IEEE Transactions on Image Processing Volume: 28 , Issue: 3 , March 2019

Towards Construction Based Data Hiding: From Secrets to Fingerprint Images

Sheng Li and Xinpeng Zhang

***Abstract*—Data hiding usually involves the alteration of a cover signal for embedding a secret message. In this paper, we propose a construction based data hiding technique which transforms a secret message into a fingerprint image directly. Unlike the conventional data hiding techniques, this scheme does not need any cover signals to participate. Instead, it generates the fingerprint image based on a piece of hologram phase constructed from the secret message. The hologram phase consists of the spiral phase and the continuous phase. Firstly, we propose to map the secret message to a polynomial and encode it into a set of points with different polarities, from which the spiral phase is computed and constructed. Then, we construct the continuous phase by decomposing a fingerprint image synthetically generated. The spiral phase and the continuous phase are combined to form the hologram phase. This is eventually used to construct a fingerprint image in a common form such as a grayscale fingerprint image, a binary fingerprint image, or a thinned fingerprint image. The secret message can be extracted by detecting the encoded points in the constructed fingerprint. We conduct the experiments by constructing fingerprint images with ordinary sizes, the results show that the secret message can be extracted accurately. It is also difficult to detect the existence of secret message from the constructed fingerprint images.**

***Index Terms*—Data hiding, fingerprint, construction**

I. Introduction

Data hiding is a technique of embedding a secret message into a cover signal by subtly altering selected locations. It is widely applied in authentication, secure communication and copyright protection. Generally speaking, the cover signal could be any meaningful digital signal including the digital audio/image/video [1]–[7], text [8], and even the 3D meshes [9], [10]. Among various data hiding techniques, image based data hiding is the most popular, where the cover signal is a digital image (i.e., the cover image) such as a natural image [2], [5], [6], a medical image [11], [12] or a biometric image [13]–[15]. Image based data hiding can be developed for images in different forms, including color/grayscale images [2], [5], [6], [13] and binary images [3], [4], [14], [15].

Most of the existing image based data hiding techniques require a cover image to participate. The pixels of the cover image will be modified to host the secret message, which inevitably causes distortions visually or statistically. Thus, it is possible to develop steganalysis tools to reveal the existence of the secret message in the stego-images (i.e., the images with hidden data) [16]–[19].

In recent years, a few data hiding techniques have been developed without the incorporation of cover images [20]–[22]. Instead of altering the pixels, these techniques perform data embedding by constructing a stego-image directly from the secret message. Meanwhile, the secret message can be extracted (or decoded) from the constructed stego-image. Such construction based data hiding techniques do not involve the alteration of pixels during the data embedding, which creates challenges for traditional steganalysis tools.

All the existing construction based data hiding techniques use texture synthesis for the construction of stego-images. This concept is initially proposed by Otori and Kuriyama [20], where the secret message is encoded into a dotted pattern. The data embedding is conducted such that the local binary patterns of all the blocks in the dotted pattern represent the secret message. The texture image with hidden data is then synthesized by painting the dotted pattern. This scheme is robust to image recapturing, however, it offers relatively low data hiding capacity. In [21], the authors propose a patch-based texture synthesis which is message-oriented. This method distributes the source texture into a composition image reversibly. The data embedding is performed by pasting proper source patches on the composition image, where the choice of source patches depends on the data to be hidden. Compared with the work in [20], this approach achieves higher data hiding capacity, but it offers no robustness when there is any change on the image content. A marbling based data hiding approach is proposed in [22]. In this scheme, the secret message is printed on a background image and deformed into different marbling textures using reversible functions. Similar to the work in [20], this scheme offers limited capacity with robustness against printing and scanning.

In this paper, instead of constructing texture images, we propose to construct fingerprint images directly from the secret messages. The reason behind is the popularity of biometrics recognition systems nowadays, and fingerprint biometrics is the most popular. A typical fingerprint recognition system requires a communication channel to transmit the fingerprint image which captures all the key features of the fingerprint [23]. There are three common forms of fingerprint images including the grayscale fingerprint image, the binary fingerprint image and the thinned fingerprint image [24], as shown in Fig. 1. The last two forms are obtained by fingerprint binarization

This work was supported in part by the National Natural Science Foundation of China under Grant 61602294, Grant U1636206, Grant 61525203, Grant 61472235, 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.

Sheng Li and Xinpeng Zhang are with the Shanghai Institute of Intelligent Electronics and Systems, School of Computer Science, Fudan University, P.R. China, 201203 (e-mail: lisheng@fudan.edu.cn; zhangxinpeng@fudan.edu.cn). Corresponding author: Sheng Li.

Fig. 1. Fingerprint images that are commonly used in fingerprint recognition systems. (a) The grayscale fingerprint image, (b) the binary fingerprint image (obtained by binarization of (a)), and (c) the thinned fingerprint image (obtained by morphological thinning of (b)).

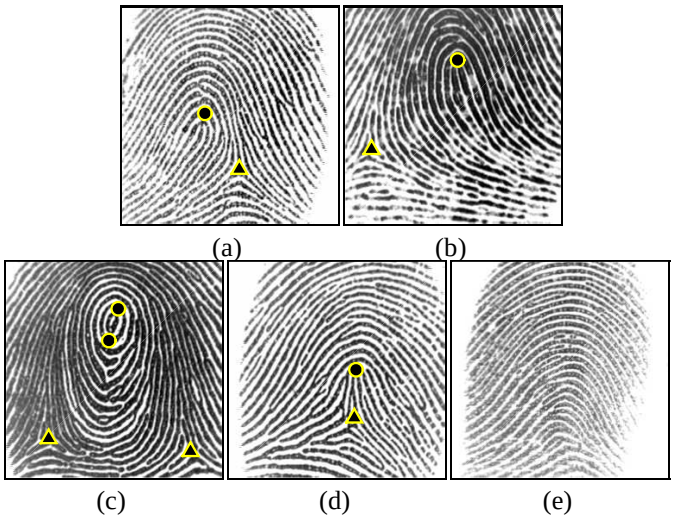


Fig. 2. Fingerprints belong to different classes. (a) Left loop, (b) right loop, (c) whorl, (d) tented arch, and (e) arch. The solid circles and triangles refer to the cores and deltas, respectively.

and thinning on the grayscale fingerprint images, which are popular fingerprint image operations during the transmission. Researchers have devoted efforts to developing data hiding techniques for these fingerprint images for various applications [13]–[15], [25]. However, these approaches require a cover fingerprint image to work and they are not robust against fingerprint binarization and thinning. The secret can not be extracted at all once the stego-images are binarized or thinned. Thus, it is necessary to develop data hiding techniques that can construct fingerprint images directly from the secret messages. In addition, they should also be robust against various attacks including popular fingerprint image operations as well as common image operations.

The proposed scheme is based on the fingerprint hologram phase which consists of the spiral phase and the continuous phase. The spiral phase corresponds to the fingerprint minutiae (i.e., the ridge endings and bifurcations). We propose to map the secret message to a polynomial and encode it into a set of two dimensional points to mimic the fingerprint minutiae, so as to construct the spiral phase. The continuous phase is related to fingerprint orientation and frequency, the construction of which is conducted by decomposing the hologram phase of a synthetic fingerprint image. We then combine the spiral phase and continuous phase to form the hologram phase, based on which we generate a fingerprint image in any of the three common forms mentioned before. In data extraction, we detect the encoded points in the fingerprint image and reconstruct the polynomial. The experimental results demonstrate the high data extraction accuracy and robustness of our proposed scheme. Furthermore, the existence of secret message is difficult to be detected using the existing steganalysis tools.

The organization of the paper is as follows. Section II introduces the background of fingerprint synthesis. Section III gives a brief review on the phase representation of the fingerprint and the corresponding phase decomposition. Section IV and Section V introduce the fingerprint image construction and the data extraction, respectively. Section VI presents the experimental results, followed by some discussions in Section VII. Our conclusions and future work are given in the last section.

II. Background: Fingerprint Synthesis

In general, there are five major fingerprint classes including left loop, right loop, whorl, tented arch and arch [26]. Fingerprints belong to different classes have different distributions of singularities (i.e., cores and deltas) as shown in Fig. 2. The aim of fingerprint synthesis is to generate a synthetic fingerprint image of a certain class based on a set of parameters. In literature, people have proposed various techniques for synthetic fingerprint generation, which are mainly designed to build large fingerprint databases or understand the rules involved in the biological process to form fingerprints.

Cappelli *et al.* [27] propose a synthetic fingerprint image generation scheme by iteratively applying Gabor filtering on a seed image. The ridge pattern of the fingerprint gradually grows during the filtering. Similarly, in [28], the ridge pattern is iteratively generated using filters of binary masks instead of Gabor filters. Besides these filtering based approaches, fingerprint can be synthesized according to the biological process of fingerprint formation. In [29], the authors argue that fingerprints are formed due to the buckling process in the basal cell layer of the epidermis, where the ridge pattern is generated by the solution of von Karman equation. Fingerprint formation can also be treated as a general biological pattern formation problem which could be solved using the Turing's (or reaction-diffusion) model [30]–[32]. The Turing's model offers the flexibility in generating various patterns we observe in nature. However, it usually requires the model parameters to be carefully selected [33]. Some other techniques are developed with the ability to restore a few missing areas in a fingerprint image [34], [35]. These approaches work well for fingerprint restoration or enhancement. However, they cannot be used to synthesize a complete new fingerprint image.

Despite the variety of synthetic fingerprint generation approaches, none of them are designed with the ability to conceal secret messages during the fingerprint construction. The data (parameters) incorporated in the fingerprint synthesis are with limited entropy and difficult to be correctly extracted. In this

paper, we take advantage of the phase representation of the fingerprint and encode the secret message as a set of spirals with some redundancy. Most of the spirals remain the same during the fingerprint construction. Such a property ensures the secret message to be correctly extracted by detecting the spirals from the constructed fingerprint image.

