Appeared in Proceedings of ICB 2007

Hardening Fingerprint Fuzzy Vault Using Password*

Karthik Nandakumar, Abhishek Nagar and Anil K. Jain

Department of Computer Science & Engineering, Michigan State University,
East Lansing, MI – 48824, USA
{nandakum, nagarabh, jain} @cse.msu.edu

Abstract. Security of stored templates is a critical issue in biometric systems because biometric templates are non-revocable. Fuzzy vault is a cryptographic framework that enables secure template storage by binding the template with a uniformly random key. Though the fuzzy vault framework has proven security properties, it does not provide privacy-enhancing features such as revocability and protection against cross-matching across different biometric systems. Furthermore, non-uniform nature of biometric data can decrease the vault security. To overcome these limitations, we propose a scheme for hardening a fingerprint minutiae-based fuzzy vault using password. Benefits of the proposed password-based hardening technique include template revocability, prevention of cross-matching, enhanced vault security and a reduction in the False Accept Rate of the system without significantly affecting the False Reject Rate. Since the hardening scheme utilizes password only as an additional authentication factor (independent of the key used in the vault), the security provided by the fuzzy vault framework is not affected even when the password is compromised.

Keywords: Biometric template security, fuzzy vault, hardening, password, fingerprint, minutiae, helper data.

1 Introduction

Biometric systems have attained popularity because they provide a convenient and reliable way to authenticate a user as opposed to traditional token-based (e.g., smart cards) and knowledge-based (e.g., passwords) authentication. However, it is now well-known that biometric systems are vulnerable to attacks. One of the most serious attacks is against the stored templates. A stolen biometric template cannot be easily revoked and it may be used in other applications that employ the same biometric trait. Table 1 presents a summary of the approaches that have been proposed for biometric template protection. We propose a hybrid approach where the biometric features are hardened using password before a secure sketch (fuzzy vault) is constructed.

1.1 Fuzzy Vault Framework: Fuzzy vault [1] is a cryptographic framework that binds the biometric template with a uniformly random key to build a secure sketch of the template. Only the secure sketch (vault) is stored and if the original template is "uniformly random", it is infeasible (or computationally hard) to retrieve either the template or the key without any knowledge of the user's biometric data.

* Research supported by ARO grant no. W911NF-06-1-0418

Table 1. Summary of biometric template protection approaches.

Template Protection Approaches	Methodology	Advantages	Limitations
Encryption	Template is encrypted using well-known cryptographic techniques	Matching algorithm and accuracy are unaffected	Template is exposed during every authentication attempt
Non-invertible transform (e.g., [2, 3])	*One-way function* is applied to the biometric features	Since transformation occurs in the same feature space, matcher need not be redesigned	Usually leads to increase in the FRR
Hardening / Salting (e.g., [4])	*User-specific external randomness* is added to the biometric features	Increases the entropy of biometric features resulting in low FAR	If the user-specific random information is compromised, there is no gain in entropy
Key generation (e.g., [5])	A key is derived directly from biometric features	Most efficient and scalable approach	Tolerance to intra-user variations is limited, resulting in high FRR
Secure sketch (e.g. [1, 6-9])	A sketch is derived from the template; sketch is secure because template can be reconstructed only if a *matching* biometric query is presented	More tolerant to intra-user variations in biometric data; can be used for securing external data such as cryptographic keys	Template is exposed during successful authentication. Non-uniform nature of biometric data reduces security
Proposed hardened fuzzy vault	A *hybrid* approach where the biometric features are hardened (using password) before a secure sketch (vault) is constructed	Hardening increases the entropy thereby improving the vault security; also enhances user privacy	Not user-friendly; user needs to provide both the password and the biometric during authentication

