Hindawi
Security and Communication Networks
Volume 2021, Article ID 5344696, 12 pages
https://doi.org/10.1155/2021/5344696

Research Article

Privacy-Preserving Fingerprint Authentication Using D-H Key Exchange and Secret Sharing

Huiyong Wang,[1,2] Mingjun Luo ,[1] and Yong Ding [2,3]

[1]*School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin 541004, China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China*
[3]*Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518055, China*

Correspondence should be addressed to Yong Ding; stone_dingy@126.com

Received 7 June 2021; Revised 10 August 2021; Accepted 28 October 2021; Published 27 November 2021

Academic Editor: Savio Sciancalepore

Biometric based remote authentication has been widely deployed. However, there exist security and privacy issues to be addressed since biometric data includes sensitive information. To alleviate these concerns, we design a privacy-preserving fingerprint authentication technique based on Diffie-Hellman (D-H) key exchange and secret sharing. We employ secret sharing scheme to securely distribute fragments of critical private information around a distributed network or group, which softens the burden of the template storage center (TSC) and the users. To ensure the security of template data, the user's original fingerprint template is stored in ciphertext format in TSC. Furthermore, the D-H key exchange protocol allows TSC and the user to encrypt the fingerprint template in each query using a random one-time key, so as to protect the user's data privacy. Security analysis indicates that our scheme enjoys indistinguishability against chosen-plaintext attacks and user anonymity. Through experimental analysis, we demonstrate that our scheme can provide secure and accurate remote fingerprint authentication.

1. Introduction

Biometric based authentication mainly depends on individual biological characteristics (such as fingerprint, face, iris, and palm print, etc.) or behavioral traits (such as speech and signature, etc.), which is convenient, fast, and less likely to be forgotten, lost, or copied compared to traditional authentication methods like password/tokens [1]. However, biometric based authentication verifies an individual's identity according to the fixed natural connection between an individual and his or her biometrics. Once biometrics are stolen or forged and misused, it will result in significant losses for individuals, businesses, and even the government [2, 3]. This is because biological characteristics are usually unique and unchangeable, and they can never be revoked or reused once leaked or stolen [4]. Furthermore, the stolen biometric information may be used to acquire sensitive information about the owner, such as ethnic groups, genetic information, medical diseases, and health records [5]. As a result, developing a biometric authentication technique with privacy protection is critical [6–8].

Since the biometric authentication system needs to store and transmit individual biometrics in different entities, it is vulnerable to a variety of attacks [9]. For example, once the hackers or other malicious attackers break the template dataset and obtain the biometric data, they can access to all applications without user authorization where the same biological characteristics have been used [10]. A good biometric template protection system should have the following characteristics [11–13]:

(1) Diversity: the same biometric data should have different template representations in different databases to resist cross-matching attacks

(2) Reusability/revocability: the damaged or stolen template should be able to be revoked and a new

template can be regenerated based on the same biometric data, which cannot be matched with the damaged or stolen one successfully

(3) Noninvertibility: it is impossible to calculate or obtain a template with reasonable similarity to the original template from the protected template, so as to prevent the adversary's biological fraud attack

(4) Performance: the performance of the authentication system cannot be greatly reduced

To ensure the template's security in a biometric authentication system [2, 14], the extracted biometric template should be protected (encrypted or transformed) being stored.

In this paper, building upon the fingerprint feature representation "FingerCode," we propose a privacy-preserving fingerprint authentication scheme, which employs secret sharing and D-H key exchange. We summarise the contributions of the proposed method as follows:

(i) The secret sharing technology effectively reduces the risk of key leakage and relieves storage pressure for users and TSC.

(ii) The D-H key exchange technology used built bilinear groups can effectively conceal the real identity of users and generate different templates, which can resist replay attacks and cross-matching attacks.

(iii) The proposed scheme can prevent attacks like chosen-plaintext attacks (CPA) and cross-matching attacks and achieve the conditions of diversity, revocability, noninvertibility, and good performance.

The rest of this paper is organized as follows. Section 2 reviews the existing related work in this field. We introduce related knowledge in Section 3. The system model and construction goals are described in Section 4. Section 5 introduces ours fingerprint protection scheme and Section 6 proves its security. In Section 7, we discuss the test results, and finally, Section 8 concludes this paper.

2. Related Work

Over the years, a number of biometric template protection schemes have been proposed [15–19]. In this section, we review some approaches that have been proposed to deal with the security and privacy issues of fingerprint authentication systems, which are relevant to the proposed method.

Tuyls [10] and Ratha [20] outlined the types of attacks that the biometric system could face at each stage, and they believed that attacks would range from template collecting to final recognition decision. For example, in the process of template collection, it may be vulnerable to spoofing attacks, brute force attacks, device replacement attacks, and denial of service attacks. In the process of sending the collected fingerprint feature templates to the matcher, there may exist replay attacks, eavesdropping attacks, man-in-the-middle attacks, and brute force attacks [12]. These surveys show us the right direction for constructing a remote authentication scheme with privacy protection.

Different biometric authentication applications in smart cities can help for human lives; thus, there are many scholars who conducted research in this field. For example, El-Latif et al. [21] introduced a score level multibiometrics fusion approach for healthcare. In their proposed approach, they treat the biometric traits of patient/user as a request for healthcare assistance, which are processed in the cloud management and received by the caregiver with valid identification/verification for further treatment. Then they use the 1D-log Gabor iris features, two-directional modified fisher principal component analysis ($(2\ D)^2$MFPCA), and Complex Gabor Jet Descriptor face features for healthcare monitoring. In [22], Sedik et al. presented a system to discriminate pristine, adulterated, and fake biometrics in 5G-based smart cities by detecting alterations to biometric modalities. This work uses a convolutional neural network (CNN) and a hybrid structure that combines CNN with convolutional long-short term memory (ConvLSTM) as DLMs for the detection of different levels of alteration in biometrics that are employed for person identification, which provides a solution for different biometric authentication applications in secure smart cities.

Other research topics of biometrics involve protecting biometric templates and preserving users' privacy. Peng et al. [23] designed a novel biometric cryptosystem scheme based on random projection (RP) and backpropagation neural network (BPNN) to solve the problems of biometric template protection. The main idea of this work is to project the original biometric feature vector onto a fix-length feature vector of random subspace using random projection. After that, a backpropagation neural network model was applied to bind a projected feature vector with a random key. Finally, based on BPNN, a robust mapping between a projected feature vector and a random key is learned to generate an error-correction based biometric cryptosystem.

