Transforming minutiae for protecting fingerprint data

Tohari Ahmad[1], Herleeyandi Markoni[2], Waskitho Wibisono[3], Royyana M. I.[4]
Department of Informatics
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya, 60111, Indonesia
{[1]tohari, [3]waswib, [4]roy}@if.its.ac.id, [2]herleeyandi@gmail.com

***Abstract*—Fingerprint is one of popular biometric identifiers for authenticating users. Its permanence characteristic has made it reliable to use. Therefore, fingerprint data should be protected and not be disclosed to unauthorized parties. Nevertheless, its instability scanning result has made conventional cryptographic algorithms do not work. In this paper, we propose an algorithm for protecting fingerprint data by transforming it before being stored in a database. Once it has been transformed, it is hard to reconstruct its original data. In the authentication process, the fingerprint is also transformed in the same way as its registration. So, the original data have never been revealed. The experimental result, which is done by using a public database, shows that the error rate caused by the transformation is relatively low.**

Keywords—fingerprint; data security; transformation; data protection

I. INTRODUCTION

Biometrics has been popular to use in an authentication process due to some reasons [1]. Its characteristics as "something you are" have more advantages than two other authentication mechanisms (i.e., "something you remember" and "something you have"). For example, users may not be comfortable to use a password because he/she has to remember it. Moreover, a user often has many user IDs and their corresponding passwords, which make it difficult to memorize. It is not to say that a "good" password must be long and random. On the other hand, a token which helps users to easily generate passwords may not be practically used. This is because it can be left behind or even stolen.

Biometrics may not be able to overcome all those problems being faced by neither passwords nor tokens. Nevertheless, it has made it easy for users to do authentication. Here, a user does not need to hold a token whenever and wherever he/she goes because biometrics is always with him/her. As a result, this minimizes the possibility of losing such authentication tools.

In general, biometrics can be classified into two groups: behaviors and traits. While the first relates to the behaviors of the users, such as signature [2] whose data is protected by using a key which is generated from a password; the second relates to parts of human body whose identifier can be voice, face, fingerprint, etc. Among those biometric identifiers, fingerprint is one of potential candidates for the authenticating purpose [3].

Beside advantages the fingerprint provides, however, there is a weakness which should be overcome. This is because, the number of fingerprint of each user is limited. Once its pattern is compromised, the fingerprint cannot be used anymore. The existing pattern must be replaced by another pattern of fingerprint. Due to this limitation, there are only ten fingers which can be used for authentication. Further, if all of them are compromised, then they may be useless. There must be a mechanism to derive a different fingerprint pattern for the authentication purpose; and if it is compromised, then a new pattern can be easily generated from the same fingerprint. In this paper, we propose such mechanism by transforming fingerprint minutia.

The rest of this paper is organized as follows. Section 2 describes works relate to the fingerprint protection. Section 3 explains the proposed mechanism whose experimental results are provided in Section 4. Lastly, Section 5 draws the conclusion.

II. RELATED WORKS

A. Fingerprints

One reason of the fingerprint popularity in an authentication process is its uniqueness. It is determined by the pattern of minutiae points which spread over the fingerprint. In general, minutiae points can be classified into two groups: ridge ending and bifurcation; which these types can be one of minutiae characteristics. The others are: the coordinate (x,y) and orientation (θ) [4]. These characteristics are often used in a fingerprint matching algorithm.

B. Fingerprint protection

As far as we know, there two general types of fingerprint protection methods: biometric cryptography and feature transformation [5]. The former protects a fingerprint pattern by employing a method-like cryptography; while the latter protects a fingerprint pattern by transforming fingerprint features into another form in order to have a new pattern. The examples of biometric cryptography are fuzzy commitment [6] [7], fuzzy vault [8] [9] and fuzzy extractor [10]; and the example of feature transformation is [11] [12] [1]. In this paper, we focus on this feature transformation mechanism.