III. The Phase Representation of Fingerprint

As indicated in [36], the structure of a fingerprint can be represented as a hologram, i.e., a phase modulated fringe pattern. Given a grayscale fingerprint image F, the intensity of each pixel (x, y) can be modeled by

$$F(x,y) = A(x,y) + B(x,y) \cdot \cos[\psi(x,y)] + N(x,y), \quad (1)$$

where $A(x,y)$ is the offset of the intensity of the image, $B(x,y)$ is the amplitude of the ridge pattern, $\psi(x,y)$ is the hologram phase of the ridge pattern, and $N(x,y)$ refers to the noise of the image. The hologram phase ψ determines the location of the ridges and minutiae of the fingerprint, which can be demodulated by:

$$\psi(x,y) = Arg\{-e^{-i\epsilon(x,y)} \cdot \Re[F(x,y) - A(x,y)] + F(x,y) - A(x,y)\}, \quad (2)$$

where $Arg(z)$ returns the principal value of the argument of z, $\epsilon(x,y)$ is the local gradient and $\Re$ is a demodulation operator such that

$$\Re[F(x,y) - A(x,y)] \cong \mathfrak{F}^{-1}\{e^{i\varphi(u,v)}\mathfrak{F}\{F(x,y) - A(x,y)\}\}, \quad (3)$$

where $\mathfrak{F}(\cdot)$ and $\mathfrak{F}^{-1}(\cdot)$ are the Fourier transform and inverse Fourier transform, and $e^{i\varphi(u,v)}$ is a spiral phase Fourier multiplier [37]:

$$e^{i\varphi(u,v)} = \frac{u + iv}{\sqrt{u^2 + v^2}}. \quad (4)$$

The fingerprint hologram phase can be decomposed into the continuous phase and the spiral phase according to the Helmholtz Decomposition Theorem [38]:

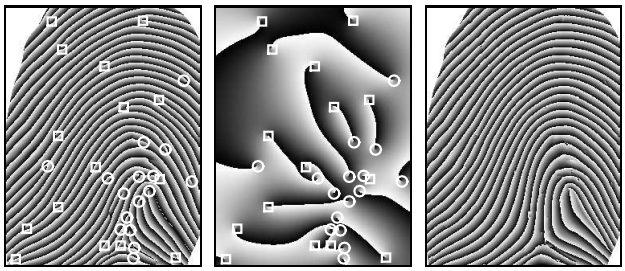
$$\psi(x,y) = \psi_c(x,y) + \psi_s(x,y), \quad (5)$$

where ψ_c is the continuous phase, ψ_s is the spiral phase. The value of different types of phase is within the range of $(0, 2\pi]$. In the following discussions, the hologram phase will also be termed as the composite phase for clarity. The continuous phase depends on the orientation and ridge frequency of the fingerprint, while the spiral phase can be calculated by a set of spirals:

$$\psi_s(x,y) = \sum_{i=1}^{n} p_i \arctan\left(\frac{y - y_i}{x - x_i}\right), \quad (6)$$

where n is the number of spirals, (x_i, y_i) is the location of the ith spiral, and $p_i \in \{-1, 1\}$ is the corresponding polarity.

It has been observed in [36] that the fingerprint minutiae (i.e., ridge endings and bifurcations) can be represented by spirals of either positive or negative polarity. And multiple minutiae points can be generated from the spirals using Eq. (6). The spirals are with abrupt phase changes and in accordance with the minutiae, which are located at the points with phase residuals of either -2π or 2π [39]. In the following discussions, both the minutiae and the spirals refer to the fingerprint ridge endings and bifurcations. Fig. 3 shows the images of different types of phase computed from the fingerprint image given in Fig. 1(a). It can be seen that the locations of the spirals are in accordance with the fingerprint minutiae, while the continuous phase image has the same ridge flow as that of the fingerprint structure.



(a) (b) (c)

Fig. 3. Different types of phase images of the same fingerprint. (a) The composite phase image, (b) the spiral phase image, and (c) the continuous phase image. The phase images are shown in grayscale for illustration purpose. The empty circles and squares refer to the minutiae points with positive and negative polarities, respectively.

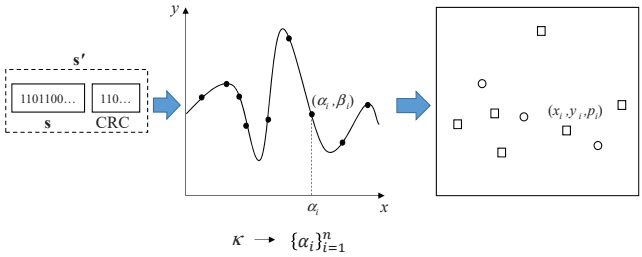


Fig. 5. The process of the secret message encoding. The empty circles and squares refer to the points with positive and negative polarities, respectively.

IV. Fingerprint Image Construction

The flowchart of our fingerprint image construction scheme is shown in Fig. 4. We propose to construct the spiral phase and the continuous phase separately from a secret message and a construction key. The constructed spiral phase and continuous phase are combined to compute the composite phase. Finally, we apply proper post processing steps to construct fingerprint images in different forms based on the composite phase. Table I gives the notations for quick reference.

A. Spiral phase construction

In order to construct the spiral phase, we propose to encode the secret message $\mathbf{s}$ into a set of n two dimensional points $\{(x_i, y_i)\}_{i=1}^n$ with the corresponding polarities $\{p_i\}_{i=1}^n$. The basic idea is to map the secret message to a polynomial. Then, we evaluate the polynomial on n different elements over a Galois field to compute x_i, y_i and p_i, as shown in Fig. 5. The details of the encoding process are summarized below.

1) Compute a set of cyclic redundancy check (CRC) bits according to $\mathbf{s}$, which is used for error detection during

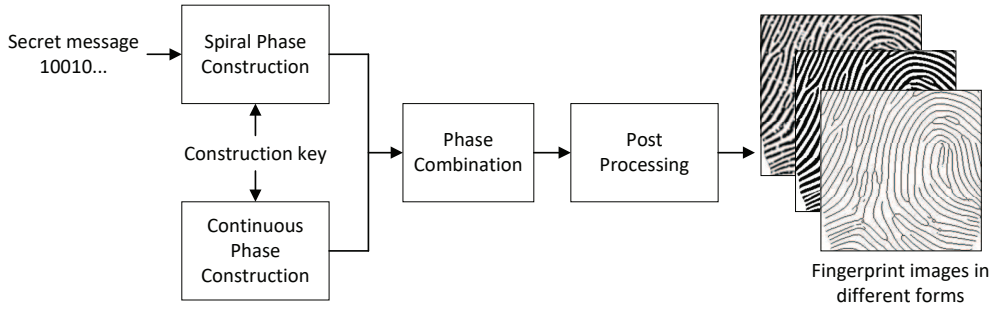


Fig. 4. The flowchart of the proposed fingerprint image construction scheme.

TABLE I
NOMENCLATURE

Notation(s)	Description
$\mathbf{s}$	Secret message
$\mathbf{s}'$	Secret message with CRC bits appended
k	Number of symbols partitioned from $\mathbf{s}'$
$\mathbf{s}'_j$	The jth of the k symbols
$\wp_k$	Polynomial constructed by the k symbols
r	Number of bits per symbol
(α_i, β_i)	Point constructed on $\wp_k$
(x_i, y_i)	Location of an encoded point (spiral)
p_i	Polarity of an encoded point (spiral)
n	Number of encoded points (spirals)
Δ	Scaling factor for mapping (α_i, β_i) to (x_i, y_i, p_i)
s_x, s_y	Displacements for mapping (α_i, β_i) to (x_i, y_i, p_i)
M, N	Width and hight of the fingerprint image
κ	Construction key
f_κ, O_κ	Frequency and orientation of the synthetic fingerprint
O^u_κ	Unwrapped orientation computed from O_κ
u	Integer for computing the unwrapped orientation
c_i, d_i	Locations of cores and deltas of the synthetic fingerprint
n_c, n_d	Number of cores and deltas of the synthetic fingerprint
λ	Parameter controlling the curvature of the arch
σ	Bandwidth of the Gabor filter
ϕ_0, ϕ_1, ϕ_2, ϕ_3	Value of composite phase of four neighboring pixels
$\gamma(x, y)$	Phase residual of the pixel located at (x, y)
$\Gamma(\phi_a, \phi_b)$	Compute the phase difference between ϕ_a and ϕ_b
ψ_s	Constructed spiral phase
ψ_c	Constructed continuous phase
ψ	Constructed composite phase
τ	Threshold for binarization
F_b	Fingerprint after binarization

data extraction. Let $\mathbf{s}'$ denote the message after appending the CRC bits to $\mathbf{s}$.

2) Partition $\mathbf{s}'$ into a group of k symbols with r bits per symbol: $\mathbf{s}' = \{\mathbf{s}'_j\}_{j=0}^{k-1}$. These symbols are mapped to a polynomial $\wp_k$ with

$$\wp_k(x) = \sum_{j=0}^{k-1} \mathbf{s}'_j x^j. \tag{7}$$

3) Evaluate the polynomial $\wp_k$ over the Galois field $\mathbb{F} = GF(2^r)$ at n ($n \geq k$) different elements: $\mathbf{x} = \{\alpha_i\}_{i=1}^n$, where α_i is the ith element of a vector containing a random permutation (based on κ) of the integers from 1 to n inclusive. As such, we have a list of evaluations $\mathbf{y} = \{\beta_i\}_{i=1}^n$, where $\beta_i = \wp_k(\alpha_i)$. Accordingly, we construct a set of points $\mathbf{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n$.

4) Map the points $\mathbf{P}$ to a set of encoded spirals $\mathbf{E}_s = \{(x_i, y_i, p_i)\}_{i=1}^n$ by

$$\begin{aligned} x_i &= \Delta\alpha_i + s_x \\ y_i &= \Delta\tfrac{\beta_i - LSB(\beta_i)}{2} + s_y \\ p_i &= 2LSB(\beta_i) - 1 \end{aligned}, \tag{8}$$

where Δ is a scaling factor to handle the noise during the fingerprint construction, $LSB(\beta_i)$ refers to the value of the least significant bit of β_i, and s_x and s_y are the displacements.

The locations and polarities of the encoded spirals $\mathbf{E}_s$ refer to that of the minutiae in the constructed fingerprint image. With the encoded spirals available, the constructed spiral phase ψ_s can be computed using Eq. (6).

We apply the IBM CRC-16 to generate a set of 16 CRC bits from the secret message $\mathbf{s}$. Therefore, the number of secret bits utilized for the spiral phase construction (i.e., the data embedding capacity) is $kr - 16$ with the maximum of k as n.

The values of n and r are constrained by the size of the fingerprint image to be constructed. Let's denote the width and height of the fingerprint image as M and N, respectively. Since the maximum of α_i and β_i are n and $2^r - 1$, the maximum of x_i and y_i can be computed as (see Eq. (8))

$$\begin{aligned} x_{max} &= \Delta n + s_x \\ y_{max} &= \Delta(2^{r-1} - 1) + s_y \end{aligned}. \tag{9}$$

Obviously, $x_{max} < \mathrm{M}$ and $y_{max} < \mathrm{N}$. Thus, we have the following constraints for n and r:

$$\begin{aligned} n &< \tfrac{\mathrm{M} - s_x}{\Delta} \\ r &< log_2\left(\tfrac{\mathrm{N} - s_y}{\Delta} + 1\right) + 1 \end{aligned}. \tag{10}$$

In other words, we can fix the width and height of the fingerprint image and determine the value of n and r. As long as $n \geq k$, secret messages with different length (i.e., different k) can be encoded to form a set of n encoded spirals, where more redundancy (i.e., $n - k$ spirals) will be added for shorter messages.

B. Continuous phase construction

Intuitively, the continuous phase can be constructed by decomposing the composite phase of an original fingerprint image according to Eq. (5). However, this will expose the continuous phase of the original fingerprint. To deal with such

an issue, we propose to construct the continuous phase by decomposing the composite phase of a synthetic fingerprint image. In particular, all the spirals in the synthetic fingerprint image will be removed to construct the continuous phase.