Zhu et al. [24] suggested a matrix based and efficient biometric outsourcing scheme with privacy protection in 2018, which has good performance in the preparation phase and the recognition phase. However, Liu et al. [25] in 2019 proved this scheme is insecure under chosen-plaintext attack (CPA) and proposed a new matrix transformation-based and privacy-preserving cloud computing, which not only can resist CPA but also has good recognition performance. Zhu et al. [26] proposed a privacy-preserving online fingerprint authentication scheme, named e-Finga, over encrypted outsourced data in 2018. e-Finga is based on bilinear mapping, which outsources the user fingerprint registered in the trust center to other servers under user's authorization, allowing the server to provide safe, accurate, and efficient identity verification services without leaking any fingerprint information. However, this method sends the user's fingerprint template in plain text to the trusted organization in the registration phase, which poses a certain security risk.

Trivedi et al. [27] proposed a noninvertible cancellable fingerprint template generation scheme based on information extracted from the Delaunay triangulation of minutiae points system in 2020. In their method, the location information of the fingerprint without minutiae points was

extracted from the triangles to solve the security problem brought by fingerprint reconstruction technology, which can handle the intraclass changes of fingerprints. In the same year, Kaur et al. [28] used cancellable biometrics and secret sharing to develop a privacy-preserving remote multiserver biometric authentication system. Their system allows user to operate safely on diverse applications and can prevent transmission attacks such as replay attacks, man-in-the-middle attacks, and database and server spoofing.

3. Preliminaries

In this section, we present the bilinear pairing technique and Decisional Diffie-Hellman assumption (DDH) problem and introduce the FingerCode-based identity matching algorithm which is the basis of our scheme.

3.1. FingerCode-Based Identity Matching Algorithm. The FingerCode-based identity matching algorithm utilizes a fixed-length "FingerCode" [29, 30] to characterize the fingerprint template, which uses a bank of Gabor filters to extract both local and global details in a fingerprint to produce a feature vector of fixed length, which is usually 640, where each element's length is 8-bit integer. When matching two fingerprints, we usually compute a certain distance between their FingerCodes and decide whether the distance is smaller than a predetermined threshold (τ). For example, given two fingerprint codes $X = [x_1, x_2, \ldots, x_n]$ and $Y = [y_1, y_2, \ldots, y_n]$, we can calculate their Euclidean distance using the following equation:

$$\|X - Y\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

The Euclidean distance is then compared to τ. If the Euclidean distance is less than τ, the two fingerprints will be judged to be from the same person. Otherwise, it is judged to be from different persons.

3.2. Bilinear Pairing. Suppose G and G_t are two cyclic groups of prime order p with generators g and g_t, respectively. Define a mapping $\hat{e}$: $G \times G \longrightarrow G_t$. We call this mapping bilinear if the following properties are satisfied:

(1) Bilinearity: $\forall \alpha, \beta \in G$ and $\forall \alpha,\ \beta \in Z_p^*$; $\hat{e}(\alpha^a, \beta^b) = \hat{e}(\alpha, \beta)^{ab}$

(2) Nondegeneracy: $\exists g,\ \hat{e}(g, g) \neq 1$

(3) Efficiency: $\forall \alpha, \beta \in G$, the mapping $\hat{e}$: $G \times G \longrightarrow G_T$ can be efficiently computed

3.2.1. The Decisional Diffie-Hellman Assumption (DDH). Let G be a cyclic group of prime order p with a generator g, and choose any triplet (g, g^a, g^b) for some random values $a, b \in_R Z_p^*$ to any probability polynomial-time (PPT) and attackers A; the advantage for $|\Pr[A(q, g, g^a, g^b, h_\chi)] - (1/2)|$ is negligible in l, where $\chi \in \{0, 1\}, h_0 = g^z, z \in_R Z_p^*, h_1 = g^{ab}$.

4. Framework and Design Goals

In this section, we formalize the architecture of the proposed scheme and summarise its security requirements as follows.

4.1. System Architecture. The proposed fingerprint template protection system consists of three types of entities (see Figure 1), which are the template storage center (TSC), the matcher (M_l), and the user (U_i). We assume that TSC is a trusted participant and M_l is a semihonest participant.

(1) TSC bootstraps the system, which generates and sends system parameters to the user and the matcher. In real-world applications, TSC can be an official entity of the government. The tasks of TSC are to store the user's encrypted fingerprint reference template and send the relevant reference template in ciphertext form to the matcher. Since all users must register in TSC, the number of registered users will be significantly large. If the fingerprint templates registered by users are stored together, TSC needs to spend a lot of time to find the reference fingerprint template corresponding to the user's query identity when users make queries. Thus, there are many substorages in TSC for user registrations and authentications. When the corresponding substorage cannot find the reference template corresponding to the user's query identity, TSC will retrieve and send the templates in all other substores to the matcher for matching.

(2) User terminal converts the user's physical fingerprints into digital fingerprint feature templates through the fingerprint sensor and uploads them to TSC for storage or poses queries to the matcher after encrypting the fingerprint feature templates.

(3) The matcher is mainly responsible for providing authentication service. Before providing fingerprint authentication services, the matcher must register in TSC to obtain the right to identify user's fingerprints. To evaluate whether the user is authenticated, the matcher compares the Euclidean distance between the user's query template and the reference template to a predefined threshold. Only the users who pass the authentication can access the related servers.

