Feature Transformation of Biometric Templates for Secure Biometric Systems Based on Error Correcting Codes

Yagiz Sutcu, Shantanu Rane, Jonathan Yedidia, Stark Draper, Anthony Vetro

Abstract

Secure storage of biometric templates is extremely important because a compromised biometric cannot be revoked and replaced an unlimited number of times. In many approaches proposed for secure biometric storage, an error correcting code (ECC) is applied to the enrollment biometric and the resulting parity or syndrome symbols are stored on the access control device, instead of the original biometric. The principal challenge here is that most standard ECCs are designed for memoryless channel statistics, whereas the variations between enrollment and probe biometrics have significant spatial correlation. To address this challenge, we propose to transform the original biometric into a feature vector that is explicitly matched to standard ECCs, thereby improving the security-robustness tradeoff of the overall biometric system. As a concrete example, we transform fingerprint minutiae maps into feature vectors compatible with ECCs designed for a binary symmetric channel. We conduct a statistical analysis of these feature vectors and show how our feature transformation algorithm may be combined with Low-Density Parity Check (LDPC) codes to obtain a secure fingerprint biometric system.

Feature Transformation of Biometric Templates for
Secure Biometric Systems based on Error Correcting Codes

Yagiz Sutcu*, Shantanu Rane†, Jonathan S. Yedidia†, Stark C. Draper‡ and Anthony Vetro†
* Polytechnic University, Brooklyn, NY 11201, `yagiz@isis.poly.edu`
† Mitsubishi Electric Research Labs, Cambridge, MA 02139, `{rane,yedidia,avetro}@merl.com`
‡ University of Wisconsin, Madison, WI 53706, `sdraper@ece.wisc.edu`

Abstract

Secure storage of biometric templates is extremely important because a compromised biometric cannot be revoked and replaced an unlimited number of times. In many approaches proposed for secure biometric storage, an error correcting code (ECC) is applied to the enrollment biometric and the resulting parity or syndrome symbols are stored on the access control device, instead of the original biometric. The principal challenge here is that most standard ECCs are designed for memoryless channel statistics, whereas the variations between enrollment and probe biometrics have significant spatial correlation. To address this challenge, we propose to transform the original biometric into a feature vector that is explicitly matched to standard ECCs, thereby improving the security-robustness tradeoff of the overall biometric system. As a concrete example, we transform fingerprint minutiae maps into feature vectors compatible with ECCs designed for a binary symmetric channel. We conduct a statistical analysis of these feature vectors and show how our feature transformation algorithm may be combined with Low-Density Parity Check (LDPC) codes to obtain a secure fingerprint biometric system.

1. Introduction

Computer-verifiable biometrics have emerged as an attractive alternative to traditional passwords and identifying documents. Their advantages include the fact that unlike passwords, they cannot be forgotten and unlike identifying documents, they are difficult to forge. One of the biggest challenges to the wide applicability of biometric systems is secure storage of biometric templates. This is because of privacy concerns as well as the fact that, unlike passwords or credit card numbers, personal biometrics cannot be renewed arbitrarily, since there is a limited number of fingers, eyes, faces, or postures available. Securely storing a biometric would greatly alleviate the privacy concerns of the public regarding biometrics. However, while passwords or ID numbers can be securely stored via a cryptographic hash, this solution is not immediately applicable to biometrics. This is because of the noisy nature of personal biometrics. Every time a biometric is measured, the observation differs slightly. For example, a fingerprint reading might change because of elastic deformations in the skin when placed on the sensor surface, dust or oil between finger and sensor, or a cut to the finger. Biometric authentication systems must be robust to such variations, which are not encountered in traditional password-based authentication systems . Currently, most biometric authentication systems solve this problem using pattern recognition. To perform recognition, the enrollment biometric is stored on the device for comparison with the probe biometric. This creates a security hole: An attacker who gains access to the device also gains access to the biometric template. This is clearly a serious problem, made worse by the fact that an individual cannot generate new biometrics if the system is compromised.

To prevent access to the original biometric, a set of features extracted from the enrollment biometric sample may be stored at the device, instead of storing the biometric sample itself. Authentication then involves a comparison between the features extracted from the enrollment and probe biometrics. Notable among such approaches are cancelable biometrics [1, 2], score matching-based approaches [3] and threshold-based biohashing [4]. In general, given only the extracted features, it is very difficult for an attacker to recover the original biometric. However, it is difficult to rigorously prove that the system is secure when the feature extraction algorithm itself is compromised.