1) Synthetic fingerprint image generation: We adopt the Garbor filtering based fingerprint synthesis scheme [27] to generate the synthetic fingerprint image, which is to obtain the overall ridge flow of the constructed fingerprint. We choose this approach because it is able to generate a full synthetic fingerprint image without ad-hoc parameter selection. It requires the following three pieces of inputs, including 1) an initial image containing several seed points; 2) the fingerprint ridge frequency (termed as the ridge frequency for simplicity); and 3) the fingerprint orientation (termed as the orientation for simplicity).

Next, we introduce how we generate the inputs based on the construction key κ. Generally speaking, the number of seed points is proportional to the number of minutiae points (i.e., the spirals) in the synthetic fingerprint image. In our case, we want the number of spirals to be as few as possible to make the phase decomposition easier for the continuous phase construction. If there are no spirals in the synthetic fingerprint image, we can construct the continuous phase directly by demodulating the synthetic fingerprint image according to Eq. (2), which does not need the phase decomposition at all. Therefore, we set the initial image with only a single seed point located at the center of the fingerprint image. The ridge frequency determines the number of ridges within an unit distance, we random set it (say f_κ) as a constant over the whole fingerprint image within the range of $[1/9, 1/6]$ (based on κ), which covers the typical range of ridge frequency in 500-dpi fingerprint images [24]. The orientation measures the directions of the fingerprint ridges from 0 to π. It is the most important information for the topology of a fingerprint, which determines the fingerprint class as shown in Fig. 6. We determine the class of the synthetic fingerprint to be generated according to κ and the categorical distribution of the five major fingerprint classes [26]. For synthetic fingerprints with singular points (i.e., left loop, right loop, whorl and tented arch), we adopt the zero-pole model [40] to compute the orientation at point (x, y):

$$O_\kappa(x,y) = \frac{1}{2}\left[\sum_{i=1}^{n_c} Arg(z-c_i) - \sum_{i=1}^{n_d} Arg(z-d_i)\right], \quad (11)$$

where $z = y + jx$ is a complex number, c_i $(i = 1, 2, ..., n_c)$ and d_i $(i = 1, 2, ..., n_d)$ refer to the locations (both are in the complex domain) of the fingerprint cores and deltas, respectively. This model is able to generate the synthetic orientation by some simple constraints on the locations of singular points [40], which can be guided easily based on κ. For synthetic fingerprints without singular points (i.e., arch), the orientation at point (x, y) is computed by the following arch orientation model [27]:

$$O_\kappa(x,y) = \arctan\left(\lambda \cos\left(\frac{x\pi}{\mathrm{M}}\right)\right), \quad (12)$$

where λ is the parameter controlling the curvature of the arch, the range of which is randomly set (based on κ) within $[0.3, 3]$.

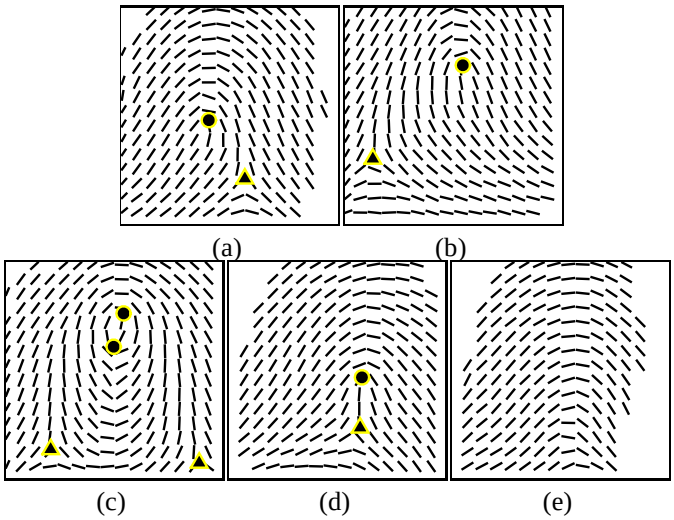


Fig. 6. The corresponding orientation estimated from the fingerprints given in Fig. 2. (a) Left loop, (b) right loop, (c) whorl, (d) tented arch, and (e) arch. The solid circles and triangles refer to the cores and deltas, respectively.

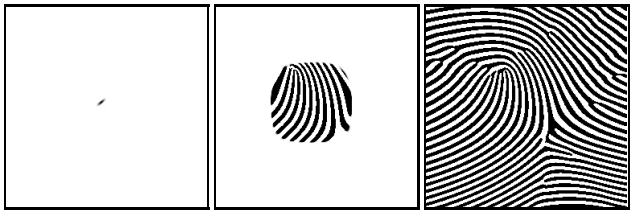


Fig. 7. Intermediate ridge patterns generated using Gabor filtering. The number of iterations increases from left to right.

With the initial image, the ridge frequency and the orientation available, we now iteratively apply a Gabor filter for each pixel located at (x, y) on the initial image [27]:

$$G(x', y'; f_\kappa, O_\kappa, \sigma) = e^{-((x'^2+y'^2)/2\sigma^2)} \cdot \cos[2\pi \cdot f_\kappa \cdot (x' \cdot \cos(O_\kappa(x,y)) + y' \cdot \sin(O_\kappa(x,y)))], \quad (13)$$

where σ is the bandwidth of the filter. The value of σ is chosen such that the filter does not contain more than three effective peaks, which is determined by the solution of the following equation [24]:

$$e^{-\left[\left(\frac{3}{2f_\kappa}\right)^2/2\sigma^2\right]} = 10^{-3}. \quad (14)$$

The iterative process will be terminated until the synthetic fingerprint is filled with an uniform ridge pattern, as shown in Fig. 7.

The orientation of such a synthetic fingerprint image is generated using classic and well known orientation models. These models ensure the smoothness and topology of the orientation. Therefore, it should be statistically similar to the original fingerprint orientation. This is to say, the ridge flow of the synthetic fingerprint image is similar to that of the original fingerprints.

2) Synthetic phase demodulation and decomposition: Once the synthetic fingerprint image is generated, we can construct the continuous phase using the phase demodulation and decomposition. According to Eq. (2), the composite phase can be demodulated if the AC component and the gradient of the

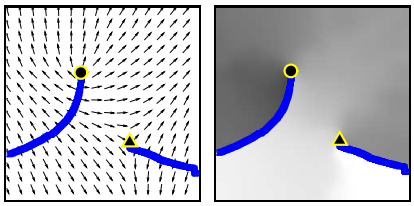


Fig. 8. Illustration of the fingerprint gradient using the arrows (on the left) and the grayscale image (on the right). The branch cut is shown in blue, and the solid circle and triangle refer to the core and delta.

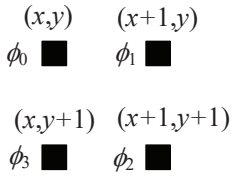


Fig. 9. The set of four adjacent pixels (the solid squares) defined for pixel (x, y) with ϕ_0 to ϕ_3 as the corresponding phase values.

fingerprint are known. We estimate the AC component by removing the mean pixel value from the synthetic fingerprint image. The gradient of the fingerprint is perpendicular to its orientation O_κ. However, the range of the local orientation is $(0, \pi]$ (see Fig. 6), while the range of the local gradient is $(0, 2\pi]$. To solve the ambiguity, we unwrap O_κ to get an unwrapped orientation O^u_κ with the range of $(0, 2\pi]$. For fingerprint without singular points, we have

$$O^u_\kappa(x, y) = O_\kappa(x, y) + u\pi, \tag{15}$$

where u is an integer satisfying the condition that the unwrapped orientation between any two adjacent pixels differs no more than $\pi/2$. Different strategies can be adopted to visit the pixels in the synthetic fingerprint image such as the depth-first or the breadth-first because the result is independent of the scan order [38]. For fingerprint with singular points, there is inevitable discontinuity around the area of the singular points. We adopt a well known branch cut based phase unwrapping algorithm [41] to mitigate this issue. This algorithm computes the branch cut by tracing the ridge of the fingerprint starting from each of the singular points. The unwrapping process is the same as in the fingerprint without any singular points, except that the pixels located at the branch cut can not be crossed and unwrapped. Given the unwrapped orientation O^u_κ, the gradient can be computed as $O^u_\kappa + \frac{\pi}{2}$.

With the AC component and the gradient available, the composite phase of the synthetic fingerprint can be demodulated according to Eq. (2). Fig. 8 illustrates the gradient of the synthetic fingerprint image shown on the right of Fig. 7. It can be seen that the range of the gradient covers from 0 to 2π, and the discontinuity appears only around the branch cut area.

In order to decompose the composite phase of the synthetic fingerprint, we first detect the spirals according to its residuals, which are calculated by summing the phase difference clockwise around each set of four adjacent pixels [41]. For each pixel located at (x, y), the set of four adjacent pixels are defined as pixel (x, y), pixel $(x + 1, y)$, pixel $(x + 1, y + 1)$ and pixel $(x, y + 1)$ as shown in Fig. 9. For simplicity, the corresponding phase values of the four adjacent pixels are termed as ϕ_0, ϕ_1, ϕ_2 and ϕ_3 (see Fig. 9). The residual $\gamma(x, y)$ for the pixel (x, y) is then computed by

$$\gamma(x, y) = \sum_{i=0}^{3} \Gamma(\phi_{(i+1) \bmod 4}, \phi_i), \tag{16}$$

where mod is the modulo operator and $\Gamma(\phi_a, \phi_b)$ is the function calculating the phase difference between ϕ_a and ϕ_b:

$$\Gamma(\phi_a, \phi_b) = (\phi_a - \phi_b + \pi) \bmod 2\pi - \pi. \tag{17}$$

It has been proved that $\gamma(x, y)$ will either be zero, 2π or -2π [39]. Pixels with residuals equal to 2π or -2π indicate positive spirals or negative spirals, respectively. Note that, the pixels around the branch cut area are not taken into account for the spirals detection due to the discontinuity of gradient.

Once the spirals are detected, we can compute the spiral phase of the synthetic fingerprint using Eq. (6). Consequently, the continuous phase ψ_c is constructed by subtracting the spiral phase from the composite phase of the synthetic fingerprint.

C. Post processing

With the constructed spiral phase ψ_s and continuous phase ψ_c, the composite phase ψ of the constructed fingerprint is computed by combining them together according to Eq. (5). During the phase combination, the local fingerprint orientation will be slightly changed (when compared with O_κ) due to the creation of minutiae points.

According to the model given in Eq. (1), the phase modulated signal $cos(\psi)$ represents an ideal fingerprint, while the other components just make the fingerprint to be realistic. The gradual change of the cosine wave forms the fingerprint ridges and valleys. The value of the composite phase also gradually increases or decreases from $(0, 2\pi]$ or $[2\pi, 0)$ within two consecutive ridges as shown in Fig. 10. This property makes it easy to obtain a binary fingerprint image directly from ψ using a single threshold. Concretely, a binary fingerprint image F_b can be computed by

$$F_b(x, y) = \begin{cases} 1 & if\ \psi(x, y) > \tau \\ 0 & otherwise \end{cases}, \tag{18}$$

where $\tau \in (0, 2\pi)$ is the threshold to construct the binary fingerprint image. The value of τ controls the thickness of the fingerprint ridges. It can be seen from Fig. 11 that the ridges become thinner when τ decreases.