4.2. Functionalities and Design Goals. In fingerprint authentication system, malicious entities can obtain user's fingerprint template information as much as possible through eavesdropping, malicious attacks, etc. In order to achieve privacy-preserving remote identification, we determine the functionalities and design goals of the proposed system as follows:

(1) User anonymity: the matcher is curious about the data entered by users to obtain additional

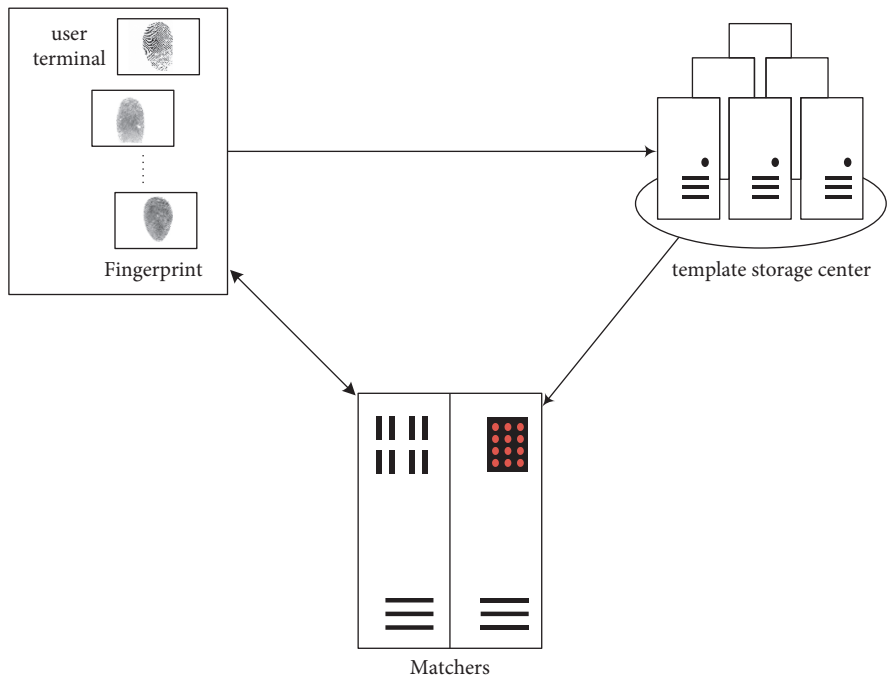

Figure 1: System architecture.

information. Thus, for each query, the original identity is always converted into a ciphertext format to guarantee privacy, and only TSC can decrypt the ciphertexts.

(2) Template security: TSC might also be attacked by an adversary. In order to prevent the user's original fingerprint template information from being leaked after the server is compromised, the user's reference template should be encrypted before being stored. Since the matcher might be curious about the templates input by the users and the templates stored in TSC, to prevent the adversary from obtaining relevant fingerprint template information through cross-matching attack, the users and TSC employ the same one-time key k_i to encrypt the query template. Thereby, even if the matcher or adversary obtained all the templates from TSC or the users, they cannot get any relevant information about the user's original fingerprint template.

(3) Efficiency: we ensure the efficiency of the system while ensuring the security and privacy of the user's fingerprint template.

4.3. Framework of Our System. Formally, the proposed system consists of nine polynomial-time computable algorithms/protocols, that is, Setup, UKgen, MKgen, Ureg, Mreg, FTenc, Aque, TSCres, and Matching.

(1) $\text{Setup}(1^l) \longrightarrow (\text{para}, x_{\text{TSC}}, y_{\text{TSC}})$: on input of a security parameter l, the fingerprint template protection system setup algorithm Setup, which is run by TSC, generates public key y_{TSC}, private key x_{TSC} for TSC, and the public parameter para for the system

(2) $\text{UKgen}(\text{para}) \longrightarrow (\text{Usk}_i, \text{Upk}_i, K_i)$: on input of the public parameter para, the user key generation algorithm UKgen, which is run by the users, generates a secret key Usk_i and a public key Upk_i and $K_i) \in Z_p^*$ for U_i

(3) $\text{MKgen}(\text{para}) \longrightarrow (\text{Msk}_l, \text{Mpk}_l)$: on input of the public parameter para, the matcher key generation algorithm MKgen, which is run by the matcher, generates a secret key Msk_l and a public key Mpk_l for M_l

(4) $\text{Ureg}(\text{para}, X_{U_i}', \text{Upk}_i, \text{ID}_{U_i}) \longrightarrow y_{T_j}$: on input of the public parameter para, the encrypted original fingerprint template X_{U_i}', the user's public key Upk_i, and the user's identity ID_{U_i}, the user registration algorithm Ureg, which is run by TSC, generates the pseudo-identity y_{T_j} for U_i

(5) $\text{Mreg}(\text{Mpk}_l, \text{ID}_{M_l})$: on input of the matcher's public key Mpk_l and identity ID_{M_l}, the matcher registration algorithm Mreg, which is run by TSC, completes matcher registration

(6) $\text{FTenc}(\text{para}, Y_i, y_{T_j}) \longrightarrow Y_i''$: on input of the public parameter *para*, the query fingerprint template Y_i, and the pseudo-identity y_{T_j} of U_i, the fresh fingerprint template encryption algorithm FTenc, which is run by U_i, generates an encrypted query fingerprint template Y_i''

(7) $\text{Aque}(\text{para}, (\text{Usk}_i, Y_i'', y_{T_j}), (\text{Msk}_l, ID_{M_l})) \longrightarrow (W, V)$: on input of the public parameter para, the fingerprint template authentication query algorithm Aque, which is jointly run by U_i and matcher with

$(\mathrm{Usk}_i, Y_i'', y_{T_j})$ and $(\mathrm{Msk}_l, ID_{M_l})$, respectively, outputs two ciphertexts (W, V)

(8) $\mathrm{TSCres}(\mathrm{para}, W, V, x_{\mathrm{TSC}}) \longrightarrow G_i$: on input of the public parameter para, the secret key x_{TSC} of TSC, and the ciphertexts (W, V), the template storage center response algorithm TSCres, which is run by *TSC*, outputs the ciphertext format G_i that includes the reference fingerprint template

(9) $\mathrm{Matching}(\mathrm{para}, G_i, \mathrm{Msk}_l) \longrightarrow \{1, \perp\}$: on input of the public parameter para, the ciphertext format G_i) that includes the reference fingerprint template, and the secret key Msk_l of the matcher, the matching algorithm, which is run by the matcher, outputs '1′ if the query is accepted; otherwise, it outputs '⊥'

4.4. Formal Security Definitions. We consider the case where malicious users may forge fingerprint templates during registration and collude with honest-but-curious matcher to get arbitrary plaintext and corresponding ciphertext of fake reference templates. Let A be a PPT adversary, who plays the following game with a challenger C.

Setup: on input of a security parameter l, the challenger C generates and publishes the public parameter para.

