Key Based Artificial Fingerprint Generation for Privacy Protection

Sheng Li, Xinpeng Zhang, Zhenxing Qian, *Member, IEEE*, Guorui Feng, and Yanli Ren

Abstract—With the widespread use of biometrics recognition systems, it is of paramount importance to protect the privacy of biometrics. In this paper, we propose to protect the fingerprint privacy by the artificial fingerprint, which is generated based on three pieces of information, i) the original minutiae positions; ii) the artificial fingerprint orientation; and iii) the artificial minutiae polarities. To make it real-look alike and diverse, we propose to generate the artificial fingerprint orientation by a model taking both the global and local fingerprint orientation into account. Its parameters can be easily guided by an user specific key with simple constraints. The artificial minutiae polarities are generated from the same key, where a block based and a function based approach are proposed for the minutiae polarities generation. These information are properly integrated to form a real-look alike artificial fingerprint. It is difficult for the attacker to distinguish such a fingerprint from the real fingerprints. If it is stolen, the complete fingerprint minutiae feature will not be compromised, and we can generate a different artificial fingerprint using another key. Experimental results show that the artificial fingerprint can be recognized accurately.

Index Terms—Artificial, fingerprint, privacy protection

1 Introduction

Nowadays, biometrics such as fingerprint, face, iris, and voice, are widely used in authentication systems. In general, biometrics needs to be stored in a database for subsequent authentication. However, templates stored in the database are at the risk of being stolen. Once the template is stolen, people's identity will be compromised forever. Furthermore, the stolen templates are difficult to be replaced like passwords, which create difficulties for authorized person to enter the system. Thus, biometric templates should be stored in the database such that both the privacy of the template and the security of the system are not compromised under various attacks.

Traditional encryption is not sufficient to protect the biometrics template because decryption is required before the biometrics matching, which exposes the original template to the attacker. Therefore, in recent years, significant efforts have been put in developing specific protection techniques for biometrics, where a lot of attention has been paid on the fingerprint. Teoh et al. [1] propose a biohashing approach by projecting the user's fingerprint feature onto an orthogonal pseudo-random matrix (i.e., the key). The accuracy of this approach mainly depends on the key, which is assumed to be never stolen or shared [2]. Ratha et al. [3] propose to generate cancelable fingerprint templates by applying non-invertible transformations on the minutiae, which requires proper fingerprint alignment. In order to create alignment-free cancelable fingerprint templates, Lee et al. [4] transform the minutiae based on some changing functions guided by a rotation and translation-free value. Ferrara et al. [5] propose to protect the fingerprint minutiae cylinder-code based on the fingerprint KL projection followed by binarization. This approach can be improved by incorporating an user specific key to permutate the protected template [6]. Besides the projection or transformation based approaches, researchers also devote efforts in generating cryptographic keys from fingerprints [7], [8], where additional chaff points (i.e., noise) are added during the encoding process.

The above mentioned approaches inevitably destroy the topology or structure of the fingerprint data. The corresponding protected fingerprint data are noise like, it would be quite easy for the attacker to differentiate these data from the real fingerprint data. Ross et al. [9] indicate that a noise like protected biometric template can cause the attacker's interest by suggesting the existence of secret data. Ratha et al. [10] suggest that a secure biometric recognition system should not give the impression to the attacker that the system is using a specific protection technique. If the attacker notices that a stolen fingerprint template has been protected. He might be interested and put more efforts to attack the template. Some work have shown that, given the cancelable fingerprint template and the corresponding transformation (i.e., the key), up to 94 percent of the original fingerprint template can be recovered [11], [12], [13], [14], [15]. In [16], the authors indicate that the key generation based approaches [7] are vulnerable to key-inversion attacks.

- S. Li, X. Zhang, and Z. Qian are with the Shanghai Institute of Intelligent Electronics and Systems, School of Computer Science, Fudan University, Shanghai 201203, P.R. China.
E-mail: {lisheng, zhangxinpeng, zxqian}@fudan.edu.cn.
- G. Feng and Y. Ren are with the Shanghai Institute for Advanced Communication and Data Science, School of Communication and Information Engineering, Shanghai University, Shanghai 200444, P.R. China.
E-mail: {grfeng, renyanli}@shu.edu.cn.

Manuscript received 20 Jan. 2017; revised 6 Feb. 2018; accepted 27 Feb. 2018. Date of publication 5 Mar. 2018; date of current version 9 July 2020.
(Corresponding author: Sheng Li.)
Digital Object Identifier no. 10.1109/TDSC.2018.2812192

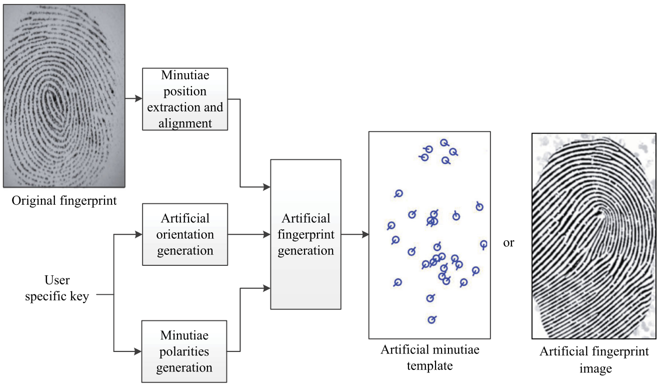

Original fingerprint
Minutiae position extraction and alignment
Artificial orientation generation
User specific key
Minutiae polarities generation
Artificial fingerprint generation
Artificial minutiae template
or
Artificial fingerprint image

Fig. 1. The proposed method for artificial fingerprint generation.

Only a few methods are able to protect the privacy of fingerprint without destroying the structure of fingerprint data, which require a pair of fingerprints to work together [17], [18]. Such a requirement is not practical in same applications. In addition, the protected fingerprint generated by these schemes can not be easily replaced. Once it is stolen, the user needs to switch the two original fingerprints or use another pair for replacement.

In this paper, a key based artificial fingerprint generation scheme is proposed for privacy protection. Given an original fingerprint, we extract the minutiae positions which capture part of the fingerprint minutiae features. An artificial fingerprint is formed based on the minutiae positions and some artificial information generated based on an user specific key, including the artificial fingerprint orientation (termed as the artificial orientation for short) and the minutiae polarities. To generate real-look alike artificial orientation with sufficient diversity, we propose an artificial orientation generation model by combining a global and a local orientation model linearly. This model is controlled by a set of parameters with simple constraints, which can be easily guided by the key. To alleviate the variations among different impressions of the same finger, we propose to compute the minutiae polarities using block partition or a smooth function. By taking the artificial orientation, the minutiae polarities, as well as the minutiae positions into account, we generate an artificial fingerprint both in the feature domain and the image domain. By storing the artificial fingerprint, the complete minutiae feature of the original fingerprint will not be compromised when the database is stolen. The performance of artificial fingerprint recognition is shown to be satisfactory using an existing fingerprint matching algorithm.

Unlike the protected fingerprints generated using the existing transformation (or projection) based approaches [1], [3], [4], [5], our artificial fingerprint is real-look alike, which is difficult to be distinguished from the real fingerprints. If it is stolen, the attacker may be fooled and treat it as a real fingerprint. Thus, the risk of the stolen fingerprint being attacked is reduced. Compared with the existing techniques that can generate protected fingerprint with similar structure to the real fingerprint data [17], [18], our scheme does not require two fingerprints to work together. When the artificial fingerprint is stolen, it can be replaced easily by issuing a different key.

The rest of this paper is organized as follows. Section 2 introduces the proposed method for generating the artificial fingerprints. Section 3 presents the experimental results. Section 4 analyzes the irreversibility of the proposed scheme. The transformation of minutiae positions is discussed in Section 5, followed by the conclusions given in the last section.

2 The Proposed Method

Fig. 1 shows the flowchart of the proposed method for generating artificial fingerprints. First, the minutiae positions of the user's original fingerprint are extracted and aligned. A piece of artificial orientation is generated from an user specific key by our proposed fingerprint orientation (termed as the orientation for short) model. Then, we proposed two approaches to generate the minutiae polarities based on the same key. Given the minutiae positions, the minutiae polarities, and the artificial orientation, an artificial fingerprint can be generated both in the feature domain (i.e., the artificial minutiae template) and the image domain (i.e., the artificial fingerprint image).

2.1 Minutiae Position Extraction and Alignment

Any of the existing minutiae extraction algorithms could be used for the minutiae position extraction. We here use the Verifinger 6.3 software [19] to extract a set of n minutia positions from the original fingerprint F. In order to align the minutiae positions, we detect the location and angle of the primary core of F using the improved complex filters for singular point detection[20], which are denoted as $\mathbf{p} = (p_x, p_y)$ and α, respectively.

Let's denote $\mathbf{m}_i = (x_i, y_i)$, $1 \leq i \leq n$, as one of the n minutiae positions. The alignment is performed by translating and rotating each minutiae position to

$$(\mathbf{m}'_i)^T = \mathbf{H}_\beta \cdot (\mathbf{m}_i - \mathbf{p})^T + (\mathbf{e})^T, \tag{1}$$

where $()^T$ is the transpose operator, $\mathbf{e} = (e_x, e_y)$ is the location of the center of F, $\beta = \alpha - \pi/2$, and $\mathbf{H}_\beta$ is the rotation matrix with β as the rotation angle, where

$$\mathbf{H}_\beta = \begin{bmatrix} \cos(\beta), \sin(\beta) \\ -\sin(\beta), \cos(\beta) \end{bmatrix}. \tag{2}$$

After such alignment, the primary core is overlapped with the center of the fingerprint with the angle of $\pi/2$.