We set $\tau = \pi$ to obtain the final binary fingerprint image and $\tau = 0.4\pi$ to get a coarsely thinned fingerprint image. Please refer to Section VI-A for the settings of τ. The final thinned fingerprint image is constructed by iteratively removing the boundary pixels from the coarsely thinned fingerprint image using the algorithm proposed in [42] (see Fig. 12(a)). To construct the grayscale fingerprint image, we treat the binary fingerprint image as the master fingerprint. Then, as suggested in [27], we perform noising on the master fingerprint by adding small white blobs of various sizes and shapes (see Fig. 12(b)).

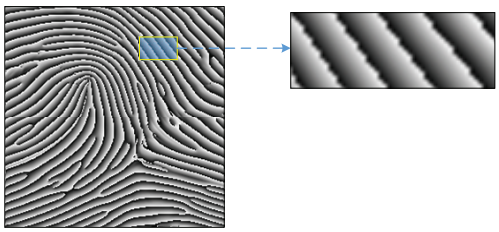


Fig. 10. An example of the constructed composite phase, where the continuous phase is constructed based on the synthetic image shown on the right of Fig. 7. Part of the composite phase is zoomed in and the phase image is shown in grayscale for illustration purpose. Dark pixels refer to the pixels with phase value close to zero, whereas the white pixels mean the pixels with phase value close to 2π.

Fig. 11. Different binary fingerprint images computed based on different thresholds from the composite phase in Fig. 10. From left to right: $\tau = 1.4\pi$, $\tau = \pi$, and $\tau = 0.4\pi$.

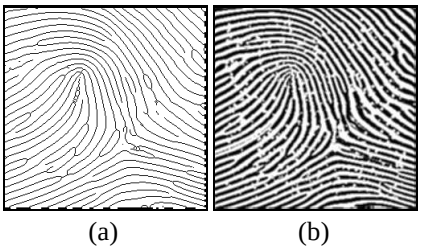


Fig. 12. The constructed thinned fingerprint image (a), and the constructed grayscale fingerprint image (b).

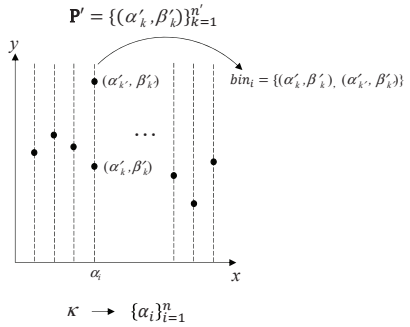


Fig. 13. Illustration of the point categorization, where $\alpha_i = \alpha'_k = \alpha'_{k'}$.

The post processing hardly affects the locations of the fingerprint minutiae, as shown in Fig. 10, Fig. 11, and Fig. 12. This is to say, the spirals of the constructed fingerprint, which encode the secret message, will be very close before and after the post processing. Please refer to Section VI-A for quantitative measures of the distortion of the constructed fingerprint before and after post processing.

V. Data Extraction

Given a constructed fingerprint image in any of the three forms, we perform the fingerprint enhancement using an existing algorithm [43], which is to remove the noise or other detailed features created during the post processing. First of all, we generate the synthetic orientation O_κ based on κ. This is used to demodulate the enhanced fingerprint image to get the composite phase. Then, we detect a set of spirals $\mathbf{E}'_s = \{(x'_k, y'_k, p'_k)\}_{k=1}^{n'}$ based on the residuals of the composite phase. In particular, spirals are located at the pixels with residuals of either 2π or -2π. Please refer to Section IV-B2 for details of the phase demodulation and spirals detection. The detected spirals are further processed to obtain the points $\mathbf{P}' = \{(\alpha'_k, \beta'_k)\}_{k=1}^{n'}$, where

$$\begin{aligned} \alpha'_k &= round(\tfrac{x'_k - s_x}{\Delta}) \\ \beta'_k &= round(\tfrac{2(y'_k - s_y)}{\Delta} + \tfrac{p'_k + 1}{2}) \end{aligned}, \tag{19}$$

where $round(\bullet)$ means the rounding operation.

In order to extract the secret $\mathbf{s}$, we have to reconstruct the polynomial $\wp_\kappa$ based on the points $\mathbf{P}'$ over the Galois field $\mathbb{F}$. According to the definition of the original view of Reed-Solomon code [44], every codeword contains a sequence of n evaluations of a polynomial of degree less than k, where the message is represented as a sequence of the polynomial coefficients. Thus, we can treat the evaluations (i.e., β'_k) as a codeword and apply an existing Reed-solomon decoder for data extraction. The advantage of the Reed-Solomon encoding/decoding is to correct the errors occur in burst. Regardless the number of bits in a evaluation are in error, it is counted as a single error. This is well-suited to correct the errors in our constructed fingerprint image, which appear in burst in terms of spurious spirals or missed spirals. The spurious spirals refer to the newly generated spirals that can not be found in the original spirals. The missed spirals are the original spirals that can not be detected from the constructed fingerprint. Each error spiral (spurious or missed) corresponds to a set of r bits.

In general, a Reed-solomon decoder is able to correct up to $(n-k)/2$ error points [45]. This is to say, as long as there are $n-(n-k)/2 = (n+k)/2$ original points in $\mathbf{P}'$, the polynomial can be reconstructed successfully. Please refer to Appendix A for an introduction of the Reed-solomon decoder and the decoding process. Given the point list $\mathbf{P}' = \{(\alpha'_k, \beta'_k)\}_{k=1}^{n'}$, we formulate the following data extraction steps.

1) Compute the same vector $\mathbf{x} = \{\alpha_i\}_{i=1}^{n}$ based on κ as what have done in the spiral phase construction (see Section IV-A).
2) Categorize the points in $\mathbf{P}'$ into n different bins bin_i, where

$$bin_i = \{pt(\alpha_i, *)\}, \tag{20}$$

where $pt(\alpha_i, *)$ refers to all the points in $\mathbf{P}'$ with x-coordinate of α_i. An illustration of the above point categorization process is shown in Fig. 13 with the algorithmic process given in Algorithm 1. If there are no points that can be found for bin_i, the point with x-coordinate nearest to α_i will be categorized into bin_i. As such, each bin contains at least one point.

Algorithm 1 Point Categorization

Input:$\mathbf{P}' = \{(\alpha'_k, \beta'_k)\}_{k=1}^{n'}$, α_1, α_2, ... , α_n
Output: bin_1, bin_2, ... ,bin_n

for $i = 1$ to n **do**
  $bin_i = NULL$
  for $k = 1$ to n' **do**
    if $\alpha_i == \alpha'_k$ **then**
      add the point (α'_k, β'_k) to bin_i
      update bin_i
    end if
  end for
end for

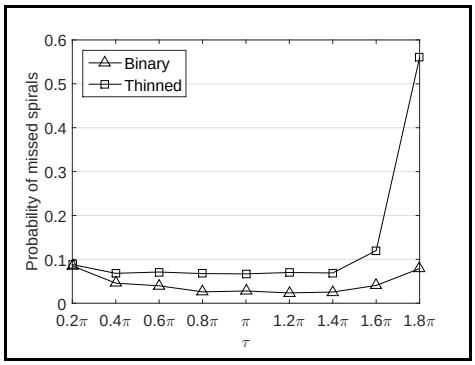


Fig. 14. The probability of missed spirals for the binary and thinned fingerprint images under different settings of τ.

3) Form a set of candidate point lists which consists of all the combinations by selecting one point from each bin. Thus, each candidate point list contains n points.
4) Select one candidate point list and forward it to the Reed-Solomon decoder to reconstruct the polynomial $\wp_\kappa$. If the decoding fails, select another one and repeat this step. Otherwise, concatenate the decoded polynomial coefficients to get a decoded message and goto the next step.
5) Compute the CRC bits of the first $kr - 16$ bits of the decoded message and compare it with the last 16 bits. If they are exactly the same, the secret message is extracted as the first $kr - 16$ bits of the decoded message. Otherwise, the data extraction has not been successful, we will select another candidate point list and goto step 4) until all the candidate point lists are visited. If there are no more candidate point lists available, the data extraction fails.

Such a data extraction process achieves high probability of successful data extraction and low probability of incorrect data extraction, please refer to Appendix B for details.

VI. Experimental results

A. Threshold settings for post processing

The post processing requires a threshold τ to determine the thickness of the fingerprint ridges. In our implementation, we adopt two thresholds for constructing the final binary fingerprint image and a coarsely thinned fingerprint image, respectively. Putting the ridge thickness aside, the setting of τ also affects the performance of data extraction. The optimal value of τ should be the one that is able to produce a binary or a thinned fingerprint image containing the most original spirals.

We construct a set of 1000 fingerprints before post processing (i.e., the constructed composite phase) of size 300×300 (i.e, $\mathrm{M} = \mathrm{N} = 300$) with $n = 40$ and $r = 6$. We set the scaling factor as $\Delta = 7$, the displacements as $s_x = 8$ and $s_y = 8$, and the value of k from 20 to 32. Such a scaling factor is able to handle three pixels of horizontal or vertical shift for the original spirals during the fingerprint image construction. This is to say, if both the horizontal and vertical shift of an original spiral are no more than three pixels, it can be correctly detected from the constructed fingerprint. We vary the value of τ to construct binary fingerprint images with different ridge thickness, from which we obtain the corresponding thinned fingerprint images. Fig. 14 gives the distortion of the constructed fingerprint images in terms of the probability of missed spirals, i.e., the probability that an original spiral can not be correctly detected. It can be seen that, as long as τ is within $[0.8\pi, 1.4\pi]$, its impact on the existence of the spirals is low with little sensitivity for the binary fingerprint images. Such range is changed to $[0.4\pi, 1.4\pi]$ for the thinned fingerprint images.

In our implementation, we set $\tau = \pi$, which corresponds to the valley of the wave $cos(\psi)$, to construct a binary fingerprint image with ridges and valleys of roughly the same width. This is in accordance with most ordinary real fingerprint images. We set $\tau = 0.4\pi$ to obtain a coarsely thinned fingerprint image for easier computation of the final thinned fingerprint image. Each of the settings is within the appropriate range of τ to achieve relatively good performance.

Besides the probability of missed spirals, we further consider two other measures given below to evaluate the distortion of the constructed fingerprints.

1) Probability of spurious spirals: the probability to detect a spiral that does not exist in the original spirals.
2) Shift of spirals: the average distance (in pixels) between the correctly detected spirals and the corresponding original spirals.

Table II reports the values of these measures estimated from the constructed fingerprints with the implemented settings of τ. It can be seen that all the original spirals can be detected in the constructed composite phase, while most of the original spirals could still be detected after the post processing. On the other hand, our fingerprint construction creates a few spurious spirals, which also slightly shifts the original spirals regardless the post processing.

