Pattern Recognition Letters 31 (2010) 733–741

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A hybrid biometric cryptosystem for securing fingerprint minutiae templates ☆

Abhishek Nagar [a],*, Karthik Nandakumar [b], Anil K. Jain [a,c]

[a] *Michigan State University, East Lansing, MI, USA*
[b] *Institute for Infocomm Research, A*STAR, Fusionopolis, Singapore 138632, Singapore*
[c] *Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seoul, 136-713, Korea*

ARTICLE INFO

Article history:
Available online 14 July 2009

Keywords:
Biometrics
Template security
Fuzzy vault
Minutiae descriptors

ABSTRACT

Security concerns regarding the stored biometric data is impeding the widespread public acceptance of biometric technology. Though a number of bio-crypto algorithms have been proposed, they have limited practical applicability due to the trade-off between recognition performance and security of the template. In this paper, we improve the recognition performance as well as the security of a fingerprint based biometric cryptosystem, called fingerprint fuzzy vault. We incorporate minutiae descriptors, which capture ridge orientation and frequency information in a minutia's neighborhood, in the vault construction using the fuzzy commitment approach. Experimental results show that with the use of minutiae descriptors, the fingerprint matching performance improves from an FAR of 0.7% to 0.01% at a GAR of 95% with some improvement in security as well. An analysis of security while considering two different attack scenarios is also presented. A preliminary version of this paper appeared in the International Conference on Pattern Recognition, 2008 and was selected as the Best Scientific Paper in the biometrics track.

1. Introduction

Pervasive use of biometrics is imminent due to its numerous advantages over the surrogate identities, e.g. passwords and smart cards. Biometric traits have very high discriminative capability to identify an individual and at the same time, unlike ID cards and passwords, one does not have to worry about losing them (Jain et al., 2008a). However, protecting stored biometric templates is important due to potential misuse of the stolen templates. Cappelli et al. (2007) and Ross et al. (2007) have shown that commonly used templates, that are supposed to be compact, informative and usable representation of biometric traits, have not been designed with the objective of their secure storage. Thus if an adversary is able to access a template, he can create a spoof biometric (e.g. *gummy finger*) from the template (Cappelli et al., 2007) and present it to the system. Due to limited liveness detection capability of current biometric readers, the spoof may be accepted by the system providing an illegitimate access to the adversary. Further, an adversary can cross link the stolen templates with other biometric databases, allowing him to track the activities of an enrolled person, thereby compromising his privacy.

Most template protection approaches can be categorized into following two classes: (i) transformation based approaches and (ii) biometric cryptosystems (Jain et al., 2008b). Transformation based approaches transform the biometric features using a user specific password such that the matching can be performed in the transformed domain. Such a technique is secure since at no time the original biometric is explicitly present in the database. Though a number of transformation based schemes have been proposed (Teoh et al., 2007; Ratha et al., 2007; Savvides and Vijaya Kumar, 2004; Boult et al., 2007), an optimal transformation that is *non-invertible* and at the same time preserves the matching accuracy to a large extent is yet to be found. Biometric cryptosystems (Dodis et al., 2006; Hao et al., 2006; Nandakumar et al., 2007; Sutcu et al., 2007), on the other hand, are techniques that associate an external key with a user's biometric to obtain helper data. The helper data should not reveal any significant information about the template or the key and at the same time it can be used to recover the key when the original biometric is presented. In this paper we shall focus on improving the matching performance and security of a fingerprint based biometric cryptosystem, called fingerprint fuzzy vault (Nandakumar et al., 2007).

A fuzzy vault is preferred to secure fingerprints because of its ability to secure biometric data that is represented as an unordered set of points; fingerprint minutiae fall in this category. In order to construct a fuzzy vault, the external key is converted into a polynomial and the minutiae are evaluated on that polynomial. These evaluations are stored along with the original minutiae as tuples. The biometric information is then secured by storing the tuples among a large number of randomly generated chaff points. During authentication, the query biometric is used to identify the legitimate minutiae in the set containing the minutiae as well as the chaff points. The evaluations, or the ordinate values, corresponding

☆ Research supported by Army Research Office Grant W911NF-06-1-0418.

* Corresponding author. Tel.: +1 517 285 3592.
E-mail addresses: nagarabh@cse.msu.edu (A. Nagar), knandakumar@i2r.a-star.edu.sg (K. Nandakumar), jain@cse.msu.edu (A.K. Jain).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.07.003

to the secured polynomial are then used to reconstruct the polynomial thereby revealing the key. Though fuzzy vault is able to effectively secure the fingerprint templates, the recognition accuracy of the resultant system is significantly lower compared to the accuracy on the original template. One reason for this is the inability of the fuzzy vault to effectively utilize salient information in a fingerprint other than minutiae. We address this limitation of fuzzy vault by incorporating minutiae descriptors in the vault construction. Figs. 1 and 4 show the fuzzy vault encoding and decoding procedures along with the technique to incorporate minutiae descriptors.

Minutia descriptors consist of the ridge frequency and ridge orientation information around a minutia point (Feng, 2008). Note that storing the descriptors along with minutiae in the vault is not recommended as the descriptors can be used to verify whether two neighboring minutiae belong to the same fingerprint or not. Thus, instead of explicitly storing the minutiae descriptors in the vault, we "encrypt" the ordinate values corresponding to the minutiae using the associated minutiae descriptors. Due to the intra-user variations in the minutiae descriptor values, standard cryptographic algorithms such as the Advanced Encryption Standard (AES) cannot be used to encrypt the ordinate values. Therefore, another bio-crypto algorithm called the fuzzy commitment technique (Juels and Wattenberg, 1999) is used to associate the ordinate value, which serves as the key, with a minutia descriptor. Since the actual matching descriptors will be able to "decrypt" the ordinate values, and hence decode the vault with a higher probability than a non-matching descriptor, the proposed hybrid cryptosystem improves the matching performance as well as security of the vault.

One limitation of a fuzzy commitment scheme designed using typical algebraic codes [e.g. BCH codes (Reed and Chen, 1999)] is that these codes do not meet the Hamming bound (see Reed and Chen, 1999, p. 105). As a result, such codes may produce a *decoding failure* when one attempts to use a non-matching descriptor to release the key. Consequently, an adversary can decrypt the ordinate values individually by trying all possible descriptors (which is not difficult given the low discriminability of descriptors). We experimentally show that such an attack is indeed feasible when the dimensionality of descriptor is high (see Section 6). Further, we show that principal component analysis (Duda et al., 2000) and sequential forward floating search (Pudil et al., 1994) can be effectively used to reduce the dimension of the descriptors thereby reducing the chances of a successful attack. A technique for encrypting ordinate values using just minutia orientation was also proposed by Mihailescu (2007) but without any implementation.

