Fingerprint Combination for Privacy Protection

Sheng Li, *Student Member, IEEE*, and Alex C. Kot, *Fellow, IEEE*

***Abstract*—We propose here a novel system for protecting fingerprint privacy by combining two different fingerprints into a new identity. In the enrollment, two fingerprints are captured from two different fingers. We extract the minutiae positions from one fingerprint, the orientation from the other fingerprint, and the reference points from both fingerprints. Based on this extracted information and our proposed coding strategies, a combined minutiae template is generated and stored in a database. In the authentication, the system requires two query fingerprints from the same two fingers which are used in the enrollment. A two-stage fingerprint matching process is proposed for matching the two query fingerprints against a combined minutiae template. By storing the combined minutiae template, the complete minutiae feature of a single fingerprint will not be compromised when the database is stolen. Furthermore, because of the similarity in topology, it is difficult for the attacker to distinguish a combined minutiae template from the original minutiae templates. With the help of an existing fingerprint reconstruction approach, we are able to convert the combined minutiae template into a real-look alike combined fingerprint. Thus, a new virtual identity is created for the two different fingerprints, which can be matched using minutiae-based fingerprint matching algorithms. The experimental results show that our system can achieve a very low error rate with FRR = 0.4% at FAR = 0.1%. Compared with the state-of-the-art technique, our work has the advantage in creating a better new virtual identity when the two different fingerprints are randomly chosen.**

***Index Terms*—Combination, fingerprint, minutiae, privacy, protection.**

I. Introduction

With the widespread applications of fingerprint techniques in authentication systems, protecting the privacy of the fingerprint becomes an important issue. Traditional encryption is not sufficient for fingerprint privacy protection because decryption is required before the fingerprint matching, which exposes the fingerprint to the attacker. Therefore, in recent years, significant efforts have been put into developing specific protection techniques for fingerprint.

Most of the existing techniques make use of the key for the fingerprint privacy protection, which creates the inconvenience. They may also be vulnerable when both the key and the protected fingerprint are stolen. Teoh *et al.* [2] propose a biohashing approach by computing the inner products between the user's fingerprint features and a pseudorandom number (i.e., the key). The accuracy of this approach mainly depends on the key, which is assumed to be never stolen or shared [3]. Ratha *et al.* [4] propose to generate cancelable fingerprint templates by applying noninvertible transforms on the minutiae. The noninvertible transform is guided by a key, which will usually lead to a reduction in matching accuracy. The work in [2] and [4] are shown to be vulnerable to intrusion and linkage attacks when both the key and the transformed template are stolen [5]. Nandakumar *et al.* [6] propose to implement fuzzy fault on the minutiae, which is vulnerable to the key-inversion attack [7]. Our work in [8] imperceptibly hide the user identity on the thinned fingerprint using a key. The user identity may also be compromised when both the key and the protected thinned fingerprint are stolen.

There are only a few schemes [9]–[13] that are able to protect the privacy of the fingerprint without using a key. Ross and Othman [9] propose to use visual cryptography for protecting the privacy of biometrics. The fingerprint image is decomposed by using a visual cryptography scheme to produce two noise-like images (termed as sheets) which are stored in two separate databases. During the authentication, the two sheets are overlaid to create a temporary fingerprint image for matching. The advantage of this system is that the identity of the biometrics is never exposed to the attacker in a single database. However, it requires two separate databases to work together, which is not practical in same applications.

The works in [10]–[12] combine two different fingerprints into a single new identity either in the feature level [10] or in the image level [11], [12]. In [10], the concept of combining two different fingerprints into a new identity is first proposed, where the new identity is created by combining the minutiae positions extracted from the two fingerprints. The original minutiae positions of each fingerprint can be protected in the new identity. However, it is easy for the attacker to identify such a new identity because it contains many more minutiae positions than that of an original fingerprint. The experiment shows that the EER of matching the new identities is 2.1% when the original minutiae positions are marked manually from the original fingerprints. A similar scheme is proposed in [13], where the minutiae positions extracted from a fingerprint and the artificial points generated from the voice are combined to produce a new identity. In this work, the EER are shown to be under 2% according to the experimental results.

In [11], [12], the authors first propose to combine two different fingerprints in the image level. First of all, each fingerprint is decomposed into the continuous component and the spiral component based on the fingerprint FM-AM model [14]. After some alignment, the continuous component of one fingerprint is combined with the spiral component of the other fingerprint, so as to create a new virtual identity which is termed as a mixed fingerprint. Compared with the work in [10], [13], such

Manuscript received March 16, 2012; revised November 08, 2012; accepted November 26, 2012. Date of publication December 20, 2012; date of current version January 09, 2013. A preliminary version of this paper appeared in the 7th International Conference on Information Assurance and Security, 2011 (IAS2011). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Fabio Scotti.

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 639798 (e-mail: lish0016@ntu.edu.sg; eackot@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2012.2234740

an image level based fingerprint combination technique has two advantages: (i) it is difficult for the attacker to distinguish a mixed fingerprint from the original fingerprints, and (ii) existing fingerprint matching algorithms are applicable for matching two mixed fingerprints. However, this approach produces a visually unrealistic mixed fingerprint due to the variations in the orientation and frequency between the two different fingerprints. Their experimental results [12] show that the EER of matching two mixed fingerprints is about 15% when two different fingerprints are randomly chosen for creating a mixed fingerprint. If the two different fingerprints are carefully chosen according to a compatibility measure, the EER can be reduced to about 4%.

In this paper, we propose a novel system for protecting fingerprint privacy by combining two different fingerprints into a new identity. During the enrollment, the system captures two fingerprints from two different fingers. We propose a combined minutiae template generation algorithm to create a combined minutiae template from the two fingerprints. In such a template, the minutiae positions are extracted from one fingerprint, while the minutiae directions depend on the orientation of the other fingerprint and some coding strategies. The template will be stored in a database for the authentication which requires two query fingerprints. A two-stage fingerprint matching process is further proposed for matching the two query fingerprints against a combined minutiae template. By using the combined minutiae template, the complete minutiae feature of a single fingerprint will not be compromised when the database is stolen. In addition, the combined minutiae template share a similar topology to the original minutiae templates, it can be converted into a real-look alike combined fingerprint by using an existing fingerprint reconstruction approach [15]. The combined fingerprint issues a new virtual identity for two different fingerprints, which can be matched using minutiae based fingerprint matching algorithms.

The advantages of our technique over the existing fingerprint combination techniques [10]–[13] are as follows:

1) Our proposed system is able to achieve a very low error rate with $\mathrm{FRR} = 0.4\%$ when $\mathrm{FAR} = 0.1\%$.
2) Compared with the feature level based technique [10], [13], we are able to create a new identity (i.e., the combined minutiae template) which is difficult to be distinguished from the original minutiae templates.
3) Compared with the image level based technique [11], [12], we are able to create a new virtual identity (i.e., the combined fingerprint) which performs better when the two different fingerprints are randomly chosen.

The organization of the paper is as follows. Section II introduces our proposed fingerprint privacy protection system. Section III explains how to generate a combined fingerprint for two different fingerprints. Section IV presents the experimental results. Section V analyzes the information leakage in a combined minutiae template, followed by the conclusions in the last section.

II. The Proposed Fingerprint Privacy Protection System

Fig. 1 shows our proposed fingerprint privacy protection system. In the enrollment phase, the system captures two fingerprints from two different fingers, say fingerprints A and B from fingers A and B, respectively. We extract the minutiae positions from fingerprint A and the orientation from fingerprint B using some existing techniques [16], [17]. Then, by using our proposed coding strategies, a combined minutiae template is generated based on the minutiae positions, the orientation and the reference points detected from both fingerprints. Finally, the combined minutiae template is stored in a database. In the authentication phase, two query fingerprints are required from the same two fingers, say fingerprints A' and B' from fingers A and B. As what we have done in the enrollment, we extract the minutiae positions from fingerprint A' and the orientation from fingerprint B'. Reference points are detected from both query fingerprints. These extracted information will be matched against the corresponding template stored in the database by using a two-stage fingerprint matching. The authentication will be successful if the matching score is over a predefined threshold.