B. Capacity and data extraction accuracy

The capacity is the amount of secret bits that can be hidden in the constructed fingerprint image. According to what we have discussed in Section IV-A, the capacity depends on the resolution of the constructed fingerprint image and the

TABLE II
QUANTITATIVE MEASURES OF THE DISTORTION OF THE CONSTRUCTED FINGERPRINTS BEFORE AND AFTER POST PROCESSING.

	before post processing	after post processing		
	constructed composite phase	binary fingerprint	thinned fingerprint	grayscale fingerpirnt
Probability of missed spirals	0	0.0276	0.0681	0.0452
Probability of spurious spirals	0.0527	0.0891	0.1387	0.1119
Shift of spirals (in pixels)	1.2475	1.5227	1.8901	1.6204

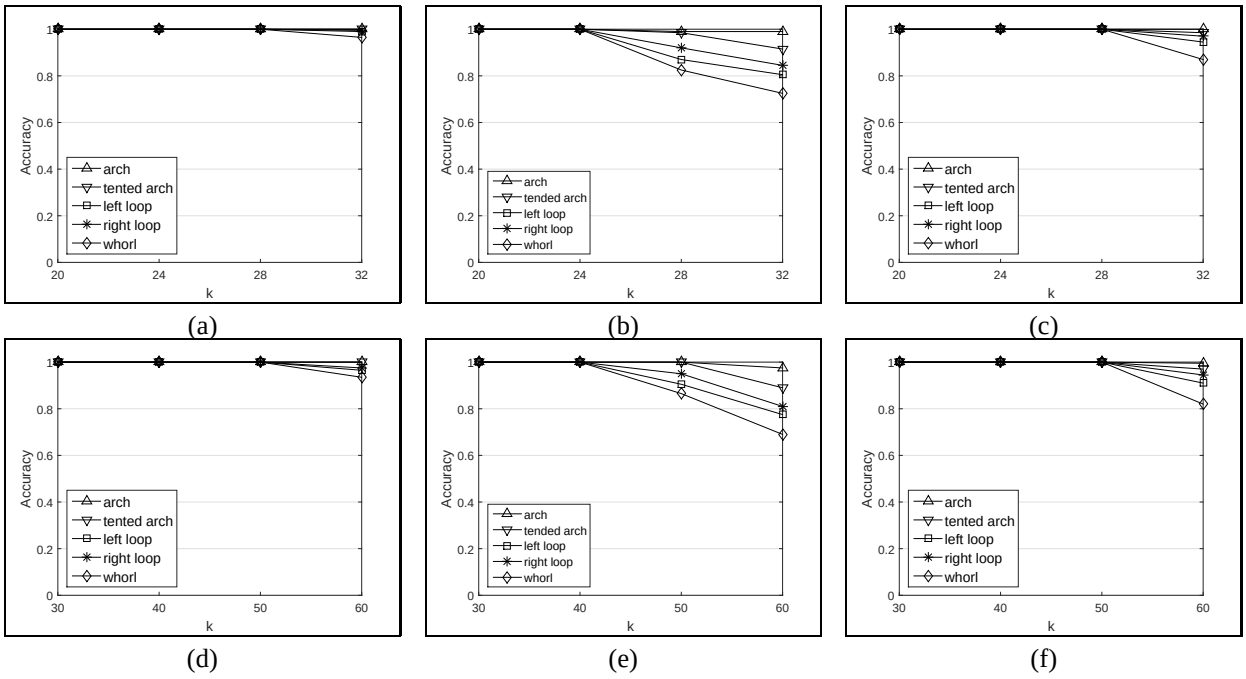


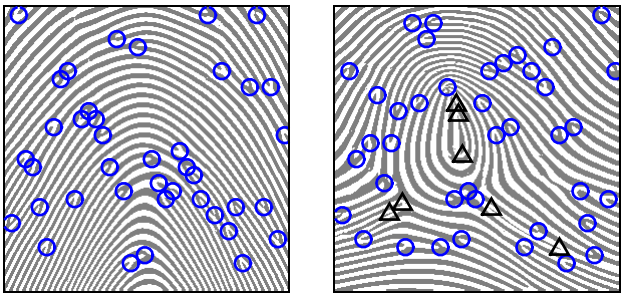
Fig. 15. Data extraction accuracy of the proposed scheme. Top: Size I with (a) binary fingerprint images, (b) thinned fingerprint images, and (c) grayscale fingerprint images; Bottom: Size II with (d) binary fingerprint images, (e) thinned fingerprint images and (f) grayscale fingerprint images.

redundancy of the encoded points. In order to make the constructed fingerprint image natural, the image resolution and the number of minutiae points should be within a normal range. In [24], the authors summarize the resolutions of the fingerprint images captured from common commercial sensors, where the maximum is 600×600 and the minimum is 95×95. The scanners for capturing multi-fingerprints can produce fingerprint images with a resolution up to 3000×3200. Furthermore, the number of minutiae points varies from 15 to over 100 for fingerprint images captured from a single finger.

We construct fingerprint images with the following two sizes: 1) Size I: 300×300, and 2) Size II: 500×500. We set $\Delta = 7$, $s_x = 8$ and $s_y = 8$ for both the two sizes. We set $n = 40$ and $r = 6$ for Size I, and $n = 70$ and $r = 7$ for Size II. This satisfies the constraints between n (or r) and the size of the constructed fingerprint image (see Eq. (10)).

We vary the value of k to construct the fingerprint images with Size I and Size II. For each size and a specific k, we construct 1000 fingerprints using different secret messages with length of $kr-16$. These fingerprints consist of five major fingerprint classes including 200 arch, 200 tented arch, 200 left loop, 200 right loop and 200 whorl. Each of the fingerprints contains three different forms including the binary fingerprint image, the thinned fingerprint image, and the grayscale fingerprint image. For each form, we perform data extraction on different fingerprint classes separately, the accuracy of which is given in Fig. 15. Since the Reed-solomon decoder is able to correct up to $(n-k)/2$ error points, smaller k (i.e., shorter length of secret message) corresponds to more tolerance of error points. It is expected to see from Fig. 15 that the data extraction accuracy increases upon the decreasing of k. It can also be seen that the data extraction accuracy varies among different fingerprint classes as well as different forms of fingerprint images. In terms of fingerprint classes, the arch achieves the highest data extraction accuracy, which appears to be the lowest for the whorl. In terms of the forms, the data extraction accuracy of the binary fingerprint images are higher than that of the other two forms (thinned and binary), with 100% data extraction accuracy for Size I at $k = 28$ and Size II at $k = 50$. The corresponding capacity is $28 \times 6 - 16 = 152$ bits and $50 \times 7 - 16 = 334$ bits, respectively.

The above phenomenon could be explained below. Our fingerprint construction may create some spurious spirals due to the discontinuity of the unwrapped orientation on the branch cuts and the post processing. Fingerprint classes with more singular points and branch cut areas have the tendency to create more spurious spirals. For arch, there is no singular point and the orientation can be unwrapped perfectly without any discontinuity. Thus, there are hardly any spurious spirals in the constructed arch fingerprint image, as shown in Fig. 16(a). For other fingerprint classes, spurious spirals may be detected around the singular point and the branch cut area, as



(a) (b)

Fig. 16. The locations of the spirals detected after the fingerprint construction, the blue circles refer to the spirals representing the secret message, and the black triangles refer to the spurious spirals. (a) A binary arch fingerprint image, and (b) a binary whorl fingerprint image. The images are made transparent for illustration purpose.

shown in Fig. 16(b). The fingerprints of whorl contain the most number of singular points (see Fig. 2) as well as the branch cut areas. It is expected to find that their data extraction accuracy is lower than other classes.

On the other hand, additional thinning or noising process is applied on the binary fingerprint images to obtain the thinned or grayscale fingerprint images, which would generate more spurious spirals. For example, thinning may create small islands in the fingerprint, which might lead to spurious spirals. The small white blobs added during the noising might also result in spurious spirals. Locations of the ridge endings and bifurcations may be slightly displaced due to the thinning or noising process. If the displacement is beyond certain distance, original spirals will be removed and spurious spirals will be generated. This explains why the binary fingerprint image achieves the highest data extraction accuracy among the three forms. It should be noted that, for the grayscale fingerprint images, the accuracy also depends on how much the noise is added. Severe noising will result in low data extraction accuracy. Some other examples of the constructed fingerprint images are given in Fig. 17.

Similar to the real fingerprint images, our constructed fingerprint images contain fine details such as pores, noise or other fine level features, which are mainly created due to the post processing. As indicated in [36], these details can not be easily modeled and a noise term has to be considered (see Eq. (1)). In the phase demodulation, the noise has to be removed beforehand as shown in Eq. (2). This is why we apply the fingerprint enhancement on the constructed fingerprints during the data extraction, which is also a necessary and important step in the phase demodulation. To verify its effectiveness, we perform data extraction on all the fingerprint images constructed before without the enhancement, the accuracy of which is shown in Fig. 18. It can be seen that, without the fingerprint enhancement, we are not able to perform any correct data extraction for the grayscale fingerprint images. Because the noising has a severe impact on the fine details of the grayscale fingerprint images, and the noise term can not be neglectable at all in the phase demodulation. For binary and thinned fingerprint images, the data can be still extracted, but the accuracy is much lower compared with that after fingerprint enhancement.

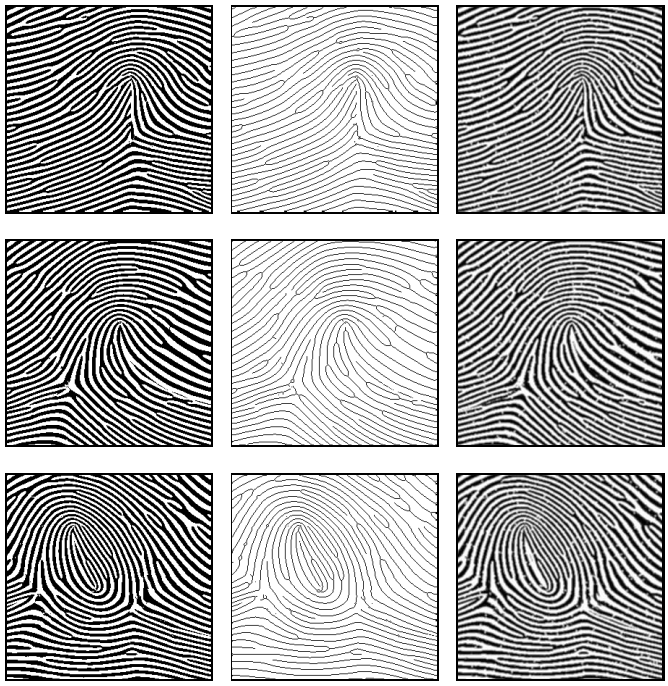


Fig. 17. Examples of the constructed fingerprint images. From left to right: binary fingerprint images, thinned fingerprint images, and grayscale fingerprint images.

