smooth functions and a small difference in the values of the feature sets extracted from the raw biometric data would lead to very large difference in the resulting encrypted features. Consequently, one cannot perform biometric matching directly in the encrypted domain. Rather, the template must be decrypted in order to be matched with the query features. As a result, the original biometric features are exposed during every authentication attempt, irrespective of whether the authentication is eventually successful. Therefore, the encryption solution is secure and revocable only under ideal conditions (key is kept secret and matching is done at a trusted location). If practical issues such as key management or susceptibility to template theft during a matching attempt are taken into account, the standard encryption technique is not good enough for securing biometric templates.

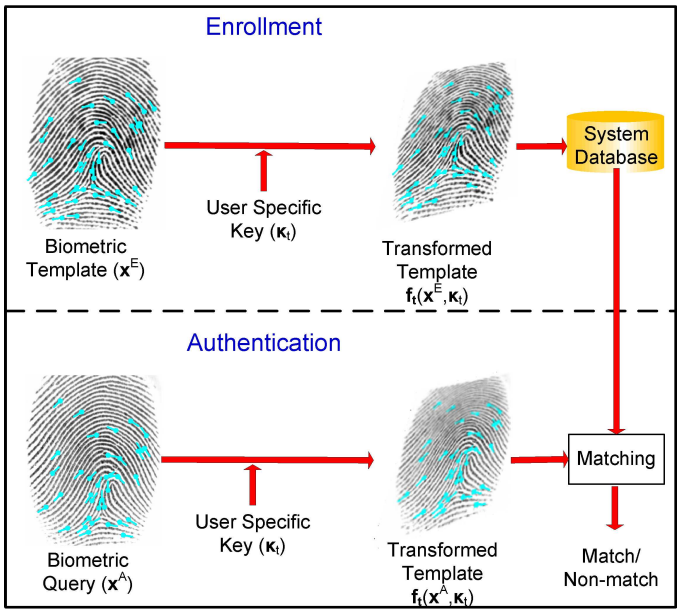
1.2 Biometric Template Protection Approaches

To overcome the limitations of the standard encryption approach, a number of techniques have been proposed to secure biometric templates (see [21] for a detailed review). These techniques can be categorized into two main classes (see Figure 3):

- **Biometric cryptosystems**: In a biometric cryptosystem, secure sketch ($\mathbf{y}_c$) is derived from the enrolled biometric template ($\mathbf{x}^E$) and stored in the system database instead of the original template. In the absence of the genuine user's biometric data, it must be computationally hard to reconstruct the template from the sketch.



(a)



(b)

Fig. 3 Biometric template protection based on (a) biometric cryptosystem and (b) template transformation.

On the other hand, given an authentication query ($\mathbf{x}^A$) that is *sufficiently close* to the enrolled template ($\mathbf{x}^E$), it should be easy to decode the sketch and recover the template. Typically, the sketch is obtained by binding the template with a codeword from an error correcting code, where the codeword itself is defined by a key (κ_c). Therefore, the sketch ($\mathbf{y}_c$) can be written as $\mathbf{f}_c(\mathbf{x}^E, \kappa_c)$, where $\mathbf{f}_c$ is the sketch generation function. The error correction mechanism facilitates the recovery of the original template and hence, the associated key. Thus, a biometric cryptosystem not only secures the biometric template, but also facilitates secure key management, which is one of the challenging issues in cryptographic systems. Examples of biometric cryptosystems include fuzzy vault [26], fuzzy commitment [27], PinSketch [14], and secret-sharing approaches [20].

- **Template transformation**: Template transformation techniques modify the template ($\mathbf{x}^E$) with a user specific key (κ_t) such that it is difficult to recover the original template from the transformed template ($\mathbf{y}_t$). During authentication, the same transformation is applied to the biometric query ($\mathbf{x}^A$) and the matching is performed in the transformed domain to avoid exposure of the original biometric template. Since the key κ_t needs to be stored in the system along with $\mathbf{y}_t$, the template security is guaranteed only if the transformation function is non-invertible even when κ_t is known to the attacker. Some well-known examples of template transformation include Bio-Hashing [49] and cancelable biometrics [42].

Different combinations of the above two basic approaches, called hybrid biometric cryptosystems, have also been proposed [45, 37]. The template protection schemes described above have their own advantages and limitations in terms of template security, computational cost, storage requirements, applicability to different kinds of biometric representations and ability to handle intra-class variations in biometric data [53].

In this chapter, we will focus on the practical issues involved in applying the available template protection algorithms to secure fingerprint templates. Features representing fingerprint images may exhibit intra-user variations due to various factors like rotation, translation, nonlinear deformation, and partial overlap between multiple impressions of the same finger. As a result, protecting fingerprint templates is a challenging task. Fingerprint recognition is typically based on the location and orientation of minutia points, which represent ridge endings or ridge bifurcations [31]. Minutia sets are unordered and there may be variations (see Figure 2) in the number and location of minutia points due to intra-user variations. The similarity between two fingerprints is measured based on the number of minutia correspondences. Furthermore, the template and query minutia sets need to be aligned before the minutia correspondences can be found. Hence, there are two key challenges in securely matching fingerprints: (i) How to align query minutia set with template without leaking information about the original minutiae template? and (ii) Even after aligning the query and the template, the minutiae in the two sets will not match exactly in location and orientation due to nonlinear deformation (hence, a simple set difference metric may not be good enough). Finding a good representation scheme for fingerprints that can overcome the above problem is a challenge.

The rest of the chapter is organized as follows. Section 2 gives a brief overview of the major template protection algorithms that have been applied for securing fingerprint templates. Next, section 3 gives some examples of how the fingerprint features need to be adapted so that biometric cryptosystems can be applied to secure them. Section 4 describes the various approaches that have been proposed for aligning the query fingerprint with the secure template. The matching performance and security of the state-of-the-art fingerprint template protection schemes are discussed in section 5. Finally, our conclusions and pointers for future research are highlighted in section 6.

2 Fingerprint Template Protection Schemes

Depending on the features used for recognition, existing solutions for fingerprint template security can be categorized as minutiae-based or pattern-based approaches. Minutiae-based template protection schemes can be further classified into three types: (i) directly secure the unordered set representation of minutiae, (ii) secure a new set of unordered features derived from the minutiae (e.g., distances between pairs of minutiae), and (iii) secure a fixed-length feature vector derived from the minutiae. On the other hand, pattern-based schemes directly derive a fixed-length feature vector based on the global texture of the fingerprint pattern. When the representation to be secured is an unordered set, a *non-invertible template transformation* approach or a biometric cryptosystem called *fuzzy vault* can be used. When the representation is a fixed-length binary vector, a biometric cryptosystem called *fuzzy commitment* can be used to secure it. We will now discuss these three schemes in detail.

2.1 Non-Invertible Fingerprint Template Transformation

Ratha et al. [42] proposed and analyzed three non-invertible transforms for generating cancelable fingerprint templates. The three transformation functions are cartesian, polar and functional. These functions were used to transform fingerprint minutiae data such that a minutiae matcher can still be applied to the transformed minutiae. In cartesian transformation, the minutiae space (fingerprint image) is tessellated into a rectangular grid and each cell (possibly containing some minutiae) is shifted to a new position in the grid corresponding to the translations set by the user-specific key. The polar transformation is similar to cartesian transformation with the difference that the image is now tessellated into a number of shells and each shell is divided into sectors. Since the size of sectors can be different (sectors near the center are smaller than the ones far from the center), restrictions are placed on the translation vector generated from the key so that the radial distance of the transformed sector is not very different from the radial distance of the original position. Exam-

ples of minutiae before and after polar and cartesian transformations are shown in Figure 4.

For the functional transformation, Ratha et al. [42] used a mixture of 2D Gaussians and electric potential field in a 2D random charge distribution as a means to translate the minutiae points. The magnitude of these functions at the point corresponding to a minutia is used as a measure of the magnitude of the translation and the gradient of these functions is used to estimate the direction of translation of the minutiae. In all the three transforms, two or more minutiae can possibly map to the same point in the transformed domain. For example, in the cartesian transformation, two or more cells can be mapped onto a single cell so that even if an adversary knows the key and hence the transformation between cells, he cannot determine the original cell to which a minutia belongs because each minutiae can independently belong to one of the possible cells. This provides a limited amount of non-invertibility to the transform. Also since the transformations used are locally smooth, the error rates are not affected significantly and the discriminability of minutiae is preserved to a large extent. Note that the key to achieving good recognition performance is the availability of an alignment algorithm that can accurately pre-align (register) the fingerprint images or minutiae features prior to the transformation (e.g., based on core and delta points in the fingerprint).

