Improving the Performance of Projection-based Cancelable Fingerprint Template Method

Tohari Ahmad
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, 60111, Indonesia
tohari@if.its.ac.id

Doni S. Pambudi
Department of Informatics
Universitas Internasional Semen Indonesia
Gresik, Indonesia
doni.pambudi@uisi.ac.id

Tsuyoshi Usagawa
HICC Laboratory
Kumamoto University
Japan

***Abstract*—Biometrics, especially fingerprint, has been popular to use for authenticating users because it is relatively permanence. This characteristic, however, is a problem because once it is compromised, fingerprint data can not be replaced. The conventional cryptographic algorithm may not be able to protect fingerprint data since the fingerprint scanning result is unstable. This paper improves the performance of the previous projection-based fingerprint protection method by removing the need of the core point and applying further minutiae checking hierarchically. The experimental result which is done in a public database produces the EER of about 1%.**

Keywords-fingerprint; security; minutiae; privacy; confidentiality

I. INTRODUCTION

In this digital era, the use of electronic devices has been a must. Consequently, an authentication process to prove that the one who would like to access is really a legitimate user is needed. There are some methods to do such authentication, including that with biometrics. Compared to other methods, such as a system with passwords, biometrics-based authentication has some advantages. For example, it is not easy to duplicate biometrics, which makes its data more secure. This requires the users to physically present when the authetication is being processed [1].

Amongst existing biometrics, fingerprint is one of popular modalities to use. As in [2], fingerprint has relatively good evaluation results. This is in terms of some key factors, such as acceptability, universality, performance and collectivity. In additon, fingerprint is relatively permanence, so it is suitable to use. Nevertheless, this last characteristic has made fingerprint data should be proteccted. This is because, once it is compromised, fingerprint data can not be changed; this is different with the password case.

There are some points where the fingerprint protection can be applied [3], such as at the sensor, extractor, matcher and database where the fingerprint template is stored. At this last point, the fingerprint data is transformed into another domain such that it is hard to recognize its original data. The transformation itself is one-way; therefore, the authentication is carried out in the transformed domain.

In order to do the transformation, it needs to extract the fingerprint to obtain some features, which can be the coordinate, orientation and type of the minutiae. By using these data along with the respective keys, the specified transformation is performed. The resulted transformed data is called cancelable template. It means that a different template can be generated by using different keys. This may occur if, for example, the template has been compromised. At the beginning, the method is introduced by Ratha et al [4]. In further development, some different methods have also been proposed [5] [6] [7] [8] with their own advantages and disadvantages.

In this paper, we propose a variation of cancelable fingerprint template method. This is done by exploring the minutiae characteristics, as an improvement of previous research, especially in [8]. The rest of the paper is structured as follows. Section 2 describes the previous works which relate to this proposed method. Section 3 explains the proposed method itself. The experimental results and conclusion are provided in Sections 4 and 5, respectively.

II. RELATED WORKS

The research in [8] uses the core point to be the center of a fingerprint area and the minutiae to be the neighboring points. These points are aligned to the position of the core point and its orientation as shown in Figs. 1(a) and 1(b), respectively.

The minutiae points are projected to a line $y = \rho x + c$, where $(x, y), \rho, c$ are abscissa and ordinate of a point in the line, the slope and the constant, respectively. Since this line crosses the coordinate (0,0), the value of $c = 0$. All minutiae points are projected into the line, parallel to the x and y axis (horizontal and vertical) as illustrated in Fig. 1(c). So, each minutia points results to two projected points. It is worth to note that it is easy to project a point to get two points but it is hard to reconstruct the original point just by using these two points due to many possibilities.

All points in the line are grouped based on their position as depicted in Fig. 1(d). In that example, there are 6 groups which contain 1, 2, 1, 2, 1 and 1 points. Each group is indexed in a specified order whose number of points constructs a set $v = \{2,1,1,2,1,1\}$, for example. This partitioning step is performed by employing a set of keys $K = \{\in, \mu, \vartheta\}$, where $\in$, μ, ϑ are the number of partitions, lenght of partitions and index of each partition, respectively. The resulted set is to be the template which is stored in the database. This set is