C. Robustness

In this section, we discuss and evaluate the robustness of the proposed scheme in two aspects: 1) the robustness of fingerprint construction, and 2) the robustness of data extraction.

1) Robustness of fingerprint construction: Our fingerprint construction contains three major steps including the spiral phase construction, the continuous phase construction and the phase combination. Among these steps, the outputs of the spiral phase construction and the phase combination are deterministic given a piece of specific secret message and continuous phase. While the construction of continuous phase is performed by demodulating and decomposing a synthetic fingerprint. Therefore, the robustness of our fingerprint construction relies on the robustness of the synthetic fingerprint generation. As what we have mentioned in Section IV-B1, we adopt the Gabor filtering based approach for synthetic fingerprint generation, where the ridge frequency f_κ and orientation O_κ are generated based on the construction key κ for input. The constraints of these two pieces of inputs are well studied in literature [24], [27], [40]. To verify the effects of f_κ and O_κ, we show in the first and second column of Fig. 19 the synthetic fingerprint image and the corresponding continuous phase image with different settings of f_κ and O_κ. It can be seen that all the continuous phase are properly constructed, where the ridge frequency depends on f_κ and the overall ridge flow depends on O_κ. These continuous phase can be combined with the spiral phase representing the same piece of secret message, which produces the composite phase (of

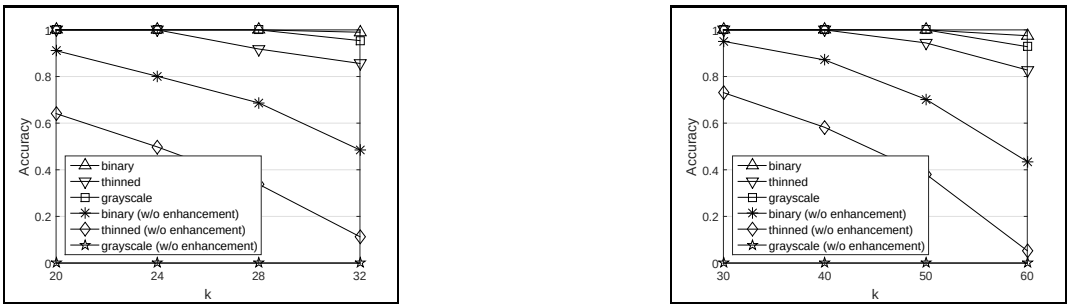


Fig. 18. The data extraction accuracy with and without the fingerprint enhancement for images of (a) Size I and (b) Size II.

the constructed fingerprint) with the same message hidden as shown in the third column of Fig. 19.

2) Robustness of data extraction: To test the robustness of our scheme in data extraction, various kinds of attacks are applied on the constructed fingerprint images using StirMark Benchmark 4.0 [46], including JPEG compression, random noise addition, filtering, rotation, scaling, shearing, linear transform, line removal and cropping. To make the StirMark Benchmark 4.0 workable on both the binary and thinned fingerprint images, we convert these images to grayscale by assigning the grayscale intensity of 0 and 255 for the black and white pixels, respectively. Besides these StirMark attacks, we consider four more types of attacks: fingerprint binarization, fingerprint thinning, salt and pepper noise addition and spiral alteration. The spiral alteration refers to the case that the attacker is aware of the fingerprint construction and uses it to attack the constructed fingerprint images. In such a case, he can conduct the phase demodulation and decomposition on the fingerprint image and modify the locations and polarities of the spirals. The modified spiral phase can be combined with the continuous phase to form a new construct fingerprint image.

Two sets of fingerprint images (constructed in Section VI-B) are incorporated in this test including: Size I with $k = 24$ and Size II with $k = 40$. The data extraction accuracy without and with different attacks are given in Table III. In this table, QF is the quality factor of JPEG compression; ST is intensity of the noise (normalized from 0 to 100); R is the angle of rotation (in degrees); SC is the scaling ratio, A_x and A_y are the horizontal and vertical shear factors; *LT1* and *LT2* are two linear transforms with $LT1 = [1.005, 0.009; 0.010, 1.006]$ and $LT2 = [1.004, 0.008; 0.009, 1.003]$; LR means one line is removed in every LR lines horizontally and vertically; C refers to the cropping ratio; and SN is the number of spirals that are altered.

It can be seen that, regardless of the forms, the constructed fingerprint images perform extremely well in resisting the popular fingerprint image operations (binarization and thinning) and the JPEG compression. For the binary and grayscale constructed fingerprint images, the data extraction accuracy remains high even when $QF = 5$. And most of them are able to resist moderate noise addition and all the filtering. For the thinned fingerprint images, however, noise addition and filtering have severe impact on the data extraction accuracy. The reason is that the thinned fingerprint images only contain ridges of one pixel width, the ridge endings and bifurcations (which represent the secret message) are sensitive to operations such as noise addition or filtering. Our scheme is less robust in resisting geometric transforms especially for images with large size. The reason is that the geometric transforms change the locations of all the spirals, which creates challenges in detecting the correct spirals for data extraction. We can also see from the table that, for most of the constructed fingerprint images, the data can be correctly extracted when a few spirals are altered.

D. Steganalysis on the constructed fingerprint images

In this section, we evaluate the performance of the existing steganalysis tools on the constructed fingerprint images. For grayscale fingerprint images, we use the rich model [17], a popular tool for steganalysis on grayscale or color images. Since both the binary and thinned fingerprint images belong to the binary images, tools for binary image steganolysis are needed for the evaluation. Unlike the grayscale or color images, there is limited work regarding the binary image stegonalysis in the literature [47]–[50]. We here apply the latest work [50] for steganalysis on the binary and thinned fingerprint images.

As suggested in [21], a set of pure synthetic images could be generated and served as the cover images. In our case, we generate the pure synthetic fingerprint images (i.e., the cover images) with the size of 300×300. The corresponding constructed fingerprint images are served as the stego-images which share the same continuous phase as the cover images. For each of the three forms, we generate 1000 cover images and 1000 stego-images, where half of them are used for training and the rest are used for testing. The performance of the steganalysis tools on these fingerprint images are given in Table IV, where $\mathbb{P}_p$ means the rate of detecting a cover image as a stego-image and $\mathbb{P}_n$ refers to the rate of detecting a stego-image as a cover image, and $\mathbb{P}_a$ is the average detection error:

$$\mathbb{P}_a = \frac{\mathbb{P}_p + \mathbb{P}_n}{2}. \tag{21}$$

It can be seen that, regardless of the form, the existing steganolysis tools are not effective in detecting the existence of secret message on the constructed fingerprint images, where the average detection error is close to 50%.

TABLE III
THE ACCURACY OF DATA EXTRACTION (IN PERCENTAGE) WITHOUT AND WITH DIFFERENT KINDS OF ATTACKS

Fingerprints		Size I			Size II		
		Binary	Thinned	Grayscale	Binary	Thinned	Grayscale
No attack		100.0	100.0	100.0	100.0	100.0	100.0
Fingerprint operation	Binarization	-	-	100.0	-	-	100.0
	Thinning	100.0	-	100.0	100.0	-	100.0
JPEG compression	$QF = 5$	100.0	78.9	97.5	100.0	80.2	100.0
	$QF = 25$	100.0	100.0	100.0	100.0	100.0	100.0
	$QF = 45$	100.0	100.0	100.0	100.0	100.0	100.0
Random noise	$ST = 5$	100.0	60.7	100.0	100.0	45.6	100.0
	$ST = 15$	100.0	26.1	97.0	100.0	6.2	96.6
	$ST = 25$	91.8	0.0	81.3	85.2	0.0	71.4
Salt and pepper noise	$ST = 5$	100.0	7.5	100.0	100.0	0.0	100.0
	$ST = 15$	100.0	0.0	100.0	100.0	0.0	100.0
	$ST = 25$	100.0	0.0	98.6	100.0	0.0	96.0
Median filter (3x3)		97.5	0.0	93.0	95.2	0.0	90.4
Gaussian filter (default setting)		96.5	0.0	91.1	94.3	0.0	88.9
Sharpening (default setting)		98.9	67.1	97.3	97.9	55.8	95.2
Rotation	$R = 0.25$	100.0	81.4	95.1	100.0	63.2	79.8
	$R = 0.50$	98.7	57.0	71.0	73.6	27.2	47.3
Scaling	$SC = 0.995$	99.1	75.6	91.8	72.1	38.3	62.0
	$SC = 1.005$	100.0	77.7	95.0	75.8	40.9	64.4
Shearing	$A_x = 0.01$, $A_y = 0.01$	97.1	60.0	68.5	0.0	0.0	0.0
	$A_x = 0.05$, $A_y = 0.05$	0.0	0.0	0.0	0.0	0.0	0.0
Linear transform	$LT1$	37.1	3.8	15.5	0.0	0.0	0.0
	$LT2$	66.8	10.1	38.0	0.0	0.0	0.0
Line removal	$LR = 150$	85.2	19.3	68.9	66.1	5.4	48.2
	$LR = 200$	92.3	27.7	80.1	78.8	10.9	61.1
Cropping	$C = 0.95$	0.0	0.0	0.0	0.0	0.0	0.0
	$C = 0.99$	97.8	71.3	89.3	80.9	50.5	71.1
Spiral alteration	$SN = 2$	99.5	73.8	98.0	100.0	85.0	100.0
	$SN = 4$	95.7	56.4	93.1	98.3	61.1	96.4

TABLE IV
PERFORMANCE OF THE EXISTING STEGANALYSIS TOOLS ON THE CONSTRUCTED FINGERPRINT IMAGES.

Fingerprint images	$\mathbb{P}_p$ (%)	$\mathbb{P}_n$ (%)	$\mathbb{P}_a$ (%)
Binary	64.4	36.0	50.2
Thinned	55.8	44.4	49.6
Grayscale	58.4	45.2	51.8

E. Security

For a constructed fingerprint image, the security of the secret depends on the secrecy of the construction key κ. During the fingerprint image construction, we encode the secret message by evaluating a polynomial on n different elements, the values of which are obtained by random permutation (based on κ) of the integers from 1 to n inclusive. In data extraction, these elements have to be correctly computed for decoding. Therefore, a brute force attack would have to try $n!$ times to extract the secret, which corresponds to $log_2(n!)$ bits of security. For $n = 40$ and $n = 70$, the security of the secret is roughly 159 and 332 bits, respectively. Of course, the attacker can also do a brute force attack to recover κ for data extraction. Assume the strength of κ is L_κ bits and $L_\kappa < log_2(n!)$, the security of the secret will be reduced to L_κ bits. Otherwise, the security of the secret remains $log_2(n!)$ bits.