2.2 Artificial Orientation Generation

There are five major fingerprint classes in general, i.e., arch, tented arch, right loop, left loop and whorl [21]. The purpose of artificial orientation generation is to compute a certain type of orientation (corresponding to one of the five fingerprint classes) based on a set of parameters guided by an user specific key κ, which has been paid little attention to in literature. In [22], the authors suggest to generate the artificial orientation using a zero-pole model [23]. The orientation at point (x, y) is computed as

$$O_\varepsilon(z) = \frac{1}{2}\left[\sum_{i=1}^{n_c} arg(z - c_i) - \sum_{i=1}^{n_d} arg(z - d_i)\right], \tag{3}$$

where $z = y + jx$ is a complex number, $arg(z)$ returns the argument of the complex number z, c_i $(i = 1, 2, \ldots, n_c)$ and d_i $(i = 1, 2, \ldots, n_d)$ refer to the locations (both are in the complex domain) of the fingerprint cores and deltas (i.e., singular points), respectively. This method is able to generate the artificial orientation by some simple constraints on the

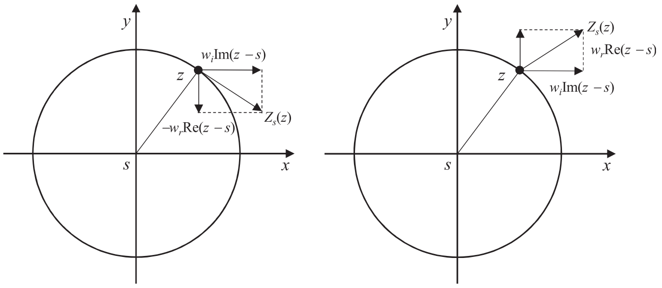


Fig. 2. Illustration of the weighted point-charge model. Left: The influence vector around a core, right: The influence vector around a delta.

locations of singular points [23]. However, the orientation generated for singular points with similar locations will also be similar, which is usually not the case for real fingerprints. To deal with this issue, a more sophisticated orientation model should be explored for artificial orientation generation. Concretely, more parameters are needed in the model to create the artificial orientation with more diversity.

People have proposed sophisticated models for estimating orientation from a fingerprint image [24], [25], [26], where the parameters are usually optimized based on the fingerprint region in the image. The distributions and constraints of these parameters for real fingerprints are yet to be explored. Without such knowledge, it is difficult to randomly choose a set of artificial parameters that can generate real-look alike orientation. Thus, these models may not be suitable for the task of artificial orientation generation. In this section, we propose to generate the artificial orientation by a model combining global orientation and local orientation. The global orientation keeps the basic topology of different fingerprint classes, which is computed directly from the zero-pole model [23]. The local orientation describes the orientation around the singular point area, which is computed by a local orientation model guided by a few parameters. With the help of the local orientation, we are able to generate diverse artificial orientation even if the singular points are with similar locations.

Next, we explain the local orientation model and the combined orientation model in detail. Since the range of fingerprint orientation is defined within $[0,\pi)$, there is an inevitable discontinuity on π. As suggested in [20], [24], representing the orientation in the complex domain would be a possible solution for this problem. Given the orientation O, the corresponding orientation in the complex domain is computed as

$$Z = \cos(2O) + j\sin(2O). \tag{4}$$

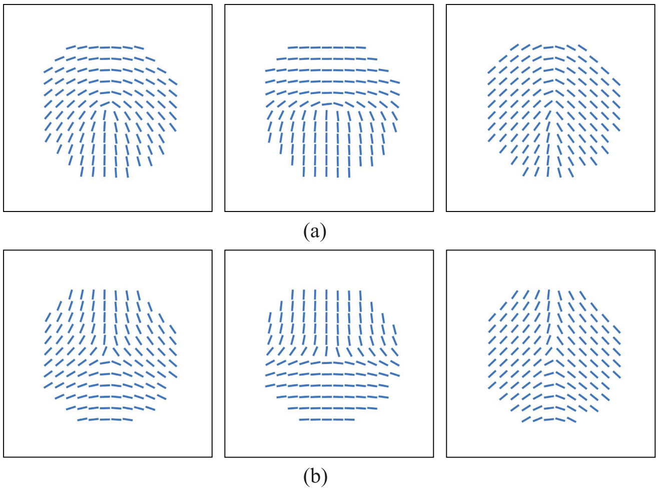
On the other hand, O can be computed from Z by

$$O = \frac{1}{2}arg(Z). \tag{5}$$

In the rest of this section, the orientation in the complex domain is termed as the complex orientation for simplicity, and all the points are located in the complex domain.

2.2.1 The Local Orientation Model

The standard local orientation can be generated using a point-charge model proposed in [24], where the quantity of electricity is assumed to be the same for the real and image part of the complex local orientation. In order to generate diverse local orientation for a singular point $s = s_y + js_x$, we propose a weighted point-charge model below:


(a)

(b)

Fig. 3. Local orientation generated for (a) a core and (b) a delta using the weighed point-charge model. From left to right: $w_r = 1$ and $w_i = 1$; $w_r = 0.9$ and $w_i = 0.2$; $w_r = 0.2$ and $w_i = 0.9$. The rotation angle θ is set as 0 for all. Both the core point and the delta point are located at the center.

$$Z_s(z) = \begin{cases} \frac{-w_r\mathrm{Re}(z-s)+jw_i\mathrm{Im}(z-s)}{v} & if\ s \in cores \\ \frac{w_r\mathrm{Re}(z-s)+jw_i\mathrm{Im}(z-s)}{v} & if\ s \in deltas, \end{cases} \tag{6}$$

where $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$ refer to the real and image part of the complex number z, w_r and w_i are the weights (i.e., the quantity of electricity) for the real and image part of $z-s$, and v is the normalization scalar

$$v = \sqrt{w_r^2\mathrm{Re}^2(z-s) + w_i^2\mathrm{Im}^2(z-s)}. \tag{7}$$

Fig. 2 illustrates the influence vector around a core and a delta on an unit circle for the weighted point-charge model. With different weights applied, the influence vector is no longer tangent to the circle (for a core) or the radial of the circle (for a delta).

We define the area with s as the central point and ϵ_s as the radius as the effective region for the singular point. By taking the effective region and rotation into consideration, the complex local orientation for s is further computed as

$$Z_s{}'(z) = \begin{cases} Z_s(z_\theta) & if\ d(z,s) \le \epsilon_s \\ 0 & otherwise, \end{cases} \tag{8}$$

where z_θ is the rotated version of point z with s as the rotation center and θ as the rotation angle (please refer to Eq. (1) for computing z_θ), $d(z,s)$ refers to the Euclidian distance between point z and s. Fig. 3 illustrates some local orientation generated using the weighted point-charge model. It can be seen that such a model is able to produce diverse local orientation by choosing different w_r and w_i.

2.2.2 The Combined Orientation Model

We use the orientation computed from the zero-pole model [23] as the global orientation, and denote its representation in complex domain as Z_g. We further denote the complex local orientation for the ith singular point s_i as Z'_{si}. The combined orientation model is obtained by

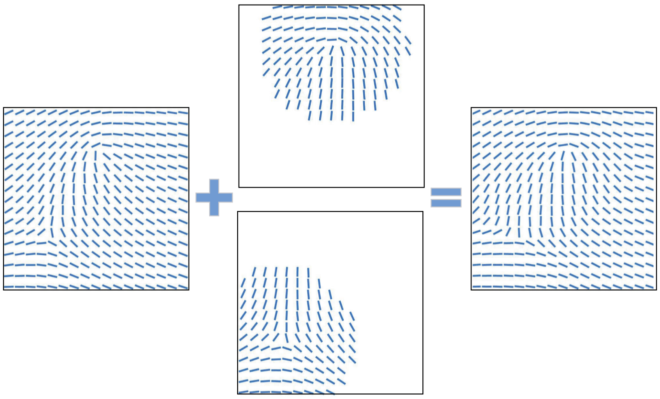


Fig. 4. Artificial orientation generated before and after the orient combination. From left to right: The global orientation; the local orientation for a core (top) and a delta (bottom); the combined orientation.

combining Z_g and Z'_{si} linearly

$$Z_\kappa(z) = \frac{G(z)Z_g(z) + \sum_{i=1}^{u} S_i(z)Z'_{si}(z)}{\left|G(z)Z_g(z) + \sum_{i=1}^{u} S_i(z)Z'_{si}(z)\right|}, \tag{9}$$

where $|z|$ computes the amplitude of the complex number z, u is the number of singular points, G and S_i are the weight maps for the global and local orientation, respectively. G and S_i define the importance of global and local orientation at each point. For real fingerprints, the importance of the local orientation usually decays when the distance between current point z and the singular point s_i increases, which could be formulated as

$$S_i(z) = 1 - \left[\frac{d(z - s_i)}{\epsilon_{si}}\right]^{\tau_i}, \tag{10}$$

where τ_i is the power measuring the slope of the decay, and ϵ_{si} is the effective region for s_i. In order to preserve the basic fingerprint topology, $G(z)$ can not be too small. We set $G(z)$ as a constant g over the whole fingerprint image with $g > g_t$.

The advantage of such a combined orientation model is that all the parameters have clear constraints with respective ranges. We will discuss later regarding the settings of these ranges. As long as the artificial parameters are chosen within the corresponding ranges, we can produce real-look alike artificial orientation thanks to the incorporation of the global orientation. By combining the local orientation, we are able to generate the artificial orientation with more diversity compared with using the global orientation only (i.e., $\tau_i = 0$), as shown in Fig. 4.