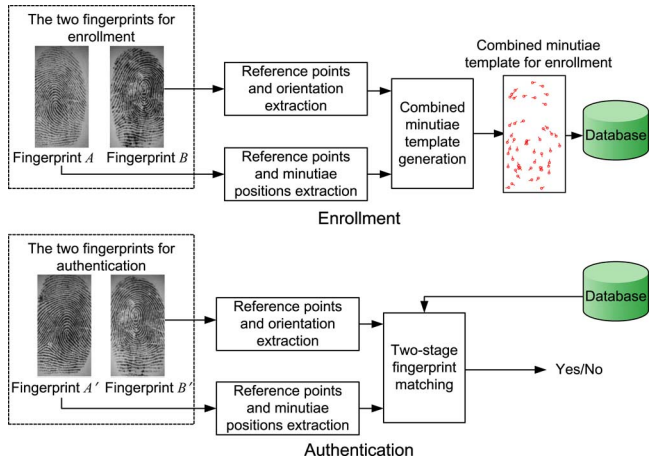


Fig. 1. Proposed fingerprint privacy protection system.

A. Reference Points Detection

The reference points detection process is motivated by Nilsson *et al.* [18], who first propose to use complex filters for singular point detection. Given a fingerprint, the main steps of the reference points detection are summarized as follows:

1) Compute the orientation O from the fingerprint using the orientation estimation algorithm proposed in [17]. Obtain the orientation Z in complex domain, where

$$Z = \cos(2O) + j\sin(2O). \tag{1}$$

2) Calculate a certainty map of reference points [18]

$$C_{ref} = Z * \bar{T}_{ref} \tag{2}$$

where "$*$" is the convolution operator and $\bar{T}_{ref}$ is the conjugate of

$$T_{ref} = (x + iy) \cdot \frac{1}{2\pi\sigma^2} \cdot \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{3}$$

which is the kernel for reference points detection.

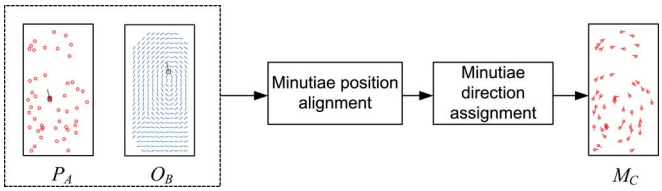


Fig. 2. Combined minutiae template generation process.

3) Calculate an improved certainty map [19]:

$$C'_{ref} = \begin{cases} C_{ref} \cdot \sin(Arg(C_{ref})) & \text{if } \mathrm{Arg}(C_{ref}) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $Arg(z)$ returns the principal value of the argument of z (defined from $-\pi$ to π).

4) Locate a reference point satisfying the two criterions: (i) the amplitude of C'_{ref} of the point (hereinafter termed as the certainty value for simplicity) is a local maximum, and (ii) the local maximum should be over a fixed threshold T. Suppose we locate a reference point at (r_x, r_y), the corresponding angle can be estimated as $Arg(C'_{ref}(r_x, r_y))$.
5) Repeat step 4) until all reference points are located.
6) If no reference point is found for the fingerprint in steps 4) and 5) (e.g., an arch fingerprint), locate a reference point with the maximum certainty value in the whole fingerprint image.

B. Combined Minutiae Template Generation

Given a set of N minutiae positions $P_A = \{\mathbf{p}_{ia} = (x_{ia}, y_{ia}), 1 \le i \le N\}$ of fingerprint A, the orientation O_B of fingerprint B and the reference points of fingerprints A and B, a combined minutiae template M_C is generated by minutiae position alignment and minutiae direction assignment, as shown in Fig. 2.

1) Minutiae Position Alignment: Among all the reference points of a fingerprint for enrollment, we define a reference point with the maximum certainty value as the primary reference point. Therefore, we have two primary reference points R_a and R_b for fingerprints A and B, respectively. Let's assume R_a is located at $\mathbf{r}_a = (r_{xa}, r_{ya})$ with the angle β_a, and R_b is located at $\mathbf{r}_b = (r_{xb}, r_{yb})$ with the angle β_b. The alignment is performed by translating and rotating each minutiae point $\mathbf{p}_{ia}$ to $\mathbf{p}_{ic} = (x_{ic}, y_{ic})$ by

$$(\mathbf{p}_{ic})^T = \mathbf{H} \cdot (\mathbf{p}_{ia} - \mathbf{r}_a)^T + (\mathbf{r}_b)^T \quad (5)$$

where $()^T$ is the transpose operator and $\mathbf{H}$ is the rotation matrix where

$$\mathbf{H} = \begin{bmatrix} \cos(\beta_b - \beta_a), \sin(\beta_b - \beta_a) \\ -\sin(\beta_b - \beta_a), \cos(\beta_b - \beta_a) \end{bmatrix}. \quad (6)$$

As such, R_a and R_b are overlapped both in the position and the angle after the minutiae position alignment.

2) Minutiae Direction Assignment: Each aligned minutiae position $\mathbf{p}_{ic}$ is assigned with a direction θ_{ic} as follows:

$$\theta_{ic} = O_B(x_{ic}, y_{ic}) + \rho_i \pi \quad (7)$$

where ρ_i is an integer that is either 0 or 1. The range of $O_B(x_{ic}, y_{ic})$ is from 0 to π. Therefore, the range of θ_{ic} will be from 0 to 2π, which is the same as that of the minutiae directions from an original fingerprint. Following three coding strategies are proposed for determining the value of ρ_i.

1) ρ_i is randomly selected from $\{0, 1\}$.
2) ρ_i is determined by

$$\rho_i = \begin{cases} 1 & \text{if } \mathrm{mod}(\theta_{ia} + \beta_b - \beta_a, \pi) - O_B(x_{ic}, y_{ic}) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where mod is the modulo operator and θ_{ia} is the original direction of a minutiae position $\mathbf{p}_{ia}$ in fingerprint A.

3) ρ_i is determined by

$$\rho_i = \begin{cases} 1 & \text{if } \mathrm{mod}(ave_b(x_{ic}, y_{ic}), \pi) - O_B(x_{ic}, y_{ic}) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where $ave_b(x_{ic}, y_{ic})$ is the average direction of the n nearest neighboring minutiae points of the location (x_{ic}, y_{ic}) in fingerprint B

$$ave_b(x_{ic}, y_{ic}) = \frac{1}{n} \sum_{k=1}^{n} \theta_b^k(x_{ic}, y_{ic}) \quad (10)$$

where $\theta_b^k(x_{ic}, y_{ic})$ means the direction of the kth nearest neighboring minutiae point of the location (x_{ic}, y_{ic}) in fingerprint B, and n is empirically set as 5 which is able to provide a good balance between the diversity and matching accuracy of the combined minutiae templates.

In the following discussions, these three coding strategies are termed as *Coding Strategy* 1, *Coding Strategy* 2 and *Coding Strategy* 3, respectively. Note that some additional information is required in *Coding Strategy* 2 and *Coding Strategy* 3, where the block diagram of our system shown in Fig. 1 should be modified accordingly.