2.2 Fingerprint Fuzzy Vault

Fuzzy vault is a cryptographic construct that is designed to work with biometric features represented as an unordered set (e.g., minutiae in fingerprints). The security of the fuzzy vault scheme is based on the computational difficulty in solving the polynomial reconstruction problem, which is a special case of the Reed-Solomon list decoding problem [2]. The fuzzy vault scheme works as follows (see Figure 5). Let $\mathbf{s}^E = \{x_1, x_2, ..., x_r\}$ denote a biometric template consisting of a set of r points from a finite field $\mathscr{F}$. In order to secure $\mathbf{s}^E$, a uniformly random cryptographic key κ_c of length L bits is generated and this key is transformed into a polynomial P of degree k ($k < r$) over $\mathscr{F}$. All the elements in $\mathbf{s}^E$ are then evaluated on this polynomial to obtain the set $\{P(x_i)\}_{i=1}^{r}$. The set of points $\{(x_i, P(x_i))\}_{i=1}^{r}$ is then secured by hiding them among a large set of q randomly generated chaff points $\{(a_j, b_j)\}_{j=1}^{q}$ that do not lie on the polynomial P (i.e., $b_j \neq P(a_j)$ and $a_j \notin \mathbf{s}^E$, $\forall\ j = 1, 2, \cdots, q$). The set of genuine and chaff points along with their polynomial evaluations constitute the sketch or vault $\mathbf{y}_c$. During authentication, if the query biometric set $\mathbf{s}^A$ is sufficiently close to $\mathbf{s}^E$, the polynomial P can be successfully reconstructed by identifying the genuine points in $\mathbf{y}_c$ that are associated with $\mathbf{s}^E$. Note that for successful reconstruction of P of degree k, a minimum of $(k+1)$ genuine points need to be identified from $\mathbf{y}_c$.

The three main parameters in the fuzzy vault scheme are r, q and k. The parameter r denotes the number of points in the vault that lie on the polynomial P and it depends on the number of features that can be extracted from the template (e.g.,

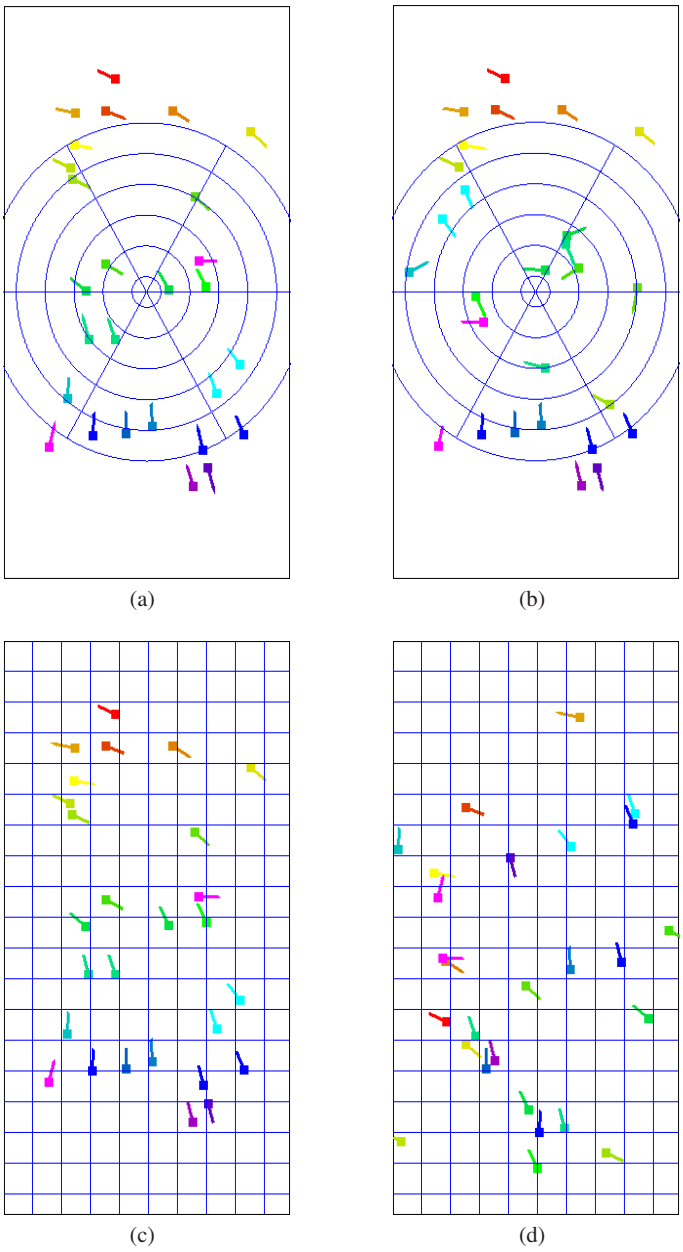


Fig. 4 Illustration of cartesian and polar transformation functions used in [42] for generating cancelable biometrics. (a) Original minutiae on radial grid, (b) transformed minutiae after polar transformation, (c) original minutiae on rectangular grid and (d) transformed minutiae after cartesian transformation.

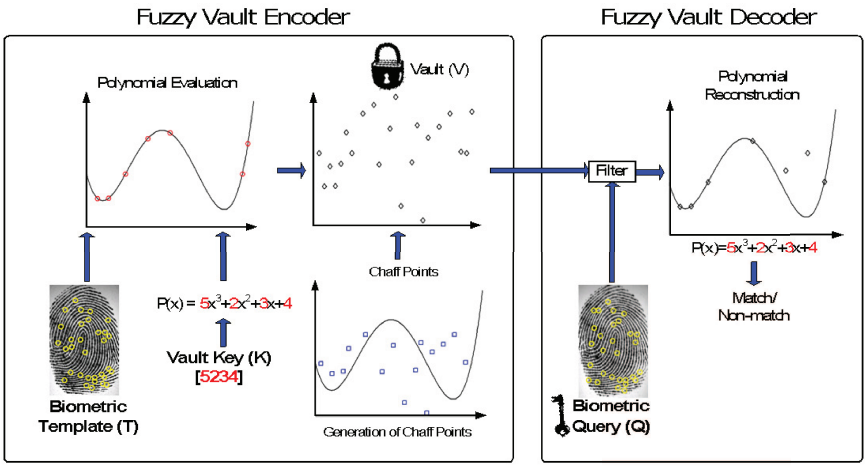


Fig. 5 Securing a fingerprint minutiae template using fuzzy vault.

number of minutia points in the user's fingerprint). The parameter q represents the number of chaff points that are added and this parameter influences the security of the vault. If no chaff points are added, the vault reveals the information about the template and the secret. As more chaff points are added, the security increases. Typically, the number of chaff points is an order of magnitude larger than the number of genuine points ($q \gg r$). Parameter k denotes the degree of the encoding polynomial and it controls the tolerance of the system to errors in the biometric data.

Since the introduction of the fuzzy vault scheme by Juels and Sudan, several researchers have attempted to implement it in practice for securing fingerprint minutiae templates. Clancy et al. [12] proposed a fuzzy vault scheme based on the location of minutia points (row and column indices in the image) in a fingerprint. They assumed that the template and query minutiae sets are pre-aligned, which is not a realistic assumption in practical fingerprint authentication systems. Further, multiple (four) fingerprint impressions of a user were used during enrollment for identifying the reliable minutia points. The error correction step was simulated without being actually implemented. The False Non-Match Rate of their system was approximately 20-30% and they claimed that retrieving the secret was 2^{69} times more difficult for an attacker than for a genuine user.

The fingerprint-based fuzzy vault proposed by Yang et al. [56] also used only the location information about the minutia points. Four impressions were used during enrollment to identify a reference minutia, and the relative position of the remaining minutia points with respect to the reference minutia was represented in the polar coordinate system. This scheme was evaluated on a small database of 10 fingers and a FNMR of 17% was reported. Chung et al. [11] proposed a geometric hashing technique to perform alignment in a minutiae-based fingerprint fuzzy vault. Uludag et al. [52] introduced a modification to the fuzzy vault scheme, which eliminated

the need for error correction coding. Uludag and Jain [51] also proposed the use of high curvature points derived from the fingerprint orientation field to automatically align the template and query minutiae sets (see Section 4 for details).

