Secure Fingerprint Matching With Generic Local Structures

Matthew Morse, Jesse Hartloff, Thomas Effland, Jim Schuler,
Jennifer Cordaro, Sergey Tulyakov, Atri Rudra, Venu Govindaraju

University at Buffalo, The State University of New York, Buffalo, NY 14260
{mjmorse, hartloff, thomasef, jcschule, jacordar, tulyakov, atri, govind}@buffalo.edu

Abstract

In this work we evaluate the performance of generic local structures as template points for secure fingerprint matching. We present a generic template structure called an n-gon that derives from a set of n neighboring minutiae points. We secure templates consisting of sets of n-gons using the fuzzy vault construct to obfuscate the data. We report the matching performance of our system in terms of the ZeroFAR for comparison with other systems. We also briefly describe a keyed version of our system for comparison with secure systems that utilize a secret user key.

1. Introduction

Template security is an important, yet sometimes overlooked, aspect of fingerprint matching systems. As fingerprints become a more popular option to protect sensitive information, a database of fingerprint templates emerges as a lucrative target for an attacker. Our goal is to simultaneously protect such databases by securing the templates from attackers and achieve reasonable levels of matching accuracy. We make progress towards this goal by utilizing templates that combine global and local matching in a single template and secure them using the fuzzy vault scheme [13].

For databases storing string-based data such as passwords, credit card information, and national identification numbers, responsible database managers solve this problem by storing only encryptions or hashes of the data. Since these strings will be the same each time they are used, they can be verified by checking if the stored encryption or hash matches that of the submitted string exactly. This enables the system to function while never revealing the plain text data. When matching fingerprints however, there is no straightforward way to encrypt the data that still allows for accurate matching in the encrypted space since two readings of the same fingerprint will not match exactly on all values.

To construct templates for secure matching, we first divide a fingerprint into a set of local structures such that readings from the same fingerprint will match exactly on a subset of the structures. There has be work using local structures including those composed of pairs of minutiae [17, 8], triplets of minutiae [26], and sets of 5 minutiae points [12, 10]. In this work we generalize the local structures to contain n minutiae points which can be adjusted as a system parameter. We name these generic constructs n-gons. In Section 4 we compare several different methods of applying these n-gons to form the fingerprint templates. These are designed to capture local information from neighboring minutia points, as well as global information by retaining the absolute orientation of each point.

To secure these templates, it would seem simple to apply a one-way hash function, such as SHA256, to each point and match based on the number hashes two templates have in common. This is fine provided there are enough possible template points to make a brute force attack infeasible, though this is not the case with n-gons which eliminates the possibility of using a straght-forward hashing scheme. For this reason, we turn to the fuzzy vault [13] to secure the templates. This scheme provides security by obfuscating the template with randomly added data called chaff points and corrects for errors using a Reed-Solomon decoder.

We use the security parameter λ from [9] to measure the security of the fuzzy vault which is based on the work of Kiayias and Yung [14]. This λ is a measure of the time it would take any algorithm to break the fuzzy vault. This proof requires the fingerprint template points to be uniformly distributed and non-correlated to achieve exactly the claimed security [18]. We use a binning technique to transform the distribution closer to uniform, though the minimum entropy is not yet ideal. Creating a uniform template point distribution is a constant focus of our work.

To measure the matching performance of our system, we report the ZeroFAR [23] which is the lowest FRR when the FAR=0. On the FVC2002-DB2 dataset, our best method resulted in a ZeroFAR of 10.47% with a failure to capture rate (FTC) of 0%. One system that provides comparison on the same problem is the popular fuzzy vault implementation

of [21]. In [21], Nandakumar et al. report a ZeroFAR of 14% with an FTC of 2% on FVC2002-DB2. Nandakumar et al. report results nearly matching ours by utilizing multiple fingerprint readings for enrollment and testing, though this would be an inappropriate comparison to our system as we use only one reading each for enrollment and testing.

We present mostly keyless systems in this work, meaning we do not rely on any secret information from the users other than their fingerprint readings. Introducing a key eliminates one of the primary advantages of biometrics in that they can't be lost of forgotten. However, we briefly present a keyed version of our system.