Sometimes, $\mathbf{p}_{ic}$ may be located outside the area of fingerprint B, where $O_B(x_{ic}, y_{ic})$ is not well defined. In such a case, we need to predict $O_B(x_{ic}, y_{ic})$ before the direction assignment. Some existing works for modeling the fingerprint orientation can be adopted to do the prediction. For example, the work in [20] can estimate the missing orientation structure even for a partial fingerprint. Here, we simply predict the value of $O_B(x_{ic}, y_{ic})$ (if it is not well defined) as the value of nearest well defined orientation in O_B.

Once all the N aligned minutiae positions are assigned with directions, a combined minutiae template $M_C = \{\mathbf{m}_{ic} = (\mathbf{p}_{ic}, \theta_{ic}), 1 \le i \le N\}$ is created for enrollment. In some cases, a global minutiae position translation may be necessary for M_C such that all the minutiae points are located inside the fingerprint image.

C. Two-Stage Fingerprint Matching

Given the minutiae positions $P_{A'}$ of fingerprint A', the orientation $O_{B'}$ of fingerprint B' and the reference points of the two query fingerprints. In order to match the M_C stored in the database, we propose a two-stage fingerprint matching process including query minutiae determination and matching score calculation as shown in Fig. 3.

1) Query Minutiae Determination: The query minutiae determination is a very important step during the fingerprint matching. In order to simplify the description of our algorithm, we first introduce the local features extracted for a minutiae

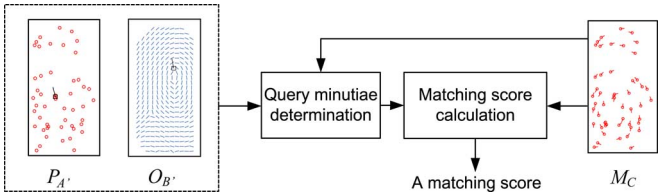


Fig. 3. Two-stage fingerprint matching process.

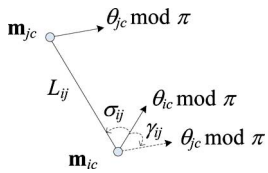


Fig. 4. Illustration of the definitions of L_{ij}, γ_{ij}, and σ_{ij}.

point in M_C. The local feature extraction is similar to the work proposed in [21]. Given a minutiae point $\mathbf{m}_{ic}$ and another minutiae point $\mathbf{m}_{jc}$ in M_C, we define

1) L_{ij} as the distance between $\mathbf{m}_{ic}$ and $\mathbf{m}_{jc}$:

$$L_{ij} = \sqrt{(x_{ic} - x_{jc})^2 + (y_{ic} - y_{jc})^2} \tag{11}$$

2) γ_{ij} as the difference between the directions (after modulo π) of $\mathbf{m}_{ic}$ and $\mathbf{m}_{jc}$:

$$\gamma_{ij} = \theta_{ic} \bmod \pi - \theta_{jc} \bmod \pi \tag{12}$$

3) σ_{ij} as a radial angle:

$$\sigma_{ij} = \Re\left(\theta_{ic} \bmod \pi, \operatorname{atan2}\left(y_{jc} - y_{ic}, x_{jc} - x_{ic}\right)\right) \tag{13}$$

where $\operatorname{atan2}(y, x)$ is a two-argument arctangent function in the range $(-\pi, \pi]$ and

$$\Re(\mu_1, \mu_2) = \begin{cases} \mu_1 - \mu_2 & \text{if } -\pi < \mu_1 - \mu_2 \le \pi \\ \mu_1 - \mu_2 + 2\pi & \text{if } \mu_1 - \mu_2 \le -\pi \\ \mu_2 - \mu_1 + 2\pi & \text{if } \mu_1 - \mu_2 > \pi. \end{cases} \tag{14}$$

An illustration of the definitions of L_{ij}, γ_{ij}, and σ_{ij} are shown in Fig. 4. For the ith minutiae point $\mathbf{m}_{ic}$ in M_C, we extract a set of local features $\mathbf{F}_i$ as follows:

$$\mathbf{F}_i = (L_{ij}, L_{ik}, L_{il}, \gamma_{ij}, \gamma_{ik}, \gamma_{il}, \sigma_{ij}, \sigma_{ik}, \sigma_{il}) \tag{15}$$

where we assume $\mathbf{m}_{jc}$ is the nearest, $\mathbf{m}_{kc}$ is the second nearest and $\mathbf{m}_{lc}$ is the third nearest minutiae point of $\mathbf{m}_{ic}$.

Suppose we detect k_1 ($k_1 \ge 1$) reference points from fingerprint A' and k_2 ($k_2 \ge 1$) reference points from fingerprint B'. The query minutiae is determined as follows:

1) Select a pair of reference points: one from fingerprint A' (say $R_{a'}$) and the other from fingerprint B' (say $R_{b'}$). Assume $R_{a'}$ is located at $\mathbf{r}_{a'} = (r_{xa'}, r_{ya'})$ with the angle $\beta_{a'}$, $R_{b'}$ is located at $\mathbf{r}_{b'} = (r_{xb'}, r_{yb'})$ with the angle $\beta_{b'}$, respectively.
2) Perturb $\beta_{a'}$ by $\tau = \beta_{a'} + \kappa \cdot \Delta$, where κ is an integer and Δ is a perturbation size. We choose $\Delta = 3 \times \pi/180$ radians (i.e., 3 degrees) and $-5 \le \kappa \le 5$. Thus, we have $K = 11$ perturbed angles for the reference point $R_{a'}$.
3) Generate a combined minutiae template $M_{C'}(\tau)$ for testing (hereinafter simply termed as a testing minutiae) from $P_{A'}$, $O_{B'}$, $R_{a'}$ (with a perturbed angle τ) and $R_{b'}$ using the proposed combined minutiae template generation algorithm. Note that the same coding strategy should be adopted for generating $M_{C'}(\tau)$ and M_C. In total, we generate K testing minutiae $M_{C'}(\tau)$.
4) Suppose $\mathbf{F}_u$ are the local features extracted for the uth minutiae point in $M_{C'}(\tau)$, while $\mathbf{F}_v$ are the local features extracted for the vth minutiae point in M_C. Calculate the difference between $\mathbf{F}_u$ and $\mathbf{F}_v$ by

$$D_\tau(u, v) = w_1 \cdot \sum_{j=1}^{3} |\mathbf{F}_u(j) - \mathbf{F}_v(j)| + w_2 \cdot \sum_{j=4}^{9} |\mathbf{F}_u(j) - \mathbf{F}_v(j)| \tag{16}$$

where $\mathbf{F}_i(j)$ refers to the jth component of $\mathbf{F}_i$, w_1 and w_2 are the weights for different features. We follow the same weight settings as in [21], where w_1 and w_2 are empirically set as $w_1 = 1$ and $w_2 = 0.3 \times 180/\pi$. Then, we define the difference between $M_{C'}(\tau)$ and M_C as

$$d_\tau = \min_{u,v} D_\tau(u, v). \tag{17}$$

5) Repeat steps 1) to 4) until all the possible pairs (in total $k_1 \times k_2$ pairs) of reference points are selected and processed. Among all the testing minutiae ($K \times k_1 \times k_2$ in total), the one which has the minimum difference from M_C (i.e., the minimum d_τ) will be considered as the query minutiae M_Q.

2) Matching Score Calculation: For the combined minutiae templates that are generated using *Coding Strategy* 1, we do a modulo π for all the minutiae directions in M_Q and M_C, so as to remove the randomness. After the modulo operation, we use an existing minutiae matching algorithm [16] to calculate a matching score between M_Q and M_C for the authentication decision. For other combined minutiae templates, we directly calculate a matching score between M_Q and M_C using an existing minutiae matching algorithm [16].