In order to generate the artificial orientation, artificial parameters need to be generated to fit the combined orientation model. We treat all the parameters as random variables with different ranges. A pseudorandom number generator is used to generate these variables with the seed being the user specific key κ. In the following discussions, all the randomly generated parameters (or numbers) refer to such pseudorandom numbers. Given an user specific key κ, the major steps of the artificial orientation generation are summarized as follows.

(1) Generate a random number t (within the range of $[0, 1]$) to determine the type of the artificial orientation according to the categorical distribution (i.e., prior possibilities) of the five major fingerprint classes [21]. As pointed out in [27], there should be one core and one delta for tented arch, left loop and right loop, and two cores and two deltas for whorl (i.e., $u = 2$ for tented arch, left loop and right loop, and $u = 4$ for whorl).

(2) Determine the locations of the singular points using the random singularities generation approach proposed in [28], the parameters of which are randomly generated.

(3) Compute the complex artificial orientation Z_κ based on the combined orientation model given in Eq. (9), where all the rest parameters are randomly generated (including w_r, w_i, θ, ϵ_s, τ_i, and g). The artificial orientation O_κ can be computed directly from Z_κ using Eq. (5).

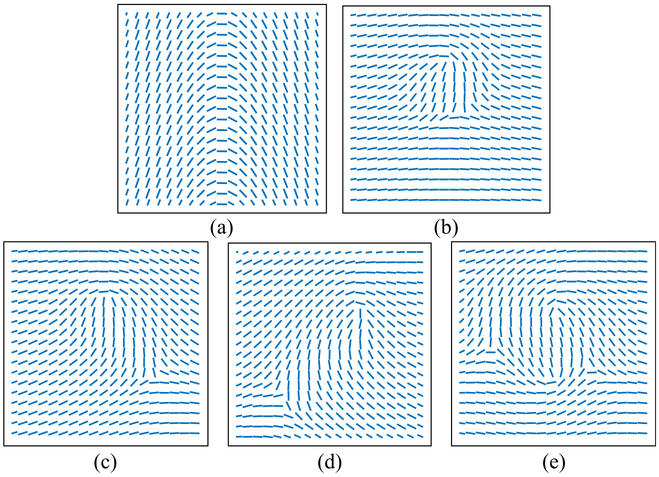


Fig. 5. Artificial orientation generated for (a) arch, (b) tented arch, (c) left loop, (d) right loop, and (e) whorl.

Sometimes, the type of the artificial orientation might be determined as arch, which does not contain any singular point (i.e., $u = 0$). In such a case, the combined orientation model does not work. We adopt the arch orientation model proposed in [22], where the orientation at point z is computed by

$$O_\kappa(z) = \arctan\left(\lambda \cos\left(\frac{\mathrm{Im}(z)\pi}{\mathrm{M}}\right)\right), \tag{11}$$

where M is the width of the fingerprint, and λ is the parameter controlling the curvature of the arch, the range of which is empirically set within $[0.3, 3]$. Thus, only one parameter is used for generating the artificial orientation of arch. We believe it is not a big issue since only 3.7 percent of the real fingerprints are arch [21]. Fig. 5 illustrates examples for each of the five types of artificial orientation generated.

2.3 Minutiae Polarity Generation

Given the artificial orientation O_κ, the artificial direction of each aligned minutiae position $\mathbf{m}'_i$ can be computed as

$$\theta'_{i\kappa} = O_\kappa(x'_i, y'_i) + \rho_{i\kappa}\pi, \tag{12}$$

where $\rho_{i\kappa}$ is an integer that is either 0 or 1. With the constraint of the artificial orientation, this scheme is able to make the topology and range (from 0 to 2π) of artificial minutiae directions close to those from real fingerprints. We term $\rho_{i\kappa}$ as the minutiae polarity of $\mathbf{m}'_i$. How to determine the value of $\rho_{i\kappa}$ is an important issue here. First of all, $\rho_{i\kappa}$ should be determined by κ only, which does not expose any information of the original fingerprint. Second, $\rho_{i\kappa}$ has to be tolerable against the variations among different impressions of the same finger. In this section, we propose two approaches for generating the minutiae polarities.

2.3.1 Block Based Minutiae Polarity

We partition the fingerprint image into a set of non-overlapping blocks with fixed size $\delta \times \delta$, each block is assigned with a random integer of value 0 or 1. The minutiae polarity is computed as the value of the block which accommodates the corresponding minutiae position, i.e.,

$$\rho_{i\kappa} = \mathrm{Blk}\left(\left\lfloor \frac{x_i{}'}{\delta} \right\rfloor, \left\lfloor \frac{y_i{}'}{\delta} \right\rfloor\right), \tag{13}$$

where $\mathrm{Blk}(a, b)$ refers to the value of the block with the two dimensional index (a, b) and $\lfloor \bullet \rfloor$ is the floor operation which gets the largest integer less than or equal to $\bullet$.

2.3.2 Function Based Minutiae Polarity

In this approach, we propose to construct a function Γ_κ for computing the minutiae polarities, where

$$\rho_{i\kappa} = \Omega(\Gamma_\kappa(x_i{}', y_i{}')), \tag{14}$$

where $\Omega(a)$ is an indicator function

$$\Omega(a) = \begin{cases} 1 & if\, a \geq 0 \\ 0 & if\, a\, <\, 0. \end{cases} \tag{15}$$

To alleviate the variations among different impressions of the same finger, Γ_κ has to be locally smooth. We mix a set of h Gaussian kernels to form Γ_κ, i.e.,

$$\Gamma_\kappa(\mathbf{x}) = \sum_{i=1}^{h} h_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i), \tag{16}$$

where $\mathbf{x}$ is a two dimensional vector, $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i)$ is the ith Gaussian kernel with $\boldsymbol{\mu}_i$ as the center and $\boldsymbol{\Lambda}_i$ as the covariance matrix, and h_i is the weight for the ith Gaussian kernel.

Similar to the generation of artificial orientation, different parameters (i.e., $\boldsymbol{\mu}_i$, h_i and $\boldsymbol{\Lambda}_i$) need to be generated to compute each kernel, where the parameter ranges should be determined. We set the center $\boldsymbol{\mu}_i$ (a two dimensional vector) to be located within the fingerprint image, and the range of the weight h_i as $[-1, 1]$. In order to make the covariance matrix $\boldsymbol{\Lambda}_i$ invertible, we set $\boldsymbol{\Lambda}_i$ as a diagonal matrix, the elements of which are within the range of $[\mathrm{L}/3, 2\mathrm{L}/3]$ with

$$\mathrm{L} = \min(\mathrm{M}, \mathrm{N}), \tag{17}$$

where M and N are the width and height of the fingerprint image, and $\min(\mathrm{M}, \mathrm{N})$ returns the minimum of M and N. With the ranges determined, these parameters can be randomly generated based on κ, and $\rho_{i\kappa}$ can be computed accordingly.

2.4 Artificial Fingerprint Generation

We obtain the artificial minutiae template M_κ by integrating the aligned minutiae positions and the artificial minutiae directions

$$M_\kappa = \{(\mathbf{m}_i', \theta_{i\kappa}'), 1 \leq i \leq n\}. \tag{18}$$

Sometimes, a global translation is necessary to be applied on M_κ such that all the minutiae points are located inside the fingerprint image. In M_κ, the minutiae positions are extracted from the original fingerprint, while the minutiae directions are computed with the constraint of the artificial orientation that follows the topology of real fingerprints. Therefore, M_κ has a similar topology to the minutiae extracted from the real fingerprints.

Next, we adopt the amplitude and frequency modulated (AM-FM) fingerprint model [29] for generating artificial fingerprint images. The AM-FM fingerprint model is initially proposed by Lakin and Fletcher [29], which is very useful in fingerprint reconstruction [30], [31]. Given the original fingerprint image F, the AM-FM fingerprint model represents the intensity of each pixel (x, y) as

$$F(x, y) = A(x, y) + B(x, y) \cdot \cos[\psi(x, y)] + N(x, y), \tag{19}$$

where $A(x, y)$ is the offset, $B(x, y)$ is the amplitude, $\psi(x, y)$ is the hologram phase, and $N(x, y)$ refers to the noise. The hologram phase ψ determines the ridges and minutiae of the fingerprint, and $\cos(\psi)$ refers to the fingerprint image without noise. It can be decomposed as

$$\psi(x, y) = \psi_c(x, y) + \psi_s(x, y), \tag{20}$$

where ψ_c is the continuous phase and ψ_s is the spiral phase. The continuous phase mainly depends on the orientation and ridge frequency of F. The spiral phase can be calculated by the n minutiae positions $\mathbf{m}_i$ of F

$$\psi_s(x, y) = \sum_{i=1}^{n} p_i \arctan\left(\frac{y - y_i}{x - x_i}\right), \tag{21}$$

where $p_i \in \{-1, 1\}$ is the bipolar polarity of $\mathbf{m}_i$.

Given the aligned minutiae positions $\mathbf{m}_i'$, the artificial orientation O_κ, and the minutiae polarities $\rho_{i\kappa}$, the artificial fingerprint image F_κ can be generated by combining its continuous phase $\psi_{c\kappa}$ and spiral phase $\psi_{s\kappa}$ as shown in Fig. 6. In order to generate $\psi_{c\kappa}$, we randomly choose a fixed fingerprint ridge frequency f_κ based on κ. The range of f_κ is within $[1/6, 1/9]$, which covers the typical range of ridgeline frequency in 500-dpi fingerprint images [32]. Based on O_κ and f_κ, $\psi_{c\kappa}$ can be generated using the continuous phase reconstruction method proposed in [31]. In order to obtain $\psi_{s\kappa}$, we transform $\rho_{i\kappa}$ into the bipolar polarity by

$$p_{i\kappa} = \begin{cases} -1 & if\; \rho_{i\kappa} = 0 \\ 1 & if\; \rho_{i\kappa} = 1. \end{cases} \tag{22}$$

Based on the minutiae positions $\mathbf{m}_i'$ and the corresponding bipolar polarities $p_{i\kappa}$, $\psi_{s\kappa}$ can be computed using Eq. (21). The artificial fingerprint image is then computed by

$$F_\kappa = \cos(\psi_{c\kappa} + \psi_{s\kappa}). \tag{23}$$

Finally, a noising and rendering step [22] is applied on F_κ to make it real-look alike (see Fig. 6).