978-1-4799-1723-5/15/$31.00 ©2015 IEEE

In the fuzzy commitment scheme, a fingerprint pattern (query) is categorized match if it is "closed" enough to that of the previous stored template. The query can be refined by employing an error correcting code (ECC) such as RS and Hamming, according to the unique pattern which was generated based on the template. It is appropriate to the fact that the scanning result of the same fingers may result to only similar data. This method has been implemented, for example in [7]. In further development, fuzzy vault [8] is introduced. Here, a secret string is locked in a vault which can only be unlocked by appropriate fingerprint. The vault consists of both the original and chaff minutia points, so that, it is relatively hard for an attacker to obtain both the secret string and the original minutia points. So, this method can be used to protect both those data. Similar to fuzzy commitment, it needs to implement ECC to correct the query fingerprint. There are some research which has implemented fuzzy vault for protecting fingerprint, such as [9]. Different from the previous research, fuzzy extractor [10] generates a string based on the biometrics (fingerprint, in more specific). Therefore, this scheme does not require to obtain the string separately. Two fingerprint patterns have to be "closed" (similar) in order to be able to produce a same string. If two fingerprints are too "far", then the generated string is different. Overall, these fingerprint protection methods can be used to protect the fingerprint pattern, the secret message, and also to do authentication.

Feature transformation is intended to make the fingerprint pattern as random as possible such that it is different enough from the original. This can be done either by transforming the fingerprint pattern into another form (domain) or transforming it into a same form. Ahmad and Hu [1] design a transformation method by mapping the minutia points into a line whose slope and direction are to be the transformation key. The line itself is partitioned, and the number of mapping points in each partition is counted. These numbers are processed by using MAE for the comparison between the template and the query purpose.

Different from the previous method which the transformation is done by utilizing both global and local features of fingerprint, some other methods only use local features for the transformation. It is intended to have a better result, by considering that it is hard to accurately detect and locate the global feature (e.g., core point). This characteristic may affect the accuracy after the transformation. Inspired by the research of Rata et al [11] and Xi and Hu [9], Ahmad and Han [12] propose a transformation method by using both Cartesian and polar coordinates [12]. Firstly, the Cartesian-based transformation is applied by rotating each minutia points according to their location/square. Secondly, the polar-based transformation is implemented by utilizing tracks and sectors which have been constructed at the beginning.

Jin et al [13] generate a fingerprint template by using Randomized Graph-based Hamming Embedding (RGHE). In this method, a minutiae descriptor is presented to obtain invariant features which is embedded into a Hamming space. In 2014, Wang and Hu [14] propose an alignment-free cancelable template which is believed to be efficient. In its development, the fingerprint template is constructed via circular convolution. For the same purpose, Prasad and Kumar [15] propose a method by developing rectangles around minutiae points. This is followed by some steps, such as calculating rotation and translation invariants.

III. PROPOSED METHOD

As previously described, in this research, we protect the fingerprint template by focusing on the fingerprint feature transformation, specifically the geometric transformation. There are some steps which construct the overall transformation. Those are: sector mapping, rotation, translation and checking the maximum distance. These steps are one of the bigger steps which are designed to be suitable for use in a fingerprint authentication process. The overall method is depicted in Fig. 1 which can be described as follows.