Nandakumar et al. [36] proposed a fuzzy vault framework that secures both minutiae locations and directions. During vault encoding a $(16 \times k)$ bit key (κ_c) is appended with a 16-bit Cyclic Redundancy Check (CRC) code and divided into $(k+1)$ blocks of 16 bits each. These $(k+1)$ values serve as the coefficients of a polynomial P of degree k in the Galois field $GF(2^{16})$. The template minutiae are sorted according to their quality and only well-separated minutiae [36] are selected for constructing the vault. If the desired number of minutiae (say r) cannot be obtained, it is counted as a Failure to Capture error (FTCR). The location and orientation of each minutia is encoded as an element in $GF(2^{16})$. Points with high ridge curvature are extracted from the fingerprint and stored along with the vault to be used for alignment during authentication.

During authentication, the high curvature points are used to align the template and query fingerprints. Then, r well separated and good quality minutiae are selected from the query and are coarsely matched with the points in the vault in order to filter out most of the chaff points. At this stage, a minutiae matcher [22] is applied to determine the corresponding pairs of minutiae in the filtered set of chaff points and the query minutiae set. To find the coefficients of a polynomial of degree k, $(k+1)$ unique projections are necessary. If the number of correspondences found is less than $(k+1)$, it results in authentication failure. Otherwise, all possible subsets of size $(k+1)$ of the obtained correspondences are selected and for each subset, a polynomial P^* is constructed using Lagrange interpolation. The coefficients of the polynomial P^* are 16-bit values which are concatenated to obtain a $16(k+1)$-bit string κ^* and CRC error detection is applied to κ^*. If an error is detected, it indicates that an incorrect key has been decoded and the same procedure is repeated for the next candidate subset. If no error is detected, it indicates that $\kappa^* = \kappa_c$ with very high probability.

2.3 Fingerprint Fuzzy Commitment

Fuzzy commitment [27] is a biometric cryptosystem that can be used to secure biometric traits represented in the form of binary vectors (see Figure 7). Suppose that the enrolled biometric template $\mathbf{b}^E$ is an N-bit binary string. In fuzzy commitment, a uniformly random key κ_c of length L ($L \leq N$) bits is generated and used to uniquely index a N-bit codeword $\mathbf{c}$ of an appropriate error correcting code. The sketch is then extracted from the template as $\mathbf{y}_c = \mathbf{c} \oplus \mathbf{b}^E$, where $\oplus$ indicates the modulo-2 addition. The sketch $\mathbf{y}_c$ is stored in the database along with $\mathbf{h}(\kappa_c)$, where $\mathbf{h}(.)$ is a cryptographic hash function. During authentication, the codeword is obtained from the query biometric $\mathbf{b}^A$ and the sketch $\mathbf{y}_c$ as follows: $\mathbf{c}^* = \mathbf{y}_c \oplus \mathbf{b}^A = \mathbf{c} \oplus (\mathbf{b}^E \oplus \mathbf{b}^A)$. This codeword $\mathbf{c}^*$, which is generally a corrupted version of the original codeword $\mathbf{c}$, can be decoded to get the key κ^*. The authentication is deemed successful if $\mathbf{h}(\kappa^*)$

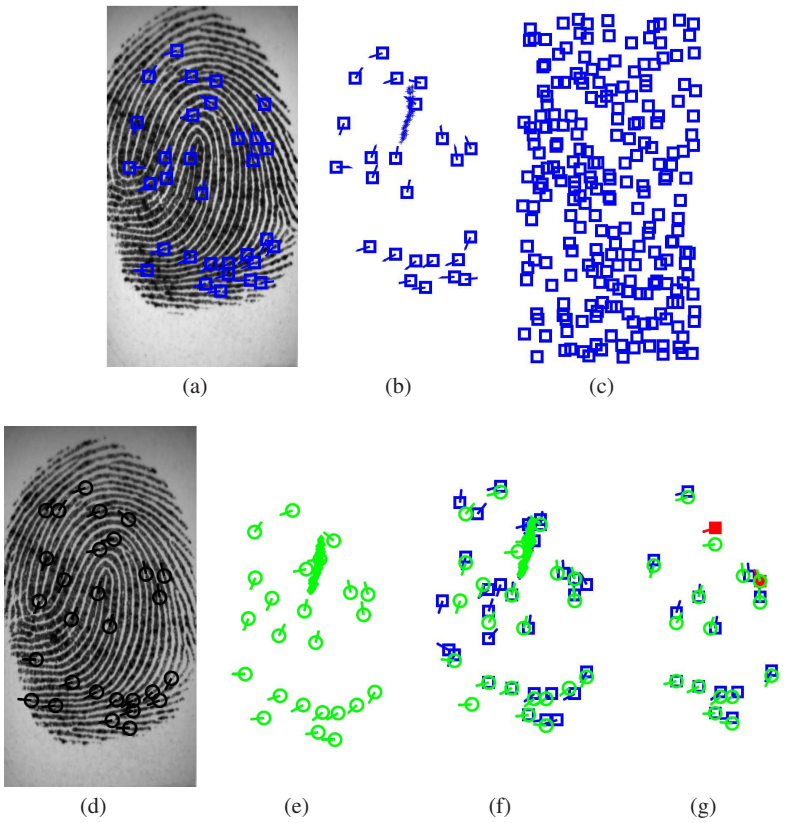


Fig. 6 An example of successful operation of the fingerprint fuzzy vault proposed in [36]. (a) Enrolled fingerprint image with minutiae template, (b) selected template minutiae and high curvature points extracted from the enrolled image, (c) vault in which the selected template minutiae are hidden among chaff points (for clarity, minutiae directions are not shown), (d) query fingerprint image with minutiae, (e) selected query minutiae and high curvature points extracted from the query image, (f) ICP alignment of template and query high curvature points and coarse filtering of chaff points, and (g) unlocking set obtained by applying a minutiae matcher which eliminates almost all the chaff points. The two points shown in filled squares in (g) are the only chaff points that remain in the unlocking set.

is the same as $\mathbf{h}(\kappa_c)$. If the Hamming distance between $\mathbf{b}^E$ and $\mathbf{b}^A$ is not greater than the error correcting capacity of the code, κ^* would be the same as κ and the matching will be successful.

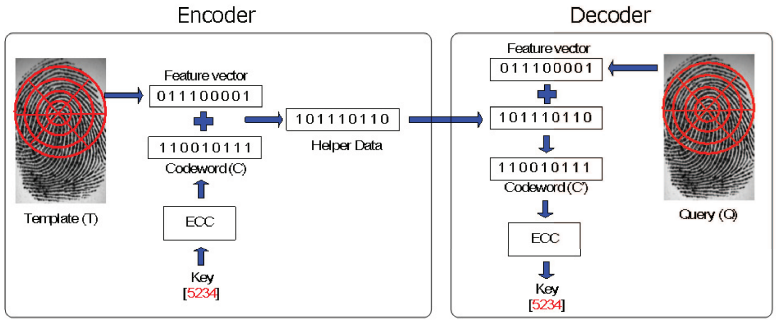


Fig. 7 Securing a fingerprint template using fuzzy commitment.

3 Adapting Fingerprint Representations for Cryptosystems

While minutiae-based schemes are widely used for fingerprint matching, the following characteristics of minutiae-based representation make it difficult to secure the minutiae templates directly.

1. **Unordered Set Representation**: Minutiae sets are *unordered* and the correspondence between individual minutiae in the enrollment and query minutiae sets are not known in advance. Furthermore, the number of minutiae in the two sets may be different (see Figure 2).
2. **Alignment Issues**: A template protection scheme for minutiae templates generally precludes the use of sophisticated minutiae matchers to align the minutiae sets. The alignment issue is handled either by using external information such as reference points or by using rotation- and translation-invariant local minutiae structures.
3. **Nonlinear distortion**: Even when two minutiae sets are aligned with respect to linear transformations like rotation and translation, the locations and directions of the corresponding minutiae do not match exactly due to nonlinear distortion. Though quantization of minutiae attributes can reduce the effect of distortion to some extent, it cannot be eliminated completely.

While some template protection schemes have been designed specifically to work with unordered sets like minutiae (e.g., fuzzy vault [26] and non-invertible transformation [42]), these schemes tend to significantly degrade the matching accuracy due to alignment issues and nonlinear distortion. Furthermore, other template protection schemes like fuzzy commitment, which have been successfully used with other biometric modalities like iris [19], cannot be directly used for securing fingerprint minutiae. On the other hand, feature representations that characterize the global texture pattern of the fingerprint image are typically fixed-length real-valued vectors, which are again difficult to secure. To overcome these limitations, several techniques have been proposed to adapt the given fingerprint representation into a

form that can be more easily secured using biometric cryptosystems like fuzzy vault and fuzzy commitment (see Table 1).