The fuzzy vault scheme can secure biometric features that are represented as an unordered set. Let $\mathbf{M^T} = \{x_1, x_2, ..., x_r\}$ denote a biometric template with r elements. The user selects a key K, encodes it in the form of a polynomial P of degree n and evaluates the polynomial P on all the elements in $\mathbf{M^T}$. The points lying on P $\left(\{(x_i, P(x_i))\}_{i=1}^{r}\right)$ are hidden among a large number (s) of random chaff points that do not lie on P $\left(\{(x_j, y_j) \mid x_j \neq x_i, \forall i = 1,...,r, y_j \neq P(x_j)\}_{j=1}^{s}\right)$. The union of genuine and chaff point sets constitutes the vault $\mathbf{V}$. In the absence of user's biometric data, it is computationally hard to identify the genuine points in $\mathbf{V}$, and hence the template is secure. During authentication, the user provides a biometric query denoted by $\mathbf{M^Q} = \{x'_1, x'_2, ..., x'_r\}$. If $\mathbf{M^Q}$ overlaps substantially with $\mathbf{M^T}$, the user can identify many points in $\mathbf{V}$ that lie on the polynomial. If the number of discrepancies between $\mathbf{M^T}$ and $\mathbf{M^Q}$ is less than (r-n)/2, Reed-Solomon decoding can be applied to reconstruct P and the authentication is successful. On the other hand, if $\mathbf{M^T}$ and $\mathbf{M^Q}$ do not have sufficient overlap, it is infeasible to reconstruct P and the authentication is unsuccessful. The vault is called *fuzzy* because it can be decoded even when $\mathbf{M^T}$ and $\mathbf{M^Q}$ are not exactly the same; this fuzziness property compensates for intra-user

variations observed in biometric data. Security of the fuzzy vault framework has been studied in [1, 6] and bounds on the entropy loss of the vault have been established.

1.2 Limitations of Fuzzy Vault Framework: Though the fuzzy vault scheme has proven security properties [1, 6], it has the following limitations.

(i) The security of the vault can be compromised if the same biometric data is re-used for constructing different vaults (with different polynomials and random chaff points) [10, 11]. If a person has access to two vaults obtained from the same biometric data, he can easily identify the genuine points in the two vaults by correlating the abscissa (x) values in the two vaults. Due to this reason, the *vault is not revocable*, i.e., if a vault is compromised, a new vault cannot be created from the same biometric data by merely binding it with a different key. Further, this vulnerability *allows cross-matching of templates across different systems*. Thus, the fuzzy vault framework does not have privacy-enhancing properties.

(ii) It is possible for an attacker to exploit the *non-uniform* nature of biometric features and develop attacks based on statistical analysis of points in the vault.

(iii) Since the number of chaff points in the vault is much larger than the number of genuine points, it is possible for an adversary to substitute a few points in the vault using his own biometric features [10, 11]. This allows both the original user and the adversary to be successfully authenticated using the same identity. Thus, an *adversary can deliberately increase the false accept rate* of the system.

(iv) As a genuine user is being authenticated, his *original template is exposed temporarily*, which may be gleaned by an attacker.

While better fuzzy vault constructions that do not involve chaff points [6] can prevent vulnerabilities (ii) and (iii), they do not address limitations (i) and (iv). By using a password as an additional factor for authentication, the above limitations of a fuzzy vault system can be easily alleviated. In this paper, we propose a scheme for hardening a fingerprint-based fuzzy vault using password. One of the main advantages of password-based hardening is enhanced user privacy. Further, the proposed scheme has been designed such that *password is only an additional layer of authentication and the security provided by the basic fuzzy vault framework is not affected even if the password is compromised*. Hence, the proposed approach provides higher level of security as long as the password is secure. When the password is compromised, template security falls to the same level as in the fuzzy vault.

2 Hardening Fuzzy Vault Using Password

Our vault hardening scheme consists of three main steps (see Fig. 1). Firstly, a random transformation function derived from the user password is applied to the biometric template. The transformed template is then secured using the fuzzy vault framework. Finally, the vault is encrypted using a key derived from the password.

Random transformation of the template using password enhances user privacy because it enables the creation of revocable templates and prevents cross-matching of templates across different applications. The distribution of transformed template is statistically more similar to uniform distribution than the distribution of original template. This provides better resistance against attacks on the vault. Furthermore, the

additional variability introduced by password-based transformation decreases the similarity between transformed templates of different users. This reduces the False Accept Rate of the system substantially. If we assume client-server architecture for the biometric system (as shown in Fig. 1) where feature extraction and transformation are applied at the client side and matching is performed at the server, the server never sees the original template. Only the transformed template would be revealed during successful vault decoding and the original template is never exposed at the server.