3 Experimental Results

3.1 Parameter Range Settings for Artificial Orientation

In our combined orientation model (see Eq. (9)), there are five types of parameters except the locations of the singular

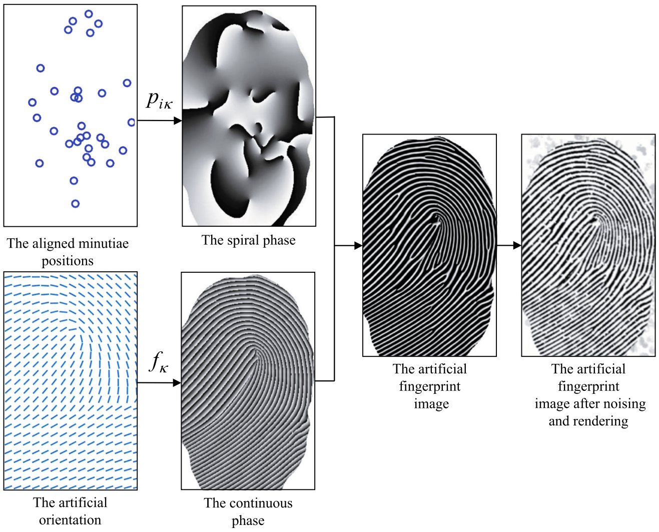

$p_{i\kappa}$
f_κ
The aligned minutiae positions
The spiral phase
The artificial orientation
The continuous phase
The artificial fingerprint image
The artificial fingerprint image after noising and rendering

Fig. 6. Artificial fingerprint image generation based on the AM-FM fingerprint model.

points, including the weights w_r and w_i, the rotation angle θ, the radius of the effective region ϵ_s, the slope of the local orientation decay τ_i, and the importance of the global orientation g. Proper ranges should be defined for these parameters to generate real-look alike artificial orientation with sufficient diversity. Next, we explain in detail regarding how to set these ranges.

The weights w_r and w_i determine the shape of the local orientation. According to Eqs. (5) and (6), the local orientation depends on the ratio between w_r and w_i. Therefore, we set $w_r \in (0,1]$ and $w_i \in (0,1]$, which covers all the possible ratios (except zero) to produce local orientation with sufficient diversity. The rotation angle θ reflects the rotation of the singular point, we set $\theta \in [-\pi/6, \pi/6]$, which is in accordance of normal fingerprint rotations. For each of the rest three types of parameters, we set it to different values and compute the difference between the combined orientation O_κ (with other parameters fixed) and the corresponding global orientation O_g, the results of which are shown in Fig. 7. For the radius of the effective region, regardless the type of the singular point, we set $\epsilon_s \in [\mathrm{L}/3, 2\mathrm{L}/3]$, which produces the rapid changes of O_κ when compared with O_g.

For τ_i or g, we observe that it can hardly produce more diversity (in O_κ) of value more than 2, with less than 0.5 degrees of additional difference between O_κ and O_g. In our implementation, we set $\tau_i \in [0,2]$ and $g \in [0.3,2]$ (thus $g_t = 0.3$). The reason we choose a relatively high lower bound for g (i.e., $g_t = 0.3$) is to make sure that the basic fingerprint topology is preserved. It should be noted that there is no harm to moderately increase the upper bound of these parameters as the basic fingerprint topology is preserved thanks to the incorporation of global orientation.

3.2 Accuracy

We evaluate the accuracy of artificial fingerprint recognition based on the first 3 impressions of the FVC2002 DB2_A database, which contains 300 fingerprints from 100 fingers (with 3 impressions per finger). The VeriFinger 6.3 [19] is used for the minutiae positions extraction and fingerprint matching. We do not use all the 8 impressions of the FVC2002 DB2_A database, the reason is that we find other impressions (4 to 8) may contain partial fingerprints without any core points. While our proposed method does not work for such fingerprints.

We assign each finger with an user specific key κ and generate the artificial fingerprint in the feature domain (i.e., the artificial minutiae template) and the image domain (i.e., the artificial fingerprint image). For each finger, the artificial fingerprint generated from the first impression is served as the gallery, while those generated from the other 2 impressions are served as the probes. We match each of the probes against the corresponding gallery, producing 2 genuine matches. Thus, we have 200 genuine matches for all the 100 fingers. The above process is repeated 10 times, each time we assign a different key for each finger. Therefore, we have 2,000 genuine matches for the 100 fingers. The 100 gallery artificial fingerprints generated from the first of the 10 repeated process are used to compute the decision thresholds. Each of the gallery is matched against the rest 99 galleries, which produces $100 \times 99/2 = 4950$ imposter matches (the repeated ones are removed).

Table 1 shows the accuracy (i.e., false rejection rate) of the artificial fingerprint recognition with different block size δ or different number of Gaussian kernels h, where the artificial fingerprint images are generated without the noising

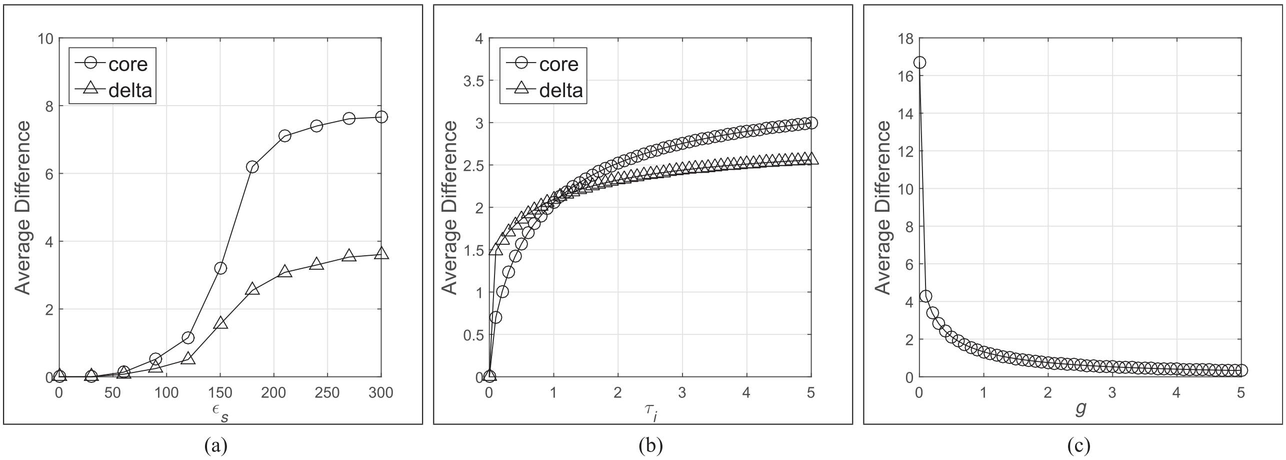


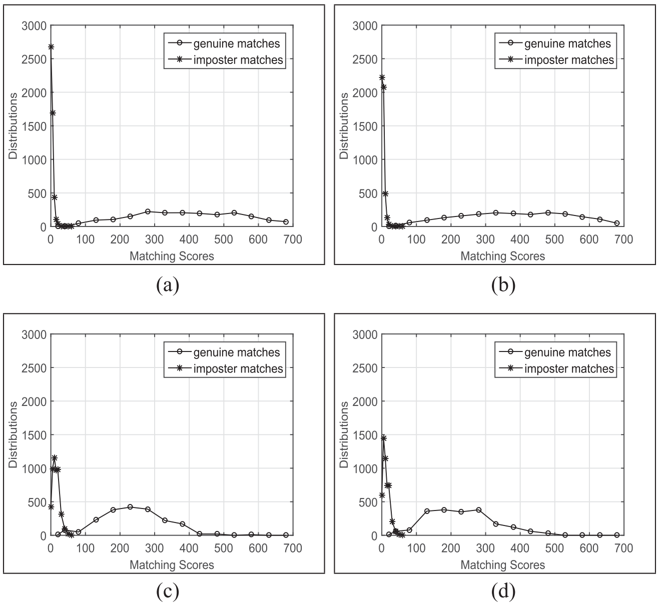
Fig. 7. The average difference (in degrees) between O_κ and O_g with different settings of (a) ϵ_s, (b) τ_i, and (c) g. The size of the orientation is with the height of 560 and width of 296 (i.e., $\mathrm{L} = 296$), which is the same as the size of the images in the FVC2002 DB2_A database.

TABLE 1
The Accuracy of Artificial Fingerprint Recognition (Values in Percentage)

	Block based artificial fingerprint						Function based artificial fingerprint					
	minutiae template			fingerprint image			minutiae template			fingerprint image		
	EER	$\mathrm{FAR}_{0.1}$	FAR_0	EER	$\mathrm{FAR}_{0.1}$	FAR_0	EER	$\mathrm{FAR}_{0.1}$	FAR_0	EER	$\mathrm{FAR}_{0.1}$	FAR_0
$\delta = 16, h = 16$	0.86	1.35	3.25	1.18	7.15	14.05	0.66	0.80	1.50	0.51	2.95	6.00
$\delta = 32, h = 32$	0.61	0.75	2.30	1.51	4.05	9.50	0.61	1.00	2.15	1.10	2.55	6.00
$\delta = 64, h = 64$	0.41	0.50	1.45	1.01	3.95	6.00	0.48	1.00	2.45	0.73	2.85	6.00

and rendering. $\mathrm{FAR}_{0.1}$ and FAR_0 refer to the case when the false acceptance rate (FAR) is 0.1 percent and 0, and EER means the equal error rate. For simplicity, we term the artificial fingerprints generated with block based minutiae polarity and function based minutiae polarity as block based and function based artificial fingerprints, respectively. Fig. 8 further gives the distributions of the genuine and imposter matching scores for these two types of artificial fingerprints with $\delta = 64$ and $h = 64$. The scores are well separated especially for the artificial minutiae templates.