Technique	Features	Transformation	Final representation
Spectral minutiae [55]	Minutiae	Fourier transform of 2D-delta functions at minutiae locations	Vector
BioPhasor [49]	FingerCode	Nonlinear	Vector
Biometric encryption [47]	Fingerprint image	Apply a secure filter	Vector
Minutiae indicator [15]	Minutiae	Minutiae locations are marked as '1'	vector
Histogram of minutiae triplets [16]	Minutiae	Hashing the histogram of minutiae triplet features	Vector
Cuboid based minutiae Aggregates [48]	Minutiae	Minutiae aggregate selection from random local regions	Vector
Symmetric hash [50]	Minutiae as complex numbers	Set of order invariant functions of minutiae	Minutiae
Cancelable fingerprints [42]	Minutiae	Image folding	Minuitae
Alignment free cancelable fingerprint[29]	Minutiae, orientation field	Transform minutiae according to surrounding orientation field	Minutiae
Minutiae structures [25]	Minutiae	Local minutiae structures	Minutiae

Table 1 Different techniques to transform fingerprint features for template protection.

We now discuss four different fingerprint feature adaptation approaches that have been proposed in the literature, namely, (i) local aggregates, (ii) spectral minutiae, (iii) local minutia structure, and (iv) quantization and reliable component selection. The goal of local aggregates and spectral minutiae approaches is to convert the minutia set into a fixed-length binary feature vector that can be secured using fuzzy commitment. The local minutia structure approach is primarily designed to overcome the alignment problem by deriving new features from the minutiae that are invariant under rotation and translation. The new features derived from the minutiae can be secured using fuzzy vault, fuzzy commitment, or other hybrid biometric cryptosystems. Quantization and reliable component selection converts a fixed-length real-valued feature vector into a compact binary vector, thereby enabling the use of a fuzzy commitment.

3.1 Local Aggregates Approach

In this approach, the fingerprint region is divided into a number of randomized local regions (could be over-lapping) and features are computed based on the minutiae falling within each local region. For example, Chang and Roy [8] consider a finite

number of lines in the fingerprint area and use the difference in the number of minutiae on the two sides of the line as the feature vector. This feature vector is further converted into a binary representation using the techniques described in Section 3.4. Note that the fingerprints need to be aligned before feature extraction in order for the local aggregates approach to work.

Sutcu et al. [48] used a set of axis-aligned variable-sized cuboids as the local region. Each cuboid is parameterized by its location and range along each of the x and y coordinates and the minutia orientation angle θ (see Figure 8 for a typical cuboid configuration). A vector consisting of the number of minutiae falling into each of the cuboids is obtained and binarized to derive the final representation. This approach was further improved in [33] by including additional statistics related to minutiae falling in each cuboid. The statistics computed are

1. *Aggregate wall distance (δ)*: For a cuboid bounded by (x_{min}, x_{max}, y_{min}, y_{max}, θ_{min}, θ_{max}), δ is computed as:

$$\delta = \sum_{i=1}^{t} \min(\delta_x^i, \delta_y^i, \delta_\theta^i, \tau_\delta) \tag{1}$$

   where t is the number of minutiae in the given cuboid, τ_δ is a threshold used for wall distance, and δ_x^i, δ_y^i, and δ_θ^i are given by $\min(|x_i - x_{min}|, |x_i - x_{max}|)$, $\min(|y_i - y_{min}|, |y_i - y_{max}|)$, and $\min(|\theta_i - \theta_{min}|, |\theta_i - \theta_{max}|)$, respectively.
2. *Minutiae Average*: Average coordinate of all the minutiae present in each cuboid in a given measurement.
3. *Minutiae Deviation*: Standard deviation of minutiae coordinates present in each cuboid in a given measurement.

Additional information related to ridge orientation as well as ridge frequency present inside a local rectangular region can also be added to the local aggregate representation [34]. To obtain the orientation-based features, the fingerprint is filtered using four different Gabor filters oriented along 0, 45, 90, and 135 degrees. Given a local aggregate region, four different values are obtained corresponding to the standard deviations of the values associated with the four Gabor responses. The ridge frequency based features are computed as the average ridge frequency inside the aggregate region.

3.2 Spectral Minutiae Representation

The spectral minutiae representation is obtained by considering the minutia set as a collection of 2-dimensional dirac-delta functions and obtaining its Fourier spectrum after low pass filtering [55]. Only the magnitude spectrum is considered and it is sampled on a log polar grid to obtain a fixed-length vector. Theoretically, the magnitude spectrum is invariant to rotation and translation due to the shift, scale, and

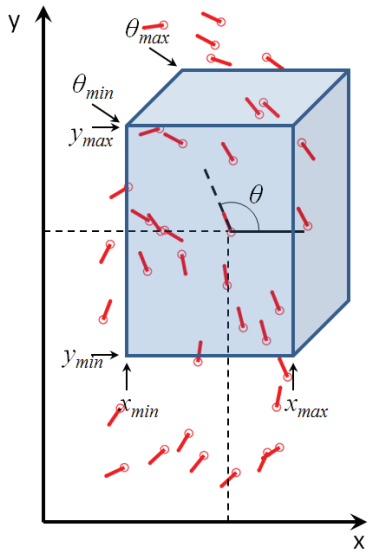


Fig. 8 A cuboid bounded by $(x_{min}, x_{max}, y_{min}, y_{max}, \theta_{min}, \theta_{max})$ overlaid over the minutia points. The local aggregate features are computed based on the statistics of minutiae that fall within the cuboid [48, 33].

rotation properties of the Fourier transform. Hence, it is possible to perform matching between two spectral minutiae vectors without aligning them first. However, in practice, alignment based on singular points is required to achieve good matching performance [55].

Another variation of the spectral minutiae approach is the Binarized Phase Spectrum (BiPS) representation proposed in [35]. To incorporate translation- and rotation-invariance, only the magnitude spectrum is considered in [55] and the phase spectrum is ignored. In [35], alignment is achieved through the use of external information such as reference points. Therefore, only the phase spectrum of the minutiae is considered. The phase spectrum can be sampled along a log-polar grid to obtain the fixed-length minutiae representation. Furthermore, these phase samples can be easily quantized into two bits depending on which quadrant they fall into. The resulting binarized phase spectrum can be directly secured using the fuzzy commitment approach.

Consider a minutiae set $\boldsymbol{M} = \{\mathbf{m}_i\}_{i=1}^{n}$, where $\mathbf{m}_i$ is the i^{th} minutiae with location (x_i, y_i) and direction θ_i, and n is the number of minutiae. We can associate a function $g(x,y)$ to each minutia $\mathbf{m}_i$ as follows.

$$g(x,y;\mathbf{m}_i) = \delta(x-x_i, y-y_i)\exp(j\theta_i). \tag{2}$$

The 2-D function $f(x,y)$ that defines the minutiae set $\boldsymbol{M}$ and its continuous Fourier transform can be expressed as

$$f(x,y) = \sum_{i=1}^{n} \delta(x - x_i, y - y_i) \exp(j\theta_i). \tag{3}$$

$$F(u,v) = \sum_{i=1}^{n} \exp(j(2\pi(ux_i + vy_i) + \theta_i)). \tag{4}$$

The phase of the Fourier spectrum of $f(x,y)$ is denoted as $\Psi(F(u,v))$ and is given by the following equation.

$$\Psi(F(u,v)) = \arctan \frac{\sum_{i=1}^{n} \sin(2\pi(ux_i + vy_i) + \theta_i)}{\sum_{i=1}^{n} \cos(2\pi(ux_i + vy_i) + \theta_i)}. \tag{5}$$

$\Psi(F(u,v))$ can take values in $[0, 2\pi]$. To binarize the phase spectrum, $\Psi(F(u,v))$ is quantized into four distinct values based on the quadrant in which it falls and is represented using two bits. Thus, the phase spectrum can be represented as a fixed length binary string $\mathbf{x} = [b_1, b_2, b_3, \cdots, b_{2N}]$ as follows.

$$\begin{aligned} b_{2j-1} &= sgn(Re(F(u_j, v_j))) \\ b_{2j} &= sgn(Im(F(u_j, v_j))), \text{where} \end{aligned} \tag{6}$$

$sgn(y) = 1$, if $y \geq 0$, zero, otherwise, $Re(.)$ and $Im(.)$ are the real and imaginary parts of a complex number, and (u_j, v_j) denotes the j^{th} frequency sample, $j = 1, \cdots, N$. On a log-polar grid, $u = \nu\cos(\phi)$ and $v = \nu\sin(\phi)$, where ν is the radial distance and ϕ is the radial angle. If we choose N_ν logarithmically spaced samples between ν_{min} and ν_{max} and N_ϕ linearly-spaced samples between 0 and π, the total number of samples is $N = N_\nu N_\phi$ and the length of the binary string obtained from a minutiae set is $2N$ bits. An illustration of the BiPS representation of minutiae is shown in Figure 9.