VII. DISCUSSIONS

Our proposed scheme can be deployed in the communication channels of fingerprint recognition systems, because it is unsuspicious to transmit fingerprint images in such channels and our scheme has the ability to resist popular fingerprint image operations. Concretely, we are able to convey short messages such as encryption/decryption keys or URLs in the channels using our constructed fingerprint images. First of all, the channel may only accept fingerprint images in one of the three common forms (see Fig. 1). Secondly, the fingerprint image may be binarized or thinned during the transmission. Our scheme is flexible in constructing fingerprint images in any of the common forms that is acceptable in the channel. On the other hand, we can still extract the message even if the fingerprint image is binarized or thinned during the transmission. It should be noted that our scheme is not workable if the fingerprint recognition system extracts the fingerprint feature locally, where the data transmitted through the channel is not the fingerprint image.

Next, we discuss the statistical distance among our constructed fingerprints, the pure synthetic fingerprints, and the original fingerprints. Regardless of the forms of the fingerprint image, there are two main levels of fingerprint features including the orientation and the minutiae. The orientation describes the ridge flow of the fingerprint. The minutiae capture the fingerprint ridge endings and bifurcations. As what we have pointed out in Section IV-B1, our constructed fingerprints and the synthetic fingerprints share the same orientation (ridge flow). While the distribution of the synthetic fingerprint orientation is similar to that of the original fingerprint orientation.

For the minutiae, some researchers indicate that they tend to cluster around the singular points, which are not uniformly distributed [51]. In our constructed fingerprint and the synthetic fingerprint, however, there are no constraints on the

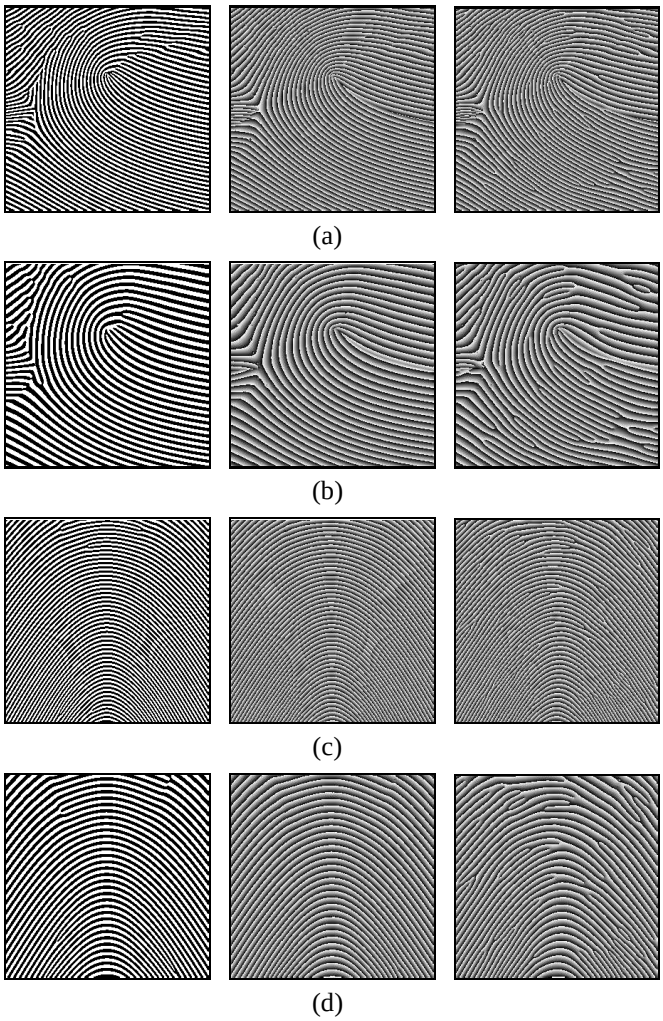


Fig. 19. Fingerprint construction based on different settings of f_κ and O_κ. From left to right: synthetic fingerprint images, the constructed continuous phase images, and the composite phase images of the constructed fingerprints. The phase images are shown in grayscale for illustration purpose. (a) and (b): Right loop fingerprints constructed with $f_\kappa = 1/6$ and $f_\kappa = 1/9$, O_κ of both fingerprints are computed based on the zero-pole model with a core located at $(150, 100)$ and a delta located at $(50, 150)$, (c) and (d): arch fingerprints constructed with $f_\kappa = 1/6$ and $f_\kappa = 1/9$, O_κ of both fingerprints are computed based on the arch orientation model with the arch curvature of 2.5.

distribution of minutiae. Therefore, the statistical distance among different fingerprints mainly depends on their minutiae, which can be roughly measured using the Kullback Liebler (KL) divergence [52]. In particular, we build three fingerprint databases including 1000 constructed fingerprint images, 1000 synthetic fingerprint images, and 800 original fingerprint images obtained from FVC2000 DB1_A [53]. All the images are in grayscale with the size of 300×300. The number of the minutiae points of the constructed or synthetic fingerprint is controlled within the range of 15 to 60, which is similar to that of the original fingerprints in FVC2000 DB1_A. We perform the fingerprint alignment such that the primary singular point of each fingerprint image is located at the center with angle of $\pi/2$ [54]. To compute the KL divergence, we partition the aligned fingerprint image into 50×50 non-overlapping blocks. Then, we accumulate the number of minutiae points in each block from all the fingerprints in each database. As such, we obtain a 6×6 probability density map indicating the minutiae distribution for each database. According to these maps, the KL divergence between the constructed fingerprints and the synthetic fingerprints is computed as 0.2112. This is increased to 0.2732 for the constructed fingerprints and the original fingerprints.

VIII. Conclusions and Future work

A novel construction based data hiding technique is proposed in this paper. Instead of constructing textures as what have been done in the literature, we propose to construct fingerprint images directly from the secret message. The proposed scheme is based on the construction of the composite phase of the fingerprint, which is the combination of the spiral phase and the continuous phase. The spiral phase is constructed by encoding the secret message to a set of two dimensional points with different polarities, while the continuous phase is constructed from a fingerprint image synthetically generated. Different fingerprint images can be generated based on the constructed composite phase, including the binary fingerprint image, the thinned fingerprint image, and the grayscale fingerprint image. The experimental results show that our scheme achieves satisfactory data extraction accuracy and robustness. In addition, we demonstrate the ineffectiveness of the existing steganalysis tools on the constructed fingerprint images.

Our scheme does not put any constraints on the distribution of the minutiae of the constructed fingerprints, which slightly differs from the synthetic fingerprints or the original fingerprints. This leaves traces for designing specific steganalysis tools. A straightforward way is to use the probability density map of the minutiae to train the classifier. We use the three fingerprint image databases adopted in Section VII. For each database, half of the images are used for training, while the rest are used for testing. For each fingerprint image, we extract a 6×6 minutiae probability density map by following the procedure described in Section VII, which serves as the feature for training and testing. The average detection error is 35.25% and 31.20% for detecting the constructed fingerprint images from the pure synthetic fingerprint images and the original fingerprint images, respectively. This demonstrates the usefulness of the fingerprint specific features for steganalysis. In the future, the traces left due to the fingerprint construction should be further studied to improve the performance of steganalysis. Meanwhile, a better spiral phase construction approach should be investigated by putting proper constraints on the distribution of the encoded spirals.

Appendix A
The Reed-Solomon decoder and the decoding processing

Let's denote a candidate point list formed from $\mathbf{P}'$ as $\mathbf{R} = \{(\alpha_i, r_i)\}_{i=1}^{n}$. Assume there are at most e errors in $\mathbf{R}$, i.e., at most e values of i such that $r_i \neq \wp_k(\alpha_i)$. The decoding relies on the following two lemmas [45].

Lemma 1: There exists non-zero polynomials $E(x)$ of degree $\leq e$ and $Q(x)$ of degree $\leq k + e - 1$ such that

$$Q(\alpha_i) = r_i E(\alpha_i) \quad for \ \ all \ \ i = 1, 2, ..., n \qquad (22)$$

Proof: Let $\mathcal{I} = \{i_1, i_2, ..., i_m\}$ be the set of error positions, i.e., $i \in \mathcal{I}$ if $r_i \neq \wp_k(\alpha_i)$. Let

$$\begin{array}{l} E(x) = \prod_{j=1}^{m}(x - \alpha_{i_j}), \\ Q(x) = \wp_k(x)E(x). \end{array} \tag{23}$$

Thus, $E(x)$ has degree $m \leq e$. Since $\wp_k(x)$ has degree $\leq k-1$, it also follows that $Q(x)$ has degree $\leq e+k-1$. If i is not an error position, i.e., $i \notin \mathcal{I}$, then $r_i = \wp_k(\alpha_i)$, so $Q(\alpha_i) = \wp_k(\alpha_i)E(\alpha_i) = r_i E(\alpha_i)$. If i is an error position, i.e., $i \in \mathcal{I}$, then $E(\alpha_i) = 0$, so $Q(\alpha_i) = \wp_k(\alpha_i)E(\alpha_i) = 0 = r_i E(\alpha_i)$. Therefore, Eq. (22) holds for every α_i. ∎

Lemma 2: If $E(x)$ and $Q(x)$ satisfy Eq. (22), and the number of errors is at most $e = (n-k)/2$, then $Q(x) = \wp(x)E(x)$. Hence, we can compute $\wp(x) = Q(x)/E(x)$.

Proof: Both $Q(x)$ and $\wp(x)E(x)$ are polynomials of degree $\leq e+k-1$, so is their difference $D(x) = Q(x) - \wp(x)E(x)$. If i is not an error position, then $\wp_k(\alpha_i) = r_i$ and we have $D(x) = 0$. Since there are at least $n-e$ non-error positions, the polynomial $D(x)$ should have at least $n-e$ distinct roots in the Galois field $\mathbb{F}$. According to the theorem of algebra [55], if the number of distinct roots is larger than the degree of the polynomial, the polynomial will be identically zero. Therefore, if $n-e > e+k-1$ (i.e., $e < (n-k+1)/2$), we must have $D(x) = 0$ identically. Thus, when $e \leq (n-k)/2$, the polynomial can be reconstructed successfully. ∎

Given a candidate point list $\mathbf{R} = \{(\alpha_i, r_i)\}_{i=1}^{n}$, the decoding starts by assuming the maximum number of errors (i.e., $e = (n-k)/2$), which then introduces variables $u_0, u_1, ..., u_{k+e-1}$ and $v_0, v_1, ..., v_e$ to stand for the coefficients of $Q(x)$ and $E(x)$, so

$$\begin{array}{l} Q(x) = \sum_{i=0}^{k+e-1} u_i x^i, \\ E(x) = \sum_{i=0}^{e} v_i x^i. \end{array} \tag{24}$$

For each α_i, we substitute $x = \alpha_i$ in $Q(x)$ and $E(x)$ given above to evaluate Eq. (22), which results in a system of n linear equations. If the equations can not be solved, e is reduced by 1 and the above process is repeated until the equations can be solved or e is reduced to 0. $Lemma$ 1 and $Lemma$ 2 guarantee that, when the actual number of errors $e \leq (n-k)/2$, we can find a non-zero solution of this system to get $Q(x)$ and $E(x)$ explicitly, and the message polynomial can be recovered by $\wp_k(x) = Q(x)/E(x)$ with a reminder of 0.