Recently, error correction coding has been proposed to deal with the joint problem of providing security against attackers while accounting for the inevitable variability of biometrics. The cryptographic primitive known as "secure sketch" proposed in [5] can also be viewed as theoretically equivalent to error correction coding. On the one hand, the error correction capability of a channel code can accommodate the slight variation between multiple measurements of

the same biometric [6, 7]. On the other hand, the check bits of the error correction code can perform much the same function as a cryptographic hash of a password on conventional access control systems. Just as a hacker cannot invert the hash and steal the password, he cannot use just the check bits to recover and steal the biometric. However, it has been found that schemes based on this principle [8, 9, 10] yield high false reject rates. One reason for this is that the statistical relationship between the enrollment biometric and probe is not accurately captured by the simple noise models assumed in the theoretical works [6, 7]. In this paper, we propose to transform fingerprint biometrics into binary feature vectors which are i.i.d. Bernoulli(0.5), independent across different users but different measurements of the same user are related by a binary symmetric channel with crossover probability p (BSC-p) where p is small. The advantage of this approach is that the BSC-p is a standard channel model for many error correcting codes. Techniques for construction, encoding and decoding of such codes are already well-understood and deeply explored. Thus, the emphasis of this paper is on the design of feature vectors which have useful statistics for secure pattern matching based on error correcting codes.

This paper is organized as follows. In Section 2, we enumerate the statistical properties that feature vectors should possess in order to be compatible with an error correcting code designed for binary symmetric channels. In Section 3, we describe two variants of a feature transformation algorithm which transforms fingerprint biometrics into feature vectors having these desired properties. In Section 4, we use this algorithm to extract feature vectors from a fingerprint database and evaluate them for security and robustness. In Section 5, we show how a practical secure biometric system is built by applying Low-Density Parity Check (LDPC) coding to the feature vectors.

2. Desired Properties of Feature Vectors

As noted in the introduction, our objective is to generate feature vectors explicitly matched with error correcting codes for binary symmetric channels (BSC). At the same time, we desire that the feature vectors must be secure in the sense that they should leak minimum information about the original biometric. To satisfy the above constraints, the feature vectors must have the following properties:

1. A bit in a feature vector representation is equally likely to be a 0 or a 1.

2. Different bits in a given feature vector are independent of each other, so that a given bit provides an attacker with no information about any other bit.

3. Feature vectors **A** and **B** from different fingers are independent of each other, so that one person's feature

vector provides no information about another person's feature vector.

4. Feature vectors **A** and **A**′ obtained from different readings of the same finger are statistically related by a BSC-p. If p is small, it means that the feature vectors are robust to repeated noisy measurements with the same finger. Then, using syndromes from an error correcting code with an appropriate rate, it is possible to estimate the enrollment biometric when provided with probe feature vectors from the enrollee.

3. Feature Transformation Algorithm

3.1. Motivation

To transform a minutiae map into a N-bit feature vector, it suffices to ask N "questions," each with a binary answer. A general framework to accomplish this is shown in Fig. 1. N operations are performed on the biometric to yield a non-binary feature representation which can then be converted to binary by thresholding. As an example, one can project the minutiae map onto N orthogonal basis vectors and quantize the positive projections to 1's and negative projections to 0's [4].

In a recent study [11], minutiae maps are transformed into integer-valued feature vectors by determining the difference in the number of minutiae points on either side of a number of randomly generated lines. Application of Principal Component Analysis (PCA) followed by a thresholding operation results in a binary output vector. A related scheme, albeit for face biometrics, is described in [12], in which a bank of Gabor filters is used to transform the face images into real-valued feature vectors. For a given enrolled face image, those vector components that are far away from their global means are deemed as reliable components in the sense that, if these components are binarized with the mean as a threshold, they will retain their binary value over multiple measurements of the face biometric.

3.2. General framework for binarization

Inspired by the above approaches, we propose a method to generate binary feature vectors from minutiae maps. Instead of using random lines as in [11], we consider random closed 3D regions (cuboids). The reasons behind using cuboids are (a) Bits can be extracted not only from the (x, y) locations of the minutiae points but also from their orientations θ (b) Since cuboids are closed regions, it is intuitively easier to see that overlapping cuboids would produce correlated bits.