3.3 Local Minutiae Structures

Local minutiae structures consist of features that characterize the relative information between two or more minutiae (e.g., distance between two minutiae) [6]. The main advantage of this approach is that since the features are relative, they are invariant to global rotation and translation of the fingerprint. Hence, no a priori alignment is needed before matching. An additional benefit is that such features are robust to nonlinear distortion. However, if the matching is based only on the local minutiae information and the global spatial relationships between minutiae (which are highly distinctive) is ignored, it may lead to degradation in the matching accuracy.

The simplest local minutiae structure is based on minutia pairs, where the distance between the pair and the orientation of each minutia with respect to the line connecting them can be used as the invariant attributes. Boult et al. [4] proposed

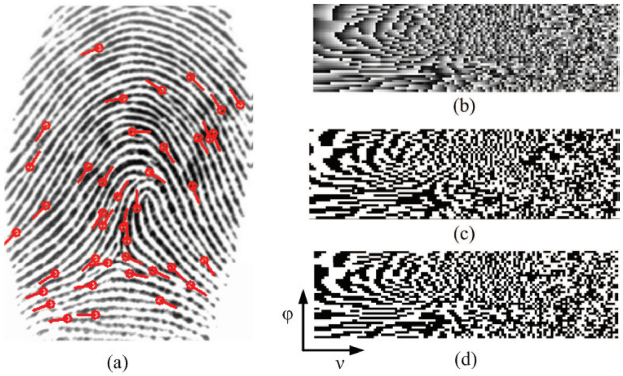


Fig. 9 Illustration of binarized phase spectrum of fingerprint minutiae [35]. (a) A fingerprint image with minutiae marked on it, (b) phase spectrum: $\Psi(F(u,v))$, (c) odd bits of the binarized phase spectrum: $sgn(Re(F(u_j,v_j)))$, and (d) even bits of the binarized phase spectrum: $sgn(Im(F(u_j,v_j)))$.

a hybrid biometric cryptosystem to secure such a representation. The fundamental idea is to split the value of each feature (relative distances and angles) into stable and unstable parts. The stable parts are encrypted, while the unstable parts are left unprotected. A robust distance measure was proposed to match minutia pairs by combining the results of the stable part matching that takes place in the encrypted domain and the unstable part matching in the plaintext domain.

Another commonly used local minutiae structure is the minutia triplet, where relative features (distances and angles) are computed from combinations of three minutiae. Farooq et al. [16] proposed a non-invertible feature transformation approach for secure fingerprint matching based on minutia triplets. The relative features in a triplet are quantized such that only a finite number of triplets (say N) are possible. A N-dimensional histogram characterizing the distribution of different triplets in the given fingerprint image is obtained. This histogram is binarized and transformed in a non-invertible manner by randomly modifying some of the bits in the binary string.

A number of other local minutiae structures have also been proposed. For example, Jeffers and Arakala [24] showed that it is possible to use a fuzzy vault to secure triplet-based, five nearest neighbor-based, and Voronoi neighbor-based minutia structures. Another interesting structure is the Minutia Cylinder Code proposed by Cappelli et al. [6]. This local minutia structure divides a cylindrical region (with its axis along the minutia orientation) around each minutia into a finite number of cells and encodes the likelihood of another minutia in the fingerprint with a specific angle difference to the reference minutia being present in the specific cell.

Finally, it is also possible to exploit additional descriptors such as ridge orientation and ridge frequency in the neighborhood of a minutia [17] for more accurate fingerprint matching. For instance, Nagar et al. [32] use the ridge orientation and ridge frequency values, which are sampled at a set of points around each minutiae,

to encrypt the polynomial evaluations of the corresponding minutia in a fuzzy vault. As a result, an attacker who only guesses the set of genuine minutia from the vault can no longer recover the key; he also needs to know the values corresponding to the associated descriptors in order to fully decode the vault.

3.4 Quantization and Reliable Component Selection

Most of the fingerprint feature adaptation techniques initially output a fixed-length real-valued feature vector. This feature vector could be either derived from the minutiae [34, 55] or based on the global texture pattern [5]. Typically, this real-valued feature vector is quantized by assigning bits to each element in order to obtain a binary representation. In some cases, only a fixed-number of reliable bits are selected to obtain the final binary representation, which is secured using a fuzzy commitment scheme.

Rohde [43] proposed two basic Binary Multidimensional Scaling techniques with the objective of obtaining a lower dimensional set of binary vectors whose pairwise distances closely follow the pairwise distances between the associated original data points. In the first approach, a singular value decomposition was performed on the original real-valued vectors and the resultant projections were binarized using unary encoding[1]. In the second technique, a projection matrix was obtained using the gradient descent method with the objective of minimizing the stress between the pairwise distances in the original space and the scaled pairwise distances in the transformed space. The original vectors were projected using the obtained projection matrix and the resultant vectors were binarized based on the sign of each vector-element.

Andoni et al. [1] proposed a technique referred to as Locality Sensitive Hashing (LSH), where the original real-valued vectors are projected using random matrices and the resultant projections are binarized using unary coding in order to obtain the final binary vector. LSH is mostly used in image retrieval applications, where the objective is to efficiently compute an approximate nearest neighbor of a query. Chen et al. [10] associated multiple bits with each real valued feature element based on its discriminability. The bit values were determined based on binary representation. Chen et al. [9] also proposed a binarization technique, where pairs of elements of real vectors were converted to polar coordinates and then quantized.

Given binarized features, it is a common practice to select a subset of reliable bits either because the specific biometric cryptosystem requires the binary vector to be of a desired length or there are a large number of unreliable bits and removing them will improve the system accuracy. Selecting a subset of bits that provides the best performance would, in general, require evaluating all the 2^n possible subsets where there are n bits in the original binary vector. However, a number of efficient

[1] A unary encoding works as follows. Suppose that a real-value a needs to be encoded using t bits. The range of a, say $[a_{min}, a_{max}]$, is quantized into $(t+1)$ bins. If a falls into the i^{th} bin, it is represented as (t-i+1) ones followed by $(i-1)$ zeros, where $i = 1, 2, \cdots, (t+1)$.

approximations have been proposed in literature. Examples include the sequential forward floating search [40], where features are successively added and removed to the selection based on the performance of the selected set of features till a stable performance is reached, mutual information based feature selection [39], and other simple selection procedures based on correlation and feature discriminability [33].

4 Alignment with Secure Fingerprint Templates

The first step in matching two fingerprint images is to align them and determine the area of overlap. Although aligning two fingerprints is a difficult problem in any fingerprint authentication system, it is much more difficult when the information about the template must not be leaked. One way to solve this problem is to use local minutiae structures, which are invariant to rotation and translation because such features are typically obtained relative to the location and orientation attributes of each minutia point. We have already discussed this approach in Section 3.3. The alternate approach is to extract and store some reference points from the enrolled fingerprint image that do not leak excessive information about the minutiae template. During authentication, the reference points can also be obtained from the query fingerprint image. The template and query minutiae sets can be aligned based on the parameters obtained by aligning the corresponding sets of reference points.

The most commonly used reference points for fingerprint alignment are the *singular points* (e.g., core and delta) [31]. There are many approaches like Poincare index method [31], geometric method [41], complex filter method [38], etc. to determine the singular points in a fingerprint image. However, the accuracy of these techniques is limited by the following three issues: (i) low quality of the captured fingerprint image, (ii) the absence of clearly defined core points in arch and tented-arch fingerprint patterns, and (iii) partial nature of many fingerprint images captured using live-scan sensors.

One promising approach for reference point detection is the focal point localization algorithm proposed by Boonchaiseree and Areekul [3], which overcomes the problems associated with singular points. The focal point is defined as the average center of curvature of a fingerprint. In other words, the focal point is the centroid of all the crossing points, where a crossing point is a point of intersection of two normal lines of curved ridges. The algorithm proposed in [3] is iterative and in each iteration, only the orientation field in the semi-circular region of a specified radius centered at the current focal point is used to generate the crossing points. The limitations of this algorithm are its iterative nature (hence high computational requirement) and the need for carefully selecting the focal point for the first iteration.

Another alternative candidate for a reference point is a stable minutia point in the given fingerprint [56]. While the alignment based on such a reference point is simple and computationally efficient, it is difficult to determine the stable minutia point reliably. Even a small error in locating the reference point could lead to a false reject.