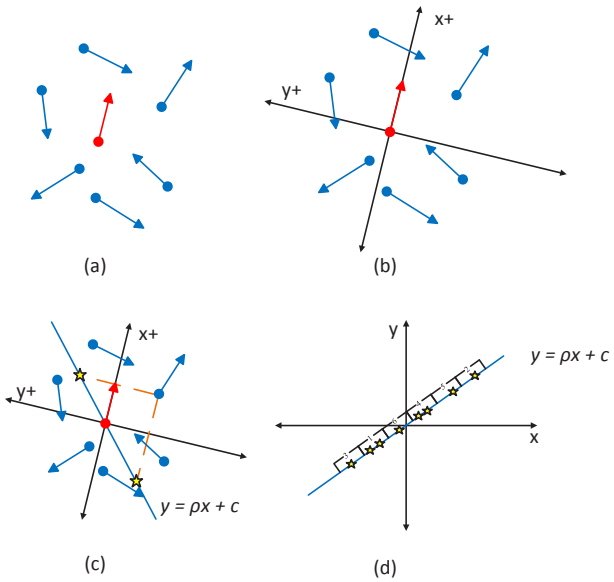

Figure 1 Points in fingerprint (a) Core and minutiae points (b) Axis in the fingerprint (c) Projected minutiae points (d) Partitioned line

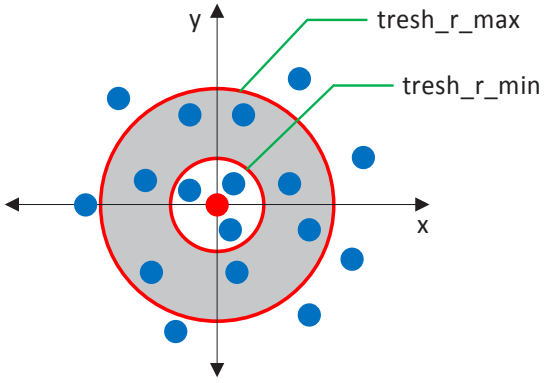

Figure 2 Minutiae selection process

compared to that of the query by using mean absolute error (MAE).

This method is simple and relatively easy to use. However, there are some disadvatages of this method. For example, it uses the core point to be the center of fingerprint. Without this core point, the set of template/query can not be generated. Furthermore, the informatioin of core point is difficult to obtain precisely; and, not all fingerprints have the core point [8]. Therefore, in this specific case, the fingerprint template can not be used.

In terms of the security, this method is relativley good since the information of the original fingerprint is not stored. Nevertheless, it is still possible to compromise it. The information of the generated template can be applied to the fake fingerprint by using v and K. Although it is not enough to reconstruct the original fingerprint just by using both data, they may be used to impersonate the authentication system. In the application, it is assumed that both v and K are confidential which may be secured by other methods.

In [9], Xi and Hu propose the hierarchical structure check (HSC) to protect fingerprint. This is carried out by comparing not only the minutia reference, but also the neighbors of neighbor points. So, there are some matching levels. In this research, different from [8], the fuzzy vault method is used for protecting the data. This method may have high accuracy, however, the processing time can be a problem.

III. Proposed Method

Different from [8], we develop a method which does not need the core point. Therefore, the existence of the core point in a fingerprint does not have an impact to the authentication process. As the result, the difficulty of extracting accurate coordinate and orientation of the core point can be ignored. Furthermore, all fingerprint types may be applicable to it. Overall, the method consists of some steps: minutiae selection, minutiae transformation, partition and matching. A minutia is randomly selected to be the reference. In turns, all minutiae are also the reference sequentially.

A. Minutiae Selection

Here, the template is only developed by some minutiae points. This is because too many minutiae may cause the authentication too long. However, too small number of minutiae may also significantly reduce the fingerprint uniqueness. Therefore, it needs to obtain an enough minutiae number which is able to represent the whole fingerprint data.