III. Combined Fingerprint Generation

In a combined minutiae template, the minutiae positions and directions (after modulo π) are extracted from two different fingerprints separately. These minutiae positions and directions share a similar topology to those from an original fingerprint. Therefore, the combined minutiae template has a similar topology to an original minutiae template. Some existing works [15], [22], [23] have shown that it is possible to reconstruct a full fingerprint image from a minutiae template. By adopting one of these fingerprint reconstruction approaches, we are able to convert our combined minutiae template into a combined fingerprint image. Fig. 5 shows our process to generate a combined fingerprint for two different fingerprints. Given any two different fingerprints as input, we first generate a combined minutiae template using our combined minutiae template generation algorithm. Then, a combined fingerprint is reconstructed from the combined minutiae template using one of the existing fingerprint reconstruction approaches.

It should be noted that the combined minutiae template generated by adopting *Coding Strategy* 1 is not appropriate for gen-

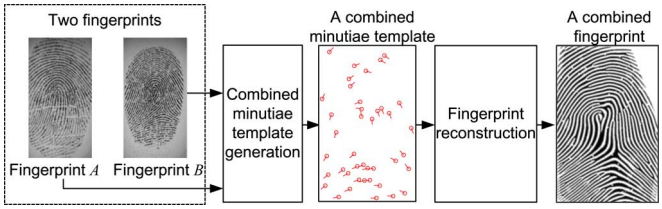


Fig. 5. Generating a combined fingerprint for two different fingerprints.

erating a combined fingerprint. The reason is that we set ρ_i as 0 or 1 randomly during the minutiae direction assignment, i.e., we add π randomly for each minutiae direction in such a coding strategy. As what has been discussed in Section II-C2, we need to perform a modulo π operation for the minutiae directions during the fingerprint matching, so as to remove such randomness. Therefore, we will not be able to match the corresponding combined fingerprint by using a general fingerprint matching algorithm. While the purpose of generating a combined fingerprint is to issue a new virtual identity for two different fingerprints, which should be matched using general fingerprint matching algorithms.

Among the existing fingerprint reconstruction approaches, our previous work [15] achieves excellent performance. We here adopt this approach for generating a combined fingerprint from a combined minutiae template. However, the work in [15] does not incorporate a noising and rendering step to make the reconstructed fingerprint image real-look alike. To create a real-look alike fingerprint image from a set of minutiae points, we further apply a noising and rendering step after adopting the work in [15], where the following 7 stages are carried through as illustrated in Fig. 6.

1) Estimate an orientation field O from the set of minutiae points by adopting the orientation reconstruction algorithm proposed in [23].
2) Generate a binary ridge pattern based on O and a predefined fingerprint ridge frequency (which is set as 0.12) using Gabor filtering.
3) Estimate the phase image ψ of the binary ridge pattern using the fingerprint FM-AM model [14].
4) Reconstruct the continuous phase image ψ_c by removing the spirals in the phase image ψ.
5) Combine the continuous phase image ψ_c and the spiral phase image ψ_s (calculated from the minutiae points), producing a reconstructed phase image ψ_f.
6) Refine the reconstructed phase image ψ_f by removing the spurious minutiae points to produce a refined phase image ψ_{fr}.
7) Apply a noising and rendering step (which is similar to the work proposed in [24]) on ψ_{fr}, so as to create a real-look alike fingerprint image.

IV. Experimental Results

The experiment is conducted on the first two impressions of the FVC2002 DB2_A database, which contains 200 fingerprints from 100 fingers (with 2 impressions per finger). The VeriFinger 6.3 [16] is used for the minutiae positions extraction and the minutiae matching. The algorithm proposed in [17] is used for the orientation extraction.

A. Parameter Settings for Reference Points Detection

The reference points detection has a significant impact on the accuracy and efficiency of our proposed system. There are two parameters need to be determined for the reference points detection, i.e., σ for the complex filtering and T which is the threshold for the reference points detection. We set $\sigma = 1.5$ as suggested in [18]. Next, we explain in detail for the setting of T. A good setting of T should meet the following two requirements for the accuracy and efficiency of our system: (i) the detected reference points should contain the true singular point which is a loop of the fingerprint and (ii) the number of the detected reference points should be small. We manually mark the location of the topmost loop (with the angle pointing upwards) for each of the first two impressions of the FVC2002 DB2_A database (in total 200 fingerprints). Note that if the fingerprint is an arch, the topmost loop is marked at the point with the highest ridge curvature. For each fingerprint, we define the reference point nearest to the marked topmost loop as the nearest reference point for simplicity. The reference points are considered to be truly detected if the Euclidian distance between the marked topmost loop and the nearest reference point is less than 30 pixels as suggested in [19]. Otherwise, the reference points are considered to be falsely detected. Table I shows the performance of the reference points detection at different settings of threshold T, where "No." refers to the total number of reference points detected among the 200 fingerprints. It can be seen that setting $T = 5$ will achieve a good balance between the accuracy and efficiency. By setting $\sigma = 1.5$ and $T = 5$, the average Euclidian distance between the marked topmost loop and the nearest reference point is 5.65 pixels. Furthermore, by adopting such settings, we will only detect one reference point for the majority of fingerprints, which is the loop as shown in Figs. 7(a) and 7(b). Figs. 7(c) and 7(d) illustrate an example with two reference points detected.

The reliability of the angle of the nearest reference point (by setting $\sigma = 1.5$ and $T = 5$) for each of the 200 fingerprints is measured as follows similar to the work in [18].

1) Rotate the original fingerprint from -30 degrees to 30 degrees with 2 degrees per step based on the topmost loop marked before.
2) Perform the reference points detection for the original fingerprint and its rotated versions. Let's denote the angle of the nearest reference point of the original fingerprint as α_0, and the corresponding angle for each of the rotated versions as α_q, where $q = -30, -28, \ldots -2, 2, 4 \ldots 30$ refers to the degree of rotation.
3) Estimate the degree of rotation for each rotated version by

$$\bar{q} = \alpha_q - \alpha_0. \tag{18}$$

4) Compute the absolute estimation error for each rotated version, i.e.,

$$e = |\bar{q} - q|. \tag{19}$$

5) Compute the mean and standard deviation of e for all the rotated versions, which are denoted as e_{mean} and e_{std}, respectively.

Among all the 200 fingerprints, the average e_{mean} and e_{std} are 1.96 degrees and 2.05 degrees, respectively.

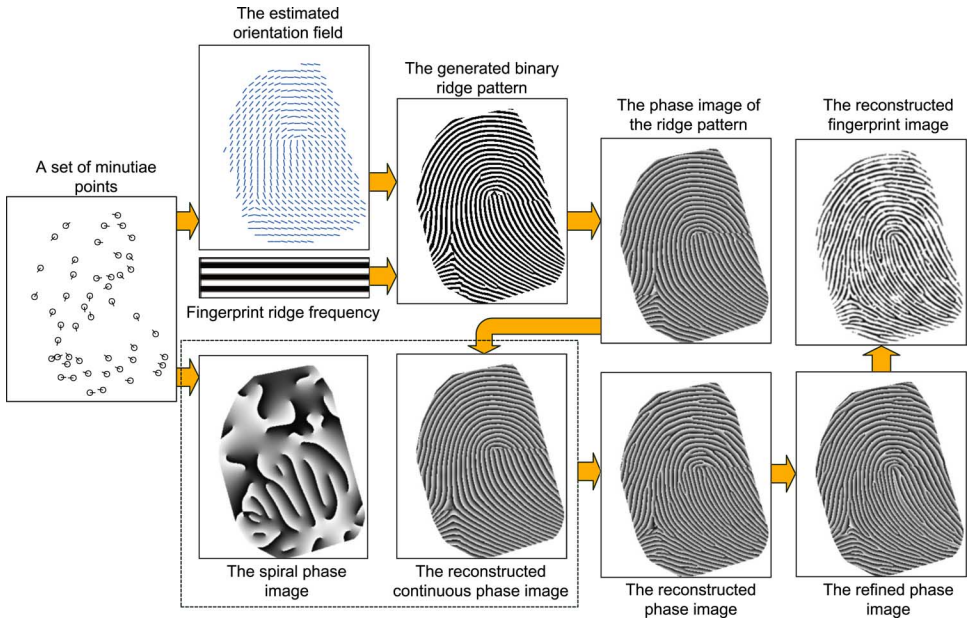


Fig. 6. Reconstructing a real-look alike fingerprint image from a set of minutiae points.