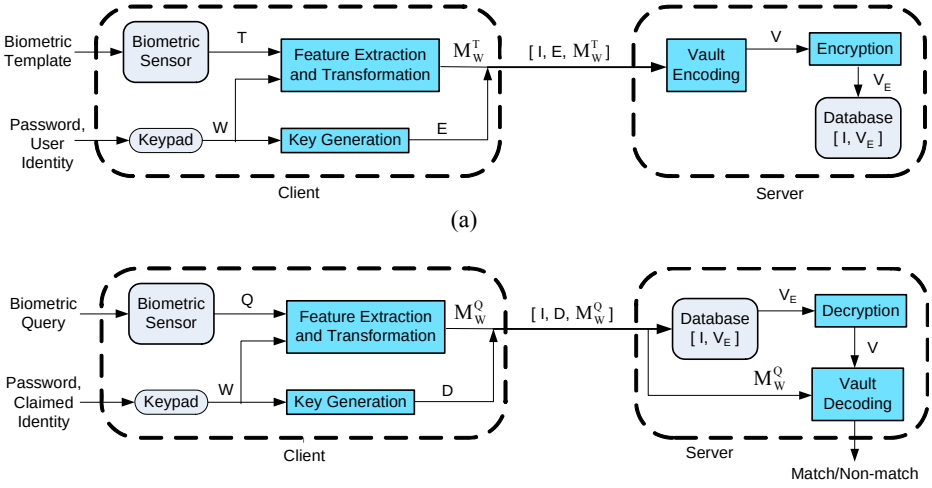


Fig. 1. Operation of the hardened fuzzy vault. (a) Enrollment and (b) authentication stages. In this figure, I represents the identity of the user, W is the user password, M^T (M^Q) represents the biometric template (query), M_W^T (M_W^Q) represents the template (query) after transformation using the password, E and D represent the encryption and decryption keys generated from the password and V and V_E represent the plaintext and encrypted vaults.

Two common methods for cracking a user password are dictionary attacks and social engineering techniques. In the proposed system, password is implicitly verified during authentication by matching the transformed biometric features. Even if an adversary attempts to guess the password, it is not possible to verify the guess without knowledge of the user's biometric data. This provides resistance against dictionary attacks to learn the password. However, it is still possible to glean the user password through social engineering techniques. Therefore, password based transformation alone is not sufficient to ensure the security of the biometric template. Due to this reason, we use the fuzzy vault framework to secure the transformed biometric template. Note that the *key used in constructing the fuzzy vault that secures the transformed template is still uniformly random and independent of the password*. Therefore, even if the password is compromised, the security of the vault is not affected and it is computationally hard for an attacker to obtain the original biometric template. Finally, the vault is encrypted using a key derived from the password. This prevents substitution attacks against the vault because an adversary cannot modify the vault without knowing the password or the key derived from it.

3 Fingerprint-based Fuzzy Vault Implementation

A number of techniques have been proposed for constructing a fuzzy vault using fingerprint minutiae (e.g., [13, 14]). The proposed hardening scheme is based on the fingerprint-based fuzzy vault implementation described in [12] which has the highest genuine accept rate and a very low false accept rate among the known implementations of fingerprint-based fuzzy vault. In this implementation, the Reed-Solomon polynomial reconstruction step is replaced by a combination of Lagrange interpolation and Cyclic Redundancy Check (CRC) based error detection. Each minutia point is represented as an element in the Galois field GF(2^{16}) by applying the following procedure. Let (u, v, θ) be the attributes of a minutia point, where u and v indicate the row and column indices in the image, and θ represents the orientation of the minutia with respect to the horizontal axis. The minutia attributes are uniformly quantized and expressed as binary strings Q_u, Q_v and Q_θ of lengths B_u, B_v and B_θ bits, respectively. The values of B_u, B_v and B_θ are chosen to be 6, 5 and 5, respectively, so that a 16-bit number can be obtained by concatenating the bit strings Q_u, Q_v and Q_θ.

A fixed number (denoted by r) of minutiae are selected based on their quality. A randomly generated key K of size $16n$ bits is represented as a polynomial P of degree n. The polynomial P is evaluated at the selected minutiae and these points constitute the locking set. A large number (denoted by s, s >> r) of chaff points are randomly generated and the combined set of minutiae and chaff is randomly reordered to obtain the vault **V**. To facilitate the alignment of query minutiae to the template, we extract and store a set of high curvature points (known as *helper data*) from the template image. The helper data itself does not leak any information about the minutiae, yet contains sufficient information to align the template and query fingerprints [12].