Keys generation: on input of the public parameter para, the challenger C runs user key generation algorithm and returns the secret key Usk_i, the public key Upk_i, and $K_i \in Z_p^*$ of U_i.

Challenge: the adversary A submits a pair of fingerprint templates plaintexts X_0 and X_1 with the same length to the challenger C. And then the challenger C chooses a bit $\chi \in \{0, 1\}$ uniformly and computes the ciphertext $X\prime = \mathrm{FTenc}(\mathrm{para}, K_i, X_\chi)$, which is given to the adversary A.

Guess: at the end of the game, the adversary A outputs a guess $\chi\prime$ and succeeds in the game if $\chi\prime = \chi$. Definition1 A fingerprint template protection scheme is ε – IND – CPA secure (indistinguishable against chosen-plaintext attacks), if any PPT adversary A has only negligible advantage in l in winning the above game; that is,

$$\mathrm{Adv}_{\epsilon,A}(l) = \left|\Pr[\chi\prime = \chi] - \frac{1}{2}\right| \le \varepsilon(l) \tag{2}$$

5. The Proposed Scheme

In this section, a concrete fingerprint template protection scheme based on bilinear groups is proposed. Table 1 summarises the frequently used notations.

5.1. System Setup. The template storage center TSC bootstraps system setup and mainly carries out two tasks:

(1) Preparatory work

(1) generates a bilinear mapping $\hat{e}$: $G \times G \longrightarrow G_t$, where G and G_T are cyclic groups with prime order p and g, h are two distinct generators of G

Table 1: Related symbols and their specific meanings.

Symbol	Specific meaning
G, G_T	Cyclic groups of prime order p
g, h	Two distinct generators of G
p	A big prime, the order of G and G_T
K_i	The key used to encrypt original template
x_{T_j}	The identity of substorages
y_{T_j}	The pseudo-identity of U_i
H_i	Cryptographic hash function for $1 \le i \le 4$
$\mathrm{ID}_{M_l}, \mathrm{ID}_{U_i}$	The identities of matcher M_l and U_i
$\mathrm{Usk}_i, \mathrm{Upk}_i$	The secret key and public key of U_i
$\mathrm{Msk}_l, \mathrm{Mpk}_l$	The secret key and public key of M_l
$x_{\mathrm{TSC}}, y_{\mathrm{TSC}}$	The secret key and public key of TSC
τ	Threshold

(2) selects random value $x_{\mathrm{TSC}} \in_R Z_p^*$, sets the x_{TSC} as secret key, and computes $y_{\mathrm{TSC}} = g^{x_{\mathrm{TSC}}}$, which is set as public key

(3) picks four cryptographic hash functions H_i and a threshold τ as H_1: $\{0,1\}^* \longrightarrow z_p^*$, H_2: $\{0,1\}^* \longrightarrow z_p^*$, H_3: $G \longrightarrow \{0,1\}^{\lambda_T + \log p}$, H_4: $G \longrightarrow \{0,1\}^{\lambda_I + 3\log p}$, where λ_T and λ_I denote the lengths of the identities of fingerprint template and the user's information, respectively

(4) makes the system parameters $\mathrm{para} = \langle G, G_T, \hat{e}, p, g, h, y_{\mathrm{TSC}}, H_1, H_2, H_3, H_4, \tau \rangle$ public

(2) Prepare substorage

The template storage center TSC prepares q (for example, $q = 5$) empty substorages for distributed storage of reference fingerprint templates registered by users.

(1) selects q random values x_{T_j}, $j = 1, 2, \ldots, q$, then computes $y_{T_j} = g^{x_{T_j}}$, and denotes the q substorages as $x_{T_1}, x_{T_2}, \ldots, x_{T_q}$

(2) For the security of x_{T_j}, TSC picks random q polynomials of degree $t-1$ like $f_j = x_{T_j} + a_{j1}x + a_{j2}x^2 + \cdots + a_{j(t-1)}x^{t-1}$, $j = 1, 2, \ldots, q$, which can translate x_{T_j} into shares for storage

5.2. User Key Generation. In the user key generation phase, U_i generates its own private key, public key, and K_i according to system parameters para. U_i selects random values $K_i, x_1, x_2 \in_R Z_p^*$, computes $y_1 = g^{x_1}$, $y_2 = g^{x_2}$, and sets its secret key and public key as $\mathrm{Usk}_i = (\mathrm{Usk}_{i,1}, \mathrm{Usk}_{i,2}) = (x_1, x_2)$ and $\mathrm{Upk}_i = (\mathrm{Upk}_{i,1}, \mathrm{Upk}_{i,2}) = (y_1, y_2)$, respectively. Finally, U_i picks a polynomial of degree $t-1$, such as $f(x) = K_i + b_1 x + b_2 x^2 + \cdots + b_{(t-1)}x^{t-1}$, which turns K_i into shares for storage.

5.3. Matcher Key Generation. In the matching key generation stage, the matcher M_l generates its own private key and public key according to the system parameters *para*. The

matcher M_l selects random value $x_l \in_R Z_p^*$ and then computes $y_l = g^{x_l}$. Finally, its secret key and public key are set as $\mathrm{Msk}_l = x_l$ and $\mathrm{Mpk}_l = y_l$, respectively.

5.4. User Registration. Each user's original fingerprint sample will be preprocessed to extract feature and obtain the corresponding fingerprint template $X_{U_i} = (x_{i1}, x_{i2}, \ldots, x_{in})$. Note that TSC might also be attacked by the adversary. Thus, the user first uses K_i to encrypt the original fingerprint template before registering in TSC:

$$\begin{cases} x_{i1}' = x_{i1} + H_1(K_i) \\ x_{i2}' = x_{i2} + H_1(K_i) \\ \vdots \\ x_{in}' = x_{in} + H_1(K_i) \end{cases} \tag{3}$$

Without loss of generality, we let $X_{U_i}' = (x_{i1}', x_{i2}', \ldots, x_{in}')$. And then the user U_i sends the public key Upk_i, its identity information ID_{U_i}, and the encrypted original fingerprint template X_{U_i}' to TSC. And TSC stores the user's information $\langle \mathrm{ID}_{U_i}, X_{U_i}' \rangle$ in q substorage randomly and sends y_{T_j} corresponding to the identity of each substorage to U_i, which is set as the user's pseudo-identity.