We define an "operation" as counting the number of minutiae points that fall in a randomly chosen cuboid in $X - Y - \Theta$ space, as shown in Fig. 2. To chose a cuboid, an origin is selected uniformly at random in $X - Y - \Theta$
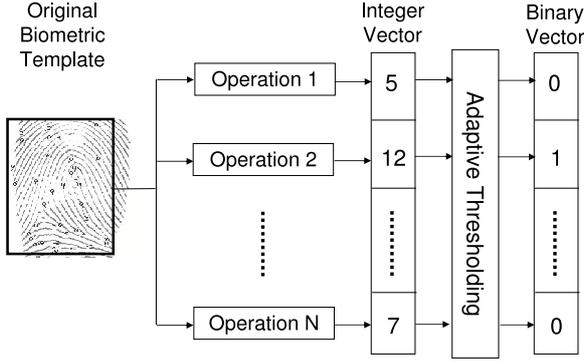
Figure 1. N questions can be asked by performing N operations on the biometric followed by thresholding. In our scheme, the operation involves counting the minutiae points in a randomly generated cuboid.
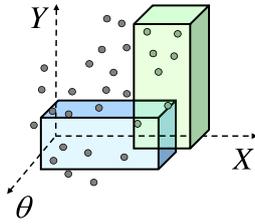

Figure 2. To obtain a binary feature vector, the number of minutiae points in a cuboid is thresholded w.r.t the median number of minutiae points in that cuboid calculated over the entire dataset. Overlapping cuboid pairs will result in correlated bit pairs.

space, and the dimensions along the three axes are also chosen at random. Next, we define the threshold as the median of the number of minutiae points in the chosen cuboid, measured across the complete training set. The threshold value may differ for each cuboid based on its position and volume. If the number of minutiae points in a randomly generated cuboid exceeds the threshold, then a 1-bit is appended to the feature vector, otherwise a 0-bit is appended. We consider the combined operation of (a) generating a cuboid and (b) thresholding as equivalent to posing a question with a binary answer. N such questions result in an N-bit feature vector.

A straightforward way to generate feature vectors is to use the same questions, i.e., the same set of cuboids, for all enrolled users. It is not difficult to see that the amount of overlap between the cuboids affects the pairwise bit correlations in the resulting binary representation. We define the overlap measure as $O_{i,j} = \frac{V_{i \cap j}}{V_{i \cup j}}$, where $V_{i \cap j}$ and $V_{i \cup j}$ are the volumes of the intersection and union of cuboids i and j.

Fig. 3(a) shows the relation between pairwise entropy of the binary feature vectors and the overlap between the corresponding pairs of cuboids used. Overlapping cuboids generate bit-pairs with high correlation, i.e., low pairwise entropy. To improve the pairwise entropy profile, we first
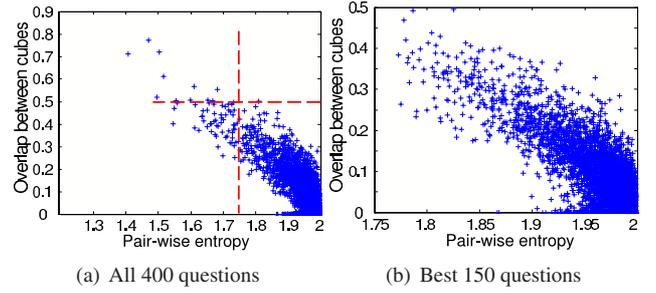

(a) All 400 questions (b) Best 150 questions

Figure 3. (a) Overlapping cubes generate bit pairs with high correlation (low pairwise entropy), simplifying the task of the attacker. (b) While choosing a subset of questions, pairs with lowest pairwise entropy are discarded.

locate the pair of cuboids with the lowest pairwise entropy. From this pair, we eliminate the cuboid which has lower pairwise entropy when paired with all the remaining cuboids. Fig. 3(b) shows the relation between the overlap between cuboids and the pairwise bit entropies after 250 out of 400 cuboids are eliminated as described above. After the elimination, pairwise independence of the bit pairs in the feature vectors is improved.