Fig. 7. Illustrations of the reference points detection. Fingerprint with only one reference point in (a) and the corresponding certainty value map in (b); fingerprint with two reference points in (c) and the corresponding certainty value map in (d).

TABLE I
PERFORMANCE OF THE REFERENCE POINTS DETECTION AT DIFFERENT SETTINGS OF THRESHOLD T

	T			
	3	4	5	6
No.	1141	650	291	207
True Detection Rate (%)	99.5	99.5	99.5	98.5
False Detection Rate (%)	0.5	0.5	0.5	1.5

B. Evaluating the Performance of the Proposed System

In order to evaluate the performance of our system, we randomly pair the 100 fingers in the FVC2002 DB2_A database to produce a group of 50 nonoverlapped finger pairs, where each finger pair contains two different fingers. The random pairing process is repeated 10 times to have 10 groups of 50 nonoverlapped finger pairs.

For the two fingerprints captured from two different fingers, we can generate two combined minutiae templates in total, where one fingerprint serves as fingerprint A, the other serves as fingerprint B or vice versa. The system designer can choose to enroll one or both of the two templates in the database, which depends on the applications. Thus, we consider the following two cases in building the system database for each group of finger pairs:

1) The first impressions of each finger pair are used to produce only one combined minutiae template for enrollment. Therefore, there are 50 templates stored in the database. To compute the False Rejection Rate (FRR), the second impressions of a finger pair are matched against the corresponding enrolled template, producing 50 genuine tests. To compute the False Acceptance Rate (FAR), the first impressions of a finger pair are matched against the other 49 enrolled templates, producing $50 \times 49 = 2450$ imposter tests.
2) The first impressions of each finger pair are used to produce two combined minutiae templates for enrollment. Thus, there are 100 templates stored in the database. Similarly, 100 genuine tests are performed to compute FRR and $100 \times 99 = 9900$ imposter tests are performed to compute FAR.

In the following discussions, the above two cases are termed as Case I and Case II, respectively.

Fig. 8 plots the average FRR (at different FAR) computed from the 10 groups of finger pairs for the two cases. We can see that our system performs similarly for the two cases. However, the error rates vary among different coding strategies, where the *Coding Strategy* 1 achieves the lowest error rate with $\mathrm{FRR} = 0.4\%$ (at $\mathrm{FAR} = 0.1\%$) for both cases. While the results of using *Coding Strategy* 3 are the worst, with over 1% FRR (at $\mathrm{FAR} = 0.1\%$) for both cases.

In order to show the effectiveness of the proposed two two-stage fingerprint matching, we evaluate the performance of our system by using a conventional minutiae matching technique [16] for the fingerprint matching. That is to say, during the authentication, we generate a combined minutiae template from two query fingerprints, which is then matched against the corresponding enrolled template by using a conventional minutiae matching algorithm [16]. Under such an assumption, the performance of our system for Case II is shown in Fig. 9. Note that the combined minutiae templates generated using *Coding Strategy* 1 can not be matched directly using a conventional minutiae

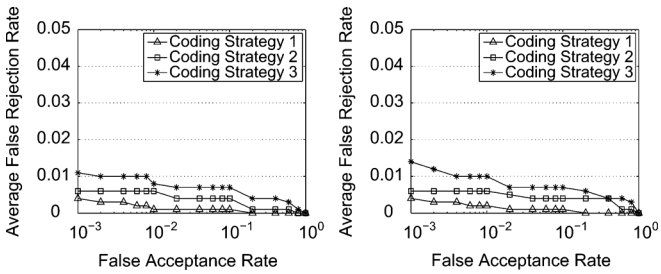


Fig. 8. Performance of the proposed system for (a) Case I, and (b) Case II.

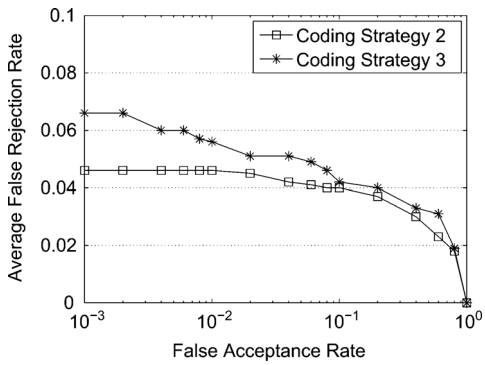


Fig. 9. Performance of the proposed system for Case II by using a conventional minutiae matching technique [16] for the fingerprint matching.

matching algorithm because of the randomness in the minutiae direction. It can be seen that the error rates significantly increase. Compared with the results shown in Fig. 8(b), there are 4.0% and 5.2% increase in FRR at $\mathrm{FAR} = 0.1\%$ for *Coding Strategy* 2 and 3, respectively.

Compared with a traditional fingerprint recognition system (hereinafter referred to as a traditional system for simplicity), our proposed system offers more choices for a single user to do the enrollment and authentication. A traditional system can only enroll 10 fingerprint templates for the ten fingers of an user. While our system is able to enroll $10 \times 9 = 90$ combined minutiae templates. Among these 90 combined minutiae templates, many share the same minutiae positions or orientations, which could be easily linked. However, they produce the diversity of the choices of the fingerprints (for the user) like passwords.

Next, we examine the difference among all the combined minutiae templates that can be created for a set of 10 fingers based on our proposed system. That is to say, we evaluate the performance of our system when the database stores all the combined minutiae templates generated for 10 fingers. We randomly separate the 100 fingers (in FVC2002 DB2_A) into 10 groups with 10 fingers per group. For each group, there are in total $\binom{10}{2} = 45$ possible finger pairs. The first impressions of each finger pair are used to produce two combined minutiae templates for enrollment. The corresponding second impressions serve as the testing fingerprints. As such, 90 combined minutiae templates are generated and stored in the system database. There are 90 genuine tests for computing FRR and $90 \times 89 = 8010$ imposter tests for computing the FAR for each group, where the average FRR for the ten groups (with 10 fingers per group) is shown in Fig. 10. We can see that the error rates of our system

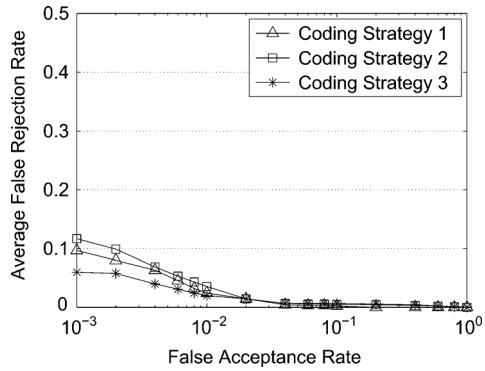


Fig. 10. Evaluating the difference among all the possible combined minutiae templates that can be generated for a set of 10 fingers based on the proposed system.

increase a lot because some templates either share the same minutiae positions or the same orientation. Among the different coding strategies, the *Coding Strategy* 3 achieves the lowest error rates with $\mathrm{FRR} = 6\%$ at $\mathrm{FAR} = 0.1\%$. While the corresponding FRR of using *Coding Strategy* 1 and *Coding Strategy* 2 are 9.67% and 11.67%, respectively.