2. Related Works

The work on privacy protection in fingerprint templates can possibly be divided in two directions: creating a non-invertible, sometimes heuristically designed, transform, and by utilizing some previously known cryptographic method. Examples of first approach include global minutia coordinate transformation in [22], locally defined shifts of individual minutia positions [16], projections [6], transformations equivalent to projections [15], and transformations implicitly losing some original data, such as keeping the geometry of minutia triplets while discarding their position in [5]. Although these methods sometimes produce good matching performance, it might be difficult to prove their non-invertibility property. In contrast, the methods relying on existing cryptographic techniques inherit the proof of their non-invertability from cryptography theory. Examples of such methods include fingerprint fuzzy vault [13, 4, 25], cryptographic hash [2], fuzzy commitment [20], fuzzy extractors [1]. The most widely used cryptography based method, fuzzy vault [13], has been developed as a method for constructing private biometric templates consisting of an unordered set of values. The most frequently used representation of fingerprints as a set of minutiae points provided the first intended application of this method.

A number of fingerprint fuzzy vault systems [4, 25, 21] utilize concatenated single minutia coordinates as encoding values for the fuzzy vault. Such methods face the problem of fingerprint alignment during matching. Indeed, two scans of the same fingerprint will rarely have corresponding minutiae occupy same image coordinates due to translation, rotation and non-affine deformations. To unlock a fuzzy vault, a sufficient number of minutia points in the test fingerprint should have the coordinates identical to the corresponding minutiae in enrolled fingerprints. The use of helper data, e.g. a stored set of high curvature points [21], has been proposed as a way to efficiently find the parameters of an alignment transformation. Note, that it is possible not to utilize helper data, but search a range of transformation parameters in brute-force approach to matching, but the matching performance is expected to decrease.

Another way to construct a fingerprint fuzzy vault is to utilize translation and rotation invariant features or descriptors extracted from the neighborhoods around minutiae. For example, [17, 8] utilize all possible minutia pairs in the fingerprint to compose vault values; the matching in such systems is equivalent to geometric hashing techniques where the origin is defined by a single minutia and its direction. [26] proposed to use translation and rotation independent parameters of minutia triplets consisting of neighboring minutia for fuzzy vault values. [12] compared the use of minutia 5-plets, triplets and Voronoi triangles for fuzzy vault construction, and concluded that triplets give best alignment results.

There are trade-offs in using both types of approaches to fuzzy vault construction. It has been noted in general research on fingerprint matching [11] that utilizing the local minutia descriptors, e.g. derived from the positions of neighboring minutia, provides better matching performance than simply relying on original minutia positions. As a result, it would be beneficial to incorporate local descriptor information in a fuzzy vault [19]. On the other hand, relying exclusively on local descriptors consisting of rotation and translation invariant features can lead to a matching performance decrease, since the global correspondence between local matches of descriptors is ignored in such cases. In the current paper, we investigate the use of both local and global information for fuzzy vault construction.

3. Security

In this section we present and discuss factors that effect the template security of our system. We first present a review of the fuzzy vault scheme along with a discussion of a security parameter for fuzzy vault as seen in [9]. We also present considerations for limitations on the scoring function of our matching schemes, such that they do not compromise the fuzzy vault security.

3.1. Fuzzy Vault

The fuzzy vault system first generates a polynomial p of degree z, then calculates $p(t)$ for each quantized template point t of the enrolling fingerprint reading. The points $(t, p(t))$ are stored, along with randomly generated chaff points. To unlock the vault, the system extracts all the points in the vault where $t = t'$ for any template value t' of the testing fingerprint. The set of extracted points is then used as input for a Reed-Solomon decoder. If enough genuine points were found, the output of the decoder will be the original polynomial, p. To confirm that the output polynomial generated by the decoder is in fact the original, we apply a cyclic redundancy check (CRC).