During authentication, the helper data extracted from the query image is aligned to the template helper data using trimmed Iterative Closest Point (ICP) algorithm [15]. Aligned query minutiae are used to coarsely filter out the chaff points in the vault. A minutiae matcher [16] is then applied to find correspondences between the query minutiae and the remaining points in the vault. Vault points having a matching minutia in the query constitute the unlocking set. For interpolation of a polynomial of degree n, at least (n+1) projections are needed. Therefore, if the size of the unlocking set is less than (n+1), it leads to authentication failure. If the unlocking set has (n+1) or more elements, all possible subsets of size (n+1) are considered. Each of these subsets gives rise to a candidate polynomial and CRC-based error detection identifies the valid polynomial. If a valid polynomial is found, the authentication is successful.

4 Hardened Fuzzy Vault Implementation

The key component of a hardened fingerprint-based fuzzy vault scheme is the feature transformation module which transforms the minutiae features using a password. We employ simple operations of translation and permutation as the transformation functions because they do not affect the intra-user variability of the minutiae features thereby maintaining the false reject rate to a great extent.

4.1 Minutiae Transformation: We assume that the password is of length 64 bits (8 characters) which is divided into 4 units of 16 bits each. We classify the minutiae into

4 classes by grouping minutiae lying in each quadrant of the image into a different class and assign one password unit to each class. We generate a permutation sequence of 4 numbers by applying a one way function on the password. Using this sequence, we permute the 4 quadrants of the image such that the relative positions of minutiae within each quadrant are not changed. Each 16-bit password unit is assumed to be in the same format as a 16-bit minutia representation described in section 3. Hence, the password unit can be divided into three components T_u, T_v and T_θ of lengths B_u, B_v and B_θ bits, respectively. The values of T_u and T_v are considered as the amount of translation along the vertical and horizontal directions, respectively, and T_θ is treated as the change in minutia orientation. The new minutiae attributes are obtained by adding the translation values to the original values modulo the appropriate range, i.e., $Q'_u = (Q_u + T_u) \bmod (2\hat{} B_u)$, $Q'_v = (Q_v + T_v) \bmod (2\hat{} B_v)$ and $Q'_\theta = (Q_\theta + T_\theta) \bmod (2\hat{} B_\theta)$. To prevent overlapping of minutiae from different quadrants, the minutia location is wrapped around in the respective quadrant if it has been translated beyond the boundary. The effect of minutiae transformation using password is depicted in Fig. 2.

4.2 Encoding Hardened Vault: The transformed minutiae are encoded in a vault using the procedure described in section 3. The vault and helper data are further encrypted using a key generated from the password. This layer of encryption prevents an impostor without knowledge of the password from modifying the vault.

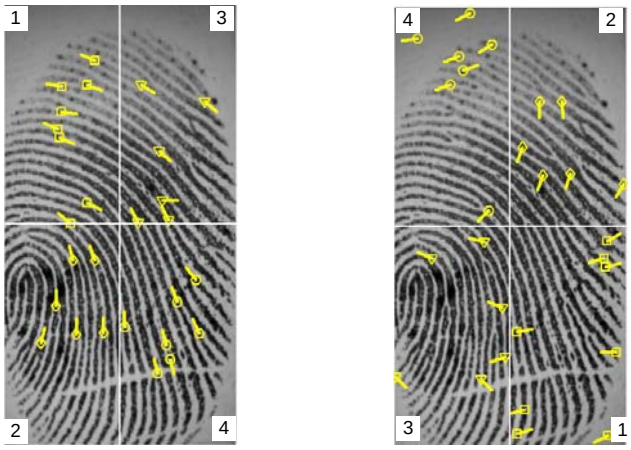
4.3 Decoding Hardened Vault: During authentication, the encrypted vault and helper data are first decrypted using the password provided by the user. The template and query helper data sets are aligned and the password-based transformation scheme described in section 4.1 is applied to the aligned query minutiae. Good quality minutiae are then selected for decoding the vault.

Apart from the well-known factors like partial overlap, non-linear distortion and noise that lead to differences in the template and query minutiae sets of the same user, the password-based transformation scheme introduces additional discrepancies. If a minutia lies close to the quadrant boundary, the same minutiae may fall in different quadrants in the template and the query due to imperfect alignment. This reduces the number of minutiae correspondences and leads to a small decrease in the genuine accept rate. Another problem arising due to imperfect alignment is that the same minutia point may appear at opposite ends of the quadrants in the template and the query after the transformation. This is because the minutiae are translated within their respective quadrants modulo the quadrant size. To address this problem, we add a border of width 15 pixels around each quadrant and minutiae within 15 pixels of the quadrant boundary are duplicated on the border at the opposite end of the quadrant.