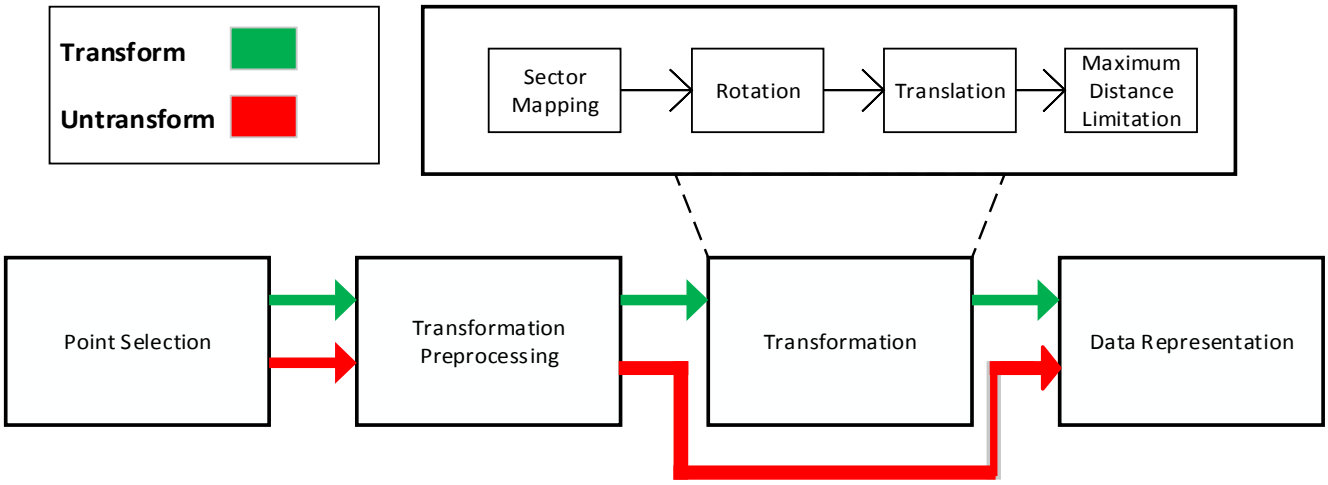

Fig. 1. The proposed method

A. Minutiae selection

When the finger is scanned, the information about minutiae points is obtained. As previously depicted, some of them is used for fingerprint matching: the coordinate of minutia point, the orientation and the type. This can be presented as in (1).

$$\begin{cases} P_o \in \Psi \\ P_o = \{(m_i)_0\}_{i=1}^{n} \\ m_i = \{x_i, y_i, \theta_i, t_i\} \end{cases} \quad (1)$$

where P_o is a set of n minutiae points m each of which comprises its abscissa x, ordinate y, orientation θ and type t. Here, Ψ represents a set of fingerprints P_o.

Not all minutiae points in P_o, however, are used in this protection scheme. Instead, only k of them are selected among n by using convex hull which is implemented by Graham Scan algorithm in [16]. This point selection is performed three times, where sets of points in the previous stages are excluded in the next selection process. Therefore, only those 1st, 2nd, and 3th outer minutiae points are used in the template development process.

B. Preprocessing

Transformation is performed to the all selected minutiae points, where each of them consecutively acts as the center of the fingerprint. Let m_c be this center of the fingerprint. The new minutiae point coordinates are constructed according to the respective orientation. In this case, the orientation of m_c is to be the abscissa of the coordinate, and the coordinate of rest (neighboring) minutiae points are recalculated by using (2) and (3), where (a, b) is the original coordinate of m_c; (x', y') and $(x", y")$ are the coordinate of neighboring points after rotation and translation (final coordinate), respectively.

$$\begin{bmatrix} x\prime \\ y\prime \end{bmatrix} = \begin{bmatrix} \cos\omega & \sin\omega \\ -\sin\omega & \cos\omega \end{bmatrix} * \begin{bmatrix} x-a \\ y-b \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix} - \begin{bmatrix} a \\ b \end{bmatrix} \quad (3)$$

C. Generating sector

As in [17], this protection method employs sectors for transformation. Different from it, the number of sectors is fixed to 16, such that its size is 22.5º, as depicted in Fig. 2.

D. Set of keys

For the transformation, it needs to have keys which is used to derive the cancelable template. In this proposed method, this set of keys is provided in (4), where comprises 16 subset of keys, according to the number of generated sectors. Each of them consists of s_i which represents the sector i to be processed; $f1_i, f2_i, f3_i$ which depict the 1st, 2nd and 3rd rotation of s_i; $x1_i, y1_i, x2_i, y2_i$ which perform the 1st and 2nd translatation of minutiae abscissa and ordinat.

$$\begin{cases} K = \{B_i\}_{i=1}^{16} \\ B_i = \{s_i, f1_i, f2_i, f3_i, x1_i, y1_i, x2_i, y2_i\} \end{cases} \quad (4)$$