3.3. User-Specific Cuboids

We now consider a second approach in which the questions are user-specific. The rationale behind using user-specific questions is that some questions are more robust (reliable) than others. In particular, a question is robust if the number of minutiae points in a cuboid is far removed from the median calculated over the entire dataset. Thus, even if there is spurious insertion or deletion of minutiae points when a noisy measurement of the same fingerprint is provided at a later time, the answer to the question (0 or 1) is less likely to change. On the other hand, if the number of minutiae points is close to the median, the 0 or 1 answer to that question is less reliable. Thus, more reliable questions result in a BSC-p intra-user channel with low p. Different users have a different set of robust questions, and we propose to use these while constructing the feature vector.

The above process of selecting reliable cuboids based on the difference between the number of minutiae points in that cuboid and the median number of minutiae points is similar to the method followed in [12]. However, this process alone is not sufficient to generate binary feature vectors with the properties listed in Section 2. For example, there is no guarantee that thresholding the number of minutaie points in the randomly chosen reliable cuboids would result in an approximately equal number of 0-bits and 1-bits. Therefore, we propose a probabilistic method to select the reliable questions. For a given user i, the *average* number of minutiae points $\bar{m}_{i,j}$ in a given cuboid $\mathcal{C}_j$ is calculated over repeated noisy measurements of the same finger-
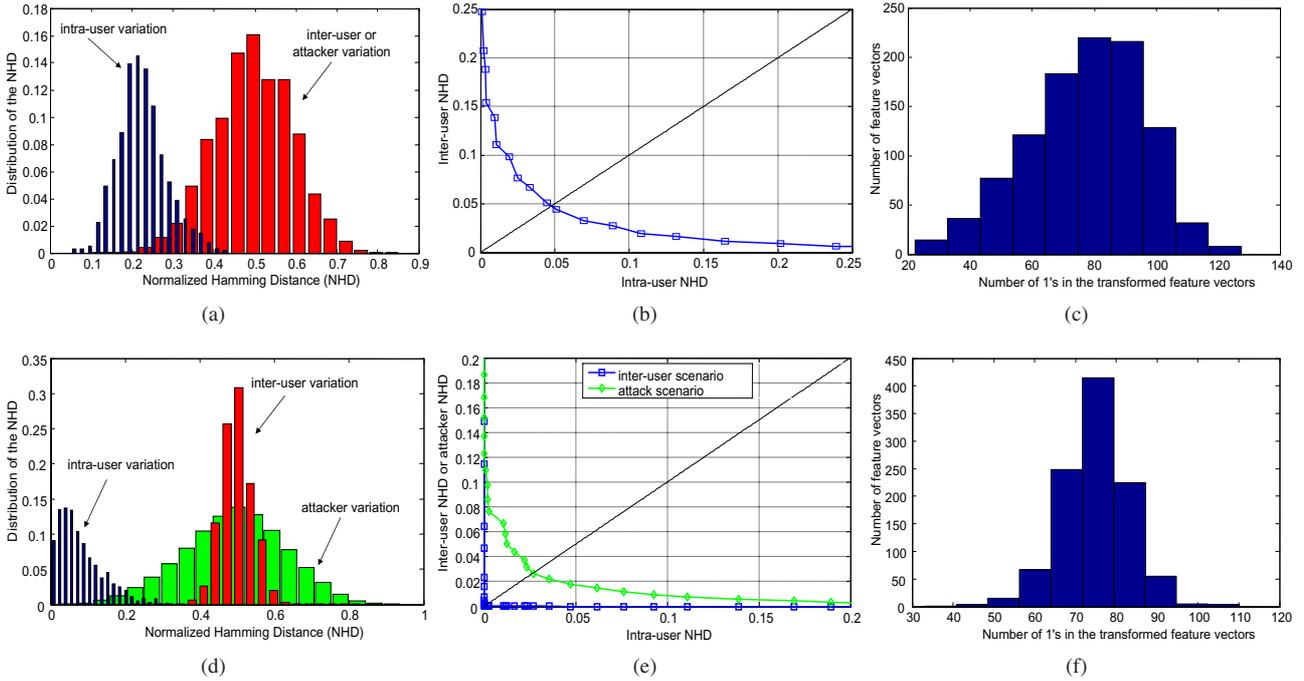
Figure 4. Statistical properties of feature vectors when the number of questions is N=150. (a,d) There is greater separation between the intra-user and inter-user distributions with user-specific questions than with common questions. (b,e) The equal error rate (EER) is 0.05 with common questions and 0.027 with user-specific questions. (c,f) The histogram of the number of ones in the feature vectors is clustered more closely around $N/2 = 75$ for the user-specific questions case compared to the common questions case.