5 Experimental Results

The proposed password-based fuzzy vault hardening scheme has been tested on the FVC2002-DB2 and MSU-DBI fingerprint databases. FVC2002-DB2 [17] is a public domain database with 800 images (100 fingers × 8 impressions/finger) of size 560×296. Only the first two impressions of each finger were used in our experiments; the first impression was used as the template to encode the vault and the second impression was used as the query in vault decoding. The MSU-DBI database [18]

consists of 640 images (160 fingers × 4 impressions/finger) of size 640×480. Two impressions of each finger collected six weeks apart were used in our experiments.



(a) (b)

Fig. 2. Minutiae transformation using password. (a) and (b) show the original and transformed minutiae, respectively. The number at each corner indicates the permutation of quadrants.

The criteria used for evaluating the performance are failure to capture rate (FTCR), genuine accept rate (GAR) and false accept rate (FAR). When the number of minutiae in the template and/or query fingerprint is less than the required number of genuine points, we call it as failure to capture. The parameters used in vault implementation were chosen as follows. Since the number of minutiae varies for different users, using a fixed value of r (the number of genuine minutiae used to construct the vault) across all users leads to large FTCR. To overcome this problem, we fix the range of r (set to 18-24 and 24-30 for the FVC and MSU databases, respectively) and determine its value individually for each user. The number of chaff points (s) is chosen to be 10 times the number of genuine points in the vault. The choice of n requires a compromise between the vault security and the acceptable values of GAR and FAR.

Table 2 shows that the proposed system leads to a small decrease in the GAR for all values of n. This is due to misclassification of a few minutiae at the quadrant boundaries and the inability of the minutiae matcher to effectively account for non-linear deformation in the transformed minutiae space. Although the minutiae matcher [16] used here can tolerate deformation to some extent by employing an adaptive bounding box, it is designed to work in the original minutiae space where the deformation is consistent in a local region. Since minutiae transformation makes the deformation inconsistent in all the regions, the number of correspondences found by the matcher decreases. Fig. 3 shows a pair of images for which the vault without hardening could be decoded, but the hardened vault could not be decoded.

From Table 2, we also observe that the FAR of the system is zero for all values of n. This is due to the transformation of minutiae using password which makes the distribution of minutiae more random and reduces the similarity between minutiae sets of different users. This enables the system designer to select a wider range of

values for n without compromising the FAR. For example, for the FVC database, the original fuzzy vault required n=10 to achieve 0% FAR (corresponding GAR is 86%). For the hardened fuzzy vault, even n=7 gives 0% FAR (corresponding GAR is 90%).

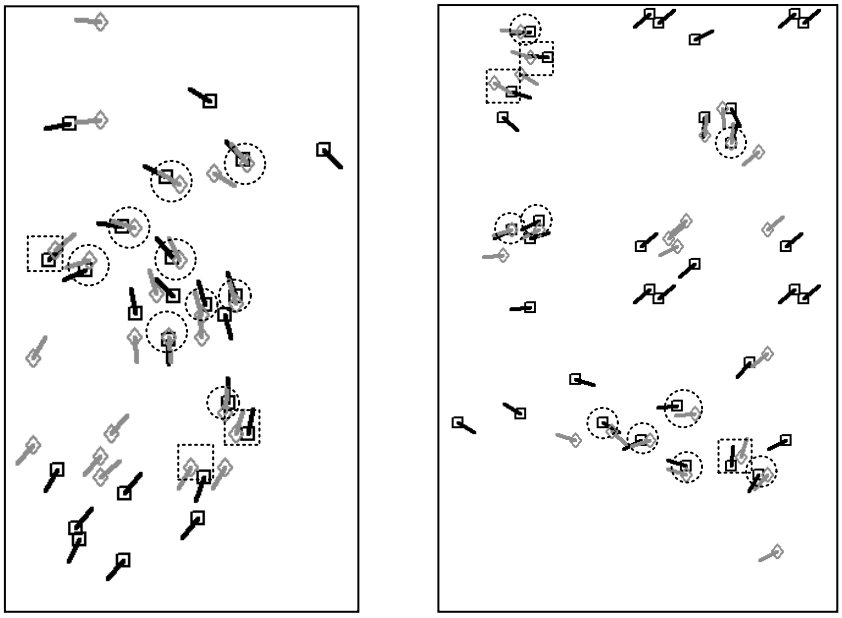
Table 2. Genuine Accept Rates (GAR), False Accept Rates (FAR) and Failure to Capture Rates (FTCR) of the hardened fuzzy vault for FVC2002-DB2 and MSU-DBI databases. Here, n represents the degree of the polynomial used in vault encoding.