5.5. Matcher Registration. In the registration phase, the matcher submits its identity information ID_{M_l} and public key Mpk_l to TSC for signature verification.

5.6. Fresh Fingerprint Template Encryption. U_i should first obtain a fresh fingerprint template $Y_{U_i} = (y_{i1}, y_{i2}, \ldots, y_{in})$ with the fingerprint sensor before initiating an authentication query to the matcher. Then U_i picks a random value $x_{U_i} \in_R Z_p^*$ and a positive random value $\alpha_i \in_R Z_p^*$, computes $y_{U_i} = g^{x_{U_i}}, k_i = y_{T_j}^{x_{U_i}}$, and encrypts the fresh fingerprint template as follows:

$$\begin{cases} y_{i1}' = y_{i1} + H_1(K_i) + H_1(k_i) \\ y_{i2}' = y_{i2} + H_1(K_i) + H_1(k_i) \\ \vdots \\ y_{in}' = y_{in} + H_1(K_i) + H_1(k_i) \end{cases} \tag{4}$$

We denote this as $Y_{U_i}' = (y_{i1}', y_{i2}', \ldots, y_{in}')$ without loss of generality. Finally, U_i extends Y_{U_i}' to a n+3-dimensional vector $Y_{U_i}'' = \alpha_i (y_{i1}', y_{i2}', \ldots, y_{in}', (1/2), -\sum_{t=1}^{n} (y_{it}')^2, (1/2)\tau)$.

5.7. Authentication Query. The authentication query stage is mainly divided into two steps: U_i initiates a fingerprint authentication query request to the matcher and the matcher asks TSC for the reference fingerprint feature template corresponding to the user's claimed identity.

Step 1. U_i chooses three random values $s_1, s_2, s_3 \in_R Z_p^*$ after encrypting his fingerprint template into Y_{U_i} and computes the ciphertext $W = (w_1, w_2, w_3, w_4, w_5)$, where

$$\begin{aligned} w_1 &= g^{s_1}, \\ w_2 &= H_3\left(Mpk_l^{s_1}\right) \oplus \left(\alpha_i y_{i1}' \| \alpha_i y_{i2}' \| \ldots \| \alpha_i y_{in}' \| \alpha_i \frac{1}{2} \right\| -\alpha_i \sum_{t=1}^{n} (y_{it}')^2 \left\| \alpha_i \frac{1}{2} \tau \right\| s_2 \Big), \\ w_3 &= H_4\left(y_{\mathrm{TSC}}^{s_1}\right) \oplus \left(y_{U_i} \| y_{T_j} \| \mathrm{ID}_{U_i} \| s_3 \right), \\ w_4 &= h^{\mathrm{Usk}_{i,1} + s_1 s_3 + \mathrm{Usk}_{i,2} H_2\left(y_{U_i} \| y_{T_j} \| \mathrm{ID}_{U_i} \| \right)}, \\ w_5 &= h^{\mathrm{Usk}_{i,1} + s_2 \mathrm{Usk}_{i,2} + s_1 H_2 (w_1 \| w_2 \| w_3 \| w_4 \| T_i)}. \end{aligned} \tag{5}$$

Then, the user U_i sends (W, T_i) to the matcher, where T_i is a time stamp.

Step 2. Upon receiving (W, T_i) from U_i, the matcher runs the following steps to decrypt ciphertext w_2 with its secret key Msk_l. The matcher computes

$$H_3\left(w_1^{Msk_l}\right) \oplus w_2 \longrightarrow \alpha_i y_{i1}' \| \alpha_i y_{i2}' \| \ldots \| \alpha_i y_{in}' \| \alpha_i \frac{1}{2} \left\| -\alpha_i \sum_{t=1}^{n} (y_{it}')^2 \right\| \alpha_i \frac{1}{2} \tau \| s_2 \tag{6}$$

and checks whether the following condition is satisfied:

$$\widehat{e}(w_5, g) \stackrel{?}{=} \widehat{e}\left(h, \mathrm{Usk}_{i,1} \cdot \mathrm{Usk}_{i,2} \cdot w_1^{H_2(w_1 \| w_2 \| w_3 \| w_4 \| T)}\right). \quad (7)$$

If it is true, the matcher saves $(\alpha_i y_{i1}', \alpha_i y_{i2}', \ldots, \alpha_i y_{in}', \alpha_i 1/2, -\alpha_i \sum_{t=1}^{n} (y_{it}')^2, \alpha_i (1/2)\tau)$. And without loss of generality, we let $Y_{U_i}'' = \alpha_i (y_{i1}', y_{i2}', \ldots, y_{in}', (1/2), -\sum_{t=1}^{n} (y_{it}')^2, (1/2)\tau)$. Then the matcher picks a random value $s_4 \in_R Z_p^*$ and computes $V_l = (v_{l,1}, v_{l,2}) = (g^{s_4}, h^{\mathrm{Msk}_l + s_4 H_2(\mathrm{ID}_{M_l} \| T_l)})$. Finally, the matcher submits $(W, V_l, ID_{M_l}, T_i, T_l)$ to TSC, where ID_{M_l} and T_l are the identity of the matcher and a time stamp, respectively.

5.8. TSCResponse. Upon receiving $(W, V_l, ID_{M_l}, T_i, T_l)$ from the matcher, TSC performs the following process. TSC first decrypts ciphertext w_3 with its secret key x_{TSC}:

$$H_4(w_1^{x_{\mathrm{TSC}}}) \oplus w_3 \longrightarrow y_{U_i} \| y_{T_j} \| \mathrm{ID}_{U_i} \| s_3, \quad (8)$$

and checks whether the following conditions are satisfied:

$$\widehat{e}(w_4, g) \stackrel{?}{=} \widehat{e}\left(h, \mathrm{Upk}_{i,1} \cdot w_1^{s_3} \cdot \mathrm{Upk}_{i,2}^{H_2\left(y_{U_i} \| y_{T_j} \| \mathrm{ID}_{U_i}\right)}\right), \quad (9)$$

$$\widehat{e}(v_{l,2}, g) \stackrel{?}{=} \widehat{e}\left(h, \mathrm{Mpk}_l \cdot v_{1,1}^{H_2\left(\mathrm{ID}_{M_l} \| T_l\right)}\right). \quad (10)$$