print. Let m_j and σ_j be the median and standard deviation of the number of minutiae points in $\mathcal{C}_j$ over the dataset of all users. Then, let $\Delta_{i,j} = (\bar{m}_{i,j} - m_j)/\sigma_j$. The magnitude, $|\Delta_{i,j}|$ is directly proportional to the robustness of the question posed by cuboid $\mathcal{C}_j$ for user i. The sign of $\Delta_{i,j}$ determines whether the cuboid $\mathcal{C}_j$ should be placed into $\mathcal{L}_{0,i}$, a list of questions with a 0 answer for user i, or into $\mathcal{L}_{1,i}$, a list of questions with a 1 answer for user i. Both these lists are sorted in the decreasing order of $|\Delta_{i,j}|$. Now, a fair coin is flipped to choose between $\mathcal{L}_{0,i}$ and $\mathcal{L}_{1,i}$ and the question at the top of the chosen list is stored on the device. After N coin flips, approximately $N/2$ of the most robust questions from each list will be stored on the device. This process is repeated for each enrolled user i.

4. Experiments and Results

4.1. Data Set

In our experiments, we use a proprietary fingerprint database which contains minutiae maps of 1035 fingers with 15 fingerprint samples taken from each finger. The minutiae maps are 240×320 pixels, with an orientation $0 \leq \theta < 2\pi$ associated with each minutiae point. The average number of minutiae points in a single map is approximately 32. All fingerprints are pre-aligned with respect to a core point and the minutiae locations and orientations have been recalcu-

lated accordingly. Normalized Hamming Distance (NHD) is used to quantify the dissimilarity of two binary feature vectors.

4.2. Statistical Analysis

To measure the extent to which the desired statistical properties in Section 2 are achieved, we examine the feature vectors obtained from the minutiae maps according to the method described in Section 3. The N most robust questions were selected to generate the feature vectors, with N ranging from 50 to 350. The statistical properties of the feature vectors constructed from N=150 cuboids is shown in Fig. 4(a,b,c) for common questions and Fig. 4(d,e,f) for user-specific questions.

Fig 4(a) has only two histograms for the intra-user and inter-user variation while Fig 4(d) has a third histogram for the attacker variation. Note that, for the case of user-specific questions, the attacker variation becomes relevant if the attacker gains access to the victim's questions. The inter-user variation is relevant if the attacker has not broken into the system and has not accessed the victim's questions, but is merely trying to pose as the victim without knowing his specific questions. In a practical biometric system, the questions would not be publicized. So, most attackers will not have access to them and therefore, in most cases, the inter-user variation will be relevant instead of the more conserva-
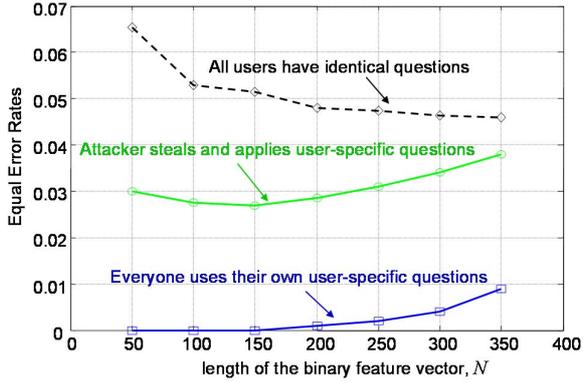
Figure 5. User-specific questions result in lower EER than common questions, even if the user-specific questions are given to the attacker.

	N	BSC crossover probability p	R_{LDPC}	No. of Bits of Security	FRR after syndrome coding	FAR after syndrome coding
User-specific cuboids	150	0.13	0.2	30	0.11	1.19×10^{-4}
	200	0.2	0.15	30	0.14	1.44×10^{-3}
	250	0.2	0.125	31.25	0.15	3.56×10^{-3}
Common cuboids	150	0.25	0.05	7.5	0.42	0.06
	200	0.25	0.05	10	0.19	0.07
	250	0.25	0.125	31.25	0.39	0.01

Figure 7. Feature vectors obtained from user-specific cuboids provide a better security-robustness tradeoff, when combined with syndrome coding than feature vectors obtained from common cuboids.

tive attacker variation.