Uludag and Jain [51] extracted a set of points with high curvature from the fingerprint orientation field. A trimmed Iterative Closest Point (ICP) algorithm was used to determine the alignment between the template and the query based on these high curvature points. Since high curvature points are global features in the fingerprint pattern, they do not reveal any information about the minutia attributes, which are local characteristics in the fingerprint. Nandakumar et al. [36] have made significant enhancements to the alignment algorithm in [51], resulting in more accurate alignment between the template and query.

The high curvature points can be extracted as follows (see Figure 10). First a set of orientation field flow curves are extracted from the fingerprint. An orientation field flow curve [13] is a set of piecewise linear segments whose tangent direction at each point is parallel to the orientation field direction at that point. Although flow curves are similar to fingerprint ridges, extraction of flow curves is not affected by breaks and discontinuities, which are commonly encountered in ridge extraction. Points of maximum curvature in the flow curves along with their curvature values can be used for alignment. These high curvature points tend to occur near the singular points in the fingerprint image. If the image has more than one singularity, high curvature points may have many clusters, which can be identified by applying a single-link clustering algorithm. While this alignment technique is more accurate than alignment based on singular points [36], it is not computationally efficient and storing many high curvature points may leak more information about the fingerprint pattern. To overcome this problem, a single focal point was estimated from each cluster of high curvature points in [35].

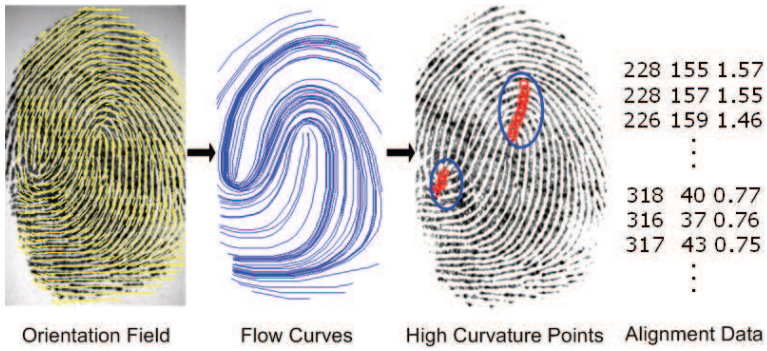


Fig. 10 Algorithm for extraction of high curvature points.

5 Matching Performance and Security

The effectiveness of a fingerprint template protection technique can be measured in terms of the resulting (i) matching performance and (ii) template security. Matching performance is usually quantified by the False Accept Rate (FAR) and the Genuine Accept Rate (GAR) of the biometric system. Security is measured in terms of the information leakage rate[2] or the computational complexity involved in recovering the original template from the secure sketch or the transformed template [28, 20]. Due to intra-user variability in fingerprint images, there is usually a trade-off between the GAR and the security in most template protection schemes. Schemes with higher security tend to have lower GAR and vice versa.

While a number of fingerprint template protection schemes have been proposed, many of them have not been carefully evaluated in terms of their matching performance and template security. For example, the matching performance of traditional fingerprint recognition systems have been evaluated on large databases containing several thousand unique fingerprints by independent third-parties (e.g., Fingerprint Vendor Technology Evaluation [54]). Such large scale independent evaluations allow us to determine whether the performance differences between competing algorithms are statistical significant. However, most fingerprint template protection schemes have been tested using small (sometimes proprietary) databases containing at most a few hundred users. Hence, it is difficult to judge the relative differences in matching performance among various fingerprint template security schemes. Similarly, accurate estimation of the security provided by a template protection scheme requires good statistical models for the distribution of fingerprint features (e.g., minutiae). Given the absence of such models, most of the schemes make unrealistic assumptions such as uniform distribution of features, resulting in over-optimistic estimates of security. Furthermore, in addition to the information leakage rate from the secure sketch or transformed template, one must also carefully analyze the security in scenarios where the adversary may get access to ancillary information (e.g., alignment information stored with a secure sketch or the user-specific key used to derive a transformed template) along with the protected template.

For illustration purposes, we evaluate implementations of the non-invertible feature transformation approach [42], fingerprint fuzzy vault [36], and fingerprint fuzzy commitment [35] on a public-domain fingerprint database, namely the FVC2002-DB2. This database [30] consists of 800 images of 100 fingers with 8 impressions per finger obtained using an optical sensor. The size of the images in this database is 560×296, the resolution of the sensor is 569 dpi and the images are generally of good quality. Our goal here is not to determine the superiority of one template protection method over the other but to simply highlight the various issues that need to be considered in implementing a template protection scheme. Of course, performance varies depending on the choice of the features, the selected feature adaptation

[2] Given the secure sketch, leakage rate quantifies the information available to adversary about the original biometric template (known as privacy leakage) or the cryptographic key associated to it (secret key leakage).

scheme, database used, and the values of the parameters used in each scheme. In our implementation, we consider only the location and orientation attributes of minutiae.

5.1 Non-invertible Transform

We implemented two non-invertible transforms, namely, polar and functional (with a mixture of Gaussian as the transformation function) defined in [42]. For the polar transform, the central region of the image was tesselated into $n = 6$ sectors of equal angular width and 30-pixel wide concentric shells. The transformation here is constrained such that it only shifts the sector number of the minutiae without changing the shell. There are $n!$ ways in which the n sectors in each shell can be reassigned. Given k shells in the image (constrained by the width of the image and ignoring the central region of radius 15 pixels), the number of different ways a transformation can be constructed is $(n!)^k$ which is equivalent to $log_2(n!)^k$ bits of security.

For the functional transformation, we used a mixture of 24 Gaussians with the same isotropic standard deviation of 30 pixels (where the peaks can correspond to +1 or -1 as used in [42]) for calculating the displacement and used the direction of gradient of the mixture of Gaussian functions as the direction of minutiae displacement. Since the mean vector of the Gaussians can fall anywhere in the image, there are 296 x 560 possible different values of means of each Gaussian component. As there are 24 Gaussian components and each one can peak at +1 or -1, there are $(296 * 560 * 2)^{24}$ possible transformations. However, two transformations with slightly shifted component means will produce two similar templates such that one template can be used to verify the other.

To analyze the security of the functional transformation, Ratha et al. [42] assumed that for each minutiae in the fingerprint, its transformed counterpart could be present in a shell of width d pixels at a distance of K pixels from the minutiae. Further, assuming that the matcher cannot distinguish minutiae that are within δr pixels and their orientations are within $\delta\theta$ degrees, each transformed minutiae encodes $I_m = log_2(\pi \frac{((K+d)^2-K^2)}{(\delta r)^2} * \frac{\pi}{\delta\theta})$ bits of information. Assuming that there are N minutiae in template fingerprint and one needs to match at least m minutiae to get accepted, the adversary needs to make $2^{I_m * m - log_2(\binom{N}{m})}$ attempts. Note that this analysis is based on the simplifying assumption that each minutiae is transformed independently. This overestimates the number of attempts needed by an adversary to guess the minutiae template.

Among the eight impressions available for each of the 100 fingers in FVC2002-DB2, we use only the first two impressions in this experiment because they have the best image quality. The results, based on the minutiae matcher in [22], are shown in Figure 11, which indicates a decrease of about 6% in the GAR at a FAR of 0.1%. In terms of security, non-invertible transformation is one of the better approaches since it is computationally hard (in terms of brute force complexity) to invert the stored template and obtain the true template. The true template is never revealed

especially in case when the transformation of the biometric template is done on a separate module (possibly a hand held device [23]), which does not save the original template in memory and is not accessible to an adversary.

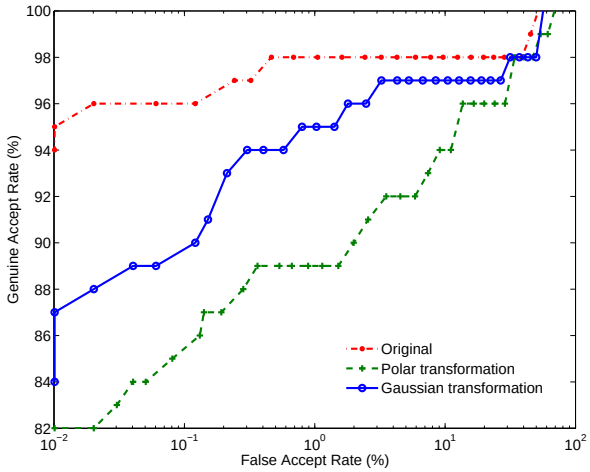


Fig. 11 ROC curves corresponding to two non-invertible transforms (Gaussian and polar) on FVC2002-DB2. The "Original" curve represents the case where no transformation is applied to the template, "Gaussian" curve corresponds to the functional transformation of the template and "Polar" corresponds to the polar transformation of the template.