To measure the fuzzy vault security, we use the parame-

ter defined by [9]

$$\lambda = \sqrt{z(c+g)} - g, \tag{1}$$

where z is the degree of the polynomial, c is the number of chaff points, and g is the number of genuine points. This security parameter is based on the hardness of Reed-Solomon decoding, which is a proposed cryptographic primitive [14]. This makes λ analogous to bits of security as it measures the logarithmic runtime required to crack a fuzzy vault.

3.2. Scoring Considerations

One limiting factor, which affects the security of a fingerprint template, is the adversary's ability to attempt to reconstruct a template by simply submitting random fingerprint templates until a match is confirmed. With this information, an adversary can look at the genuine points provided by the unlocked vault and reassemble the fingerprint.

For this reason, we note that we cannot claim better security than the False Accept Rate (FAR) [23]. Thus, we require an FAR on of $2^{-\lambda}$. Since, for any reasonably secure value of λ, we do not have a database large enough to measure such a small FAR, in our experiments we require the FAR to be zero to ensure a secure mode of operation.

4. Fingerprint Matching

We present an evolution of fingerprint matching schemes based on generic local structures of minutia points. We begin with clusters of n minutia points extracted from a fingerprint. We then extend this notion by rotating the clusters through a range of angles for enrollment and testing. Varying enrollment and testing clusters further improves our scheme. In particular, we draw attention to the trade off between matching performance and number of features required to lock in the fuzzy vault to provide security. It makes intuitive sense that locking more fingerprint features in the fuzzy vault will improve matching performance, but will require an increasingly large number of chaff points to keep the fuzzy vault secure. Section 4.1 is a general purpose scheme that provides decent matching performance and creates a medium number of features from a given fingerprint. Section 4.2 has slightly worse matching performance with less features to store in the fuzzy vault, while Section 4.3 provides excellent matching performance at the expense of storing many more features.

4.1. n-gons

From a given fingerprint template $T(f)$ with m minutia points, we create a set $C_k(n)$ of n-clusters that characterize the fingerprint, each of which we define as an n-gon. To create this set of n-gons, we proceed in the following manner. For some fixed k such that $n \leq k$ and each minutia point p_0^i in $T(f)$, we find the k minutia points with the smallest Euclidean distance in x and y from p_0^i, namely $p_1^i, p_2^i, \ldots, p_k^i$, and form the set S_i of all possible subsets of size n from the set $\{p_j^i\}_{j=0}^k$ of size $k+1$. For the ith minutia point, there are exactly $\binom{k+1}{n}$ n-gons. The union of all the S_i's produces $C_k(n)$.

We note that the ith minutia point p_i is defined by x_i for the x-coordinate of p_i, y_i for the y-coordinate of p_i, and θ_i for the orientation of p_i. Thus, a given n-gon has $3n$ attributes. Within each n-gon, we define the left-most minutia point as the origin and scale the other minutia points into this new reference coordinate system. This reduces the number of attributes defining an n-gon to $3n-2$, yet eliminates all spurious data. For each n-gon in $C_k(n)$, we quantize each x_i,y_i, and θ_i of each minutia point and concatenate them into a value $t = x_0 \circ x_1 \circ \ldots \circ x_n \circ y_0 \circ y_1 \circ \ldots \circ y_n \circ \theta_0 \circ \theta_1 \circ \ldots \circ \theta_n$. The resulting t value is an element of a field whose size is determined by the number of discrete bins used in the quantization method. The number of bins may be different among parameters. Our optimal matching of fingerprints utilized n-gons quantized to $t \in \mathbb{F}_{2^{27}}$, where $\mathbb{F}$ is a finite field. An example of an n-gon is seen in Figure 1.

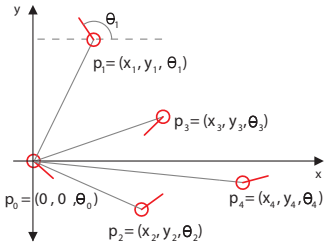

Figure 1. An n-gon where $n = 5$

4.2. n-gons Rotated Testing

The performance of naive n-gons is surprisingly effective as seen in Section 5.3, yet we aim to improve upon this scheme. The primary drawback of the matching algorithm in Section 4.1 is its sensitivity to rotations, since a $45°$ rotation will likely change the left-most point in the n-gon and, as a result, all resulting scaled values in the cluster. This is clearly an issue, since rotational variation of up to $\pm 50°$ between fingerprint readings is to be expected.