To better evaluate the inter-user and intra-user separation from the histograms in Fig. 4(a,d) we plot the inter-user NHD (or attacker NHD for user-specific questions) against the intra-user NHD as shown in Fig. 4(b,e). The point at which these quantities are equal is defined as the Equal Error Rate (EER) for the feature vectors. User-specific questions yield a lower EER which indicates a superior security-robustness tradeoff. Fig. 5 plots the EER for various values of N. Observe that user-specific questions provide a significantly lower EER than using the same questions for all users. Even if the attacker is provided with the user-specific questions, the resulting EER is lower than the case in which everybody has the same questions.

5. A Secure Fingerprint Biometrics System

For a practical implementation in which a human biometric template is transformed into a feature vector compatible with a suitable error correcting code, consider the secure fingerprint biometrics system shown in Fig. 6. During enrollment, the user provides a fingerprint from which the system first determines a minutiae map $\mathbf{M}$. Next, the feature transformation algorithm from Section 3 maps the minutiae array into a binary feature vector $\mathbf{A}$ of fixed preset length N. Then, a syndrome encoder maps the binary feature vector into a syndrome $\mathbf{S}$, which serves as the secure biometric. In the proposed scheme, syndrome encoding is performed with the graph of an LDPC code $\mathbb{C}$. The access control system stores $\mathbf{S}$, $\mathbb{C}$ and a cryptographic hash of the binary feature vector $f_{\text{hash}}(\mathbf{A})$. It does not store $\mathbf{M}$ or $\mathbf{A}$ or the image of the original fingerprint.

During authentication, a user or attacker requests access by providing a probe fingerprint from which the authenticator obtains a minutiae map $\mathbf{L}$. Next, it transforms $\mathbf{L}$ into a probe feature vector $\mathbf{B}$. Now, the LDPC decoder assumes that the probe feature vector $\mathbf{B}$ is an error prone version of the enrollment feature vector $\mathbf{A}$. It combines the secure biometric $\mathbf{S}$ (syndrome) and the probe feature vector $\mathbf{B}$ and performs belief propagation decoding. The result of belief propagation is either an estimate $\widehat{\mathbf{A}}$ of enrollment feature vector $\mathbf{a}$, or a special symbol $\emptyset$ indicating decoder failure. Now, it is possible that $\widehat{\mathbf{A}} \neq \mathbf{A}$, yet $\widehat{\mathbf{A}}$ satisfies the syndrome $\mathbf{S}$. To protect against this possibility, and more importantly to protect against an attacker using a stolen set of syndromes to construct his own estimate $\widehat{\mathbf{A}}$ which satisfies the syndromes but is not the true biometric, access is granted if and only if $f_{\text{hash}}(\widehat{\mathbf{A}}) = f_{\text{hash}}(\mathbf{A})$.

The overall tradeoff between robustness, as measured by the False Reject Rate (FRR) and security, as measured by the False Accept Rate (FAR) is shown in Fig. 7 for feature transformation carried out both with common cuboids and user-specific cuboids. We observe that user-specific cuboids provide a better performance both in terms of the FRR-FAR tradeoff as well as in terms of the number of bits of security. The number of bits of security is calculated as NR_{LDPC}. In particular, the best overall performance is obtained for the case of 150 user-specific cuboids. We emphasize that for the purposes of security analysis, the set of questions used in the system is assumed public. An attacker who steals a set of syndromes and poses falsely as a user will be given the set of questions appropriate to that user. Security is *not* based on the obscurity of the questions, but rather on the information-theoretic difficulty of recovering the biometric given only the stolen syndromes.