C. Evaluating the Performance of the Combined Fingerprints

In this section, we compare the performance of our combined fingerprints with the mixed fingerprints generated by the proposed technique in [12] (hereinafter referred to as the mixed fingerprints for simplicity). The VeriFinger 6.3 [16] is adopted for matching two combined fingerprints or two mixed fingerprints. We use the same 10 groups of 50 nonoverlapped finger pairs that are randomly paired at the beginning of Section IV-B. For each group of finger pairs, we consider the same two cases for enrollment as in Section IV-B, i.e.,

1) The first impressions of each finger pair are used to produce only one combined fingerprint for enrollment. The corresponding second impressions are used to generate a query combined fingerprint. The query combined fingerprint will be matched against its counterpart enrolled in the database to compute the FRR, producing 50 genuine tests. The FAR is computed by matching an enrolled combined fingerprint against other 49 enrolled combined fingerprints, producing $50 \times 49/2 = 1225$ imposter tests, where the symmetric imposter tests are not executed.
2) The first impressions of each finger pair are used to produce two combined fingerprints for enrollment. The corresponding second impressions are used to generate two query combined fingerprints. Similarly, we have 100 genuine tests and $100 \times 99/2 = 4950$ imposter tests.

The above evaluation is also performed by using the mixed fingerprint approach [12] for comparison. Note that the work in [12] does not incorporate a noising and rending step to create the mixed fingerprints. Therefore, in order to do a fair comparison, all our combined fingerprints are created without noising and rendering.

Fig. 11 shows the performance comparison between the combined fingerprints and the mixed fingerprints. It can be seen that our combined fingerprint achieves a lower error rate than the mix fingerprint. Especially for Case II, our work performs much

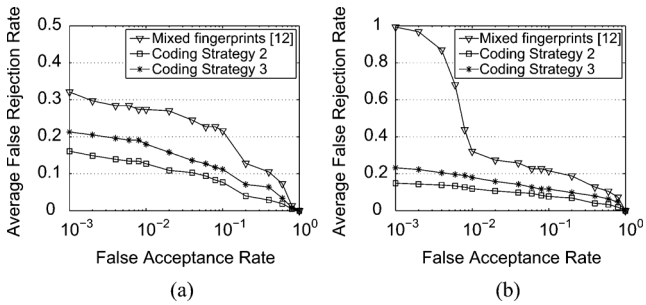


Fig. 11. Performance comparison between the combined fingerprints and the mixed fingerprints for (a) Case I, and (b) Case II.

better when the FAR is less than 1%. The combined fingerprints of *Coding Strategy* 2 perform the best, with FRR around 15% at FAR = 0.1% for the two cases. A visual comparison among different types of new identities is shown in Fig. 12.

The poor performance of the mixed fingerprints approach for Case II is due to the small overlapping area of some finger pairs. The two mixed fingerprints generated by such a finger pair will be quite similar, which may produce a very high imposter matching score. However, it should be noted that the performance of the mixed fingerprints approach can be much improved by incorporating a compatibility measure, which is used to determine whether two different fingerprints are suitable to generate a visually realistic mixed fingerprint (EER is around 4% according to the results reported in [12]). Our combined fingerprint scheme is applicable for minutiae based fingerprint matching algorithms. On a separate note, generating a combined fingerprint would cost more time than creating a mixed fingerprint because of the fingerprint reconstruction.

D. Evaluating the Probability to Attack Other Systems by Using the Combined Minutiae Templates

In case the combined minutiae templates are stolen, the attacker can use them to attack other traditional systems which store the original fingerprints. He can reconstruct a fingerprint image from a stolen combined minutiae template and make a fake finger based on the reconstructed fingerprint. By scanning the fake finger, the attacker may be able to break into other traditional systems. Similarly, if a combined fingerprint or a mixed fingerprint is stolen, the attacker can directly make a fake finger from the fingerprints and launch the attack.

In this section, we evaluate the successful rates to attack other traditional systems by using the combined minutiae templates. Assume that the attacker can do a perfect job to reconstruct a full fingerprint image from a combined minutiae template, i.e., the minutiae of the reconstructed fingerprint is exactly the same as the combined minutiae template. Under such an assumption, the successful attack rate by using a combined minutiae template should be higher than using the corresponding combined fingerprint because our current fingerprint reconstruction approach is not perfect.

Generally speaking, the attacker can launch the following two types of attacks based on a combined minutiae template:

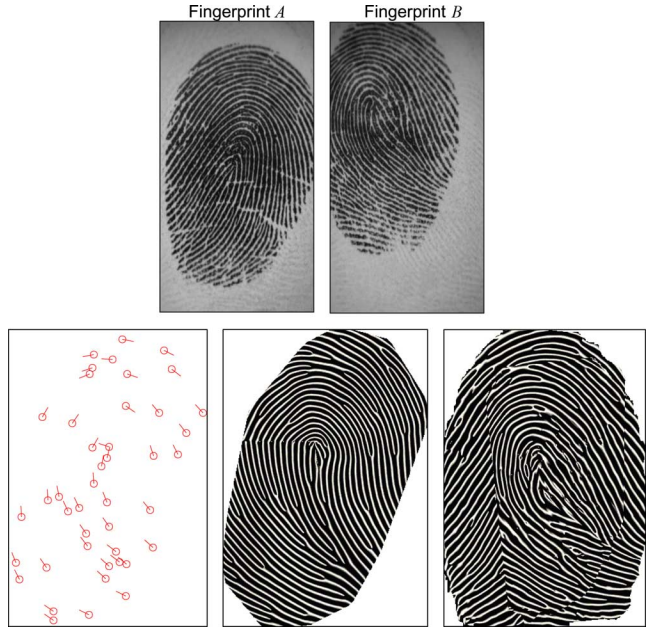


Fig. 12. Different types of new identities that are generated from two different fingerprints. The second row (from left to right): the combined minutiae template, the combined fingerprint from our proposed method (without noising and rendering), and the mixed fingerprint obtained using the approach proposed in [12].

1) The combined minutiae template is used to attack the system which stores the corresponding fingerprint A (mainly provides the minutiae positions).
2) The combined minutiae template is used to attack the system which stores the corresponding fingerprint B (mainly provides the minutiae directions).

For simplicity, the above two types of attacks are termed as *Attack Type* A and *Attack Type* B, respectively. Suppose the 10 databases built for Case II in Section IV-B are stolen, where each database contains 100 combined minutiae templates. To evaluate the successful rates of the two types of attacks, each stolen template is matched against the corresponding fingerprint A and fingerprint B using the VeriFinger 6.3 [16], respectively. Thus, we have 1000 matches for *Attack Type* A and 1000 matches for *Attack Type* B. Again, in order to compare with the work in [12], the same evaluation is performed by using the mixed fingerprint approach.