Section 2 describes the helper data extraction procedure which consists of (i) fuzzy vault encoding and (ii) securing ordinate values using fuzzy commitment. Section 3 describes the authentication procedure. In Section 4, different stages involved in obtaining a binary vector from a minutia descriptor are described. Section 5 provides the experimental results corresponding to different binarization schemes used for descriptors. Section 6 describes techniques to measure the security improvement using the proposed approach. In this section, we also discuss some strategies that an adversary can use to compromise the system and the security of our system against those strategies. Section 7 summarizes our conclusions and the future enhancements.

2. Helper data extraction (enrollment)

In the proposed fingerprint cryptosystem, helper data extraction consists of two main steps: (i) fuzzy vault encoding and (ii) securing ordinate values (see Fig. 1). The first step consists of securing the minutiae location and direction using the fuzzy vault framework as described in (Nandakumar et al., 2007). In the second step, the ordinate values of the vault are secured using the minutiae descriptors through the fuzzy commitment approach.

2.1. Fuzzy vault encoder

During vault encoding a 16-bit Cyclic Redundancy Check (CRC) code is appended to a $16n$-bit key K and divided into $(n+1)$ blocks of 16 bits each. These $(n+1)$ values serve as the coefficients of a polynomial f of degree n in the Galois field $GF(2^{16})$. The template minutiae are sorted according to their quality and well-separated minutiae (Nandakumar et al., 2007) are selected for constructing the vault. If the desired number of well-separated minutiae (say r) cannot be obtained, we count it as a Failure to Capture Error (FTCR). The location and orientation of each minutia is encoded as an element in $GF(2^{16})$. The minutiae $x_i, i = 1, \ldots, r$ along with their corresponding polynomial evaluations $f(x_i), i = 1, \ldots, r$ are stored in the vault V. A set of s chaff points $\{(y_j, z_j), j = 1, \ldots, s\}$ is generated randomly such that $y_j \neq x_i, \forall i = 1, \ldots, r; j = 1, \ldots s$ and $z_j \neq f(y_j), \forall j = 1, \ldots, s$. The chaff point set is added to the vault V which can now be represented as $V = (A, B)$, where A and B are the sets of $(r+s)$ abscissa and ordinate values in the vault, respectively. Points with high ridge curvature are extracted from the fingerprint and stored along with the vault to be used for alignment during authentication.

2.2. Securing ordinate values

The security of the vault described in Section 2.1 depends only on the difficulty in identifying the genuine points in the set A. Once $(n+1)$ genuine points are identified, Lagrange interpolation can be used to reconstruct the polynomial f, thereby revealing the key K. But if ordinate values corresponding to each point in the vault are encrypted, an adversary will not be able to reconstruct the polynomial even if he correctly guesses the genuine points from the vault. We use minutiae descriptors (Feng, 2008) in order to encrypt the ordinate values using fuzzy commitment approach.

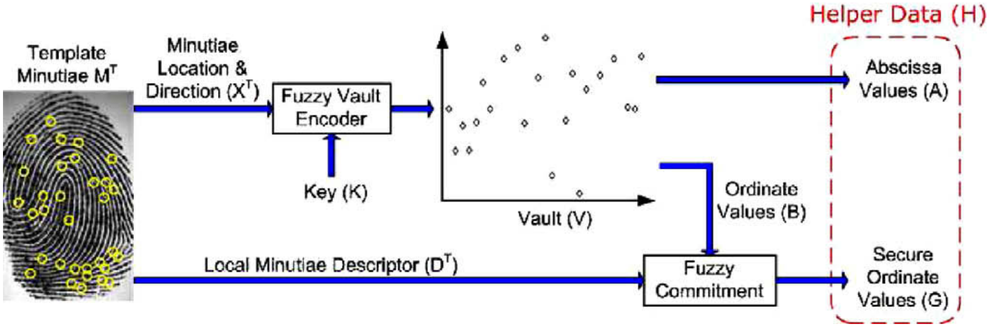

Fig. 1. Helper data extraction in the proposed fingerprint cryptosystem.

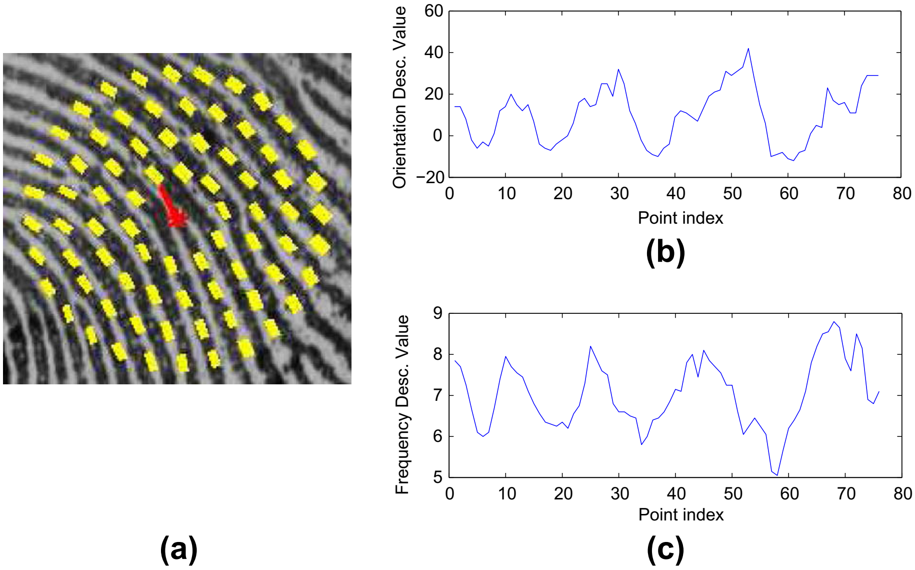

Fig. 2. Minutiae descriptor: (a) positions of 76 points around a minutiae; thickness of each line and its orientation correspond to frequency and orientation descriptors, (b) orientation and (c) frequency descriptors.