6. Conclusions and Outlook

The broad objective of this paper is to propose and illustrate that human biometric templates can be transformed so
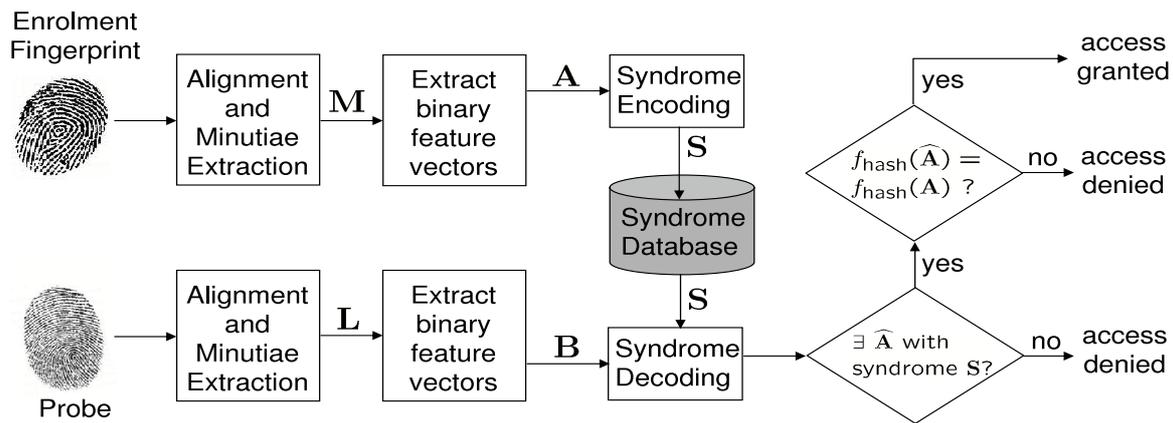
Figure 6. Robust feature extraction is combined with syndrome coding to build a secure fingerprint biometrics system.

as to possess a statistical profile that is compatible with the tools used to perform secure pattern recognition. We described an algorithm that transforms minutiae-based fingerprint templates into binary feature vectors whose properties are matched with error correcting codes designed for standard channel models. These feature vectors account for the location and orientation of the minutiae points and are robust to the variation in minutiae maps derived from repeated noisy measurements from the same finger. We showed a practical implementation of a secure biometric storage system in which syndromes obtained via LDPC coding of these feature vectors serve as secure biometrics. In this way, fingerprint-based access control is implemented without the need to store the original fingerprint template at the device.

The focus of our ongoing work is on improving the security-robustness tradeoff in fingerprint biometrics by (a) developing efficient methods to eliminate correlated question pairs for the user-specific feature transformation and (b) incorporating other modalities such as fingerprint ridge maps into the feature transformation.

References

[1] N. Ratha, J. Connell, R.M. Bolle, and S. Chikkerur. Cancelable biometrics: A case study in fingerprints. In *Intl. Conf. on Pattern Recognition*, pages 370–373, 2006.

[2] Nalini K. Ratha, Sharat Chikkerur, Jonathan H. Connell, and Ruud M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):561–572, 2007.

[3] Koji Sakata, Takuji Maeda, Masahito Matsushita, Koichi Sasakawa, and Hisashi Tamaki. Fingerprint authentication based on matching scores with other data. In *Lecture Notes in Computer Science*, volume 3832 of *LNCS*, pages 280–286, 2005.

[4] A.B.J. Teoh, A. Gho, and D.C.L. Ngo. Random multi-space quantization as an analytic mechanism for biohashing of biometric and random identity inputs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1892–1901, 2006.

[5] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Eurocrypt*, volume 3027 of *LNCS*, pages 523–540. Springer-Verlag, 2004.

[6] G. Davida, Y. Frankel, and B. Matt. On enabling secure applications through off-line biometric identification. In *IEEE Symp. on Security and Privacy*, pages 148–157, 1998.

[7] Ari Juels and Madhu Sudan. A fuzzy vault scheme. In *IEEE Intl. Symp. on Information Theory*, 2002.

[8] T. Charles Clancy, Negar Kiyavash, and Dennis J. Lin. Secure smartcardbased fingerprint authentication. In *ACM SIGMM workshop on biometrics methods and applications*, 2003.

[9] Shenglin Yang and Ingrid Verbauwhede. Automatic secure fingerprint verification system based on fuzzy vault scheme. In *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, pages 609–612, 2005.

[10] U. Uludag and A.K. Jain. Fuzzy fingerprint vault. In *Workshop on Biometrics: Challenges Arising from Theory to Practice*, pages 13–16, August 2004.

[11] E. C. Chang and S. Roy. Robust extraction of secret bits from minutiae. In *Proceedings of the IAENG International Conference on Biometrics (ICB 2007)*, pages 750–759, Hong Kong, March 2007.

[12] T.A.M. Kevenaar, G.J. Schrijen, M. Van der Veen, A.H.M. Akkermans, and F. Zuo. Face recognition with renewable and privacy preserving binary templates. *Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pages 21–26, 2005.