Fig. 13 shows the successful rates of the two types of attacks by using the combined minutiae templates and the mixed fingerprints. Note that the security thresholds (FAR) of the tradition system are computed over FVC2002 DB2_A based on the FVC2002 protocol. We can see that the coding strategies do not have a significant impact on the successful rates of the two types of attacks. Compared with using the mixed fingerprints, it is more difficult to launch the attacks by using the combined minutiae templates. For *Attack Type* A, the successful rate is around 25% at $\mathrm{FAR} = 0.1\%$ using the combined minutiae templates. While the corresponding successful rate is 57.5% for the mixed fingerprints. For *Attack Type* B, the successful rates significantly reduce for both the combined minutiae templates and the mixed fingerprints. At FAR = 0.1%, the successful rate is almost 0% by using the combined minutiae templates. While

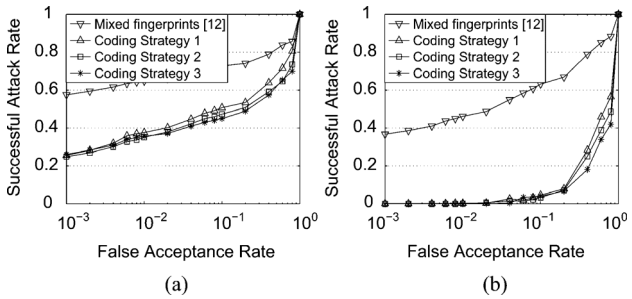


Fig. 13. Attack a traditional system using a combined minutiae template or a mixed fingerprint. (a) *Attack Type* A; (b) *Attack Type* B.

the corresponding successful rate is 36.7% for the mixed fingerprints, such a reduction shows that it is more dangerous to lose the minutiae positions than losing the orientation of the fingerprint.

If the attacker knows that a stolen template has been protected by using our technique, he would try to launch the aforementioned attacks based on the minutiae positions only, i.e., he would try to modify the minutiae matcher such that the minutiae directions are ignored during the matching. We implement a minutiae matcher based on the work proposed in [21], where we only use the minutiae positions for the matching. By using this matcher, the successful rates of *Attack Type* A and *Attack Type* B are 86.0% and 0.3% at FAR $= 0.1\%$, respectively.

V. Information Leakage in a Combined Minutiae Template

In this section, we analyze how much the information of an original fingerprint is exposed in a combined minutiae template. Suppose the two original minutiae templates of fingerprints A and B are $M_A = \{\mathbf{m}_{ia} = (\mathbf{p}_{ia}, \theta_{ia}),\ 1 \le i \le N\}$ and $M_B = \{\mathbf{m}_{ib} = (\mathbf{p}_{ib}, \theta_{ib}),\ 1 \le i \le N_1\}$, respectively. The corresponding combined minutiae template is $M_C = \{\mathbf{m}_{ic} = (\mathbf{p}_{ic}, \theta_{ic}),\ 1 \le i \le N\}$, where $\mathbf{p}_{ic}$ is a translated and rotated version of $\mathbf{p}_{ia}$. Given M_C, we define its information leakage as the probability $\wp_A$ and $\wp_B$ to recover M_A and M_B, respectively. In what follows, we will roughly estimate the value of $\wp_A$ and $\wp_B$. Note that the analysis would be the same for the information leakage in a corresponding combined fingerprint.

A. Recovering M_A From M_C

As all the minutiae positions of M_A are exposed in M_C, the attacker only needs to recover the minutiae direction θ_{ia} in M_A. Uludag [25] suggests to find the minutiae directions by using the ground truth orientation field of different fingerprint classes. Once the orientation field (in the range from 0 to π) is established, the direction of each minutiae point (in the range from 0 to 2π) could be randomly selected from one of the two directions corresponding to the orientation of the minutiae point.

Wilson *et al.* [26] indicate that there are five major fingerprint classes in general, i.e., arch, tented arch, right loop, left loop and whorl, where the prior class probabilities are 0.037, 0.029, 0.317, 0.338 and 0.279, respectively. For the attacker, the probability to correctly guess the class of fingerprint A will approximately equal to the corresponding prior class probability, which is denoted as ξ_A for simplicity. Note that the position of a loop of fingerprint A is exposed in M_C because the two primary reference points (usually are the loops) of fingerprints A and B are overlapped during the creation of M_C (see Section II-B1). While a rough orientation O_B of fingerprint B can be estimated from M_C using some existing orientation reconstruction algorithms [20], [23]. According to O_B, the attacker is able to obtain a rough location of the loop in fingerprint A. Therefore, once the class of fingerprint A is correctly guessed (with probability ξ_A), the attacker is able to compute a rough orientation O_A of fingerprint A based on the location of the loop. Then, θ_{ia} can be either recovered as $O_A(x_{ic}, y_{ic})$ or $O_A(x_{ic}, y_{ic}) + \pi$ by following the suggestion given in [25]. The probability of recovering M_A from M_C can be computed as

$$\wp_A = \xi_A \cdot \left(\frac{1}{2}\right)^N. \tag{20}$$

B. Recovering M_B From M_C

Recovering M_B from M_C is very difficult because the minutiae positions in M_B are not exposed at all in M_C. The attacker can only obtain a rough orientation O_B of fingerprint B from M_C[20], [23], based on which he knows the class of fingerprint B. Some existing works [25], [27] have shown that the distribution of the minutiae positions is not uniform and dependent on the fingerprint class. Such a property could be utilized for recovering the minutiae positions of fingerprint B. For simplicity, we term the probability to recover a minutiae position $\mathbf{p}_{ib} = (x_{ib}, y_{ib})$ of fingerprint B as $\zeta(x_{ib}, y_{ib})$. Assume the class of fingerprint B is right loop, $\zeta(x_{ib}, y_{ib})$ can be estimated as follows [25]

$$\zeta(x_{ib}, y_{ib}) = \frac{1}{N_r} \sum_{t=1}^{N_r} \frac{1}{2\pi\sigma_r^2} \cdot \exp\left(-\frac{(x_{ib} - x_t)^2 + (y_{ib} - y_t)^2}{2\sigma_r^2}\right) \tag{21}$$

where N_r is the total number of training minutiae points (extracted from right loop fingerprints), (x_t, y_t) refers to the position of a training minutiae point and ${\sigma_r}^2$ (${\sigma_r}^2 = 3$ as suggested in [25]) is the variance of the Gaussian window. We pick out all the right loop fingerprints (in total 50) from the first two impressions of the FVC2002 DB2_A database, where the width and height of each fingerprint are $W = 296$ and $H = 560$, respectively. As suggested in [25], these fingerprints are aligned manually such that the topmost loop of each fingerprint is located at the image center. We randomly select 25 of the 50 aligned fingerprints to extract the training minutiae points. In total, there are 1189 training minutiae points extracted. The minutiae positions are further quantized by a quantization step $\nu = 16$ which is assumed to be the estimation tolerance. The remaining 25 aligned fingerprints are served as the fingerprint B providing the testing minutiae points. There are in total 1172 testing minutiae points, where the average probability to recover a quantized minutiae position is 0.0025 based on (21). Note that if we consider an uniform distribution, the probability to recover a quantized minutiae position of fingerprint B will be $\nu^2/(W \cdot H) = 0.0015$. Thus, based on the knowledge of the distribution of the minutiae positions, the probability to recover all the N_1 minutiae positions of fingerprint B from M_C is

approximately 0.0025^{N_1}. The probability to recover M_B from M_C can then be estimated as

$$\wp_B = 0.0025^{N_1} \cdot \left(\frac{1}{2}\right)^{N_1}. \quad (22)$$

It should be noted that our analysis of the information leakage is based on the recovering of the original minutiae template. If the target is to obtain an approximate version of the original minutiae template (from the combined minutiae template) so that the attacker can launch a successful attack (*Attack Type* A or *Attack Type* B), it is expected that $\wp_A$ and $\wp_B$ will increase significantly as shown in [25].

VI. Conclusions

In this paper, we introduce a novel system for fingerprint privacy protection by combining two fingerprints into a new identity. In the enrollment, the system captures two fingerprints from two different fingers. A combined minutiae template containing only a partial minutiae feature of each of the two fingerprints will be generated and stored in a database. To make the combined minutiae template look real as an original minutiae template, three different coding strategies are introduced during the combined minutiae template generation process. In the authentication process, two query fingerprints from the same two fingers are required. A two-stage fingerprint matching process is proposed for matching the two query fingerprints against the enrolled template. Our combined minutiae template has a similar topology to an original minutiae template. Therefore, we are able to combine two different fingerprints into a new virtual identity by reconstructing a real-look alike combined fingerprint from the combined minutiae template. The experimental results show that our system achieves a very low error rate with FRR $= 0.4\%$ at FAR $= 0.1\%$. It is also difficult for an attacker to break other traditional systems by using the combined minutiae templates. Compared with the state-of-the-art technique, our technique can generate a better new virtual identity (i.e., the combined fingerprint) when the two different fingerprints are randomly chosen. The analysis shows that it is not easy for the attacker to recover the original minutiae templates from a combined minutiae template or a combined fingerprint.