The selection is done by measuring the distance between the reference and all minutiae by using (1) which should meet the requirement in (2), where $dis(m_i, m_j)$ is the distance between minutiae m_i and m_j ; $thres\_r\_min$ and $thres\_r\_max$ are the allowed minimum and maximum distance, respectively. In this case, the minimum distance is specified in order to avoid too close projected points in the transformation process (see Section III.B). This process can be depicted in Fig. 2.

$$dis(m_i, m_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

$$thres\_r\_min \leq dis(m_i, m_j) \leq thres\_r\_max \quad (2)$$

B. Minutiae Transformation

As in our previous research [8], the minutiae are projected to the line which crosses the reference point (in this case is a minutia point). Different from it, we use a set of keys $K = \{d_{diff}, \propto_v, \propto_h, \propto_{mod}\}$. Firstly, the minutiae are projected into a line (see Fig. 3). The resulted vertical and horizontal projected points along with their orientation are transformed separately. Further transformation is applied to the orientation of the vertical projected points. Next, the respective point is translated d_{diff} according to the obtained orientation, as depicted in Fig. 4. The resulted point is projected back to the line and the distance between this point and the previous one is calculated by using (3). It is worth noting that this translation can be either positive or negative, according to the orientation. The same process is also applied to the horizontal projected point. These vertical and horizontal projected points construct a new point (represented by green circle in Fig. 5).

$$d'_{diff} = \cos(\theta_{trans}) * d_{diff} \quad (3)$$

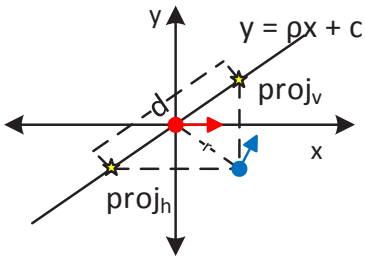

Figure 3 Projected minutiae

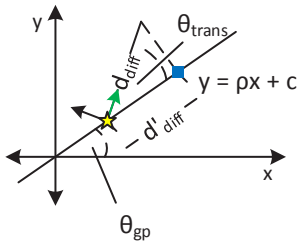

Figure 4 Translated projected minutiae

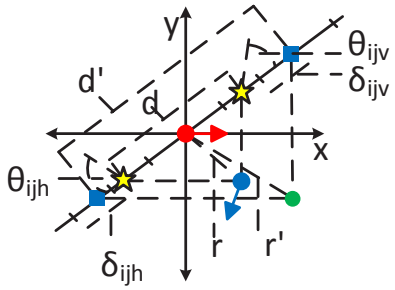

Figure 5 Features of projected minutiae

Let i and j be the reference and its neighbor point, respectively; r'_{ij} and d'_{ij} are the distance between (0,0) and the point constructed by vertical and horizontal projected points, and the distance between those vertical and horizontal projected points; θ_{ijv} and θ_{ijh} are the orientation of vertical and horizontal projected points; δ_{ijv} and δ_{ijh} are the index of the partition from where vertical and horizontal projected points are located (see Fig. 5). The generated template v of a fingerprint can be defined as in (4), where n is the number of neighbor points.

$$\{v\}_{i=1}^{n} = \begin{Bmatrix} (r'_{i1}, d'_{i1}, \theta_{i1v}, \theta_{i1h}, \delta_{i1v}, \delta_{i1h}) \\ \dots \\ (r'_{n1}, d'_{n1}, \theta_{n1v}, \theta_{n1h}, \delta_{n1v}, \delta_{n1h}) \end{Bmatrix} \quad (4)$$

C. Partition

Different from [8], the index of the partition is not randomized. This is because the order of partitions does not influence the accuracy. In addition, the number of partitions is fixed to 100 in order to cover all resulted points. Also, there is no threshold of how far those point from the center because we do not count the number of points in a partition. Instead, a partition is to estimate the position of those points.

The index of the partitions δ is given by using (5), where L_0 is the distance between a point and the center; and L_1 is the length of each partition, which is fixed to 20.

$$\delta = \begin{cases} 50 - \frac{L_0}{L_1} + 1, & x < 0 \\ 50 + \frac{L_0}{L_1}, & x \geq 0 \end{cases} \quad (5)$$

D. Matching

This process is carried out by comparing the set of vector v of template and query. In general, this comparison follows (6), where var is all component of vector in (4), and Δ_{var} is the difference of each of those components between query and template. So, we obtain $\Delta_r, \Delta_d, \Delta_{\theta ijv}, \Delta_{\theta ijh}, \Delta_{\delta v+h}$. Inspired by [9], these values are combined along with their respective threshold as depicted in (7), where f is the distance between template and query. Matching is performed to all vectors of template and query.

$$\Delta_{var} = \frac{abs(var_{template} - var_{query})}{var_{template}} * 100\% \quad (6)$$

$$f = \Delta r * Wr + \Delta d * Wd + \Delta\theta_{ijv} * W\theta_v + \Delta\theta_{ijh} * W\theta_h + \Delta\delta_{v+h} * W\delta \quad (7)$$

By employing the method in [9], the resulted values of f are selected. This means that the corresponding template and query are possibly matched. In addition, at least 40% of the minutiae template have to match to those of query with condition that 40% is more than 1 minutiae.