It can be seen from Table 1 that the function based artificial fingerprints have a similar performance when compared with the block based ones in the feature domain, but they perform better in the image domain. For block based artificial fingerprints, larger δ achieves higher accuracy. The block with the size 64 (i.e., $\delta = 64$) performs the best, with false rejection rate (FRR) of 1.45 percent (at FAR_0) in the feature domain and FRR of 6.00 percent (at FAR_0) in the image domain. The reason is that larger δ provides larger blocks, which is more robust for the variations among different impressions of the same finger. For function based artificial fingerprints, the accuracy does not vary much with different h. The number of kernels of 16 (i.e., $h = 16$) performs the best, with FRR of 1.50 percent (at FAR_0) in the feature domain and FRR of 6.00 percent (at FAR_0) in the image domain. The recognition accuracy in the image domain is consistently lower compared with that in the feature domain. This is due to the imperfection of artificial fingerprint image generation, where a few spurious minutiae points might be produced. In the following discussions, we only use the artificial minutiae templates for evaluation unless otherwise stated.



(a) (b)

(c) (d)

Fig. 8. The distributions of the genuine and imposter matching scores for (a) block based artificial minutiae templates, (b) function based artificial minutiae templates, (c) block based artificial fingerprint images, and (d) function based artificial fingerprint images. Both δ and h are set as 64.

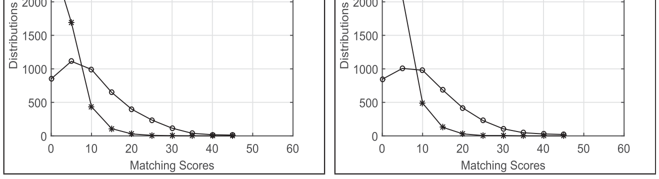
3.3 Attacks

In case the user specific key is stolen, unauthorized people may try to enter the system using the stolen key and his own fingerprint. Let's assume all the 100 fingers share the same key. The first impression of each finger is used to generate a gallery. Each gallery is matched against the other 99 galleries, producing $100 \times 99/2 = 4950$ imposter matches. The successful attack rates of these imposter matches are given in Table 2, where the decision thresholds are the same as that computed in the previous section. In addition, we show in Fig. 9 the distributions of the imposter matching scores with the same and different keys, where a significant portion of the distributions is overlapped.

We can see from Table 2 that it is more difficult to attack the block based artificial fingerprints than the function based ones using the stolen key. The successful attack rate for the block based artificial fingerprints is less than 2.00 percent at $\mathrm{FAR}_{0.1}$ (when $\delta = 16$ or 32), while the corresponding successful attack rate for the function based ones is around 8.00 percent.

TABLE 2
Successful Attack Rates Using the Stolen Key Against the Artificial Fingerprint (Values in Percentage)

	block based		function based	
	$\mathrm{FAR}_{0.1}$	FAR_0	$\mathrm{FAR}_{0.1}$	FAR_0
$\delta = 16, h = 16$	1.39	0.00	6.79	0.14
$\delta = 32, h = 32$	1.66	0.00	7.97	0.18
$\delta = 64, h = 64$	4.82	0.00	5.01	0.12



(a) (b)

Fig. 9. The distributions of the imposter matching scores using the same and different keys for (a) block based artificial fingerprints, and (b) function based artificial fingerprints. Both δ and h are set as 64.

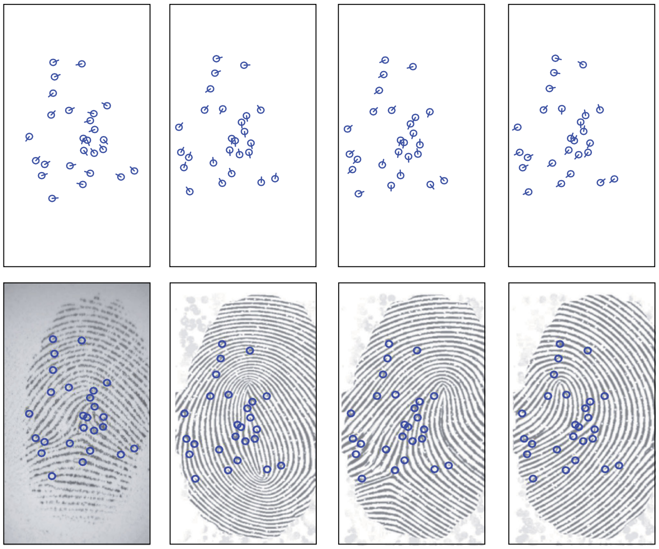


Fig. 10. The original fingerprint (the first column) and the corresponding artificial fingerprints (the second to the fourth columns) generated using different keys. Top: The original and the artificial minutiae templates; Bottom: The original and the artificial fingerprint images. The blue circles refer to the original minutiae positions, and all the images are made transparent for illustration purpose.

Overall, the successful attack rate is low even if the key is shared among all the fingers. It would be difficult for unauthorized people to enter the system using a stolen key.

3.4 Fingerprint Replacement

Once the user specific key is stolen, we can reissue the user a different key and generate a different artificial fingerprint, as shown in Fig. 10. In this section, we evaluate the difference among different artificial fingerprints generated based on the same fingerprint with different keys. We randomly choose 10 out of the 100 fingers. For each finger, we generate 100 artificial fingerprints based on the first impression using 100 different keys, each of the 100 artificial fingerprints is matched against the other 99, so we have $100 \times 99/2 = 4950$ matches. In total, we conduct $4950 \times 10 = 49500$ matches for the 10 fingers chosen. In order to show the effectiveness of the proposed artificial orientation generation scheme, we generate the following three different versions of artificial fingerprints:

(1) Version A: the artificial fingerprints generated based on the proposed artificial orientation generation scheme;

(2) Version B: the artificial fingerprints generated based on the global orientation model (i.e., the zero-pole model) only;

(3) Version C: the artificial fingerprints generated based on the real orientation extracted from another fingerprint database (FVC2002 DB1_A).

The successful match rates of these three versions of artificial fingerprints are listed in Table 3, where both δ and h are set as 64. It can be seen that the artificial fingerprints with Version A have lower successful match rate when compared with the Version B ones (over 5 percent lower at $\mathrm{FAR}_{0.1}$ for function based artificial fingerprints), while those with Version C perform the best. This means our proposed artificial orientation generation scheme achieves better diversity when compared with using the global orientation model only. The function based artificial fingerprints perform significantly better than the block based ones, with over 20 percent lower successful match rate at $\mathrm{FAR}_{0.1}$.

TABLE 3
Successful Match Rates Among the Replaced Artificial Fingerprints (Values in Percentage)

	block based		function based	
	$\mathrm{FAR}_{0.1}$	FAR_{0}	$\mathrm{FAR}_{0.1}$	FAR_{0}
Version A	43.04	27.93	20.69	9.14
Version B	46.78	32.80	25.52	12.05
Version C	39.82	26.51	19.27	8.28

3.5 Fingerprintness

We then quantitatively evaluate how real-look alike it is of the artificial fingerprint. The evaluation is based on the measure of fingerprintness which is initially proposed in [33] for differentiating fingerprint images from non-fingerprint images or altered fingerprint images. This scheme estimates the image fingerprintness using a support vector machine (SVM) classifier based on the gray-level co-occurrence matrix (GLCM) features extracted from the orientation difference map (i.e., the difference between the image orientation and the modeled orientation). The fingerprintness is defined from 0 to 1, higher fingerprintness means more likely the image is a fingerprint image. First of all, we construct a database with 800 non-fingerprint images and 800 real fingerprint images. The non-fingerprint images are collected from the ImageNet database [34], where 100 object classes are randomly selected with 8 images per class (all converted to grayscale). The real fingerprint images are chosen from the FVC2002 DB2_A database. We select 400 non-fingerprint images and 400 real fingerprint images to train an SVM classifier with radial basis function. The rest of the images serve as the test images, where the rate of detecting a non-fingerprint image as a real fingerprint image is 1.25 percent, and the rate of detecting a real fingerprint image as a non-fingerprint image is 5 percent. By using the SVM classifier, we measure the fingerprintness of 1,000 gallery artificial fingerprint images generated in Section 3.2 (with the noising and rendering applied). The average fingerprintness of these images is 0.94 with 4.6 percent of them detected as non-fingerprint images. This indicates that our artificial fingerprint images are difficult to be differentiated from the real fingerprint images.

However, for evaluating the fingerprintness of the fingerprint in the feature domain (i.e., the minutiae template in our case), the approach in [33] can not be directly applied because there is no image available to compute the image orientation. To deal with this issue, we estimate the image orientation from the minutiae templates using an existing orientation reconstruction scheme [30]. With the orientation estimated and the orientation difference map computed, the fingerprintness of a minutiae template can be measured using the SVM classifier trained before. Two sets of minutiae templates are incorporated in this test, including 400 original minutiae templates and 1,000 artificial minutiae templates. The original minutiae templates are extracted from the 400 real fingerprint images for testing, and the other set is obtained from the gallery templates generated in

TABLE 4
The Detection Rates (D_r) and Average Fingerprintness (A_i) Between the Original Minutiae Template and the Artificial Minutiae Template

Minutiae templates	Original	Artificial
D_r (%)	6.75	5.20
A_i	0.91	0.93

Section 3.2. Table 4 gives the detection results and average fingerprintness of these two sets of minutiae templates, where the detection rate refers to the percentage detected as non-fingerprints. The artificial minutiae templates perform similarly to the original minutiae templates, less than 6 percent of which are detected as non-fingerprints with average fingerprintness of 0.93. Thus, it is also difficult to differentiate the artificial minutiae templates from the real minutiae templates. In other words, the protection of fingerprint is well concealed in the artificial fingerprints.