If both are true, TSC accepts y_{U_i}, y_{T_j}, and ID_{U_i}. Then *TSC* finds the substorage x_{T_j} that corresponds to y_{T_j} and computes $k_i = y_{U_i}^{x_{T_j}}$. TSC should encrypt reference fingerprint templates corresponding to the identity ID_{U_i} in the substorage x_{T_j} with k_i

$$\begin{cases} x_{i1}'' = x_{i1}' + H_1(k_i) \\ x_{i2}'' = x_{i2}' + H_1(k_i) \\ \vdots \\ x_{in}'' = x_{in}' + H_1(k_i) \end{cases} \quad (11)$$

Without loss of generality, we denote it as $X_{U_i}'' = (x_{i1}'', x_{i2}'', \ldots, x_{in}'')$. Then TSC selects two random values $s_5, s_6 \in_R Z_p^*$ and a positive random value $\beta_M \in_R Z_p^*$, extends X_{U_i}'' to $(n+3)$-dimensional vector as $X_{U_i}'' = \beta_M (x_{i1}'', x_{i2}'', \ldots, x_{in}'', -\sum_{t=1}^{n} (x_{it}'')^2, (1/2), \tau)$, and computes the ciphertext $G_{j,i} = (G_{j,i,1}, G_{j,i,2}, G_{j,i,3})$, where

$$\begin{aligned} G_{j,i,1} &= g^{s_5}, \\ G_{j,i,2} &= H_3\left(\mathrm{Mpk}_l^{s_5}\right) \oplus \left(\beta_M x_{i1}'' \| \beta_M x_{i2}'' \| \ldots \| \beta_M x_{in}'' \| - \beta_M \sum_{t=1}^{n} \left(x_{it}''\right)^2 \| \beta_M \frac{1}{2} \| \beta_M \tau \| s_6\right), \\ G_{j,i,3} &= h^{x_{\mathrm{TSC}} + s_5 s_6 H_2\left(G_{j,i,1} \| G_{j,i,2} \| T_{j,i}\right)}. \end{aligned} \quad (12)$$

Finally, TSC sends $(G_{j,i}, T_{j,i})$ to the matcher, where $T_{j,i}$ is a time stamp.

5.9. Matching. Upon receiving $(G_{j,i}, T_{j,i})$, the matcher decrypts $G_{j,i,2}$ with its secret key Msk_l as

$$H_3\left(G_{j,i,1}^{\mathrm{Msk}_l}\right) \oplus G_{j,i,2} \longrightarrow \beta_M x_{i1}'' \| \beta_M x_{i2}'' \| \cdots \| \beta_M x_{in}'' \| - \beta_M \sum_{t=1}^{n} \left(x_{it}''\right)^2 \| \beta_M \frac{1}{2} \| \beta_M \tau \| s_6 \quad (13)$$

and checks whether the following condition is satisfied:

$$\widehat{e}\left(G_{j,i,3}, g\right) \stackrel{?}{=} \widehat{e}\left(h, y_{\mathrm{TSC}} \cdot G_{j,i,1}^{s_6 H_2\left(G_{j,i,1} \| G_{j,i,2} \| T_{j,i}\right)}\right). \quad (14)$$

If it is true, the matcher saves $(\beta_M x_{i1}'', \beta_M x_{i2}'', \ldots, \beta_M x_{in}'', -\beta_M \sum_{t=1}^{n} (x_{it}'')^2, \beta_M (1/2), \beta_M \tau)$. We denote it as X_{U_i}.

Then the matcher matches X_{U_i}'' with query fingerprint template Y_{U_i}''

$$
\begin{aligned}
D &= X''_{U_i} \cdot Y''_{U_i}, \\
&= \beta_M \left(x''_{i1}, x''_{i2}, \ldots, x''_{in}, -\sum_{t=1}^{n} \left(x''_{it} \right)^2, \frac{1}{2}, \tau \right), \\
&\quad \alpha_i \left(y'_{i1}, y'_{i2}, \ldots, y'_{in}, \frac{1}{2}, -\sum_{t=1}^{n} \left(y'_{it} \right)^2, \frac{1}{2}\tau \right), \\
&= \frac{1}{2}\alpha_i\beta_M \left(2\sum_{t=1}^{n} x''_{it}\, y'_{it} - \sum_{t=1}^{n} \left(x''_{it} \right)^2 - \sum_{t=1}^{n} \left(y'_{it} \right)^2 + \tau^2 \right), \\
&= \frac{1}{2}\alpha_i\beta_M \left(-\sum_{t=1}^{n} \left(\left(x''_{it} - y'_{it} \right)^2 + \tau^2 \right), \right. \\
&= \frac{1}{2}\alpha_i\beta_M \left(\tau^2 - \sum_{t=1}^{n} \left((x_{it} - y_{it})^2 \right) \right.
\end{aligned}
\tag{15}
$$

Since α_i and β_M are both positive, the matcher outputs "1" when the results of the user's query fingerprint template and the reference fingerprint template are greater than or equal to 0; otherwise, it outputs "$\perp$."

6. Soundness and Security

In this section, we show that our scheme is sound and enjoys various security and privacy as discussed below. Theorem 1 The proposed fingerprint template protection scheme is sound.

Proof. We only need to show equations (7)–(14) hold.

For a public key Upk_i of U_i, equation (7) holds as below:

$$
\begin{aligned}
\widehat{e}(w_5, g) &= \widehat{e}\left(h^{\mathrm{Usk}_{i,1} + s_2 \mathrm{Usk}_{i,2} + s_1 H_2 (w_1 \| w_2 \| w_3 \| w_4 \| T_i)}, g \right), \\
&= \widehat{e}\left(h, g^{\mathrm{Usk}_{i,1}} \left(g^{\mathrm{Usk}_{i,2}} \right)^{s_2} \left(g^{s_1} \right)^{H_2 (w_1 \| w_2 \| w_3 \| w_4 \| T_i)} \right), \\
&= \widehat{e}\left(h, \mathrm{Upk}_{i,1} \cdot \mathrm{Upk}_{i,2}^{s_2} \cdot w_1^{H_2 (w_1 \| w_2 \| w_3 \| w_4 \| T_i)} \right).
\end{aligned}
\tag{16}
$$