Under this improved method, a user still enrolls their fingerprint as a set of quantized n-gons and locks it in a fuzzy vault. However, to test a fingerprint's validity, we construct a set of n-gons from the tested fingerprint in addition to those generated from rotated copies of the fingerprint at two degree intervals over the range $[-50°, 50]$. An example of the rotation of one n-gon is seen in Figure 2.

We quantize this much larger set of n-gons and use it to attempt to unlock the vault. Since minor rotations in a user's fingerprint reading are common, always rotating the testing fingerprint over a set range increases the likelihood of creating n-gons that are oriented closely to those locked in the

vault. As a result, the system parameters which achieve optimal performance for this method use a smaller value of k. This provides a significantly smaller number of features per enrolling template, which in turn allows us to decrease the number of chaff points in the fuzzy vault while maintaining the security parameter λ.

4.3. n-gons All Rotations

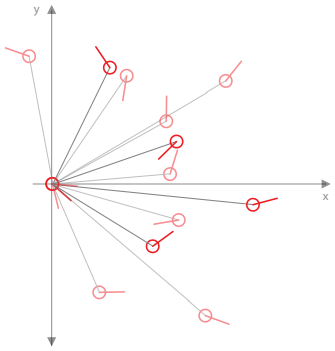

Figure 2. Two rotations of a 5-gon

In order to further improve matching performance, we instead lock the set of n-gons created from a spectrum of rotations described in Section 4.2 inside of the fuzzy vault, and validate a user's fingerprint by attempting to unlock the fuzzy vault in the same manner. As one could imagine, equal error rates for this scheme are better, as seen in Section 5.3, since we are taking rotation invariance into account for enrollment and testing rather than only the latter, allowing for far more overlap when attempting to unlock the vault.

The downside of this method is the sheer quantity of t-values that must be stored in the fuzzy vault. In order to maintain security, we must appropriately increase the number of chaff points to compensate for the greater number of genuine points. A fuzzy vault for an individual user in this scheme is on the order of several hundred kilobytes and can even reach several megabytes, thus care must be taken when choosing quantization parameters for this scheme to make it feasible in practice.

5. Results

In this section we analyze the experimental performance of the matching schemes described in Section 4 relative to one another. We illustrate the trade-offs between the matching performance and security of n-gons, n-gons with rotation testing (n-gons RT), n-gons with all rotations (n-gons AR). In particular, we show that n-gons provides a reasonable trade-off between matching performance and feature set sizes, which affect the security of the fuzzy vault. n-gons with rotation testing provides the smallest feature set sizes, which are ideal for fuzzy vault security measures, while n-gons with all rotations provides the best matching performance at the cost of increased vault size with more strict parameter ranges for practical implementation.

We conducted the experiments on the first two fingerprint databases from the Second International Fingerprint Verification Competition (FVC2002/DB1-DB2). The minutiae points were extracted by finding large curvature points on the contours of the binarized fingerprint images [7]. Additionally, to improve the performance of minutia extraction, the fingerprints were enhanced by the short time Fourier transform algorithm [3]. We also remove suspect spurious minutia points by removing all points within a euclidean distance of 3 from each other. A standard testing protocol for calculating all possible 2800 genuine (100 persons with $\frac{8\cdot7}{2}$ matches) and 4950 (1 print of each person matched against 1 print of another, or $\frac{100\cdot99}{2}$) impostor matches were used.

5.1. Matching Performance

By the nature of our matching schemes, we require a minimum number of minutia points to create at least one feature. Because of this, we introduce a Failure To Capture (FTC) mechanism that eliminates fingerprints which do not have enough minutia to create at least one feature.