3.6 Comparisons

We first compare our proposed scheme with the existing fingerprint privacy protection techniques which are able to preserve the topology of the fingerprint [17], [18]. Note these existing schemes require two original fingerprints to work together, which are not able to perform the fingerprint replacement. We term the protected fingerprints generated using the method in [17], [18] as the mixed fingerprint and the combined fingerprint, respectively. For fair comparison, we generate 1,000 gallery mixed/combined fingerprints in the image domain from the first impressions in the FVC2002 DB2_A database, and 2,000 probe mixed/combined fingerprints from the corresponding second and third impressions. Both the mixed fingerprint image and the combined fingerprint image are generated based on the strategies that are able to provide the best accuracy, and the VeriFinger 6.3 [19] is served as the matcher for evaluation. The comparison results are reported in Table 5, where the results of our scheme (in the image domain) are duplicated from previous sections. It can be seen that, compared with the work in [17], [18], our scheme performs significantly better in terms of accuracy and maintains higher fingerprintness. In addition, we can generate difference artificial fingerprints for the same fingerprint with different keys.

TABLE 5
Comparisons Among Different Privacy Protection Techniques with the Ability to Preserve the Fingerprint Topology

Schemes	Proposed block based ($\delta = 64$)	Proposed function based ($h = 64$)	Ross et al. [17]	Li et al. [18]
Accuracy (at FAR$_0$)	6.00%	6.00%	16.70%	18.50%
Fingerprintness	0.94 (with various settings of δ and h)		0.81	0.90

Next, we compare the accuracy of the proposed scheme with other fingerprint privacy protection algorithms [5], [6], [7], [8], [35], [36] which are not able to preserve the topology of the fingerprint. For fair comparison, we adopt two common protocols (the 1vs1 and the standard FVC protocol [5], [8]) and evaluate our scheme on five different databases including FVC2002 DB1_A, FVC2002 DB2_A, FVC2002 DB3_A, FVC2002 DB4_A and FVC2006 DB2_A. Each of the FVC2002 databases contains 800 fingerprint images from 100 fingers with 8 impressions per finger. The FVC2006 DB2_A database contains 1,680 fingerprint images from 140 fingers with 12 impressions per finger. In the 1vs1 protocol, the first impression of each finger is matched against the corresponding second impression to compute the FRR, which is then matched against the first impressions of the other fingers to compute the FAR. In the standard FVC protocol, each impression of each finger is matched against the other impressions of the same finger to compute the FRR, while the computation of the FAR is the same as that of the 1vs1 protocol.

For the four FVC2002 databases, only the 1vs1 protocol is conducted because some of the impressions (except the first two impressions) are partial without any core points. There are $100 \times 1 = 100$ genuine matches to compute the FRR and $100 \times 99/2 = 4950$ imposter matches to compute the FAR. For the FVC2006 database, both the two protocols are conducted as this database contains no partial fingerprint images. There are $140 \times 1 = 140$ and $(12 \times 11/2) \times 140 = 9240$ genuine matches to compute the FRR for the 1vs1 protocol and the standard FVC protocol, respectively. The number of imposter matches is $140 \times 139/2 = 9730$ for both protocols.

Table 6 shows the accuracy of various fingerprint privacy protection techniques using the 1vs1 protocol, where the results of our scheme is evaluated on the feature domain. The “same key” means that all the artificial fingerprints are generate using the same key, which is to evaluate the effect of the user specific key on the accuracy. It can be see that using the same key slightly reduces the accuracy of the proposed scheme. Compared with the most recent work [8], our scheme (block based) performs the same or better in all the databases except in FVC2002 DB1_A. Compared with the work in [5], our scheme performs better in FVC2002 DB2_A, FVC2002 DB3_A and FVC2002 DB4_A. The comparisons among different techniques using the standard FVC protocol on the FVC2006 database is given in Table 7. Our scheme (block based) performs better than all the other schemes except the work in [5] and [6]. It should be noted that both the work in [5] and [6] adopt FVC2006 DB2_B for training to compute the protected template, which leads to the best accuracy among all the techniques on the FVC2006 database using both protocols.

On the other hand, the work in [6] randomly permutates the P_MCC template obtained from the work in [5] using an user specific key. The accuracy of the “same key” case for [6] should be similar to the accuracy of the work in [5] (see Table 7). From Table 6, we can see that the proposed scheme (block based and “same key”) performs better than the work in [5] on the FVC2002 DB2_A, FVC2002 DB3_A and FVC2002 DB4_A using the 1vs1 protocol. This indicates that, in the “same key” case, the proposed scheme (block based) should also perform better than the work in [6] on these databases using the 1vs1 protocol.

4 Irreversibility

We analyze the irreversibility of the artificial fingerprint in terms of the possibility to recover the original fingerprint

TABLE 6
Comparisons of the Accuracy of the Proposed Scheme and Other Privacy Protection Techniques which are Not Able to Preserve the Fingerprint Topology Using the 1vs1 Protocol (Percentage Values)

Schemes	FVC2002 DB1_A			FVC2002 DB2_A			FVC2002 DB3_A			FVC2002 DB4_A			FVC2006 DB2_A		
	EER	$FAR_{0.1}$	FAR_0	EER	$FAR_{0.1}$	FAR_0	EER	$FAR_{0.1}$	FAR_0	EER	$FAR_{0.1}$	FAR_0	EER	$FAR_{0.1}$	FAR_0
Nandakumar et al. [7]	-	-	14.00	-	-	-	-	-		-	-	-	-	-	-
Yang et al. [35]	3.38	-	-	0.59	-	-	9.80	-		16.52	-	-	-	-	-
Yang et al. [36]	-	-	4.00	1.02	-	2.00	8.63	-	-	-	-	-	4.83	-	-
Li et al. [8]	0.83	-	2.00	**0.00**	-	**0.00**	4.77	-	-	10.05	-	-	2.74	-	16.43
Ferrara et al. [5] P_MCC_{64}[1]	**0.00**	**0.00**	**0.00**	0.37	1.00	2.00	4.94	9.00	12.00	7.00	19.00	24.00	**0.69**	**2.86**	**2.86**
Proposed block based $\delta = 64$	1.00	1.00	1.00	**0.00**	**0.00**	**0.00**	**1.00**	**1.00**	**5.00**	**6.01**	**17.00**	**21.00**	2.38	4.29	5.71
Proposed block based $\delta = 64$ (same key)	1.00	1.00	1.00	**0.00**	**0.00**	**0.00**	1.15	2.00	**5.00**	6.05	18.00	22.00	3.49	5.71	7.08
Proposed function based $h = 64$	0.47	1.00	2.00	**0.00**	**0.00**	**0.00**	3.01	7.00	12.00	6.14	23.00	27.00	3.20	8.57	9.29
Proposed function based $h = 64$ (same key)	1.00	1.00	2.00	**0.00**	**0.00**	**0.00**	3.30	11.00	15.00	6.52	24.00	30.00	3.50	9.29	10.00

[1] *The recommended configuration as indicated in [5].*

(minutiae) from the artificial fingerprint. Let's assume the attacker already knows this is an artificial fingerprint as well as the corresponding generation process (i.e., the key). In order to recover the fingerprint minutiae, he needs to guess the minutiae direction of each minutiae point. Next, we discuss the possibility to recover the original fingerprint (from the artificial fingerprint) in terms of full-leakage irreversibility and authorized-leakage irreversibility [37].

4.1 Full-Leakage Irreversibility

Full-leakage irreversibility refers to the difficulty (possibility) to exactly recover the original fingerprint from the protected fingerprint (i.e., the artificial fingerprint in our case). In [38], the authors indicate that the minutiae directions can be estimated based on the ground truth orientation of different fingerprint classes. Once the ground truth orientation (within $[0, \pi)$) is established, the direction of each minutiae point (within $[0, 2\pi)$) could be randomly selected from one of the two directions corresponding to the orientation of the minutiae point (i.e., O or $O + \pi$). Such a direction selection procedure may not be the best strategy for the recovery. The reason is that the minutiae have the tendency to form clusters, and neighboring minutiae points tend to have similar directions of either O or $O + \pi$ [39]. Therefore, to conduct the recovery, the attacker could estimate the ground truth orientation and cluster the minutiae positions. Then, the recovery can be performed cluster by cluster based on the similarity among the neighboring minutiae directions.