In the fingerprint template storage center response phase, ciphertexts w_4 and V_l can pass the verification of equations (9) and (10), respectively:

$$
\begin{aligned}
\widehat{e}(w_4, g) &= \widehat{e}\left(h^{\mathrm{Usk}_{i,1} + s_1 s_3 + \mathrm{Usk}_{i,2} H_2 \left(y_{U_i} \| y_{T_j} \| \mathrm{ID}_{U_i} \right)}, g \right), \\
&= \widehat{e}\left(h, g^{\mathrm{Usk}_{i,1}} \left(g^{s_1} \right)^{s_3} \left(g^{\mathrm{Usk}_{i,2}} \right)^{H_2 \left(y_{U_i} \| y_{T_j} \| \mathrm{ID}_{U_i} \right)} \right), \\
&= \widehat{e}\left(h, \mathrm{Upk}_{i,1} \cdot w_1^{s_3} \cdot \mathrm{Upk}_{i,2}^{H_2 \left(y_{U_i} \| y_{T_j} \| \mathrm{ID}_{U_i} \right)} \right), \\
\widehat{e}\left(v_{l,2}, g \right) &= \widehat{e}\left(h^{\mathrm{Msk}_l + s_4 H_2 \left(\mathrm{ID}_{M_l} \| T_l \right)}, g \right), \\
&= \widehat{e}\left(h, g^{\mathrm{Msk}_l} \left(g^{s_4} \right)^{H_2 \left(\mathrm{ID}_{M_l} \| T_l \right)} \right), \\
&= \widehat{e}\left(h, \mathrm{Mpk}_l \cdot v_{l,1}^{H_2 \left(\mathrm{ID}_{M_l} \| T_l \right)} \right).
\end{aligned}
\tag{17}
$$

For a valid ciphertext $G_{j,i}$, equation (14) holds in matching stage as follows:

$$
\begin{aligned}
\widehat{e}\left(G_{j,i,3}, g \right) &= \widehat{e}\left(h^{x_{\mathrm{TSC}} + s_5 s_6 H_2 \left(G_{j,i,1} \| G_{j,i,2} \| T_{j,i} \right)}, g \right), \\
&= \widehat{e}\left(h, g^{x_{\mathrm{TSC}}} \left(g^{s_5} \right)^{s_6 H_2 \left(G_{j,i,1} \| G_{j,i,2} \| T_{j,i} \right)} \right), \\
&= \widehat{e}\left(h, y_{\mathrm{TSC}} \cdot G_{j,i,1}^{s_6 H_2 \left(G_{j,i,1} \| G_{j,i,2} \| T_{j,i} \right)} \right).
\end{aligned}
\tag{18}
$$

□

Theorem 2. *The proposed scheme is secure. That is, suppose the DDH assumption holds; the fingerprint template of the developed scheme is ε − IND − CPA security.*

Proof. If there exists an adversary A that can break the fingerprint template protection scheme with nonnegligible probability $\varepsilon(l)$, then we can construct a PPT algorithm C to solve the underlying DDH problem.

We consider the following PPT algorithm C attempts to solve the DDH problem. Suppose C receives (G, q, g, h_1, h_2, h_3), $K \in_R Z_p^*$, and the system parameters para $= \langle G, G_T, \widehat{e}, p, g, h, y_{\mathrm{TSC}}, H_1, H_2, H_3, H_4, \tau \rangle$, where $h_1 = g^x$, $h_2 = g^y$, and h_3 is either g^{xy} or g^z (for uniform $x, y, z \in_R Z_p^*$). The goal of C is to determine which is the case. □

Algorithm C:. The algorithm is given (G, q, g, h_1, h_2, h_3), $K \in_R Z_p^*$, and the system parameters para $= \langle G, G_T, \widehat{e}, p, g, h, y_{\mathrm{TSC}}, H_1, H_2, H_3, H_4, \tau \rangle$ as input.

(1) Sets $PK = (G, q, g, h_1, h_2)$ and runs $A(PK)$ to obtain two fingerprint templates X_0, X_1 with $|X_0| = |X_1|$

(2) Chooses a bit $\chi \in \{0, 1\}$ uniformly, and sets $X'_\chi = X_\chi + H_1(K) + H_1(h_3)$

(3) Gives the ciphertext X'_χ to A and obtains an output bit χ'. If $\chi' = \chi$, outputs 1; otherwise, output 0

There are two cases about the behavior of C to consider:

Case1. The challenger C chooses random values $x, y, z \in_R Z_p^*$ and sets $h_1 = g^x, h_2 = g^y, h_3 = g^z$, and then C runs A on a public key constructed as (G, q, g, h_1, h_2) and returns a ciphertext as $X'_\chi = X_\chi + H_1(K) + H_1(g^z)$.

In this case, we can see that the view of A when run as a subroutine by C is distributed identically to $A's$ view in the game of Section 4.4. Since C outputs 1 exactly when the output χ' of A is equal to χ, we have

$$\Pr\left[C\left(G, q, g, g^x, g^y, g^z\right) = 1\right] = \Pr[\chi\prime = \chi] = \frac{1}{2}. \quad (19)$$

Case2. The challenger C chooses random values $x, y \in_R Z_p^*$ and sets $h_1 = g^x, h_2 = g^y, h_3 = g^{xy}$; then C runs A on a public key constructed as (G, q, g, h_1, h_2) and returns a ciphertext as $X'_\chi = X_\chi + H_1(K) + H_1(g^{xy})$.

In this case, we can see that the view of A when run as a subroutine by C is distributed identically to $A's$ view in the game of Section 4.4. Since C outputs 1 exactly when the output χ' of A is equal to χ, we have

$$\Pr\left[C\left(G, q, g, g^x, g^y, g^{xy}\right) = 1\right] = \Pr[\chi\prime = \chi] = \frac{1}{2}. \quad (20)$$

Under the assumption that the DDH problem is hard, there is a negligible function ϵ such that

$$\varepsilon(l) \geq \left|\Pr\left[C\left(G, q, g, g^x, g^y, g^z\right) = 1\right] - \Pr\left[C\left(G, q, g, g^x, g^y, g^{xy}\right) = 1\right]\right| = \left|\frac{1}{2} - \Pr[\chi' = \chi]\right|. \quad (21)$$

This implies that $\mathrm{Adv}_{\epsilon,A}(l) = |\Pr[\chi' = \chi] - (1/2)| \leq \varepsilon(l)$, which demonstrates the proposed fingerprint template scheme is $\varepsilon - \mathrm{IND} - \mathrm{CPA}$ secure.