References

[1] S. Li and A. C. Kot, "A novel system for fingerprint privacy protection," in *Proc. 7th Int. Conf. Inform. Assurance and Security (IAS)*, Dec. 5–8, 2011, pp. 262–266.

[2] B. J. A. Teoh, C. L. D. Ngo, and A. Goh, "Biohashing: Two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognit.*, vol. 37, no. 11, pp. 2245–2255, 2004.

[3] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You, "An analysis of biohashing and its variants," *Pattern Recognit.*, vol. 39, no. 7, pp. 1359–1368, 2006.

[4] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle, "Generating cancelable fingerprint templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 561–72, Apr. 2007.

[5] A. Nagar, K. Nandakumar, and A. K. Jain, "Biometric template transformation: A security analysis," in *Proc. SPIE, Electron. Imaging, Media Forensics and Security*, San Jose, Jan. 2010.

[6] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 4, pp. 744–57, Dec. 2007.

[7] W. J. Scheirer and T. E. Boult, "Cracking fuzzy vaults and biometric encryption," in *Proc. Biometrics Symp.*, Sep. 2007, pp. 34–39.

[8] S. Li and A. C. Kot, "Privacy protection of fingerprint database," *IEEE Signal Process. Lett.*, vol. 18, no. 2, pp. 115–118, Feb. 2011.

[9] A. Ross and A. Othman, "Visual cryptography for biometric privacy," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 70–81, Mar. 2011.

[10] B. Yanikoglu and A. Kholmatov, "Combining multiple biometrics to protect privacy," in *Proc. ICPR- BCTP Workshop*, Cambridge, U.K., Aug. 2004.

[11] A. Ross and A. Othman, "Mixing fingerprints for template security and privacy," in *Proc. 19th Eur. Signal Proc. Conf. (EUSIPCO)*, Barcelona, Spain, Aug. 29–Sep. 2, 2011.

[12] A. Othman and A. Ross, "Mixing fingerprints for generating virtual identities," in *Proc. IEEE Int. Workshop on Inform. Forensics and Security (WIFS)*, Foz do Iguacu, Brazil, Nov. 29–Dec. 2, 2011.

[13] E. Camlikaya, A. Kholmatov, and B. Yanikoglu, "Multi-biometric templates using fingerprint and voice," *Proc. SPIE*, vol. 69440I, pp. 69440I-1–69440I-9, 2008.

[14] K. G. Larkin and P. A. Fletcher, "A coherent framework for fingerprint analysis: Are fingerprints holograms?," *Opt. Express*, vol. 15, pp. 8667–8677, 2007.

[15] S. Li and A. C. Kot, "Attack using reconstructed fingerprint," in *Proc. IEEE Int. Workshop on Inform. Forensics and Security (WIFS)*, Foz do Iguacu, Brazil, Nov. 29–Dec. 2, 2011.

[16] VeriFinger 6.3. [Online]. Available: http://www.neurotechnology.com

[17] L. Hong, Y. F. Wan, and A. Jain, "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 777–789, Aug. 1998.

[18] K. Nilsson and J. Bigun, "Localization of corresponding points in fingerprints by complex filtering," *Pattern Recognit. Lett.*, vol. 24, no. 13, pp. 2135–2144, 2003.

[19] S. Chikkerur and N. Ratha, "Impact of singular point detection on fingerprint matching performance," in *Proc. Fourth IEEE Workshop on Automat. Identification Advanced Technologies*, Oct. 2005, pp. 207–212.

[20] Y. Wang and J. Hu, "Global ridge orientation modeling for partial fingerprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 72–87, Jan. 2011.

[21] X. Jiang and W. Yau, "Fingerprint minutiae matching based on the local and global structures," in *Proc. 15th Int. Conf. Pattern Recognition*, 2000, vol. 2, pp. 1038–1041.

[22] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, "Fingerprint image reconstruction from standard templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1489–1503, Sep. 2007.

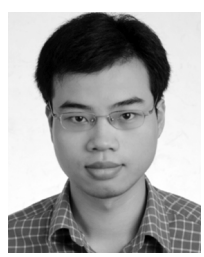
[23] J. Feng and A. K. Jain, "Fingerprint reconstruction: From minutiae to phase," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 209–223, Feb. 2011.

[24] R. Cappelli, A. Erol, D. Maio, and D. Maltoni, "Synthetic fingerprint-image generation," in *Proc. 15th Int. Conf. Pattern Recognition*, Sep. 3–7, 2000, vol. 3, pp. 471–474.

[25] U. Ulugdag, "Secure Biometric Systems," Ph.D. thesis, Michigan State Univ., East Lansing, MI, 2006.

[26] C. L. Wilson, G. T. Candela, and C. I. Watson, "Neural network fingerprint classification," *J. Artif. Neural Netw.*, vol. 1, no. 2, pp. 203–228, 1994.

[27] Y. Chen, "Extended Feature set and Touchless Imaging for Fingerprint Matching," Ph.D. thesis, Michigan State Univ., East Lansing, MI, 2009.



Sheng Li (S'11) received the B.Eng. degree in communication engineering and the M.Eng. degree in communication and information system from Shanghai University, China, in 2005 and 2008, respectively. He is currently working toward the Ph.D. degree at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

His current research interests include biometric template protection, pattern recognition, multimedia forensics and security.

Mr. Li is the recipient of the IEEE WIFS Best Student Paper Silver Award.

Alex C. Kot (F'06) has been with the Nanyang Technological University, Singapore since 1991. He headed the Division of Information Engineering at the School of Electrical and Electronic Engineering for eight years and served as Associate Chair Research and Vice-Dean Research for the School of Electrical and Electronic Engineering. He is currently Professor and Associate Dean for College of Engineering. He is also Director of the Rapid-Rich Object SEarch (ROSE) Laboratory. He has published extensively in the areas of signal processing for communication, biometrics, data-hiding, image forensics, and information security.

Dr. Kot served as Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE SIGNAL PROCESSING LETTERS,*IEEE Signal Processing Magazine*, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (IEEE CSVT), and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS as well as IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS. He also served as Guest Editor for the Special Issues for the IEEE CSVT and the EURASIP*Journal of Advanced Signal Processing* (JASP). He is currently Associate Editor for the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and IEEE TRANSACTIONS ON IMAGE PROCESSING. He is also Editor for the EURASIP JASP, and the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING. He served in the IEEE SPS Image and Video Multidimensional Signal Processing and he is now serving in the IEEE CAS Visual Signal Processing and Communication, and IEEE SPS Information Forensic and Security technical committees. He has served the IEEE Society in various capacities such as the General Cochair for the 2004 IEEE International Conference on Image Processing (ICIP) and Chair of the worldwide SPS Chapter Chairs and the Distinguished Lecturer program. He served as IEEE Fellow Evaluation Committee. He received the Best Teacher of the Year Award and is a coauthor for several Best Paper Awards including ICPR, IEEE WIFS, and IWDW. He was the IEEE Distinguished Lecturer in 2005 and 2006 and is a Fellow of Academy of Engineering, Singapore and IES. He is the Vice-President Elect for the IEEE Signal Processing Society.