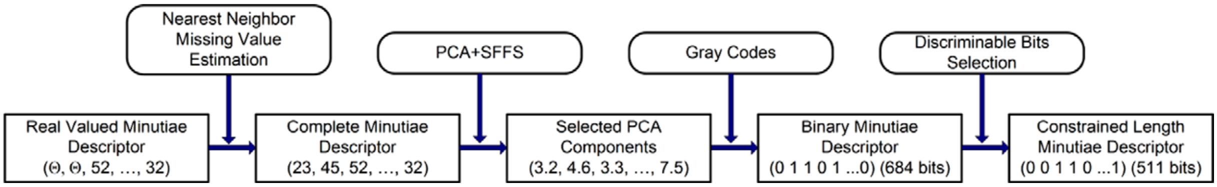
Nearest Neighbor Missing Value Estimation
PCA+SFFS
Gray Codes
Discriminable Bits Selection
Real Valued Minutiae Descriptor (Θ, Θ, 52, ..., 32)
Complete Minutiae Descriptor (23, 45, 52, ..., 32)
Selected PCA Components (3.2, 4.6, 3.3, ..., 7.5)
Binary Minutiae Descriptor (0 1 1 0 1 ...0) (684 bits)
Constrained Length Minutiae Descriptor (0 0 1 1 0 ...1) (511 bits)

Fig. 3. Different stages involved in obtaining a binary vector of desired length from raw minutiae descriptors.

A minutia descriptor consists of ridge orientation and frequency at 76 equidistant points, uniformly spaced on four concentric circles around a minutia. The four concentric circles, with radius 27, 45, 63 and 81 pixels, contain 10, 16, 22 and 28 points, respectively (see Fig. 2). This configuration of points is based on the criteria that the difference between radii of two consecutive concentric circles and that between two sampled points on a circle should be twice the ridge period. Sampling the points in this manner captures maximum information contained in the neighborhood of a minutia (Tico and Kuosmanen, 2003).

In order to use a minutia descriptor in a fuzzy commitment scheme, the descriptor needs to be converted to an m-bit binary vector, say D^b. Length of D^b is decided based on availability of an efficient error correcting code of the same length. The binarization procedure consists of four stages designed to capture the maximum possible discriminability in the minutiae descriptors: (i) missing value estimation, (ii) dimensionality reduction, (iii) binary encoding using Gray codes, and (iv) discriminable bits selection. We explain the above stages in detail in Section 4.

Next, the ordinate value for each minutia is used to obtain a codeword from an error correcting code. If the dimension of the code being used is larger than the number of bits in the ordinate value, required number of randomly generated bits are appended to the ordinate value. In case the dimension of the code is smaller than the number of bits in ordinate value, first k bits are used as the *message*, where k is the dimension of the code.

Let $D_i^b, i = 1, \ldots, (r+s)$ be the descriptor in binary format and C_i be a codeword generated from the corresponding 16-bit ordinate value B_i. Now, instead of the ordinate value B_i, only the secure ordinate value, $G_i (= (D_i^b \oplus C_i))$, is stored in the vault.[1] Note that the descriptors for the chaff points are chosen at random from the set of all the descriptors in the database. The set of abscissa values A, the set of secure ordinate values G and the high curvature points together constitute the helper data in our fingerprint cryptosystem.

3. Authentication

During authentication (see Fig. 4), the query fingerprint is first aligned using the high curvature points of the template and query fingerprints as described in (Nandakumar et al., 2007). Then, r well separated and good quality minutiae are selected from the query and matched with the points in the vault in order to filter out most of the chaff points. Further, the minutiae descriptors are extracted from the fingerprint and are binarized using the same procedure as in the enrollment stage. An XOR operation is applied between the descriptor (say D'^b) associated with each selected query minutia and the corresponding secure ordinate value to obtain the corresponding word C'. This word is then decoded to obtain the *message*, which represents the ordinate value corresponding to that minutia (see Fig. 5). If the ordinate value is correctly decoded for some minimum number $(n+1)$ of genuine points in the vault, the polynomial f is correctly reconstructed indicating a successful match.

4. Descriptor binarization

The fuzzy commitment scheme requires the biometric features to be in the form of binary vectors. Further, it is desirable that the Hamming distance between the matching and non-matching descriptors be as far apart as possible. In order to achieve this, we follow a four stage binarization scheme consisting of (see Fig. 3)

[1] $\oplus$ denotes the XOR operation.

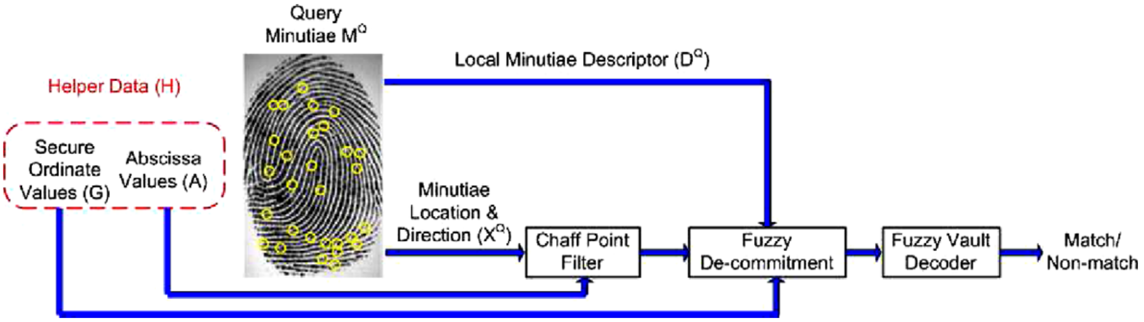

Fig. 4. Authentication using the proposed fingerprint cryptosystem.

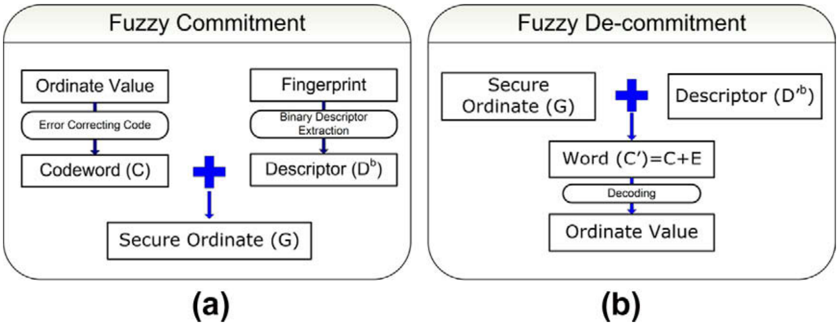

Fig. 5. Fuzzy commitment and de-commitment procedure for securing ordinate values.