APPENDIX B
PROBABILITY OF SUCCESSFUL AND INCORRECT DATA EXTRACTION

In this appendix, we first empirically estimate the probability of successful data extraction, which is a measure of how likely the secret message can be correctly extracted from a constructed fingerprint image. It is also a good indicator of the performance of the data extraction.

Let's denote the candidate point list with the most original points as the optimal point list. This list contains all the original points that can be detected. The probability to get an error point in the list equals to the probability that an original point can not be detected (i.e., the probability of missed spirals), an empirical estimation of which can be found in Table II. For each of the n points in the optimal point list, let's consider whether it is an error point or not as an event, so we can get n independent events from the list. The number of successes of the n events (i.e., the number of error points) is a random variable (say X) following the binomial distribution. Thus, the probability of getting e error points is given by

$$\Pr(X = e) = \binom{n}{e} d_s^e (1 - d_s)^{(n-e)}, \tag{25}$$

where d_s refers to the probability of missed spirals. The probability of successful reconstruction is the probability that the optimal point list contains at most $(n-k)/2$ error points, i.e.,

$$\mathbb{P} = \sum_{e=0}^{(n-k)/2} \Pr(X = e). \tag{26}$$

According to Table II, $\mathbb{P}$ is 1 for the fingerprint without post processing. When $n = 40$ and $k = 20$, $\mathbb{P}$ is $1 - 7.80 \times 10^{-9}$ for the binary fingerprint images, $1 - 5.28 \times 10^{-5}$ for the thinned fingerprint images and $1 - 1.10 \times 10^{-6}$ for the grayscale fingerprint images.

It is possible that the Reed-Solomon decoder incorrectly decodes the polynomial due to the limitation of error correction. In general, this can be verified by comparing the CRC bits. However, it may happen that the CRC bits of the extracted message are the same as the last 16 bits of the message decoded directly from the polynomial, even though the polynomial is incorrectly decoded. In such a case, we will extract an incorrect message, the probability of which depends on the capability of the decoder and the length of the CRC bits, which can be roughly estimated as

$$\mathbb{P}_e = (1 - \mathbb{P})2^{-16}. \tag{27}$$

For a binary fingerprint image with $\mathbb{P} = 1 - 7.80 \times 10^{-9}$, the value of $\mathbb{P}_e$ is 1.2×10^{-13}.

REFERENCES

[1] M. A. Akhaee, M. J. Saberian, S. Feizi, and F. Marvasti, "Robust audio data hiding using correlated quantization with histogram-based detector," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 834–842, 2009.

[2] X. Zhang, S. Wang, Z. Qian, and G. Feng, "Reference sharing mechanism for watermark self-embedding," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 485–495, 2011.

[3] W. Zhang, B. Chen, and N. Yu, "Improving various reversible data hiding schemes via optimal codes for binary covers," *IEEE Transactions on Image Processing*, vol. 21, no. 6, pp. 2991–3003, 2012.

[4] H. Cao and A. C. Kot, "On establishing edge adaptive grid for bilevel image data hiding," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1508–1518, 2013.

[5] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1905–1917, 2015.

[6] F. Huang, X. Qu, H. J. Kim, and J. Huang, "Reversible data hiding in jpeg images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1610–1621, 2016.

[7] D. Xu, R. Wang, and Y. Q. Shi, "Data hiding in encrypted h.264/avc video streams by codeword substitution," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 596–606, 2014.

[8] P. V. K. Borges, J. Mayer, and E. Izquierdo, "Robust and transparent color modulation for text data hiding," *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1479–1489, 2008.

[9] V. Itier, W. Puech, and J. P. Pedeboy, "Highcapacity data-hiding for 3d meshes based on static arithmetic coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 4575–4579.
[10] Y.-M. Cheng and C.-M. Wang, "A high-capacity steganographic approach for 3d polygonal meshes," *The Visual Computer*, vol. 22, no. 8.
[11] M. Fallahpour, D. Megias, and M. Ghanbari, "Reversible and high-capacity data hiding in medical images," *IET Image Processing*, vol. 5, no. 2, pp. 190–197, 2011.
[12] M. K. Kundu and S. Das, "Lossless roi medical image watermarking technique with enhanced security and high payload embedding," in *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2010, pp. 1457–1460.
[13] A. K. Jain and U. Uludag, "Hiding biometric data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1494–1498, 2003.
[14] S. Li and A. C. Kot, "Privacy protection of fingerprint database using lossless data hiding," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2010, pp. 1293–1298.
[15] ——, "Privacy protection of fingerprint database," *IEEE Signal Processing Letters*, vol. 18, no. 2, pp. 115–118, 2011.
[16] H. Wang and S. Wang, "Cyber warfare: Steganography vs. steganalysis," *Communications of the ACM*, vol. 47, no. 10, pp. 76–82, 2004.
[17] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.
[18] F. Li, X. Zhang, B. Chen, and G. Feng, "Jpeg steganalysis with high-dimensional features and bayesian ensemble classifier," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 233–236, 2013.
[19] T. H. Thai, R. Cogranne, and F. Retraint, "Statistical model of quantized dct coefficients: Application in the steganalysis of jsteg algorithm," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 1980–1993, 2014.
[20] H. Otori and S. Kuriyama, "Texture synthesis for mobile data communications," *IEEE Computer Graphics and Applications*, vol. 29, no. 6, pp. 74–81, 2009.
[21] K.-C. Wu and C.-M. Wang, "Steganography using reversible texture synthesis," *IEEE Transactions Image Process*, vol. 24, no. 1, pp. 130–139, 2015.
[22] J. Xu, X. Mao, X. Jin, A. Jaffer, S. Lu, L. Li, and M. Toyoura, "Hidden message in a deformation-based texture," *The Visual Computer*, vol. 31, no. 12, pp. 1653–1669, 2015.
[23] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Biometrics break-ins and band-aids," *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2105–2113, 2003.
[24] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition (Second Edition)*. Springer-Verlag, 2009.
[25] M. Vatsa, R. Singh, and A. Noore, "Feature based RDWT watermarking for multimodal biometric system," *Image and Vision Computing*, vol. 27, no. 3, pp. 293–304, 2009.
[26] C. L. Wilson, G. T. Candela, and C. I. Watson, "Neural network fingerprint classification," *Journal of Artificial Neural Networks*, vol. 1, no. 2, pp. 203–228, 1994.
[27] R. Cappelli, A. Erol, D. Maio, and D. Maltoni, "Synthetic fingerprint-image generation," in *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2000, pp. 471–474.
[28] J. L. Araque, M. Baena, B. E. Chalela, D. Navarro, and P. R. Vizcaya, "Synthesis of fingerprint images," in *Proceedings of IEEE International Conference on Pattern Recognition (ICPR)*, 2002.
[29] M. Kucken and K. C. Newell, "Fingerprint formation," *Journal of Theoretical Biology*, vol. 235, no. 1, pp. 71–83, 2005.
[30] A. M. Turing, "The chemical basis of morphogenesis," *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 237, no. 641, pp. 37–72, 1952.
[31] A. Witkin and M. Kass, "Reaction-diffusion textures," *ACM SIGGRAPH Computer Graphics*, vol. 25, no. 4, pp. 299–308, 1991.
[32] D. A. Garzón-Alvarado and A. M. R. Martinez, "A biochemical hypothesis on the formation of fingerprints using a turing patterns approach," *Theoretical Biology and Medical Modelling*, vol. 8, no. 1, p. 24, 2011.
[33] P. K. Maini, T. E. Woolley, R. E. Baker, E. A. Gaffney, and S. S. Lee, "Turing's model for biological pattern formation and the robustness problem," *Interface focus*, vol. 2, no. 4, pp. 487–496, 2012.
[34] S. T. Acton, D. P. Mukherjee, J. P. Havlicek, and A. C. Bovik, "Oriented texture completion by am-fm reaction-diffusion," *IEEE Transactions on Image Processing*, vol. 10, no. 6, pp. 885–896, 2001.
[35] M. Rahmes, J. D. Allen, A. Elharti, and G. B. Tenali, "Fingerprint reconstruction method using partial differential equation and exemplar-based inpainting methods," in *Biometrics Symposium, 2007*, 2007.
[36] K. G. Larkin and P. A. Fletcher, "A coherent framework for fingerprint analysis: are fingerprints holograms?" *Optics Express*, vol. 15, pp. 8667–8677, 2007.
[37] K. G. Larkin, D. J. Bone, and M. A. Oldfield, "Natural demodulation of two-dimensional fringe patterns. i. general background of the spiral phase quadrature transform," *Journal of the Optical Society of America A (Optics, Image Science and Vision)*, vol. 18, pp. 1862–70, 2001.
[38] D. Ghiglia and M. Pritt, *Two-dimentional phase unwrapping: Theory, Algorithms, and Software*. New York: John Wiley and Sons, 1998.
[39] D. J. Bone, "Fourier fringe analysis: the two-dimensional phase unwrapping problem," *Applied Optics*, vol. 30, pp. 3627–3632, 1991.
[40] B. Sherlock and D. Monro, "A model for interpreting fingerprint topology," *Pattern Recognition*, vol. 26, no. 7, pp. 1047 – 1055, 1993.
[41] R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry - two-dimensional phase unwrapping," *Radio Science*, vol. 23, no. 4, pp. 713–720, 1988.
[42] R. W. Zhou, C. Quek, and G. S. Ng, "A novel single-pass thinning algorithm and an effective set of performance criteria," *Pattern Recognition Letters*, vol. 16, no. 12, pp. 1267–1275, 1995.
[43] L. Hong, Y. F. Wan, and A. Jain, "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777–789, 1998.
[44] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.
[45] R. E. Blahut, *Algebraic codes on lines, planes, and curves*. Cambridge University Press, 2008.
[46] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *The Workshop on Information Hiding*, 1998, pp. 218–238.
[47] J. Cheng, A. C. Kot, J. Liu, and H. Cao, "Steganalysis of binary text images," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.
[48] J. Cheng, A. C. Kot, and S. Rahardjat, "Steganalysis of binary cartoon image using distortion measure," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007.
[49] J. Cheng and A. C. Kot, "Steganalysis of halftone image using inverse halftoning," *Signal Processing*, vol. 89, no. 6, pp. 1000–1010, 2009.
[50] K. L. Chiew and P. Josef, "Binary image steganographic techniques classification based on multi-class steganalysis," *Lecture Notes in Computer Science*, vol. 6047, pp. 341–358, 2010.
[51] Q. Zhao, Y. Zhang, A. K. Jain, N. G. Paulter, and M. Taylor, "A generative model for fingerprint minutiae," in *Processings of International Conference on Biometrics (ICB)*, 2013.
[52] S. Kullback, *Information Theory and Statistics*. John Wiley and Sons, 1959.
[53] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "Fvc2000: Fingerprint verification competition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 3, pp. 402–412, 2002.
[54] K. Nilsson and J. Bigun, "Localization of corresponding points in fingerprints by complex filtering," *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2135–2144, 2003.
[55] J. Von Zur Gathen and J. Gerhard, *Modern computer algebra*. Cambridge university press, 2013.