Securing Fingerprint Template: Fuzzy Vault with Minutiae Descriptors*

Abhishek Nagar
Michigan State Univ.
East Lansing, MI, USA
nagarabh@cse.msu.edu

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

Anil K. Jain
Michigan State Univ.
East Lansing, MI, USA
jain@cse.msu.edu

Abstract

Fuzzy vault has been shown to be an effective technique for securing fingerprint minutiae templates. Its security depends on the difficulty in identifying the set of genuine minutiae points among a mixture of genuine and chaff points and reconstructing the secure polynomial using the evaluations (ordinate values) available for each point in the vault. We show that the security of fuzzy vault can be improved by "encrypting" these polynomial evaluations using a fuzzy commitment scheme. This encryption makes it difficult for an adversary to decode the vault even if the correct set of minutiae is selected. We use minutiae descriptors, which capture orientation and ridge frequency information in a minutia's neighborhood, for securing the polynomial evaluations. This modification leads to a significant increase in both the security (number of tries an adversary has to make in order to guess the secure key) and matching accuracy of the vault. We validate our results on FVC2002 DB2 and show that false accept rate (FAR) is reduced from 0.7% *to* 0.01% *at a genuine accept rate (GAR) of 95%. At the same time, vault security as measured in terms of min-entropy, is increased from 31 bits to 47 bits in case a perfect code is used.*

1. Introduction

The issue of biometric template security is gaining importance due to concerns about the potential misuse of stolen templates. There are two major concerns regarding a stolen biometric template: (i) spoofing and (ii) privacy intrusion. If an adversary is able to access the stored templates, he can create a spoof biometric (e.g. *gummy finger*) from the template [2] and present it to the system. Due to limited liveness detection capability of current biometric systems, spoofing is a major security vulnerability. Further, an adversary can cross-link the stolen templates with other biometric databases, allowing him to track the activities of a person covertly.

Template protection approaches can be categorized into two classes: (i) Transformation based approaches - the template is stored in a transformed form so that the original biometric template cannot be easily recovered. But, finding a suitable non-invertible transformation that allows accurate matching in the transformed domain is a challenge. Though a number of transforms have been proposed [1, 9, 10, 12], there is a need for better transformation functions that provide high security without comprising on the matching accuracy. (ii) Biometric cryptosystems [3, 5, 8, 11] - *helper data* extracted from the biometric template, with or without the use of an external key, is stored in the database instead of template. The helper data should not reveal any information about the template or the key, but the key should be recoverable when another instance of the biometric trait that closely matches the template is presented.

We focus here on improving the security and performance of fuzzy vault [6], which is a popular biometric cryptosystem. Fuzzy vault can effectively utilize the natural representation of fingerprint minutiae i.e. an unordered set [8]. In addition to minutiae position and orientation, we utilize additional attributes extracted from a minutia's neighborhood to improve the vault. In particular, we show that minutiae descriptors [4], which contain local ridge orientation and ridge frequency information, have sufficient saliency to reduce the FAR of a fingerprint fuzzy vault. Moreover, we also show that "encrypting" the polynomial evaluations in the vault using the minutiae descriptors increases the vault security.

2. Helper Data Extraction (Enrollment)

In our fingerprint cryptosystem, helper data extraction consists of two main steps (see Figure 1). The fuzzy vault framework described in [8] is first used to secure

*This research was supported by Army Research Office grant W911NF-06-1-0418.

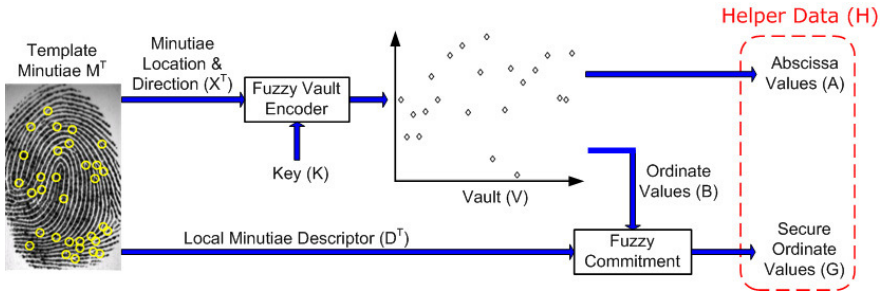


Figure 1. Helper data extraction in descriptor-based fingerprint cryptosystem.

the minutiae locations and directions. The ordinate values of the vault are further secured using the minutiae descriptors through the fuzzy commitment approach.

2.1. Fuzzy Vault Encoder

During vault encoding a $(16 \times n)$ bit key (K) is appended with a 16-bit Cyclic Redundancy Check (CRC) code 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 only well-separated minutiae [8] are selected for constructing the vault. If the desired number of 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 $x_i, i = 1, \cdots, r$ along with their corresponding polynomial evaluations $f(x_i), = 1, \cdots, r$ are stored in the vault V. A set of s chaff points $\{(y_j, z_j), j = 1, \cdots, s\}$ is generated randomly such that $y_j \neq x_i, \forall i = 1, \cdots, r; j = 1, \cdots s$ and $z_j \neq f(y_j), \forall j = 1, \cdots, s$. 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. However, 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 in order to encrypt the ordinate values.