TABLE 7
Comparisons of the Accuracy of the Proposed Scheme and Other Privacy Protection Techniques Which Are Not Able to Preserve the Fingerprint Topology Using the Standard FVC Protocol on the FVC2006 DB2_a Database (Percentage Values)

Schemes	EER	$FAR_{0.1}$	FAR_0
Yang et al. [36]	3.07	-	-
Li et al. [8]	1.59	-	5.78
Ferrara et al. [5] P_MCC_{64}[1]	0.32	0.47	1.07
Ferrara et al. [6] $2P\_MCC_{64,64}$	**0.10**	**0.10**	**0.20**
Ferrara et al. [6] $2P\_MCC_{64,64}$ (same key)	0.30	0.50	1.00
Proposed block based $\delta = 64$	1.38	2.43	3.47
Proposed block based $\delta = 64$ (same key)	1.96	2.84	5.17
Proposed function based $h = 64$	1.67	3.73	5.25
Proposed function based $h = 64$ (same key)	2.12	4.22	5.84

[1] *The recommended configuration as indicated in [5].*

The establishment of the ground truth orientation requires the following two pieces of information: i) the original fingerprint class, and ii) the location of the primary core of the original fingerprint. Let's denote the probability to correctly guess the original fingerprint class as ξ_o, which can be estimated based on the prior probabilities of the five major fingerprint classes mentioned before [21], i.e., 0.037 for arch, 0.029 for tented arch, 0.317 for right loop, 0.338 for left loop, and 0.279 for whorl. During the artificial fingerprint generation, the primary core of the original fingerprint is initially translated to the center due to the minutiae position alignment (see Section 2.1), which might be further translated because of the global translation (see Section 2.4). Therefore, the primary core should be located around the image center. Assume it is located at the center, the probability to establish the ground truth orientation is roughly ξ_o for a certain fingerprint class.

As suggested in [39], the distribution of the minutiae positions can be estimated using the Gaussian mixture model (GMM), where one component represents the distribution of the minutiae positions from one cluster, and the clusters are independently distributed. Thus, the attacker can cluster the minutiae positions based on the GMM and form a set of C clusters. For the cth cluster, let's denote the number of the minutiae positions and those with original directions of O as

n_c and o_c, respectively. If all the original directions are similar, o_c should be equal to 0 or n_c. When $o_c = i$ ($i \in [0, n_c]$), the probability for the attacker to perform the full recovery is

$$\mathbb{P}_c^i = \frac{P(o_c = i)}{\binom{n_c}{i}}, \tag{24}$$

where $P(o_c = i)$ refers to the prior probability of a cluster with $o_c = i$. The value of $P(o_c = i)$ can be roughly estimated based on a set of clusters extracted from the original fingerprints of the same database. Since the events of a cluster with $o_c = i$ are mutually disjoint with prior probability of $P(o_c = i)$, the probability to fully recover the cth cluster (with any possible value of o_c) is estimated as

$$\mathbb{P}_c = \sum_{i=0}^{n_c} P(o_c = i) \cdot \mathbb{P}_c^i. \tag{25}$$

By considering all the C clusters, the probability to fully recover the original minutiae (of a certain fingerprint class) is given by

$$\mathbb{P}_{rf} = \xi_o \cdot \prod_{c=1}^{C} \mathbb{P}_c. \tag{26}$$

Next, we empirically assess the full-leakage irreversibility using the 100 gallery artificial fingerprints generated in Section 3.2. We randomly split these fingerprints into two parts, each of which contains 50 fingerprints. One part (say Part I) is used to roughly estimate $P(o_c = i)$. Since n_c varies among different clusters, we compute $P(o_c = i)$ based on the frequency in the ith bin of the histogram of o_c/n_c. The other part (say Part II) is used to estimate $\mathbb{P}_{rf}$ given $P(o_c = i)$. For each fingerprint, we vary the value of C from 1 to 6 to fit a set of 6 GMMs using the Expectation Maximization algorithm [40], where the Akaike Information Criterion [41] is adopted for the model selection (i.e., selecting the optimal C). The difficulty to fully recover each of the original fingerprints in Part II is computed as $log_2 \mathbb{P}_{rf}$ bits with an average of 27.6 bits, where the average number of minutiae points is $n = 45.82$. This is around 18 bits lower than using the strategy of choosing one of the two directions independently for each minutiae position (the difficulty is roughly $n = 45.82$ bits in such a case).

We would like to point out that knowing the key does not provide any advantage for the recovery. Since the artificial orientation and minutiae polarities (which determine the artificial minutiae directions) are generated purely based on the key, there is no relationship between the original minutiae directions and the artificial fingerprint (or the key). This is to say, the information of the original minutiae directions is completely discarded in the artificial fingerprint.

4.2 Authorized-Leakage Irreversibility

Authorized-leakage irreversibility means the difficulty (possibility) to recover a fingerprint that would "match" the unprotected fingerprint in a traditional fingerprint recognition system [37]. In the following discussions, we term such recovery as the authorized-leakage recovery. We assume the attacker has correctly guessed the original fingerprint class with the possibility ξ_o. Given an artificial fingerprint and the corresponding ground truth orientation of the original fingerprint class, the task of the authorized-leakage recovery becomes to find a set of minutiae polarities and generate a recovered fingerprint that would "match" the unprotected fingerprint.

TABLE 8
Successful Attack Rates of the Authorized-Leakage Recovery (Values in Percentage)

	$\text{FAR}_{0.1}$	FAR_0
Against the first impression	30.08	7.31
Against the second impression	16.29	3.11

We empirically assess the difficulty of the authorized-leakage recovery using the first two impressions in the FVC2002 DB2_A database. Given a stolen artificial fingerprint generated from one of the first impressions, we recover 100 fingerprints based on the aligned minutiae positions, the ground truth orientation, and a set of minutiae polarities of 0 or 1. For the minutiae positions belonging to the same cluster (please refer to Section 4.1 for the clustering), we randomly select i minutiae positions and assign their polarities with 0 (corresponding to the directions that are recovered as O) with probability of $P(o_c = i)$, while the polarities of the rest are assigned with 1. Each recovered fingerprint is matched against the corresponding first (or the second) impression, which results in 10,000 authorized-leakage recovery attacks. The successful attack rates of these recovered fingerprints are given in Table 8, where $\text{FAR}_{0.1}$ and FAR_0 refer to the operating points (thresholds) of the traditional fingerprint recognition system with false acceptance rates of 0.1 percent and 0, respectively. It can be seen that it is easier to attack a traditional fingerprint recognition system that stores the first impression (i.e., the original fingerprint of the same impression) based on the artificial fingerprint, the successful rates of which are 30.08 percent at $\text{FAR}_{0.1}$ and 7.31 percent at FAR_0. To attack the same system that stores a different impression of the same finger, the successful rates are 16.29 percent at $\text{FAR}_{0.1}$ and 3.11 percent at FAR_0. Let's denote the successful attack rate as ξ_s at a certain operating point, the possibility to perform a successful authorized-leakage recovery can be estimated as

$$\mathbb{P}_{ra} = \xi_o \xi_s. \tag{27}$$

The results in Table 8 indicate that it is much more difficult for an attacker to break a traditional fingerprint recognition system by using the artificial fingerprint compared with using the original fingerprint, especially when the FAR of the system is low. Meanwhile, it is visually difficult for the attacker to differentiate an artificial fingerprint from the real fingerprints. The privacy of the user is thus protected by using the artificial fingerprint. In real-world applications, we can build a fingerprint recognition system by storing the artificial fingerprint generated from a new capture of the user's finger. If the artificial fingerprint is stolen, the attacker may treat it as an original fingerprint and use it to generate a fake finger (or conduct a direct data injection) to attack some other systems (with the same finger enrolled) that still use the original fingerprint for authentication. In such a case, it would be difficult for the attacker to launch a successful attack, with around 0.9 percent successful attack rate (i.e., $\mathbb{P}_{ra} \approx 0.009$) at FAR_0 for loop or whorl fingers, where the corresponding ξ_o is around 0.3 (see Section 4.1) and the

TABLE 9
Performance of the Artificial Fingerprint Recognition with Transformed Minutiae Positions Using the 1vs1 Protocol (Values in Percentage)

	block based			function based		
	EER	$\mathrm{FAR}_{0.1}$	FAR_0	EER	$\mathrm{FAR}_{0.1}$	FAR_0
$\delta = 16, h = 16$	1.00	1.00	2.00	0.25	1.00	1.00
$\delta = 32, h = 32$	1.00	1.00	2.00	1.00	1.00	1.00
$\delta = 64, h = 64$	0.00	0.00	1.00	0.00	0.00	3.00

successful attack rate (with the ground truth orientation known) is 3.11 percent (see Table 8).

Obviously, authorized-leakage recovery is a much easier task compared with the full-leakage recovery (i.e., exactly recovery). As a matter of fact, the authors in [38] show that, even if the attacker knows nothing about the original fingerprint, he can launch a successful attack by trying 182 times on average at $\mathrm{FAR}_{0.1}$.

5 Discussions

The artificial fingerprint exposes the original minutiae positions. This is a known issue in the existing techniques with the ability to preserve the fingerprint topology [17], [18]. In our case, thanks to the incorporation of the key, it can be addressed by transforming the original minutiae positions spatially before the artificial fingerprint generation. People have proposed various schemes that are able to transform the minutiae spatially [3], [4]. Here, we apply the changing function approach proposed in [4] for the transformation. Given an aligned minutiae position $\mathbf{m}_i = (x'_i, y'_i)$, its movement $(\Delta x_i, \Delta y_i)$ is computed as

$$\begin{aligned}\Delta x_i &= L_1(v_i)\cos(\mathrm{mod}(\theta'_i, \pi) + L_2(v_i))\\ \Delta y_i &= L_1(v_i)\sin(\mathrm{mod}(\theta'_i, \pi) + L_2(v_i)),\end{aligned} \tag{28}$$

where mod is the modulo operation, θ'_i is the aligned minutia direction, $L_1(\cdot)$ and $L_2(\cdot)$ are two changing functions, and v_i is the invariant value computed by

$$v_i = \mathbf{f}_i \circ \mathbf{u}_{pin}, \tag{29}$$

where $\circ$ denotes the inner product between two vectors, $\mathbf{f}_i$ is a normalized vector computed based on θ'_i and the original orientation, and $\mathbf{u}_{pin}$ is a normalized random vector generated by the user specific key. Details of the construction of the changing functions as well as the computation of the invariant value can be found in [4]. In our implementation, we take advantage of our artificial orientation (which is always the same for the same key) and adopt it to compute $\mathbf{f}_i$ instead of using the original orientation.