(i) Missing value estimation
(ii) Dimensionality reduction
(iii) Binary encoding using Gray codes
(iv) Discriminable bits selection

4.1. Estimating missing values for minutiae descriptors

The descriptors corresponding to minutiae near the fingerprint boundary tend to have many missing values because only a part of the neighborhood of such minutiae lies within the fingerprint region (foreground). We estimate the missing values from the *k*-nearest descriptors of a given descriptor in the database. The nearest neighbor based approach is expected to provide realistic and reliable estimates as it selects the missing values from similar descriptors.

The missing values in the descriptors are estimated in the following manner. First, we find the *k*-nearest neighbors of the given descriptor among a set of desired descriptors in the database. A set of desired descriptors is selected such that 75% of the values available in the given descriptor are available in the selected descriptors as well. The missing values are then estimated as the average of the available values in the *k*-nearest neighbors. The nearest neighbors and missing values are computed separately for the orientation values and the frequency values. In case of orientation values, distance between two descriptors is computed as the normalized sum of the distance between the individual values. Let the set of orientation values of two descriptors D^1 and D^2 being matched be $D_o^1 = \{d_{o1}^1, d_{o2}^1, \ldots, d_{om}^1\}$ and $D_o^2 = \{d_{o1}^2, d_{o2}^2, \ldots, d_{om}^2\}$. The distance between two orientation descriptors is given by

$$d(D_o^1, D_o^2) = \frac{\sum_{i=1}^{m} min(|d_{oi}^1 - d_{oi}^2|, 180 - |d_{oi}^1 - d_{oi}^2|) mask_{oi}}{\sum_{i=1}^{m} mask_{oi}}, \quad (1)$$

where $mask_{oi}$ has a value 1 if both the d_{oi}^1 and d_{oi}^2 are inside the fingerprint region (foreground) and 0 otherwise. If the k nearest neighbors of ith descriptor are $D_o^{(1)}, D_o^{(2)}, \ldots, D_o^{(k)}$ then the estimated orientation descriptors are given by:

$$d_{oj}^i = \frac{1}{2} atan\left(\frac{\sum_{l=1}^{k} \sin(2d_{oj}^{(l)}) mask_{oj}^{(l)}}{\sum_{l=1}^{k} \cos(2d_{oj}^{(l)}) mask_{oj}^{(l)}}\right), \quad (2)$$

where $mask_{oj}^{(l)}$ has value 1 if $d_{oj}^{(l)}$ is in the foreground.

The missing values for the ridge frequency are also computed in a similar way by changing the distance measure between descriptors and the function that combines multiple descriptors to estimate the missing value. Distance between two frequency descriptors is given by

$$d(D_f^1, D_f^1) = \frac{\sum_{i=1}^{m} |d_{fi}^1 - d_{fi}^2| mask_{fi}}{\sum_{i=1}^{m} mask_{fi}} \quad (3)$$

and frequency values estimated from the k neighbors of i^{th} descriptor are given by

$$d_{fj}^i = \frac{\sum_{l=1}^{k} d_{fj}^{(l)} mask_{fj}^{(l)}}{\sum_{l=1}^{k} mask_{fj}^{(l)}}, \quad (4)$$

where $mask_{fj}^{(l)}$ has value 1 if $d_{fj}^{(l)}$ is in foreground. A small fraction of the descriptor values that could not be estimated using the above procedure can be interpolated as weighted average of the neighboring values. Fig. 6 compares the orientation component of the descriptors where missing values were estimated using the nearest neighbor approach and the simple interpolation scheme. We observe that the values estimated using the nearest neighbor based technique is more similar to the real descriptor values in a matching descriptor (obtained from the same minutiae in a different impression of the same finger) compared to the simple interpolation scheme.

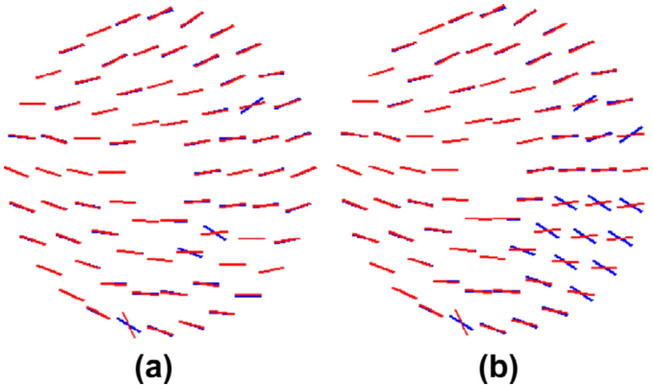

Fig. 6. Estimating missing values in descriptors: (a) orientation of two matching descriptors overlaid where missing values were estimated using the nearest neighbor approach; (b) orientation of the same descriptors when simple interpolation is used for estimating the missing values. It can be observed that there are very few inconsistent orientation values in case nearest neighbor approach is used.

4.2. Dimensionality reduction for minutiae descriptors

As noted in (Nagar et al., 2008), use of long binary minutiae descriptors degrade the system security. This is because the error correcting codes compatible with such descriptors are more likely to produce decoding failures when an adversary attempts to decrypt an ordinate value using a non-matching descriptor (see Section 6). A decoding failure occurs when the distance between the corrupted word and any codeword is larger than the error correction capability of the code.

To mitigate this problem, we reduce the dimensionality of the minutiae descriptor using principal component analysis (PCA) (Duda et al., 2000) and sequential forward floating search (Pudil et al., 1994) in order to detect the most informative components. One of the motivations for using PCA is the fact that the different elements of minutiae descriptors are highly correlated resulting in strongly correlated bits in the binarized descriptor. The use of PCA is expected to lead to uncorrelated bits in the binarized descriptor.