A minutiae descriptor [4] consists of ridge orientation and frequency at 76 equidistant points, uniformly spaced on 4 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 Figure 2). The radius and the number of points on each circle are selected in such a way that the descriptor values capture the maximum information contained in the neighborhood of a minutia. For minutiae on the fingerprint boundary, missing values in the descriptors are estimated using extrapolation. Descriptors for the chaff points can be randomly sampled from a pool of all the descriptors available in the database.

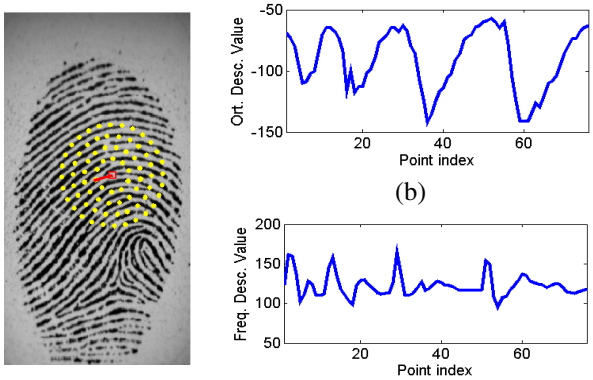


Figure 2. Minutiae descriptor: (a) positions of 76 points around a minutiae, (b) orientation and (c) frequency descriptors.

Minutiae descriptors are used in a fuzzy commitment scheme to secure the ordinate values in the vault as follows. First, the descriptor is binarized by quantizing each orientation and frequency value into 2^{α} and 2^{β} val-

ues, respectively ($\alpha = 5$ and $\beta = 4$). Gray codes are used to convert the quantization indices into bits having the property that adjacent values differ only in single bit. Then, we select 511 bits among the 684 (76×9) bits based on their variation. Let D_i be the descriptor in binary format and C_i be a codeword generated by adding error correction bits to the 16-bit ordinate value B_i, $i = 1, \cdots, (r+s)$. Now, instead of the ordinate value B_i, only the secure ordinate value $G_i = (D_i \oplus C_i)$ is stored. Here, $\oplus$ denotes the XOR operation. 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 Figure 3), first the template and query fingerprints are aligned using the high curvature points as described in [8]. 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. An XOR operation is applied between the descriptor (say $D^{'}$) associated with each selected query minutia and the corresponding secure ordinate value to obtain a word C'. This word is then decoded to obtain the message, which represents the ordinate value corresponding to that minutia. If the ordinate value is correctly decoded for some minimum number ($n+1$) of genuine points in the vault, the polynomial f can be correctly reconstructed indicating a successful match.

4. Experimental Results

We used FVC2002 DB2 to compare the fuzzy vault performance with and without minutiae descriptors. As in [8], only the first two impressions of the 100 different fingers were used in the experiments, one as template and the other as query. BCH(511,19) error correcting scheme is used in our implementation for encoding the ordinate values which can correct upto 119 errors. Figure 4 shows the GAR and FAR corresponding to the original fuzzy vault implementation in [8] (without descriptors) and the proposed cryptosystem. FTCR 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 reduced from 0.7% to 0.01% when the proposed cryptosystem is used. It was also experimentally observed that only 0.9% of the descriptors mismatched given the minutiae correspondences were correct.

5 Security Analysis

Nandakumar [7] showed that the min-entropy [3] of the minutiae template M^T given the vault V can be computed as

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

if both the minutiae location and minutiae orientation are uniformly distributed. Here, r, n and s have the same meaning as defined earlier. The fuzzy vault implementation in [8] 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 [8] is approximately 31 bits.

In the proposed fingerprint fuzzy vault the true ordinate values can be obtained in two ways. (i) Directly guessing the 16-bit ordinate values. Since the ordinate values of the genuine points are obtained through 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). Also since the adversary has to simultaneously guess $(n+1)$ ordinate values correctly, this corresponds to approximately $16(n+1)$ bits of security. (ii) Guess the descriptors associated with each minutia. Although the length of descriptor is N bits ($N = 511$ here), there is a strong correlation between the descriptor bits. Suppose that the entropy of a minutia descriptor D is I_D bits and $\rho = (119 \times I_D)/511 \approx I_D/4$ bits should be corrected. As shown by Hao et al. [5], the difficulty in guessing a minutiae descriptor is approximately $R = \log\left(2^{I_D}/\binom{I_D}{\lceil\rho\rceil}\right)$ bits. Since the adversary has to simultaneously guess $(n+1)$ minutiae descriptors correctly, using minutiae descriptors provides approximately $(n+1)R$ bits of security. For instance, 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 16 bits and the overall security is 47 (31 + 16) bits.