The performance of the artificial fingerprint recognition (with the transformed minutiae positions) on FVC2002 DB2_A using the 1vs1 protocol is given in Table 9. Compared with the results in Table 6, the transformation slightly reduces the accuracy of the artificial fingerprint recognition, which is still acceptable with more protection offered.

6 Conclusions

In this paper, artificial fingerprints are proposed in order to protect the privacy of fingerprints. The artificial fingerprint is generated based on the minutiae positions extracted from the original fingerprint as well as the artificial orientation and minutiae polarities computed from an user specific key. We propose to generate real-look alike and diverse artificial orientation by a combined orientation model guided by a set of parameters with simple constraints. Then, a block based approach and a function based approach are proposed for generating the minutiae polarities, the artificial fingerprint in the feature domain (i.e., the artificial minutiae template) can be generated accordingly. By using the AM-FM fingerprint model, we can also generate an artificial fingerprint image. The proposed artificial fingerprint is real-look alike, which can not be easily identified from the real fingerprints. This reduces the chance of the artificial fingerprint to get attacked. If it is stolen, only a partial information of the original fingerprint minutiae is exposed to the attacker. Furthermore, we can assign different keys to generate different artificial fingerprints for the replacement. The experimental results indicate that the artificial fingerprint can be recognized at high accuracy, and they are visually similar to the real fingerprints.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61602294, Grant U1636206, Grant 61525203, Grant 61472235, Grant 61572308, Grant U1536109, Grant 61373151, Grant U1536108, Grant U1736213 and Grant 61572309, in part by the Shanghai Sailing Program under Grant 16YF1404100, in part by the Young Oriental Scholar under Shanghai Institutions of Higher Education, in part by the Shanghai Dawn Scholar Plan under Grant 14SG36 and in part by the Shanghai Excellent Academic Leader Plan under Grant 16XD1401200.

References

[1] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: Two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognit.*, vol. 37, no. 11, pp. 2245–2255, 2004.

[2] 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.

[3] 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. 561–572, Apr. 2007.

[4] C. Lee, J.-Y. Choi, K.-A. Toh, S. Lee, and J. Kim, "Alignment-free cancelable fingerprint templates based on local minutiae information," *IEEE Trans. Syst. Man Cybern. Part B: Cybern.*, vol. 37, no. 4, pp. 980–992, Aug. 2007.

[5] M. Ferrara, D. Maltoni, and R. Cappelli, "Noninvertible minutia cylinder-code representation," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 6, pp. 1727–1737, Dec. 2012.

[6] M. Ferrara, D. Maltoni, and R. Cappelli, "A two-factor protection scheme for MCC fingerprint templates," in *Proc. Int. Conf. Biometrics Special Interest Group*, 2014, pp. 1–8.

[7] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," *IEEE Trans. Inf. Forensics Secur.*, vol. 2, no. 4, pp. 744–757, Dec. 2007.

[8] C. Li and J. Hu, "A security-enhanced alignment-free fuzzy vault-based fingerprint cryptosystem using pair-polar minutiae structures," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 3, pp. 543–555, Mar. 2016.

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

[10] N. K. Ratha, M. A. Figueroa-Villanueva, J. H. Connell, and R. M. Bolle, "A secure protocol for data hiding in compressed fingerprint images," in *Proc. Int. Workshop Biometric Authentication*, 2004, pp. 205–216.

[11] Q. Feng, F. Su, A. Cai, and F. Zhao, "Cracking cancelable fingerprint template of ratha," in *Proc. Int. Symp. Comput. Sci. Comput. Technol.*, 2008, pp. 572–575.

[12] S. W. Shin, M.-K. Lee, D. Moon, and K. Moon, "Dictionary attack on functional transform-based cancelable fingerprint templates," *ETRI J.*, vol. 31, no. 5, pp. 628–630, 2008.

[13] A. Nagar and A. K. Jain, "On the security of non-invertible fingerprint template transforms," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2009, pp. 81–85.

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

[15] K. Simoens, C. Chang, and B. Preneel, "Reversing protected minutiae vicinities," in *Proc. IEEE Int. Conf. Biometrics: Theory Appl. Syst.*, 2010, pp. 1–8.

[16] W. J. Scheirer and T. E. Boult, "Cracking fuzzy vaults and biometric encryption," in *Proc. Biometrics Symp.*, 2007, pp. 1–6.

[17] A. Othman and A. Ross, "On mixing fingerprints," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 1, pp. 260–267, Jan. 2013.

[18] S. Li and A. C. Kot, "Fingerprint combination for privacy protection," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 2, pp. 350–360, Feb. 2013.

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

[20] S. Chikkerur and N. Ratha, "Impact of singular point detection on fingerprint matching performance," in *Proc. IEEE Workshop Autom. Identification Adv. Technol.*, 2005, pp. 207–212.

[21] 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.

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

[23] B. Sherlock and D. Monro, "A model for interpreting fingerprint topology," *Pattern Recognit.*, vol. 26, no. 7, pp. 1047–1055, 1993.

[24] J. Zhou and J. Gu, "A model-based method for the computation of fingerprints' orientation field," *IEEE Trans. Image Process.*, vol. 13, no. 6, pp. 821–835, Jun. 2004.

[25] J. Feng, J. Zhou, and A. K. Jain, "Orientation field estimation for latent fingerprint enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 925–940, Apr. 2013.

[26] 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.

[27] J. Zhou, J. Gu, and D. Zhang, "Singular points analysis in fingerprints based on topological structure and orientation field," in *Proc. Int. Conf. Biometrics*, 2007, pp. 261–270.

[28] R. Cappelli and D. Maltoni, "On the spatial distribution of fingerprint singularities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 742–448, Apr. 2009.

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

[30] 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.

[31] S. Li and A. C. Kot, "Attack using reconstructed fingerprint," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 29 Nov.–2 Dec. 2011, pp. 1–6.

[32] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd ed. Berlin, Germany: Springer-Verlag, 2009.

[33] S. Yoon and A. K. Jain, "Is there a fingerprint pattern in the image?" in *Proc. Int. Conf. Biometrics*, 2013, pp. 1–8.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[35] W. Yang, J. Hu, S. Wang, and M. Stojmenovic, "An alignment-free fingerprint bio-cryptosystem based on modified Voronoi neighbor structures," *Pattern Recognit.*, vol. 47, no. 3, pp. 1309–1320, 2014.

[36] W. Yang, J. Hu, and S. Wang, "A delaunay quadrangle-based fingerprint authentication system with template protection using topology code for local registration and security enhancement," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 7, pp. 1179–1192, Jul. 2014.

[37] K. Simoens, B. Yang, X. Zhou, and F. Beato, "Criteria towards metrics for benchmarking template protection algorithms," in *Proc. Int. Conf. Biometrics*, 2012, pp. 498–505.

[38] U. Ulugdag, "Secure biometric systems," PhD Thesis, Dept. Comput. Sci. Eng., Michigan State Univ., East Lansing, MI, USA, 2006.

[39] Y. Zhu, S. C. Dass, and A. K. Jain, "Statistical models for assessing the individuality of fingerprints," *IEEE Trans. Inf. Forensics Secur.*, vol. 2, no. 3, pp. 391–401, Sep. 2007.

[40] G. J. Mclachlan and T. Krishnan, *The EM Algorithm and Extensions*, 2nd ed. Hoboken, NJ, USA: Wiley, 2007.

[41] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. 19, no. 6, pp. 716–723, Dec. 1974.

Sheng Li received the PhD degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2013. From 2013 to 2016, he was a research fellow with Rapid Rich Object Search (ROSE) Lab, Nanyang Technological University. He is currently with the faculty of the School of Computer Science, Fudan University, China. His research interests include biometric template protection, pattern recognition, multimedia forensics, and security. He is the recipient of the IEEE WIFS Best Student Paper Silver Award.

Xinpeng Zhang received the BS degree in computational mathematics from Jilin University, China, in 1995, and the ME and PhD degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a professor. He is also with the faculty of the School of Computer Science, Fudan University. He was with the State University of New York, Binghamton as a visiting scholar from January 2010 to January 2011, and Konstanz University as an experienced researcher sponsored by the Alexander von Humboldt Foundation from March 2011 to May 2012. He is an associate editor of the *IEEE Transactions on Information Forensics and Security*. His research interests include multimedia security, image processing, and digital forensics. He has published more than 200 papers in these areas.

Zhenxing Qian (M'12) received the BS and PhD degrees from the University of Science and Technology of China (USTC), in 2003 and 2007, respectively. He is currently a professor in the School of Computer Science, Fudan University, Shanghai, China. His research interests include data hiding and multimedia security. He is a member of the IEEE.

Guorui Feng received the BS and MS degrees in computational mathematic from Jilin University, China, in 1998 and 2001 respectively, and the PhD degree in electronic engineering from Shanghai Jiaotong University, China, in 2005. From January 2006 to December 2006, he was an assistant professor with the East China Normal University, China. During 2007, he was a research fellow with the Nanyang Technological University, Singapore. Now he is with the School of Communication and Information Engineering, Shanghai University, China. His current research interests include image processing, image analysis, and computational intelligence.

Yanli Ren received the MS degree in applied mathematics from Shaanxi Normal University, China, in 2005 and the PhD degree in computer science and technology from Shanghai Jiao Tong University, China, in 2009. She is an associate professor with the School of Communication and Information Engineering, Shanghai University, China. Her research interests include applied cryptography, secure outsourcing computing, and network security.