Fingerprint template and query are considered to be matched if there are at least 12 points match, as specified in [10]. In case the number of matched points is between 5 and 12, this conditional matched point is further checked by improving the capability of [9] to get the depth level D of the minutiae. Here, we propose the steps as follows:

1. Let T_1 and T_2 be the transformed template and query, respectively. Reset $D = 0$.
2. Take a minutia m_1 and m_2 from T_1 and T_2 with their respective v (see (4))
3. Take v_1 and v_2 from m_1 and m_2, respectively.
4. Compare v_1 and v_2 by applying (6). If the resulted value is less than the specified threshold, then they match and m_1 and m_2 are noted.
5. If step 4 does not meet, then repeat step 3 until all minutiae of m_1 are processed.
6. If step 4 meets, then take a neighbor point and go to step 2.

Once this process has been applied, we obtain the depth level of the minutiae. If it is higher than the specified threshold, then the minutiae are considered to be matched. An example of this process is provided in Fig. 6. Firstly, m_1 and m_2 of template and query are compared. If they matched, then depth level = 1 (see Fig. 6(a)), proceed to one of the neighbor (e.g., m_5) and compare it with others. If m_2 and m_5 of template and query match, then depth level = 2 (see Fig. 6(b)), proceed to one of neighbor of m_5, and soon. This process finishes when there is no more minutiae match.

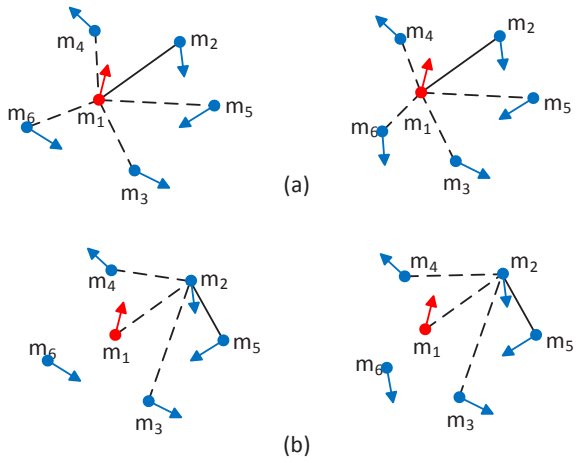

Figure 6 Depth level checking (a) Depth level=1 (b) Depth level=2

IV. Experimental Result

Similar to [1] [8] [9], we use FVC2002Db2a for testing which is extracted by using Verifinger [11]. As these research, we use 2 impression, each of which comprises 100 fingerprints. In order to evaluate the capability to accept a legitimate fingerprint, we compare those in the first impression to the corresponding fingerprint in the second impression; and for evaluating the capability to reject an illegitimate fingerprint, we compare those in the first impression to all fingerprints in the second impression except the corresponding one. Therefore, there are 10000 fingerprint comparison. Those two evaluation represent Genuine Acceptance Rate (GAR) and False Acceptance Rate (FAR), respectively.

A. Performance

Let n be the number of selected minutiae resulted from Section III(A). A minutia is represented by $n-1$ vectors; in turns, a fingerprint is represented by $n(n-1)$ vectors. The value of n depends on the minimum (*Rmin*) and maximum (*Rmax*) distance values being determined by the user. In the experiment, we specify the minimum distance by 11 as depicted in Table I. It shows the minimum (*Nmin*), maximum (*Nmax*) and average (*Navg*) numbers of selected minutiae. A high number of generated vectors may need more spaces to store in a database. The size of the template, nevertheless, is only kilo bytes. We believe that this size is not a significant problem for today technology.

The result of the experiment is provided in Table II. It is shown that the best result is obtained when the depth level is 7; that is, GAR and FAR are 98% and 0, respectively. Decreasing the depth level raises both GAR and FAR; while increasing depth level reduces GAR and FAR.