Since the orientation values do not belong to Euclidean space, direct application of PCA is not expected to produce meaningful components. Thus, orientation descriptor is now represented as:

$$D_{o'} = [\cos(2d_{o1})\sin(2d_{o1})\cos(2d_{o2})\sin(2d_{o2})\ldots\cos(2d_{om})\sin(2d_{om})]. \tag{5}$$

The complete descriptor thus becomes

$$D = [\cos(2d_{o1})\sin(2d_{o1})\cos(2d_{o2})\sin(2d_{o2})\ldots\cos(2d_{om}) \times \sin(2d_{om})d_{f1}d_{f2}\ldots d_{fm}]. \tag{6}$$

PCA is now applied to the descriptor represented in Eq. (6) to obtain the uncorrelated components. Further, since certain components might be very noisy, we apply a supervised feature selection. Among the various feature selection algorithms available (Peng et al., 2005; Jain and Zongker, 1997), sequential forward floating selection algorithm (SFFS) (Pudil et al., 1994) is simple to implement and provides good performance. We use the False Accept Rate (FAR) at the 98% Genuine Accept Rate (GAR) as the objective function for selecting salient features using SFFS. The descriptors are matched using the Euclidean distance. Once the desired number of features are selected, they are binarized using the scheme described in Section 4.3.

4.3. Binarizing minutiae descriptors

The main objective of a minutia descriptor binarization scheme is to generate binary descriptors having small dimensionality that retain the maximum possible discriminability. Large discriminability will allow correct decoding of ordinate values using descriptors for genuine matches and at the same time limit the correct decoding for an impostor pair. Small dimensionality on the other hand will limit the decoding failures leading to greater security against impostor attacks.

Binarization of a continuous value feature consists of two parts: (i) quantization, and (ii) bit assignment. We perform uniform quantization for the descriptor values based on their maximum and minimum values. The number of bins used for quantization is of the form 2^α where α is a positive integer. Due to the intra-user variation in the minutiae descriptors, it is desirable that the number of bit differences among the representations of the adjacent quanta should be minimum. That is, if a value lies close to the boundary of a quantum and due to intra class variation, it shifts to the next quantum in the matching descriptor, the penalty paid in terms of the Hamming distance of the binarized descriptors should be minimum i.e. only 1 bit. Gray codes (Gray, 1953) are well known codes designed specifically for this purpose. Table 1 shows a 3-bit Gray code.

A noticeable fact about the gray codes is that the first and the last quanta also have only a single bit difference. This fact is beneficial in case the relative orientation values are directly binarized without dimensionality reduction since the relative orientation value of −90° is the same as the orientation value of 90°. Since the total number of bits generated from a minutia descriptor may not be the same as the length of a desirable error correcting code, it is desirable to appropriately select certain bits. To this end, we employ a supervised bit selection procedure. This procedure selects a set of bits such that they have minimum variation among the matching descriptors and maximum variation among the non-matching descriptors.

In order to obtain a measure of the intra-class and inter-class variations of a particular bit, we first obtain the genuine and impostor matching minutiae from the database. The variation in genuine matching descriptors corresponds to the fraction of genuine matches that have different values for the particular bit in consideration. The variation in impostor matching descriptors thus corresponds to the fraction that have different bit values for the impostor matches. Let the intra-class variation of the ith bit be σ_G and inter-class variation be σ_I. The discriminability index for each bit is given by

$$\Gamma = \alpha_d\sigma_I - (1 - \alpha_d)\sigma_G, \tag{7}$$

where $\alpha_d \in \{0, 1\}$ is constant. A total of γ bits having the largest discriminability index are selected to constitute the binary descriptor.

5. Experiments

We used the FVC2002 DB2 fingerprint database to compare the fuzzy vault performance with and without minutiae descriptors. As in (Nandakumar et al., 2007), only the first two impressions of the 100 different fingers the database were used in the experiments, one as the template and the other as the query. During both

Table 1
3-Bit Gray code. Note that adjacent quanta differ in only a single bit.

Quantum index	Gray code
1	000
2	001
3	011
4	010
5	110
6	111
7	101
8	100

enrollment as well as authentication, the missing descriptor values are estimated using the 10-nearest neighbor approach as described in Section 4.1. The nearest neighbors are found among the descriptors corresponding to all the minutiae extracted from all images in FVC02 DB2; there are around 27,000 descriptors in all. The orientation and frequency values of the descriptors are quantized separately into 2^5 and 2^4 values, respectively, and binarized using Gray codes as described in Section 4.3 to obtain 684 bits. From these 684 bits, 511 bits are selected using the bit selection scheme as described in Section 4.3. The BCH(511,19) error correcting scheme is used for generating the fuzzy commitment that can correct up to 119 errors. Fig. 7 shows the GAR and FAR values corresponding to the fuzzy vault implementation in (Nandakumar et al., 2007) (without descriptors) and the proposed implementation where minutiae descriptors are used (Desc (511,19)). Failure to capture rate in both cases is 2%. We observe that the use of minutiae descriptors reduces the FAR of the system significantly, while the GAR remains nearly the same. For instance, when the degree of the polynomial is 6, the GAR is 95% for both the scenarios. However, the FAR is 0.7% when the descriptors are not used and 0.01% when the proposed cryptosystem is used. These estimates of GAR and FAR are based on 100 genuine matches and 9900 impostor matches.

The principal component analysis (PCA) is further used to reduce the dimensionality of the descriptors as described in Section 4.2. The covariance matrix of the descriptors values, that is required for computing the principal components, is computed using the descriptors available in the database. First 10 principal components are retained and each one is quantized into 2^7 bins. A 7-bit Gray code is used to binarize each of the 10 components. Note that PCA and the desired components can be computed off-line once for all. Fig. 7 shows the FAR as well as the GAR corresponding to 31-bit as well as 15-bit descriptors obtained by selecting 31 and 15 bits, respectively, from the available 70bits. It can be seen there is slight degradation in the GAR because of dimensionality reduction from 95% to 94% and 93%, respectively, for 31 and 15 bits descriptors. However, as described in Section 6, the security is increased by around 10 bits in case a 15-bit descriptor is used.