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 codeword C such that the Hamming distance between C and $C^{'}$ is at most w bits). 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. Note that even if all the incorrect descriptors lead to decoding failure, which is very unlikely, the security is at least as good as the security of

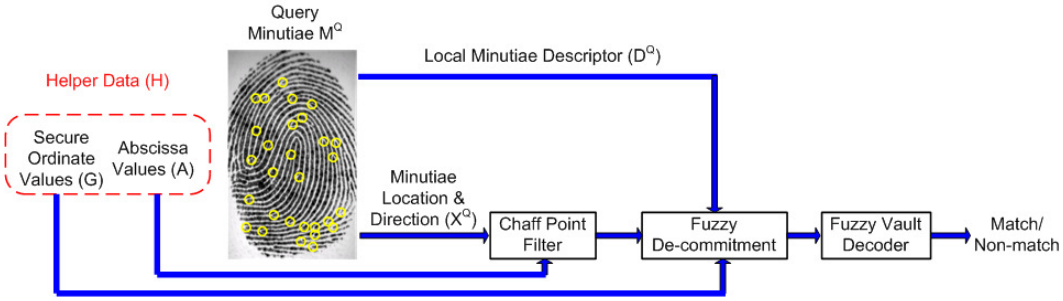
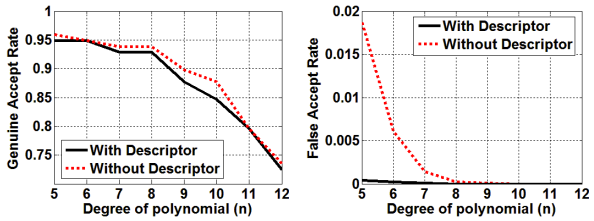


Figure 3. Authentication in descriptor-based fingerprint cryptosystem.



(a) (b)

Figure 4. (a) Genuine accept rate (GAR) and (b) False accept rate (FAR) of the fingerprint fuzzy vault with and without minutiae descriptors.

the original fuzzy vault. Non-perfect codes may reduce the effective value of R, thereby limiting the additional security that can be achieved using descriptors. While our current implementation uses non-perfect codes, we are exploring the possibility of using available perfect codes (the class of Hamming codes and Golay code) to achieve the required error correction capability.

6. Conclusions

We have shown that both the performance and security of a fingerprint fuzzy vault can be improved by incorporating minutiae descriptors. The descriptor, consisting of ridge frequency and ridge orientation information in a minutia's neighborhood, helps us encrypt the ordinate values of the polynomial during vault construction. Experimental results on a public domain database, FVC2002 DB2, demonstrate that the use of minutiae descriptors leads to an order of magnitude reduction in the false accept rate without affecting the genuine accept rate. Also, the security in terms of number of tries an adversary has to make in order to guess the secure key is significantly increased.

References

[1] T. E. Boult, W. J. Scheirer, and R. Woodworth. Fingerprint Revocable Biotokens: Accuracy and Security Analysis. In *Proc. CVPR*, pages 1–8, Minneapolis, 2007.

[2] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni. Fingerprint Image Reconstruction From Standard Templates. *IEEE Trans. PAMI*, 29(9):1489–1503, 2007.

[3] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. Technical Report 235, Cryptology ePrint Archive, February 2006.

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

[5] F. Hao, R. Anderson, and J. Daugman. Combining Crypto with Biometrics Effectively. *IEEE Trans. on Computers*, 55(9):1081–1088, September 2006.

[6] A. Juels and M. Sudan. A Fuzzy Vault Scheme. In *Proceedings of IEEE International Symposium on Information Theory*, page 408, Lausanne, Switzerland, 2002.

[7] K. Nandakumar. *Multibiometric Systems: Fusion Strategies and Template Security*. PhD thesis, Department of Computer Science and Engineering, Michigan State University, January 2008.

[8] K. Nandakumar, A. K. Jain, and S. Pankanti. Fingerprint-based Fuzzy Vault: Implementation and Performance. *IEEE Trans. Information Forensics and Security*, 2(4):744–757, 2007.

[9] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle. Generating Cancelable Fingerprint Templates. *IEEE Trans. PAMI*, 29(4):561–572, April 2007.

[10] M. Savvides and B. V. K. Vijaya Kumar. Cancellable Biometric Filters for Face Recognition. In *Proc. ICPR*, volume 3, pages 922–925, Cambridge, August 2004.

[11] Y. Sutcu, Q. Li, and N. Memon. Protecting Biometric Templates with Sketch: Theory and Practice. *IEEE Trans. Information Forensics and Security*, 2(3):503–512, September 2007.

[12] A. B. J. Teoh, K.-A. Toh, and W. K. Yip. 2^N Discretisation of BioPhasor in Cancellable Biometrics. In *Proc. ICB*, pages 435–444, Seoul, 2007.