5.2 Fingerprint Fuzzy Vault

We implemented the fuzzy vault as proposed in [36] using the first two impressions of each of the 100 fingers in the FVC2002-DB2. Table 2 shows the error rates corresponding to different key sizes used in binding. Compared to the "original" ROC curve in Figure 11, we observe that the fuzzy vault scheme has a lower genuine accept rate by about 4%. Further, this scheme also has failure to capture errors if the number of minutiae in the fingerprint image is not sufficient for vault construction (18 in our implementation).

Suppose an attacker launches a brute-force attack on the fuzzy vault by trying to decode the key using all possible combinations of $(k+1)$ points in the vault. If $k = 10$ and the number of genuine and chaff points in the vault are 24 and 200, respectively, the security of the minutiae template is approximately 39 bits. Note that

Table 2 Performance summary of the fuzzy vault implementation for FVC2002-DB2 database. Here, k denotes the degree of the encoding polynomial used in vault construction. The maximum key size that can be bound to the minutiae template is $16k$ bits. FTCR refers to the failure to capture rate, which is the proportion of fingerprints having very small number of minutiae that is not sufficient for vault construction.

FTCR	$k=7$		$k=8$		$k=10$	
	GAR	**FAR**	**GAR**	**FAR**	**GAR**	**FAR**
2%	91%	0.13%	91%	0.01%	86	0%

this estimate is based on the assumption that minutia points are distributed uniformly over the fingerprint image area, which is not true in practice. Moreover, there are some specific attacks that can be staged against a fuzzy vault, e.g., *attacks via record multiplicity*, *stolen key inversion attack* and *blended substitution attack* [46]. If an adversary has access to two different vaults (say from two different applications) obtained from the same biometric data, he can easily identify the genuine points in the two vaults and decode the vault. Thus, the fuzzy vault scheme does not provide revocability. In a stolen key inversion attack, if an adversary somehow recovers the key embedded in the vault, he can decode the vault to obtain the biometric template. Since the vault contains a large number of chaff points, it is possible for an adversary to substitute a few points in the vault using his own biometric features. This allows both the genuine user and the adversary to be successfully authenticated using the same identity and such an attack is known as blended substitution.

5.3 Fingerprint Fuzzy Commitment

The fingerprint fuzzy commitment scheme based on the Binarized Phase Spectrum representation of minutiae proposed in [35] was implemented with the following parameter settings: $\nu_{min} = 0.01$, $\nu_{max} = 0.25$, $N_\nu = 128$, $N_\phi = 37$. At the time of enrollment, $N = 2,048$ most reliable bits are selected from the available bits using the bit selection algorithm described in [35]. During enrollment, we select a codeword $\mathbf{c}$ of the same length N by adding error correction bits to a uniformly random key (κ_c) of length L bits generated independently. The length of the key (L) is varied from 224 to 256 bits to obtain different false accept rates (FAR). A turbo encoder with a recursive convolutional code of rate $1/4$ as the component encoder is used for error correction. For these settings, the turbo code can recover the key κ from the secure sketch even if approximately 30% of bits in $\mathbf{b}^E$ and $\mathbf{b}^A$ are different.

Table 3 Genuine Accept Rate (GAR) of fingerprint fuzzy commitment based on Binarized Phase Spectrum representation of minutiae proposed in [35]. Here, FAR denotes the false accept rate.

0% FAR	0.02% FAR	0.1% FAR
87.4%	90.4%	91.1%

The genuine accept rate (GAR) at zero, 0.02%, and 0.1% FAR are shown in Table 3. Note that these GAR values are based on all the impressions for each user. In the case of fingerprint fuzzy vault (see Table 2), the GAR is 86% at zero-FAR compared to a GAR of 87.4% for the fingerprint fuzzy commitment. However, only the 100 genuine matches based on the first two impressions for each user were considered in the fuzzy vault. For this subset of FVC2002-DB2, the GAR of the fuzzy commitment is 94% at zero-FAR. Even after considering the correlation between the bits, a security of 43 bits was reported in [35].

6 Conclusions and Future Research Directions

Among the various vulnerabilities of a biometric system, leakage of biometric template information is a major security and privacy concern due to the strong linkage between a user's template and his identity and the irrevocable nature of biometric templates. In this chapter, we briefly reviewed the three basic theoretical frameworks for biometric template protection, namely, encryption, template transformation, and biometric cryptosystems and discussed the practical issues involved in applying these techniques to secure a fingerprint template. Due to variations in finger placement and pressure applied on the sensor, there are two fundamental challenges in any fingerprint template protection scheme. First, we need to automatically align or register the fingerprints obtained during enrollment and matching, without revealing excessive information about the features that uniquely characterize a fingerprint. Secondly, we need to select an appropriate representation scheme that captures most of the discriminatory information, but is relatively invariant to changes in finger placement. Finally, specific implementations of three different template protection schemes on a common fingerprint database was presented to illustrate the issues concerning matching accuracy and template security.

We believe that as yet, there is no "best" approach for template protection that completely satisfies the three main requirements of template security, matching accuracy, and revocability. The application scenario and requirements play a major role in the selection of a template protection scheme. In an airport watch list application, non-invertible transform may be a more suitable approach because it provides both template security and revocability without relying on any other input from the user. Biometric cryptosystems may be more appropriate in match-on-card applications because such systems typically release a key to the associated application in order to indicate a successful match. In general, more than one template protection scheme may be admissible and the choice of the suitable approach may be based on a number of factors such as matching performance, computational complexity, memory requirements, and user acceptance and co-operation. Further research in the area of fingerprint template security is expected to progress along the following three main directions.

1. What is the "optimal" feature transformation function or biometric cryptosystem for matching fingerprints securely? Optimality generally refers to the best tradeoff between template security and matching performance.
2. Suppose that there is a good template protection algorithm for a specific feature type (e.g., a binary string); what is the best way to embed other types of features (e.g., minutia set) in the desired feature domain? This question is also relevant in case there is a need to secure templates from multiple biometric traits as a single entity. Note that the best representation should be compact, preserve accuracy, and preferably have a uniform distribution.
3. Finally, one of the important but difficult tasks in the design of a template protection algorithm is: how to quantify the security provided by the algorithm? Most of the existing methodologies for security analysis are based on unrealistic assumptions (e.g., uniform distribution of minutiae). A related issue is the need to quantify the inherent entropy in (or the individuality of) a fingerprint or the features extracted from it.

References

1. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Communications of the ACM **51**(1), 117–122 (2008)
2. Bleichenbacher, D., Nguyen, P.Q.: Noisy Polynomial Interpolation and Noisy Chinese Remaindering. In: Proc. Nineteenth IACR Eurocrypt, pp. 53–69. Bruges, Belgium (2000)
3. Boonchaiseree, N., Areekul, V.: Focal Point Detection Based on Half Concentric Lens Model for Singular Point Extraction in Fingerprint. In: Proceedings of International Conference on Biometrics, pp. 637–646 (2009)
4. Boult, T.E., Scheirer, W.J., Woodworth, R.: Fingerprint Revocable Biotokens: Accuracy and Security Analysis. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
5. Bringer, J., Chabanne, H., Cohen, G., Kindarji, B., Zmor, G.: Theoretical and Practical Boundaries of Binary Secure Sketches. IEEE Trans. on Info. Forensics & Security **3**(4), 673–683 (2008)
6. Cappelli, R., Ferrara M. Maltoni, D.: Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(12), 2128 – 2141 (2010)
7. Cappelli, R., Lumini, A., Maio, D., Maltoni, D.: Fingerprint Image Reconstruction From Standard Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(9), 1489–1503 (2007)
8. Chang, E.C., Roy, S.: Robust Extraction of Secret Bits From Minutiae. In: Proc. Second International Conference on Biometrics, pp. 750–759. Seoul, South Korea (2007)
9. Chen, C., Veldhuis, R.: Binary Biometric Representation through Pairwise Polar Quantization. In: Proc. International Conference on Biometrics, pp. 72–81 (2009)
10. Chen, C., Veldhuis, R.N.J., Kevenaar, T.A.M., Akkermans, A.H.M.: Biometric Quantization through Detection Rate Optimized Bit Allocation. EURASIP Journal on Advances in Signal Processing (2009)
11. Chung, Y., Moon, D., Lee, S., Jung, S., Kim, T., Ahn, D.: Automatic Alignment of Fingerprint Features for Fuzzy Fingerprint Vault. In: Proc. Conference on Information Security and Cryptology, pp. 358–369. Beijing, China (2005)