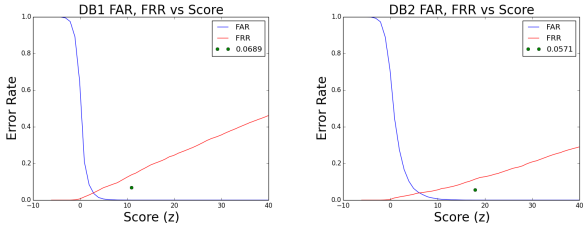

Figure 3. FAR vs FRR curves for our best performing matching scheme - n-gons all rotations. The ZeroFARs are .1266 and .1048 and the score thresholds are 10 and 17 respectively.

We present summaries of our best results from the matching schemes in Section 4. We note that the FTC for n-gons and n-gons rotation testing are .125% and the FTC for n-gons all rotations is 0. Table 1 presents results concerning the ZeroFAR and the scoring thresholds complying with the parameter constraints necessary for security guarantees as discussed in Section 3.2.

As discussed in Section 3.2, by matching fingerprints at the threshold where FAR $= 0$, we prevent an adversary from reconstructing a template by simply submitting random fingerprints until access is granted. Since FAR $= 0$, we can increase λ further by increasing the number of chaff points used in the fuzzy vaults, however this has potential practical limitations due to the space required by fuzzy vaults with so many points. This is discussed in detail in Section 5.2.

5.2. Security and Space Analysis

The λ values provided in Table 1 are reasonable for template security and can be increased by adding additional chaff points to the fuzzy vaults. This is ideal for increasing the template security for the system.

DB1	FAR	FRR	z	c	λ
n-gons	0.0	0.3667	6	50,000	259.3
n-gons RT	0.0	0.3385	9	20,000	133.4
n-gons AR	0.0	0.1379	10	170,000	166.2
DB2	FAR	FRR	z	c	λ
n-gons	0.0	0.2304	6	50,000	185.7
n-gons RT	0.0	0.3311	14	20,000	167.0
n-gons AR	0.0	0.1143	18	170,000	276.9

Table 1. Results of various parameters for our different methods. The difference in λ values between the databases can be accounted for by Equation (1) and the average number of genuine features from Table 2, g, being higher for DB2 than for DB1, as well as the different values for z.

However, there is one practical caveat to this model. Storing a fuzzy vault with, for example, 1500 genuine points and 250,000 chaff points can require a large amount of space. To calculate the size of a fuzzy vault in kilobytes, we introduce the following equation:

$$Size = \frac{q \cdot (g + c)}{8 \cdot 1024} \tag{2}$$

where q is the number of bits needed to quantize a genuine point. Using Equation 2, we calculate the average fuzzy vault sizes for the experiments conducted in Table 1.

DB1	g	c	q	Size [KB]	HTER
n-gons	290	50,000	27	165.75	0.1834
n-gons RT	294	20,000	27	66.89	0.1693
n-gons AR	1142	170,000	27	564.07	0.0690
DB2	g	c	q	Size [KB]	HTER
n-gons	364	50,000	27	165.99	0.1152
n-gons RT	367	20,000	27	67.13	0.1656
n-gons AR	1480	150,000	27	565.18	0.0567

Table 2. Here we demonstrate the space trade-off for the three schemes. n-gons rotation testing provides the smallest fuzzy vault size at 66.89 KB and 67.13 KB for DB1 and DB2 respectively. n-gons all rotations requires larger file sizes of 564.07 KB and 565.18 KB for DB1 and DB2 respectively. This shows the trade-off in storing more hashes in the fuzzy vault. Higher matching performance can be achieved by increasing the number of hashes at the cost of large file sizes.

From Table 2, we see there is a trade-off between the matching performance and the required size of the fuzzy vault, caused by the number of genuine and chaff points stored inside. For the schemes with smaller feature set sizes, n-gons and n-gons with rotation testing, we can increase λ to be much higher by adding in as many more chaff points as desired. Since the chaff points are generated randomly from such a large field, they are not likely to match with genuine points and so matching performance would remain virtually the same.