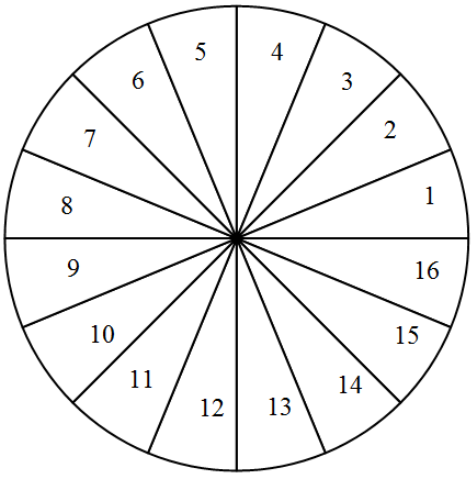

Fig. 2. An exampleof generated sectors

E. Transformation

The transformation is carried out in three steps. In the first two steps, the selected minutiae points (obtained in Section 3.A) are rotated and translated; while in the last steps, only rotation is performed. Since this transformation is applied to all selected points, all of them are to be the center of the transformation, consecutively. So, the number of generated template is same as the points, and the number points in each template is the number of points minus one. Let ϕ_o be the set of generated template; T_i and (d_i, β_i) be the template and its respective fingeprint fiture (see Section 3.F), respectively, the generated template can depicted in (5).

$$\begin{cases} \phi_o = \{T_i\}_{i=1}^{n} \\ T = \{(d_i, \beta_i)\}_{i=1}^{n-1} \end{cases} \quad (5)$$

The transformation is started by mapped minutiae to the respective sector. This is done by calculating the angle (α) between the line of minutia – center of the transformation (m_c) and the ordinat line x'. The value of α is determined by (6). This is done to all minutiae such that all of them are in the appropriate sector.

$$\alpha = \arctan\left(\frac{y}{x}\right) \quad (6)$$

A sector, including minutiae in it, is rotated according to the value of f. Once the rotation has been done, the minutiae points are translated according to the values of x_i, y_i. Let P, P' and P'' be the coordinate of a minutia point before transformation, after rotation and after translation, respectively, which can be presented in (7). The angle after rotation is in (8), while the rotation and translation are carried out by using (9) and (10), respectively.

$$\begin{cases} P_i = \{x_i, y_i\} \\ P'_i = \{x'_i, y'_i\} \\ P"_i = \{x''_i, y''_i\} \end{cases} \quad (7)$$

$$\omega = 22{,}5° f_i \quad (8)$$

$$\begin{bmatrix} x\prime_i \\ y\prime_i \end{bmatrix} = \begin{bmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{bmatrix} * \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} x''_i \\ y''_i \end{bmatrix} = \begin{bmatrix} Txi \\ Tyi \end{bmatrix} + \begin{bmatrix} x'_i \\ y'_i \end{bmatrix} \quad (10)$$

It needs to limit the maximum distance of translated minutiae, x''_i, y''_i because it may be too far from m_c. This is done by applying modulus operation to the distance value. If $rmax$ and $rtrans$ are the specified maximum distance and the actual minutia distance values, then the obtained result is determined by (11) and (12) whose angle is calculated by (13). In the case the result of (12) is 0, it should be changed to $rmax$, as in (14).

$$rtrans = \sqrt{(x'')^2 + (y'')^2} \quad (11)$$

$$finalr = mod(rtrans, rmax) \quad (12)$$

$$\sigma = \arctan\left(\frac{y''}{x''}\right) \quad (13)$$

$$finalr'f(x) = \begin{cases} rmax, & \text{if } finalr = 0 \\ finalr, & \text{if } finalr \neq 0 \end{cases} \quad (14)$$

F. Feature representation

The transformed data is represented in a specified form before being stored as a template. There are some purposes of this data representation, e.g., hardening the template security. In this research, the template is represented in two forms: arc and sum of angles, which can be described as follows.

1) Arc (d): The value of d is obtained by using (15) as below, where $\propto'''$ and r are the angle after the 3rd rotation and the distance between the respective point and the center.

$$d = \frac{\min(\propto''', 360^o - \propto''')}{360^0} * 2\pi r \quad (15)$$

2) Sum of angles (β): Let θ, $\Delta\propto'$ and $\Delta\propto''$ be the orientation of minutiae and the minimum angle obtained from the first and second transformation, respectively. The value of β can depicted in (16).

$$\beta = \Delta\alpha' + \Delta\alpha'' + \theta \quad (16)$$

All generated template and query are represented by (d, β).

G. Verification

Verification is conducted by comparing the template and the query. Similar to our previous research [12] [17], the verification process is performed by calculating the minutiae information (local) and the number of matched minutiae (global). In this proposed method, the local verification is as follows.