		FTCR (%)	**n = 7**		**n = 8**		**n = 10**	
			GAR(%)	**FAR(%)**	**GAR(%)**	**FAR(%)**	**GAR(%)**	**FAR(%)**
FVC2002 – DB2	Vault without hardening	2	91	0.13	91	0.01	86	0
	Hardened vault	2	90	0	88	0	81	0
		FTCR (%)	**n = 10**		**n = 11**		**n = 12**	
			GAR(%)	**FAR(%)**	**GAR(%)**	**FAR(%)**	**GAR(%)**	**FAR(%)**
MSU-DBI	Vault without hardening	5.6	85	0.08	82.5	0.02	78.8	0
	Hardened vault	5	80.6	0	75.6	0	73.8	0

6 Security Analysis

The hardened fuzzy vault system has two independent layers of security, namely, password and biometric. An impostor can gain access to the system only if both these layers of security are compromised simultaneously. Now, we shall analyze the security of the system if one of the layers is compromised.

Compromised Password: Suppose an impostor gains access to the password of a genuine user. The impostor can at most generate the decryption key that allows him to decrypt the vault. However, to be successfully authenticated, he still would have to decode the vault by identifying the genuine minutia points from the vault, which is computationally hard. Suppose an attacker attempts a brute-force attack on the proposed system by trying to decode the vault using all combinations of (n+1) points in the vault. If n = 10, r = 30 and s = 300, the total number of possible combinations is C(330,11); among these combinations, C(30,11) combinations will successfully decode the vault. The expected number of combinations that need to be evaluated is 2×10^{12} which corresponds to ~ 40 bits of security. Security can be improved by adding a larger number of chaff points (e.g., when s = 600 in the above system, we can achieve ~ 50 bits of security) at the expense of increased storage requirements. Further improvement in the template security can be achieved by replacing the random permutation and translation functions (that are invertible) by a non-invertible transform [2]. Though non-invertible transforms usually result in a decrease in GAR, they can provide an additional 50-70 bits of security [2] if the goal is to prevent an attacker from learning the original biometric template of the genuine user.



(a) (b)

Fig. 3. An example of false reject in password-based vault hardening. (a) Template and aligned query minutiae prior to hardening and the corresponding minutiae matches found by the matcher, (b) Template and query minutiae after vault hardening and the corresponding minutiae matches found by the matcher. While the minutiae matches marked with circles were found in both (a) and (b), the matches marked with squares were detected *only* in (a) and *not* in (b). Since the number of minutia correspondences prior to hardening is 12, the vault can be successfully decoded because n was set to 10. After hardening, the number of minutia matches is only 9; hence, the vault cannot be decoded for $n = 10$.

Compromised Biometric: Suppose an impostor gains access to the biometric template of a genuine user through covert means (e.g., lifting a fingerprint impression of the genuine user without his knowledge), he will still have to guess the password to be authenticated. The guessing entropy of an 8-character password is between 18-30 bits [19]. Although this level of security may be insufficient in practical applications, it is still better than the fuzzy vault framework and most of the other approaches presented in Table 1 which offer no security when the biometric is compromised.

When an adversary does not have any knowledge of the user password and biometric data, then the security of the hardened fuzzy vault is the combination of the security provided by the password and biometric layers. If n = 10, r = 30, s = 300 and password is 8-character long, the security of the hardened vault is between 58-70 bits.

7 Summary

We have proposed an algorithm to harden a fingerprint-based fuzzy vault based on user password. Based on permutations and translations generated from the user password, we modify the minutiae in a fingerprint before encoding the fuzzy vault.