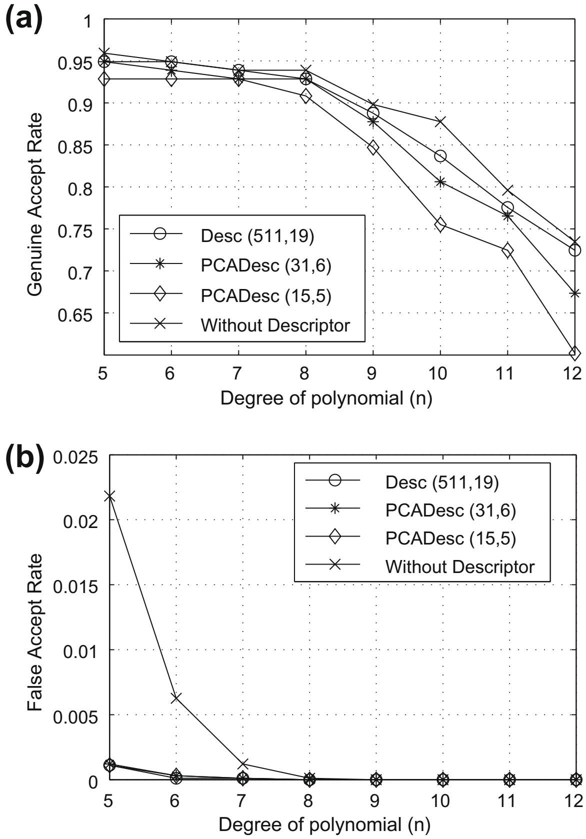

Fig. 7. GAR (a) and FAR (b) for the fuzzy vault with and without descriptors. "Desc (511,19)" corresponds to case when orientation values are quantized into 2^5 quanta, ridge frequency values are quantized into 2^4 quanta and 511 bits are extracted from them. Here the fuzzy commitment scheme is constructed using BCH(511,19) code. "PCADesc (31,6)" and "PCADesc (15,5)" correspond to cases when 10 principal components are extracted and each value is divided into 2^7 quanta. In "PCADesc (31,6)", 31 bits are extracted and BCH(31,6) code is used for fuzzy commitment whereas in "PCADesc (15,5)" 15 bits are extracted and BCH(15,5) code is used. BCH(511,19) corrects up to 119 errors, BCH(31,6) corrects up to seven errors and BCH(15,5) corrects up to three errors.

6. Security analysis

Nandakumar (2008) showed that the min-entropy (Dodis et al., 2006) of the minutiae template M^T given the vault V can be computed as

$$H_\infty(M^T|V) = -\log\left(\frac{\binom{r}{n+1}}{\binom{r+s}{n+1}}\right), \tag{8}$$

where r, n and s have the same meaning as in Section 2.1. This derivation is based on the assumption that both the minutiae location and minutiae orientation are uniformly distributed. The fuzzy vault implementation in (Nandakumar et al., 2007) uses the values of $r = 24$, $s = 200$ and $n = 8$ for the typical vault construction. Based on the above analysis, the security of the fingerprint fuzzy vault implementation in (Nandakumar et al., 2007) is approximately 31 bits. This is equivalent to a randomly chosen four character password which requires around a billion trials on average to break the system.

In the proposed fingerprint fuzzy vault the true ordinate values can be obtained in two ways: (i) directly guessing the 16-bit ordinate values and (ii) guessing the descriptors associated with each minutia. Since the ordinate values of the genuine points are obtained through an evaluation of a randomly generated secure polynomial, it is reasonable to assume that the difficulty of directly guessing an ordinate value is approximately 16 bits (assuming there are more than 16 information bits in the error correcting code, otherwise it is the number of information bits of the code). Also since the adversary has to simultaneously guess $(n+1)$ ordinate values correctly, this corresponds to approximately $16(n+1)$ bits of security.

In order to estimate security against guessing the descriptor, let the entropy of a minutia descriptor D be I_D bits and say ρ bits out of these should be corrected. As shown by Hao et al. (2006), the difficulty in guessing a minutiae descriptor is approximately $R = \log(2^{I_D} \Big/ \binom{I_D}{\lceil\rho\rceil})$ bits. Since the adversary has to simultaneously guess $(n+1)$ minutiae descriptors correctly, using minutiae descriptors provides approximately $(n+1)R$ bits of security. Although the length of descriptor is N bits, there is a strong correlation between the descriptor bits leading to reduction in effective entropy of the descriptor bits i.e. I_D. We empirically determine that approximately $N/4$ bit errors need to be corrected in order to preserve the GAR to a large extent. Thus ρ can be approximated as $I_D/4$. Thus, if $n = 8$ and $I_D = 6$ bits, then $R \approx 2$ bits. In this scenario, the proposed scheme increases the security of the fuzzy vault by approximately 18 bits so the overall security now becomes $49(31+18)$ bits. This is equivalent to a six character password.

The above security analysis assumes the use of a *perfect* error correction coding scheme (a w-error correcting binary code of size 2^N is said to be perfect if for every word C', there is a unique

Table 2
The values corresponding to π_{df}, π_0, $\max_i(\pi_i)$ and T_a for the different representations of descriptors considered.

Descriptor format	Min	Max	Median
π_{df}			
Desc (511,19)	0.941	0.999	0.991
PCADesc (31,6)	0.761	0.896	0.826
PCADesc (15,5)	0.394	0.448	0.418
π_0			
Desc (511,19)	0.000	0.059	0.001
PCADesc (31,6)	0.032	0.194	0.103
PCADesc (15,5)	0.014	0.168	0.070
$max_i(\pi_i)$			
Desc (511,19)	0	0	0
PCADesc (31,6)	0.007	0.050	0.016
PCADesc (15,5)	0.048	0.128	0.074
T_a			
Desc (511,19)	0	0	0
PCADesc (31,6)	2.74	17.09	6.63
PCADesc (15,5)	16.61	48.41	27.55

codeword C such that the Hamming distance between C and C' is at most w bits). It has, however, been proven that any non-trivial perfect code over a prime-power alphabet has the parameters of a Hamming code or a Golay code (Hill, 1988). Note that Hamming codes correct only single error whereas Golay codes correct only up to three errors in code of length 24 bits and thus would not be applicable to the current problem.

If the coding scheme is not perfect, some of the words may result in a decoding failure which would indicate an incorrect minutia descriptor being used to de-commit the ordinate value. Due to unknown distribution of biometric features, it is important to empirically estimate the number of decoding failure and incorrect decodings while using a particular error correcting code. Note that even if all the incorrect descriptors lead to decoding failure, the security is at least as good as the security of the original fuzzy vault.

When an adversary applies a descriptor to decode the secure ordinate value, following situations can arise:

(i) a decoding failure is detected,
(ii) the correct codeword c is obtained and
(iii) an incorrect codeword $c_i(\neq c)$ is obtained.