1) Comparing all minutiae in the template and query: Calculating the difference as in (17) and (18).

$$s_d = dT - dQ \quad (17)$$

$$s_\beta = \beta T - \beta Q \quad (18)$$

2) Checking s_d and s_β: different from [17] which sums the values in Δf, here only s_β is used for further verification (see (19)) whose result is stored in a matrix, according to the respective row and column.

$$f = \begin{cases} -1, & s_\beta > t1, s_d > t2 \\ s_\beta, & s_\beta \leq t1, s_d \leq t2 \end{cases} \quad (19)$$

The global verification is carried out by counting the number of match minutiae as resulted by local verification. If the number of corresponding minutiae is higher than the specified threshold, then the template and the query are matched.

IV. Experimental Result

The experiment is carried out by using a public database fvc2002db2a [18] which consists of 100 pairs of fingerprint. It is performed by comparing each pair of fingeprint and counted the matched ones. This is to measure the ability of the method to accept the correct fingerprint which is represented by True Positive Rate (TPR). In addition, we also measure the capability to reject the incorrect fingerprints which is represented in False Acceptance Rate (FAR). It is done by comparing all templates to all queries except its corresponding fingerprint. So, this results to 10000 comparison in total. In addition, [16] is used for the convex hull implementation.

A. Performance of Transformed Data

The value of $t1$ is varied with fixed $t2 = 9$. As shown in Table I, $t1 = 7.5$ gives better performance than others, where TPR and FPR are relatively balanced. That is, the error is around 9%. There are some possibilities why this error level and $t1$ are obtained. For example, after the transformation is done, we have d along with $\Delta\alpha'''$ and r. There may be similar d with different $\Delta\alpha'''$ and r. Once the threshold is applied, those similar d may be considered same, so the result is affected. This spurious matching condition can occur because of the instability of fingerprint scanning or inaccurate tolerant value (threshold) [19].

B. Performance of Untransformed Data

The result of the transformation is evaluated by comparing it to the performance of the unstranformed data by using the same variable values. This is to measure how much the increase of the error rate. This experiment result is provided in Table II. It is depicted that for $t1 = 7.5$, there is an increase about 5% of FAR and a decrease of 3% of TPR. Overall, there is positive and negative trends for TPR and FAR, respectively.

C. Effect of Convex Hull

This experiment is to find the level of convex hull and its effect to the performance. As shown in Table III, the level 1, 2, and 3 provide better result than others. The level 4 and higher are not used because, in some cases, all minutiae are included.

D. Security Analysis

The template which is stored in the database is the tuple (d, β). In order to find d, it needs to have $\Delta\alpha'''$ and r; while $\Delta\alpha'''$ itself needs the minutiae coordinate after the third rotation and the key. In case the key can be found, it is still hard to find the minutiae because the information about sector and the post third rotation are not stored. Furthermore, not all minutiae points are used in the transfomation because of the use of convex hull. This removes some information of the original minutiae.

V. Conclusion

A fingerprint protection method has been presented in this paper. The protection is done by transforming the fingerperprint information. As in the experimental result, it is hard to reconstruct the original fingerprint by using this transformed

TABLE I PERFORMANCE OF TRANSFORMED DATA WITH VARIED T1

TPR (%)	FAR (%)	T1
81	2.929293	6
81	4.414141	6.5
82	6.515152	7
91	9.474747	7.5
92	12.58586	8
95	16.79798	8.5
97	21.20202	9

TABLE II PERFOMANCE OF UNTRANSFORMED DATA WITH VARIED T1

TPR (%)	FAR (%)	T1
85	0.9393939	6
85	1.7676768	6.5
87	2.9393939	7
87	4.2424242	7.5
87	6.1717172	8
90	8.1919192	8.5
92	10.676768	9

TABLE III EFFECT OF CONVEX HULL

Level of Convex Hull	TPR(%)	FAR(%)
1	0	0
2	0	0
3	0	0
1 and 2	18	0
1 and 3	6	0
2 and 3	19	0
1, 2, and 3	56	0
All points	88	10.64646

information. Moreover, the transformation is one way and the keys are secret. In case they are revealed, a new transformed template can be generated by using new keys.