5.3. Discussion

In this paper, we have presented multiple variations of a secure fingerprint matching scheme that uses quantized generalized substructures of minutia points. All three schemes presented are similar in that they form subsets of n points and quantize the subsets in a translation-invariant and rotation-variant way, but with slight changes in how rotational variation is addressed. By maintaining this partial rotation invariance, we allow for more global matching of fingerprint templates than totally invariant schemes, and this allows us to deter impostor matches, which is vital to maintaining template security.

From these experimental results, we find that n-gons with all rotations provides optimal matching performance at the cost of large fuzzy vault sizes to secure the template. n-gons with rotation testing provides optimal vault sizes and provides room for increasing the fuzzy vault security parameter λ greatly, but does not achieve the low HTER values of n-gons with all rotations. n-gons represents a compromise of the other two schemes in that it treads the line between accuracy and vault size. To successfully use these schemes, one must find an optimal balance of the matching performance, the number of chaff points necessary for the desired λ security, and size of the fuzzy vault.

As described in Section 1, uniformity of the template points is a persistent challenge for secure fingerprint matching. All of our particular quantization techniques and binning methods are an attempt to decrease the level of non-uniformity in the data, but do not produce perfectly uniform quantized values. Quantization to a unique distribution remains an elusive target of this area of research. We will continue to search for methods to achieve this uniformity.

6. Keyed-System Comparison

We briefly discuss a keyed version of our system in this section. In the keyed version, each user is required to maintain a secret key k separate from their fingerprint template that will be used for verifying their identity. This can be a large burden for the user, which is why we focus primarily on the keyless system. The keyed system is very similar to the keyless system, but with the following changes: (i) The user concatenates the secret key with each template point t generated from their fingerprint to form $t \circ k$. (ii) The user then applies a hash function h to each of these new template points and builds a fuzzy vault using these hashed values $h(t \circ k)$.

We implemented this using an 80 bit random key and SHA256 as the hash function. Since the nature of the system requires a user to match on both the fingerprint and

the key simultaneously, every impostor score was 0, as expected, since the probability of matching the key itself is 2^{-80}. We chose a set of parameters that didn't result in any genuine scores of 0 which yielded an $FRR = 0$ and $FAR = 0$. In addition, the system has 80 bits of security from k which can be increased by increasing the key size.

One drawback of this system is that it cannot be used as an identification system since each user's template will be hashed using a different key.

We note that if a users key is compromised, the users template is still protected by the fuzzy vault and the system effectively becomes the keyless system described in the rest of this paper. Thus, our system holds up in the stolen-token scenario described in [24]. If a user's fingerprint is compromised, the system is still secured by the secret key. If a users fingerprint and key are both compromised, the user can re-enroll using a different key.

Acknowledgments. This research is supported by NSF grants TC 1115670 and I/UCRC 1266183.

References

[1] A. Arakala, J. Jeffers, and K. Horadam. Fuzzy extractors for minutiae-based fingerprint authentication. In *ICB 2007, LNCS 4642*, pages 760–769. Springer-Verlag, 2007.

[2] T. E. Boult, W. J. Scheirer, and R. Woodworth. Revocable fingerprint biotokens: Accuracy and security analysis. In *CVPR*, 2007.

[3] S. Chikkerur, A. N. Cartwright, and V. Govindaraju. Fingerprint enhancement using STFT analysis. *Pattern Recognition*, 40(1):198–211, 2007.

[4] T. Clancy, D. Lin, and N. Kiyavash. Secure smartcard-based fingerprint authentication. In *WBMA*, Berkeley, CA, USA, 2003.

[5] F. Farooq, R. Bolle, T.-Y. Jea, and N. Ratha. Anonymous and revocable fingerprint recognition. In *CVPR*, pages 1–7, June 2007.

[6] M. Ferrara, D. Maltoni, and R. Cappelli. Noninvertible minutia cylinder-code representation. *Information Forensics and Security, IEEE Transactions on*, 7(6):1727–1737, 2012.