Our hardening technique addresses some of the major limitations of a fingerprint-based fuzzy vault framework and provides enhanced security and privacy. Experiments on two fingerprint databases show that proposed algorithm reduces the False Accept Rate of the system with some loss in the Genuine Accept Rate. An impostor cannot circumvent the hardened fuzzy vault system as long as both the password and the biometric features are not compromised simultaneously.

References

1. A. Juels and M. Sudan, "A Fuzzy Vault Scheme," in *Proceedings of IEEE International Symposium on Information Theory*, Lausanne, Switzerland, 2002, p. 408.
2. N. Ratha, S. Chikkerur, J. H. Connell and R.M. Bolle, , "Generating Cancelable Fingerprint Templates", *IEEE Trans. on PAMI*, Vol. 29, No. 4, pp. 561–572, April 2007.
3. M. Savvides, B.V.K.V. Kumar and P.K. Khosla, "Cancelable biometric filters for face recognition", in *Proceedings of ICPR*, Vol. 3, Cambridge, UK, August 2004, pp. 922–925.
4. A.B.J. Teoh, A. Goh and D.C.L. Ngo, "Random Multispace Quantization as an Analytic Mechanism for BioHashing of Biometric and Random Identity Inputs", *IEEE Trans. on PAMI*, Vol. 28, No. 12, pp. 1892-1901, December 2006.
5. F. Monrose, M. K. Reiter, Q. Li and S. Wetzel, "Cryptographic Key Generation from Voice", in *Proc. IEEE Symp. Security and Privacy*, Oakland, May 2001, pp. 202-213.
6. Y. Dodis, L. Reyzin and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," in *Proceedings of International Conference on Theory and Applications of Cryptographic Techniques*, pp. 523-540, May 2004.
7. F. Hao, R. Anderson and J. Daugman, "Combining Crypto with Biometrics Effectively," *IEEE Trans. on Computers*, Vol. 55, No. 9, pp. 1081–1088, September 2006.
8. Y. Sutcu, Q. Li and N. Memon, "Protecting Biometric Templates with Sketch: Theory and Practice", to appear in IEEE Trans. on Information Forensics and Security, 2007.
9. S. C. Draper, A. Khisti, E. Martinian, A. Vetro and J. S. Yedidia, "Using Distributed Source Coding to Secure Fingerprint Biometrics", to appear in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2007.
10. T. E. Boult, W. J. Scheirer and R. Woodworth, "Fingerprint Revocable Biotokens: Accuracy and Security Analysis", to appear in Proc. of CVPR, Minneapolis, June 2007.
11. W. J. Scheirer and T. E. Boult, "Cracking Fuzzy Vaults and Biometric Encryption", Univ. of Colorado at Colorado Springs, Tech. Rep., February 2007.
12. K. Nandakumar, A.K. Jain and S. Pankanti, "Fingerprint-based Fuzzy Vault: Implementation and Performance", Michigan State Univ., Tech. Rep., TR-06-31, 2006.
13. S. Yang and I. Verbauwhede, "Automatic Secure Fingerprint Verification System Based on Fuzzy Vault Scheme", in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, Philadelphia, USA, March 2005, pp. 609–612.
14. U. Uludag, S. Pankanti and A. K. Jain, "Fuzzy Vault for Fingerprints," in *Proceedings of Fifth International Conference on AVBPA*, Rye Town, USA, July 2005, pp. 310–319.
15. D. Chetverikov, D. Svirko, D. Stepanov and P. Krsek, "The Trimmed Iterative Closest Point Algorithm," in *Proc. of ICPR*, Quebec City, Canada, August 2002, pp. 545–548.
16. A. K. Jain, L. Hong and R. Bolle, "On-line Fingerprint Verification", *IEEE Trans. on PAMI*, Vol. 19, No. 4, pp. 302–314, April 1997.
17. D. Maio, D. Maltoni, J.L. Wayman and A.K. Jain, "FVC2002: Second Fingerprint Verification Competition," in *Proc. of ICPR*, Quebec City, Aug. 2002, pp. 811–814.
18. A. K. Jain, S. Prabhakar and A. Ross, "Fingerprint Matching: Data Acquisition and Performance Evaluation," Michigan State Univ., Tech. Rep., TR99-14, 1999.
19. W. E. Burr, D. F. Dodson and W. T. Polk, "Information Security: Electronic Authentication Guideline", NIST Special Report 800-63, April 2006.