ACKNOWLEDGMENT

This research is supported by PUPT Research Grant, Directorate of Higher Education, Ministry of Research, Technology and Higher Education.

REFERENCES

[1] T. Ahmad and J. Hu, "Generating Cancelable Biometric Templates Using a Projection Line," in *11th International Conference Control, Automation, Robotics and Vision (ICARCV)*, Singapore, 2010.

[2] S. H. Khan, M. A. Akbar, F. Shahzad, M. Farooq and Z. Khan, "Secure biometric template generation for multi-factor authentication," *Pattern Recognition,* vol. 48, no. 2015, pp. 458-472, 2014.

[3] S. Prabhakar, S. Pankanti and A. Jain, "Biometric Recognition: Security and Privacy Concerns," *IEEE SECURITY & PRIVACY,* vol. 1, no. 2, pp. 33-42, 2003.

[4] P. Gutierrez, M. Lastra, F. Herrera and J. Benitez, "A High Performance Fingerprint Matching System for Large Databases Based on GPU," *IEEE Transactions on Information Forensics and Security,* vol. 9, no. 1, pp. 62-71, 2014.

[5] A. K. Jain, K. Nandakumar and A. Nagar, "Biometric Template Security," *EURASIP Journal on Advances in Signal Processing,* vol. 8, no. 2, pp. 1-17, 2008.

[6] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *The 6th ACM Conference on Computer and Communications Security*, 1999.

[7] E. Maiorana and P. Campisi, "Fuzzy Commitment for Function Based Signature Template Protection," *IEEE SIGNAL PROCESSING LETTERS,* vol. 3, no. 249-252, p. 17, 2010.

[8] A. Juels and M. Sudan, "Designs, Codes and Cryptography," *A fuzzy vault scheme,* vol. 38, no. 2, p. 237–257, 2006.

[9] K. Xi and J. Hu, "Biometric Mobile Template Protection: A Composite Feature Based Fingerprint Fuzzy Vault," in *IEEE International Conference on Communications*, 2009.

[10] Y. Dodis, L. Reyzin and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," *Advances in Cryptology (EUROCRYPT'04), LNCS 3027,* p. 523–540, 2004.

[11] N. K. Ratha, S. Chikkerur, J. H. Connell and R. M. Bolle, "Generating cancelable fingerprint templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, no. 4, p. 561–572, 2007.

[12] T. Ahmad and F. Han, "Cartesian and Polar Transformation-based Cancelable Fingerprint Template," in *37th Annual Conference on IEEE Industrial Electronics Society - IECON 2011*, Melbourne, 2011.

[13] Z. Jin, M.-H. Lim, A. B. J. Teoh and B.-M. Goi, "A non-invertible Randomized Graph-based Hamming Embedding for generating cancelable fingerprint template," *Pattern Recognition Letters,* vol. 42, no. 2014, p. 137–147, 2014.

[14] S. Wang and J. Hu, "Design of alignment-free cancelable fingerprint templates via curtailed circular convolution," *Pattern Recognition,* vol. 47, no. 2014, p. 1321–1329, 2014.

[15] M. V. Prasad and C. S. Kumar, "Fingerprint template protection using multiline neighboring relation," *Expert Systems with Applications,* vol. 41, no. 2014, p. 6114–6122, 2014.

[16] [Online]. Available: http://webs.cs.berkeley.edu/tos/dist-1.1.0/snapshot-1.1.5Mar2004cvs/tools/matlab/contrib/kamin/localizationSimulation /testCases/. [Accessed October 2014].

[17] T. Ahmad, J. Hu and S. Wang, "Pair-polar coordinate-based cancelable fingerprint templates," *Pattern Recognition,* vol. 44, no. 2011, p. 2555–2564, 2011.

[18] FVC2002, "International Competition for Fingerprint Verification Algorithms," International Competition for Fingerprint Verification Algorithms, 2002. [Online]. Available: http://bias.csr.unibo.it/fvc2002/databases.asp.

[19] C. Wen and T. Guo, "An Efficient Algorithm for Fingerprint Matching Based on Convex Hulls," in *International Conference on Computational Intelligence and Natural Computing*, Wuhan, China, 2009.