We are interested in estimating the relative frequency of these three events as they provide an estimate of ambiguity about the

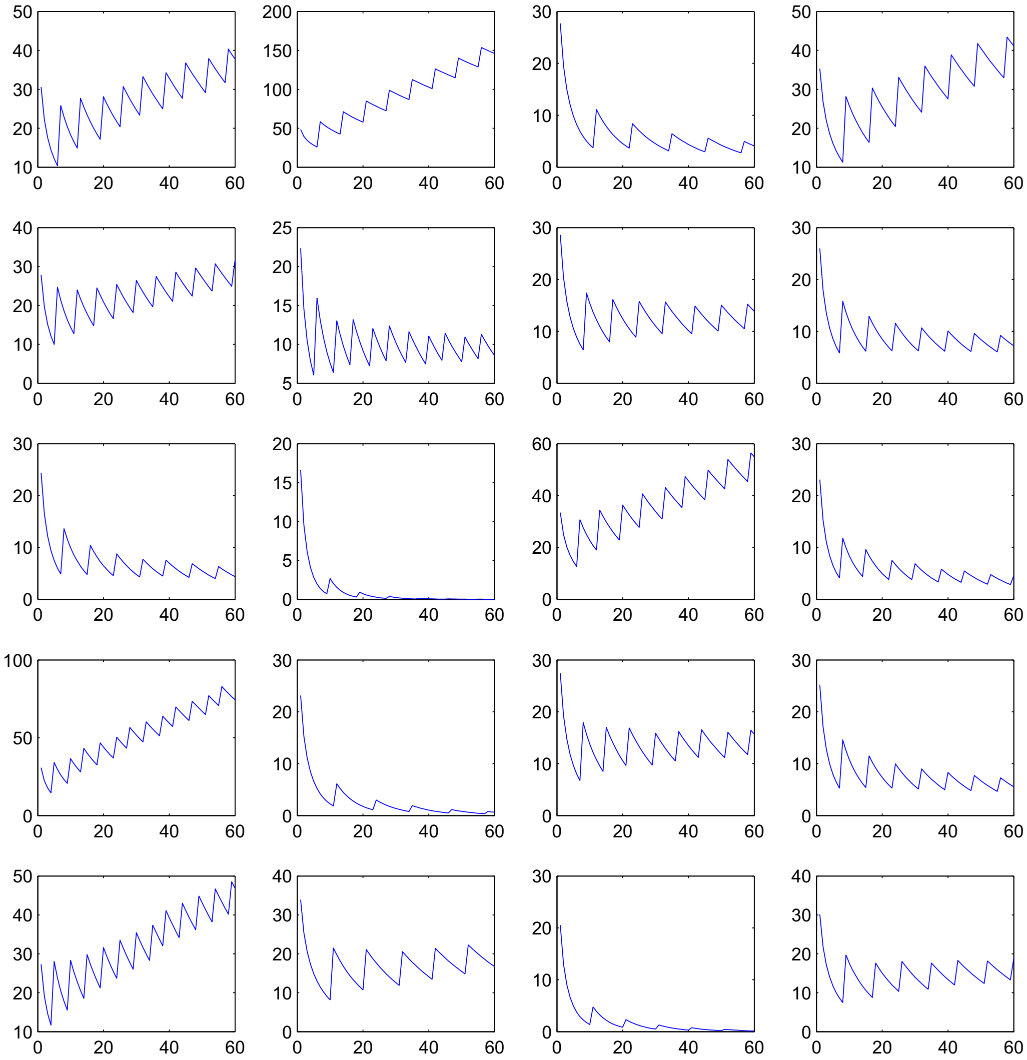

Fig. 8. Variation of bits of security induced by descriptors as $u = t(1 - \pi_{df})$ increases for "PCADesc (15,5)". Here u is the number of descriptors tried and π_{df} is the fraction of descriptors leading to decoding failure. Abscissae represents u and the ordinates represents the number of bits of security. The 20 different graphs show the variation for the 20 different randomly selected descriptors. In many cases a minimum security of around 10 bits can be imparted to the system leading to a total security of 41 (31 + 10) bits.

true codeword. Let the relative frequency of the three events be: π_{df}, π_0, and $\pi_i, i = 1, 2, \ldots$ in order. One strategy an adversary might employ would be to try decoding the ordinate value with a large number of descriptors one by one and select the first ordinate value decoded. Here the adversary would be successful with probability

$$p_a = \left\{ \frac{\pi_0}{\sum_i \pi_i} \right\}^{(n+1)} = p_0^{(n+1)}, \tag{9}$$

where $p_0 = \frac{\pi_0}{\sum_i \pi_i}$. Thus the number of bits of security added would be equal to $T_a = -\log_2 p_a$.

In order to estimate π_{df}, π_0 and π_i, we randomly selected 20 different descriptors and tried to decode those using the database containing 27,000 descriptors. Table 2 shows the values corresponding to π_{df}, π_0, $max_i \pi_i$ and T_a for the different representations of descriptors considered. It can be seen that BCH(31,6) provides around 7 bits of security on average whereas that BCH(15,5) provides around 28 bits.

In our experiments with the imperfect codes having high dimension e.g. BCH(511,19) or BCH(31,5), it has been observed that $\pi_0 \gg \pi_i$. This can be explained by the fact that if the difference between two matching descriptors is less than the error correction capacity of the code, which is often the case, there are an acceptable number of errors introduced in the codeword leading to correct decoding. On the other hand, when a randomly selected descriptor is used to decode the fuzzy commitment, a large number of errors beyond the error correcting capacity are introduced into the codeword. Due to this the codeword is shifted to a non-decodable region with high probability leading to a decoding failure.

Note that in case of BCH(511,19), theoretical estimate of the fraction of space that is not decodable is $\sim 1 - 10^{-19}$, that for BCH(31,6) is ~ 0.9 and for BCH(15,5), it is ~ 0.44 which is consistent with the probabilities of decoding failure reported in Table 2. Also no incorrect decoding was detected in case of using BCH(511,19) due to large fraction of non-decodable region.