[7] V. Govindaraju, Z. Shi, and J. Schneider. Feature extraction using a chaincoded contour representation of fingerprint images. In *4th international conference on Audio- and video-based biometric person authentication*, volume 2688, pages 268–275, Guildford, UK, 2003. Springer-Verlag.

[8] X. Q. Guo and A. Q. Hu. The automatic fuzzy fingerprint vault based on geometric hashing: Vulnerability analysis and security enhancement. *Int. Conference on Multimedia Information Networking and Security*, 1:62–67, 2009.

[9] J. Hartloff, M. Bileschi, S. Tulyakov, J. Dobler, A. Rudra, and V. Govindaraju. Security analysis for fingerprint fuzzy vaults. In *Proc. SPIE*, volume 8712, pages 871204–871204–12, 2013.

[10] J. Hartloff, J. Dobler, S. Tulyakov, A. Rudra, and V. Govindaraju. Towards fingerprints as strings: Secure indexing for fingerprint matching. In *ICB*, pages 1–6, 2013.

[11] T.-Y. Jea and V. Govindaraju. A minutia-based partial fingerprint recognition system. *Pattern Recognition*, 38(10):1672–1684, 2005.

[12] J. Jeffers and A. Arakala. Fingerprint alignment for a minutiae-based fuzzy vault. In A. Arakala, editor, *Biometrics Symposium, 2007*, pages 1–6, 2007.

[13] A. Juels and M. Sudan. A fuzzy vault scheme. *Des. Codes Cryptography*, 38(2):237–257, 2006.

[14] A. Kiayias and M. Yung. Cryptographic hardness based on the decoding of Reed-Solomon codes. *IEEE Transactions on Information Theory*, 54(6):2752–2769, 2008.

[15] G. Kumar, S. Tulyakov, and V. Govindaraju. Combination of symmetric hash functions for secure fingerprint matching. In *ICPR*, pages 890 –893, aug. 2010.

[16] C. Lee, J.-Y. Choi, K.-A. Toh, and S. Lee. Alignment-free cancelable fingerprint templates based on local minutiae information. *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, 37(4):980–992, 2007.

[17] S. Lee, D. Moon, S. Jung, and Y. Chung. Protecting secret keys with fuzzy fingerprint vault based on a 3d geometric hash table. In *Adaptive and Natural Computing Algorithms*, volume 4432 of *LNCS*, pages 432–439. 2007.

[18] J. Merkle, M. Niesing, M. Schwaiger, H. Ihmor, and U. Korte. Security capacity of the fuzzy fingerprint vault. *International Journal on Advances in Security*, 3(3 & 4):146–168, 2010.

[19] A. Nagar, K. Nandakumar, and A. K. Jain. Securing fingerprint template: Fuzzy vault with minutiae descriptors. In *ICPR*, pages 1–4, 2008.

[20] K. Nandakumar. A fingerprint cryptosystem based on minutiae phase spectrum. In *IEEE WIFS*, pages 1–6, 2010.

[21] K. Nandakumar, A. K. Jain, and S. Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Transactions on Information Forensics and Security*, 2(4):744–757, 2007.

[22] N. Ratha, S. Chikkerur, J. Connell, and R. Bolle. Generating cancelable fingerprint templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(4):561–572, 2007.

[23] W. Scheirer and T. Boult. Bipartite biotokens: Definition, implementation, and analysis. In M. Tistarelli and M. Nixon, editors, *Advances in Biometrics*, volume 5558 of *LNCS*, pages 775–785. Springer Berlin Heidelberg, 2009.

[24] A. Teoh, B. Jin, T. Connie, D. Ngo, and C. Ling. Remarks on biohash and its mathematical foundation. *Inf. Process. Lett.*, 100(4):145–150, Nov. 2006.

[25] U. Uludag, S. Pankanti, and A. K. Jain. Fuzzy vault for fingerprints. In *in Proc. AVBPA, Lecture Notes in Computer Science 3546*, pages 310–319. Springer, 2005.

[26] S. Yang and I. Verbauwhede. Automatic secure fingerprint verification system based on fuzzy vault scheme. In *ICASSP*, volume 5, pages 609–612, 2005.