The Equal Error Rate (EER) of the same data can be plotted in Fig. 7. The False Rejection Rate (FRR) is defined as (100%-GAR), which represents the number of legitimate fingerprints that can not be recognized by the system. It is depicted that EER is 1% which is reached at about 1.2 of depth level. This is lower than that of [1][8] which is more than 2%;

TABLE 1 Number of Selected Minutiae

Rmin	Rmax	Neighbor minutiae		
		Nmin	Nmax	Navg
11	20	1	5	0.36
11	30	1	8	1.13
11	40	1	13	2.19
11	50	1	19	3.47
11	60	1	21	4.9
11	70	1	24	6.48
11	80	1	28	8.22
11	90	1	32	10.07
11	100	1	37	11.97
11	110	1	39	13.98
11	120	1	46	15.97
11	130	1	50	18.02
11	140	1	52	20

TABLE II GAR and FAR values of Various Depth Levels

Depth Level	GAR (%)	FAR(%)
1	99	6.6869
2	99	1.303
3	99	0.1919
4	98	0.0202
5	98	0.0101
6	98	0.0101
7	98	0
8	98	0
9	96	0
10	96	0
11	95	0

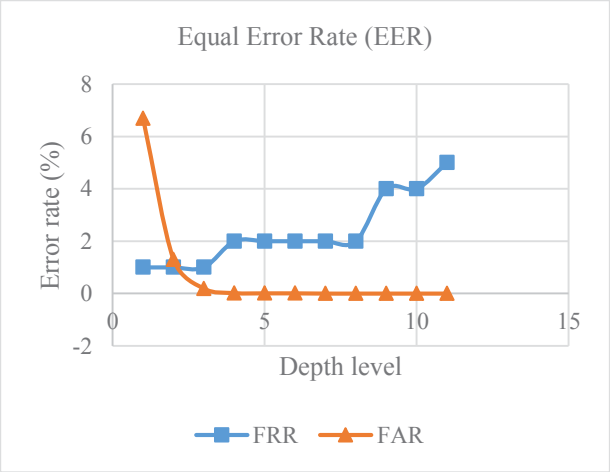

Figure 7 Equal error rate of the proposed method

while in [9], the EER is not provided. In the real world, the selection of depth level depends on the application purpose. It may focus on either a high GAR, low FAR or equal error level.

B. Security

The transformation is one way. Eventhough the set of keys is given, it is hard to reconstruct the original fingerprint. This is because there is no exact position of the resulted

transformed point. Instead, only the distance of the point is stored in the database. Moreover, not all minutiae are used in the transformation. So, their information is not in the template at all. In the worst case that the stored template can be revealed, not all fingerprint information can be obtained by attackers.

V. CONCLUSION

In this paper, the improvement of the projection-based fingerprint cancelable method is proposed. This method works by comparing the set of template and query vectors. In case the number of matched minutiae is higher than the specified threshold, then the fingerprint matched. Nevertheless, there is still a possibility to further evaluate the query whose value is less than the threshold by using the depth level.

This method, which is evaluated by using a public database, has been able to increase the performance, which is represented by GAR, FAR and EER. In terms of the security, the stored template is only generated by some minutiae. This has made the transfomed minutiae does not contain all fingerprint information.

REFERENCES

[1] T. Ahmad and F. Han, "Cartesian and polar transformation-based cancelable fingerprint template," in *The 37th Annual Conference of the IEEE Industrial Electronics Society*, Melbourne, Australia, 2011.

[2] D. Maltoni, D. Maio, A. Jain and S. Prabhakar, Handbook of Fingerprints, London: Springer, 2009.

[3] N. K. Ratha, J. H. Connell and R. M. Bolle, "An Analysis of Minutiae Matching Strength," in *Third International Conference on Audio- and Video-Based Biometric Person Authentication*, 2001.

[4] 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. 27, no. 4, pp. 561-572, 2007.

[5] 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.

[6] C. Lee and J. Kim, "Cancelable fingerprint templates using minutiae-based bit-strings," *Journal of Network and Computer Applications,* vol. 33, pp. 236-246, 2010.

[7] 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.

[8] T. Ahmad and J. Hu, "Generating cancelable biometric templates using a projection line," in *The 11th IEEE International Conference on Control Automation Robotics & Vision*, Singapore, 2010.

[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] S. Pankanti, S. Prabhakar and A. Jain, "On the Individuality of Fingerprints," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 8, pp. 1010-1025.

[11] Neurotechnology, "Verifinger 5.0," [Online]. Available: http://www.neurotechnology.com.