Another strategy that an adversary can employ is to apply t different descriptors for decoding each secure ordinate value and get the ordinate value that occurred maximum number of times. Note that, on average, there would be $u = t(1 - \pi_{df})$ different descriptors that will not produce decoding failures. Thus the adversary will succeed if there are more than up_i^{max} correctly decoded ordinate values among the set of u decoded values, where

$$p_i^{max} = \left\{ \frac{\max_i \{\pi_i; i = 1, 2, \ldots\}}{\sum_i \pi_i} \right\}. \tag{10}$$

Thus the probability of successful attack is given by

$$p'_a = \{p(\#(\text{correct codewords}) > \lceil up_i^{max} \rceil)\}^{(n+1)}, \tag{11}$$

where

$$p(\#(\text{correct codewords}) > l) = \sum_{i=l+1}^{u} \binom{u}{l} p_0^i (1 - p_0)^{u-i}. \tag{12}$$

Note that the security in terms of number of bits is given by $T'_a = -\log_2 p'_a$.

Fig. 8 shows the variation of bits of security as u increases corresponding to the case when the descriptor is represented as a 15-bit vector. It is noted that in more than half of the cases around 10 bits of security can be imparted to the fuzzy vault in case the degree of polynomial secured by the fuzzy vault, i.e. n, is 8. The "sawtooth" shape of the curves is due to the fact that in Eq. 11, $\lceil up_i^{max} \rceil$ remains the same even if u increases and that change in $\lceil up_i^{max} \rceil$ leads to larger reduction in the probability of attack as compared to the effect of increasing u. Note that the curves corresponding to descriptors that have larger values of p_i^{max} have sharper "teeth" and have high chances of leading to greater security. Although, the security depends on p_0 as well.

The increase in the number of descriptors tried by the adversary, i.e. t, also leads to increased computational complexity of the attack. Thus even though no additional information theoretic security is imparted in case of "BCH(511,19)", there is significant computational cost to the adversary in order to compromise the system due to large π_{df} leading to improvement in security to a certain extent. Note that given u, t is directly proportional to π_{df}.

7. Conclusions

Template security is critical to the integrity of a biometric system. In this paper we have shown that both the matching performance and security of a fingerprint fuzzy vault can be improved by incorporating minutiae descriptors. Experiments on a public domain fingerprint database demonstrates that the use of minutiae descriptors leads to an order of magnitude reduction in the False Accept Rate without significantly affecting the Genuine Accept Rate. Further, the vault security measured in terms of number of tries an adversary has to make in order to guess the secure key is increased. As future work, we plan to investigate nearest neighbor decoding implementations of certain error correcting codes in order to reduce the decoding failures. Since the descriptors corresponding to neighboring minutiae are correlated, we plan to investigate if an adversary can leverage such information to increase his chances of a successful attack and to what extent.

References

Boult, T.E., Scheirer, W.J., Woodworth, R., 2007. Fingerprint revocable biotokens: Accuracy and security analysis. In: Proc. CVPR. Minneapolis, pp. 1–8.

Cappelli, R., Lumini, A., Maio, D., Maltoni, D., 2007. Fingerprint image reconstruction from standard templates. IEEE Trans. Pattern Anal. Machine Intell. 29 (9), 1489–1503.

Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A., 2006. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. Tech. Rep. 235, Cryptology ePrint Archive.

Duda, R.O., Hart, P.E., Stork, D.G., 2000. Pattern Classification. Wiley-Interscience.

Feng, J., 2008. Combining minutiae descriptors for fingerprint matching. Pattern Recognition 41 (1), 342–352.

Gray, F., 1953. Pulse code communication. US Patent 2,632,058.

Hao, F., Anderson, R., Daugman, J., 2006. Combining crypto with biometrics effectively. IEEE Trans. Comput. 55 (9), 1081–1088.

Hill, R., 1988. A First Course in Coding Theory. Oxford University Press, p. 102.

Jain, A.K., Flynn, P., Ross, A.A., 2008a. Handbook of Biometrics. Springer.

Jain, A.K., Nandakumar, K., Nagar, A., 2008b. Biometric template security. EURASIP J. Adv. Signal Process. 2008, Article ID 579416, 17pp.

Jain, A.K., Zongker, D., 1997. Feature selection: Evaluation, application, and small sample performance. IEEE Trans. Pattern Anal. Machine Intell. 19 (2), 153–158.

Juels, A., Wattenberg, M., 1999. A fuzzy commitment scheme. In: Proc. 6th ACM Conf. on Computer and Comm. Security, Singapore, pp. 28–36.

Mihailescu, P., 2007. The Fuzzy Vault for Fingerprints is Vulnerable to Brute Force Attack. Available at: http://arxiv.org/abs/0708.2974v1.

Nagar, A., Nandakumar, K., Jain, A.K., 2008. Securing fingerprint template: Fuzzy vault with minutiae descriptors. In: Internat. Conf. for Pattern Recognition, Tampa.

Nandakumar, K., 2008. Multibiometric systems: Fusion strategies and template security. Ph.D. Thesis, Department of Computer Science and Engineering, Michigan State University.

Nandakumar, K., Jain, A.K., Pankanti, S., 2007. Fingerprint-based fuzzy vault: Implementation and performance. IEEE Trans. Inform. Forensics Security 2 (4), 744–757.

Peng, H.C., Long, F., Ding, C., 2005. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. 27 (8), 1226–1238.

Pudil, P., Novovicova, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern Recognition Lett. 15, 1119–1125.

Ratha, N.K., Chikkerur, S., Connell, J.H., Bolle, R.M., 2007. Generating cancelable fingerprint templates. IEEE Trans. PAMI 29 (4), 561–572.

Reed, I.S., Chen, X., 1999. Error-Control Coding for Data Networks. Kluwer Academic Publishers.

Ross, A.K., Shah, J., Jain, A.K., 2007. From templates to images: Reconstructing fingerprints from minutiae points. IEEE Trans. Pattern Anal. Machine Intell. 29 (4), 544–560.

Savvides, M., Vijaya Kumar, B.V.K., 2004. Cancellable biometric filters for face recognition. In: Proc. ICPR, vol. 3, Cambridge, pp. 922–925.

Sutcu, Y., Li, Q., Memon, N., 2007. Protecting biometric templates with sketch: Theory and practice. IEEE Trans. Inform. Forensics Security 2 (3), 503–512.

Teoh, A.B.J., Toh, K.-A., Yip, W.K., 2007. 2^N discretisation of biophasor in cancellable biometrics. In: Proc. ICB, Seoul, pp. 435–444.

Tico, M., Kuosmanen, P., 2003. Fingerprint matching using an orientation-based minutia descriptor. IEEE Trans. Pattern Anal. Mach. Intell. 25 (8), 1009–1014.