12. Clancy, T., Lin, D., Kiyavash, N.: Secure Smartcard-Based Fingerprint Authentication. In: Proc. ACM SIGMM Workshop on Biometric Methods and Applications, pp. 45–52. Berkley, USA (2003)
13. Dass, S.C., Jain, A.K.: Fingerprint Classification Using Orientation Field Flow Curves. In: Proc. Indian Conference on Computer Vision, Graphics and Image Processing, pp. 650–655. Kolkata, India (2004)
14. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. Tech. Rep. 235, Cryptology ePrint Archive (2006). A preliminary version of this work appeared in EUROCRYPT 2004
15. Draper, S.C., Khisti, A., Martinian, E., Vetro, A., Yedidia, J.S.: Using Distributed Source Coding to Secure Fingerprint Biometrics. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 2, pp. 129–132. Hawaii, USA (2007)
16. Farooq, F., Bolle, R., Jea, T., Ratha, N.: Anonymous and revocable fingerprint recognition. In: Proc. IEEE Computer Vision and Pattern Recognition (2007)
17. Feng, J.: Combining minutiae descriptors for fingerprint matching. Pattern Recognition **41**(1), 342352 (2008)
18. Feng, J., Jain, A.: FM model based fingerprint reconstruction from minutiae template. In: International conference on Biometrics (ICB) (2009)
19. Hao, F., Anderson, R., Daugman, J.: Combining Crypto with Biometrics Effectively. IEEE Transactions on Computers **55**(9), 1081–1088 (2006)
20. Ignatenko, T., Willems, F.M.J.: Biometric systems: Privacy and secrecy aspects. IEEE Trans. on Information Forensics and Security **4**(4), 956–973 (2009)
21. Jain, A., Nandakumar, K., Nagar, A.: Biometric template security. EURASIP Journal on Advances in Signal Processing (2008)
22. Jain, A.K., Hong, L., Bolle, R.: On-line Fingerprint Verification. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(4), 302–314 (1997)
23. Jain, A.K., Pankanti, S.: A Touch of Money. IEEE Spectrum **3**(7), 22–27 (2006)
24. Jeffers, J., Arakala, A.: Minutiae-based structures for a fuzzy vault. In: Proc. Biometric Symposium, BCC, pp. 1–6. Baltimore, MD (2006)
25. Jeffers, J., Arakala, A.: Fingerprint alignment for a minutiae-based fuzzy vault. In: Proc. Biometric Symposium, BCC. Baltimore, MD (2007)
26. Juels, A., Sudan, M.: A Fuzzy Vault Scheme. In: Proc. IEEE International Symposium on Information Theory, p. 408. Lausanne, Switzerland (2002)
27. Juels, A., Wattenberg, M.: A Fuzzy Commitment Scheme. In: Proc. Sixth ACM Conference on Computer and Communications Security, pp. 28–36. Singapore (1999)
28. Lai, L., Ho, S.W., Poor, H.V.: "privacy-security tradeoffs in biometric security systems. In: Annual Allerton Conference on Communication, Control, and Computing, pp. 23–26. Monticello, IL (2008)
29. Lee, C., Choi, J.Y., Toh, K.A., Lee, S.: Alignment-free cancelable fingerprint templates based on local minutiae information. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics **37**(4), 980–992 (2007)
30. Maio, D., Maltoni, D., Wayman, J.L., Jain, A.K.: FVC2002: Second Fingerprint Verification Competition. In: Proc. International Conference on Pattern Recognition (ICPR), pp. 811–814. Quebec City, Canada (2002)
31. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition, second edn. Springer-Verlag (2009)
32. Nagar, A., Nandakumar, K., Jain, A.K.: Securing Fingerprint Template: Fuzzy Vault with Minutiae Descriptors. In: Proc. IEEE International Conference on Pattern Recognition. Tampa, FL (2008)
33. Nagar, A., Rane, S., Vetro, A.: Privacy and security of features extracted from minutiae aggregates. In: Proceedings IEEE International Conf. on Acoustics, Speech and Signal Processing, pp. 524–531. Dallas, TX (2010)
34. Nagar, A., Rane, S.D., Vetro, A.: Alignment and bit extraction for secure fingerprint biometrics. In: SPIE Conference on Electronic Imaging (Special Collection), vol. 7541 (2010)

35. Nandakumar, K.: A Fingerprint Cryptosystem based on Minutiae Phase Spectrum. In: Proc. of Second IEEE Workshop on Info. Forensics & Security. Seattle, USA (2010)
36. Nandakumar, K., Jain, A.K., Pankanti, S.: Fingerprint-based Fuzzy Vault: Implementation and Performance. IEEE Transactions on Information Forensics and Security **2**(4), 744–757 (2007)
37. Nandakumar, K., Nagar, A., Jain, A.K.: Hardening Fingerprint Fuzzy Vault Using Password. In: Proc. Second Intl. Conf. on Biometrics, pp. 927–937. Seoul, South Korea (2007)
38. Nilsson, K., Bigun, J.: Complex Filters Applied to Fingerprint Images Detecting Prominent Symmetry Points Used for Alignment. In: Proceedings of International Conference on Biometric Authentication, pp. 39–47 (2002)
39. Peng, H.C., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(8), 1226–1238 (2005)
40. Pudil, P., Novovicova, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognition Letters **15**, 1119–1125 (1994)
41. Ramo, P., Tico, M., Onnia, V., Saarinen, J.: Optimized Singular Point Detection Algorithm for Fingerprint Images. In: Proceedings of ICIP, pp. 242–245 (2001)
42. Ratha, N.K., Chikkerur, S., Connell, J.H., Bolle, R.M.: Generating Cancelable Fingerprint Templates. IEEE Trans. on Pattern Analysis and Machine Intelligence **29**(4), 561–572 (2007)
43. Rhodes, D.: Methods for Binary Multidimensional Scaling. Neural Computation **14**, 1195–1232 (2002)
44. Ross, A.K., Shah, J., Jain, A.K.: From Templates to Images: Reconstructing Fingerprints From Minutiae Points. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(4), 544–560 (2007)
45. Scheirer, W., Boult, T.: Bio-cryptographic protocols with bipartite biotokens. In: Proc. Biometric Symposium (2008)
46. Scheirer, W.J., Boult, T.E.: Cracking Fuzzy Vaults and Biometric Encryption. In: Proceedings of Biometrics Symposium (2007)
47. Soutar, C., Roberge, D., Stoianov, A., Gilroy, R., Kumar, B.V.K.V.: Biometric encryption using image processing. In: Proc. of SPIE, vol. 3314, pp. 178–188 (1998)
48. Sutcu, Y., Rane, S., Yedidia, J., Draper, S., Vetro, A.: Feature extraction for a slepian-wolf biometric system using ldpc codes. In: Proc. the IEEE International Symposium on Information Theory. Toronto, Canada (2008)
49. Teoh, A.B.J., Toh, K.A., Yip, W.K.: 2^N Discretisation of BioPhasor in Cancellable Biometrics. In: Proc. Second Intl. Conf. on Biometrics, pp. 435–444. Seoul, South Korea (2007)
50. Tulyakov, S., Farooq, F., Mansukhani, P., Govindaraju, V.: Symmetric hash functions for secure fingerprint biometric systems. Pattern Recognition Letters **28**(16), 2427–2436 (2007)
51. Uludag, U., Jain, A.K.: Securing Fingerprint Template: Fuzzy Vault With Helper Data. In: Proc. CVPR Workshop on Privacy Research In Vision, p. 163. New York, USA (2006)
52. Uludag, U., Pankanti, S., Jain, A.K.: Fuzzy Vault for Fingerprints. In: Proc. Fifth International Conference on Audio- and Video-based Biometric Person Authentication, pp. 310–319. Rye Town, USA (2005)
53. Uludag, U., Pankanti, S., Prabhakar, S., Jain, A.K.: Biometric Cryptosystems: Issues and Challenges. Proc. IEEE (Special Issue on Multimedia Security for Digital Rights Management) **92**(6), 948–960 (2004)
54. Wilson, C., Hicklin, A.R., Bone, M., Korves, H., Grother, P., Ulery, B., Micheals, R., Zoepfl, M., Otto, S., Watson, C.: Fingerprint Vendor Technology Evaluation 2003: Summary of Results and Analysis Report. Technical Report NISTIR 7123, NIST (2004)
55. Xu, H., Veldhuis, R.N.J., Bazen, A.M., Kevenaar, T.A.M., Akkermans, T.A.H.M., Gokberk, B.: Fingerprint Verification Using Spectral Minutiae Representations. IEEE Trans. on Info. Forensics & Security **4**(3), 397–409 (2009)
56. Yang, S., Verbauwhede, I.: Automatic Secure Fingerprint Verification System Based on Fuzzy Vault Scheme. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 609–612. Philadelphia, USA (2005)