In addition to satisfying the above security, a secure fingerprint template protection scheme also needs to satisfy diversity, noninvertibility, and performance. The experimental results and analysis in Section 7 have proved that the fingerprint template protection scheme proposed in this paper will not affect the performance of the fingerprint authentication system. Next, we analyze the security of the proposed scheme in other aspects.

(1) Noninvertibility: An adversary is impossible to calculate or recover a template with reasonable similarity to the original template from the transformed template. In our scheme, the original fingerprint template is first encrypted to $X\_U_i\left\{\hat{\prime}\right\}$ by adding each element of the original fingerprint template X_{U_i} to the hash value of K_i randomly selected by the user in registration phase. And in the query stage, each element of $X\_U_i\left\{\hat{\prime}\right\}$ is added to the hash value of a random number k_i again, which is generated by the Diffie-Hellman key exchange. Thus, to obtain the original template, the adversary must first obtain the value of k_i. However, to obtain k_i, it must solve the DDH assumption problem, which is hard. In this case, our scheme satisfies noninvertibility.

(2) Diversity and Revocability: If a stored template is damaged or stolen, we should be able to revoke it and issue a new one based on the same original template to replace it, and it cannot match the new template successfully. In this paper, we encrypt fingerprint template by exploiting the hash values of the random keys K_i and k_i, which is selected by user. Thus, different transformed fingerprint templates can be generated from the same sample by changing only K_i or only k_i or both of them. So our scheme satisfies diversity and revocability.

(3) User Anonymity: For each query, the original identity is always converted into a ciphertext format to guarantee privacy, where only TSC can decrypt the ciphertexts. Therefore, an adversary is infeasible to trace the original identity making the transaction.

Existing literature on biometric template protection suffers from the problem of key management; that is, whether the key is stored in the server or is kept by the user, there is a risk of key leakage or loss. In this paper, the secret sharing can safely distribute confidential data in an efficient, secure, and private manner without storing it on a centralized server, which can reduce the risk of information leakage. The proposed scheme satisfies the $\varepsilon - \mathrm{IND} - \mathrm{CPA}$ security, which is attributed to the D-H key exchange. The D-H key exchange allows the user and template storage center to encrypt the query fingerprint template with different random number k_i in each query, thus strengthening security.

7. Experimental Analysis

We implement our system by using MATLAB (2019b) as the programming language. All experimental results were run on Windows 10 with 8-core 3.00 GHz Intel i7 CPU and 16 GB RAM. We use the public fingerprint dataset (http://www.neurotechnology.com) composed of 408 grayscale fingerprint images acquired by a CrossMatch Verifier 300 sensor for performance testing. The dataset contains 8 images for each individual and each image is of 512 × 480 pixel size with a resolution equal 500 dpi [31]. Figure 2 shows two image samples in the dataset.

(a) (b)

Figure 2: Test image samples.

Table 2: Fingerprint template information with different feature numbers.

N_d	N_f	N_b	b (pixel)	N_k
640	8	5	20	16
384	8	4	25	12
192	8	3	20	8
96	4	3	33	8

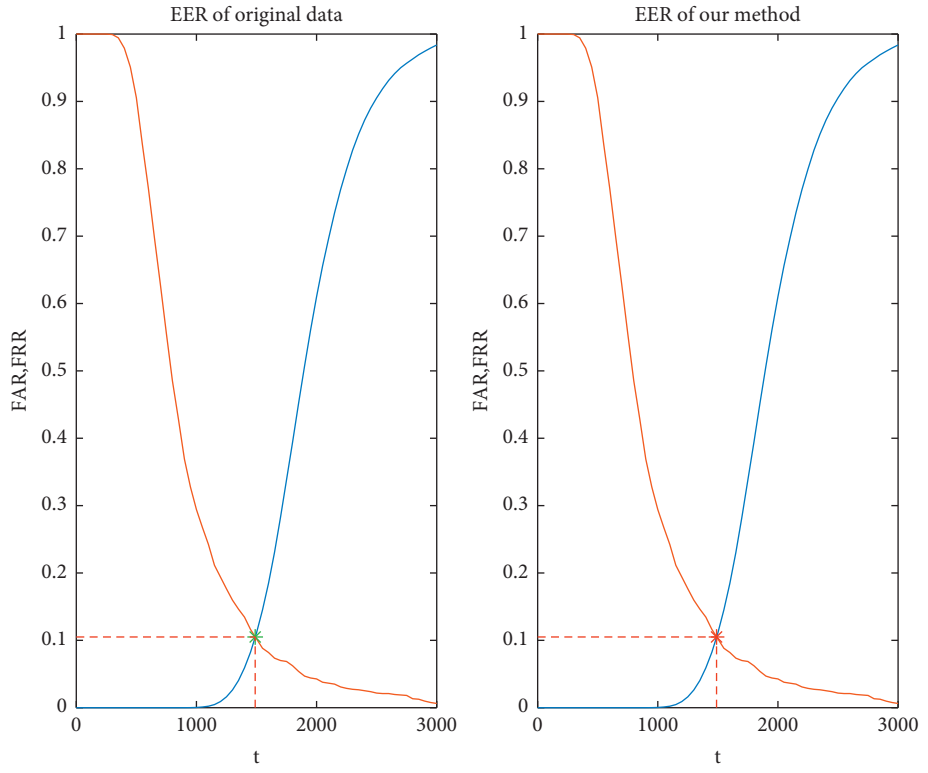

Figure 3: EER comparison before and after fingerprint template protection.

Our main purpose is to study the effectiveness of the fingerprint template protection, not the fingerprint recognition algorithm, so the final recognition accuracy is somewhat different from the accuracy obtained by the relevant fingerprint recognition algorithm. The performance of the proposed system on accuracy is measured as equal

Table 3: EER before and after protection (the results run on fingerprint template with 640 dimensions).

	Original data	Our method
Threshold (t)	1489	1489
EER	0.1050	0.1050

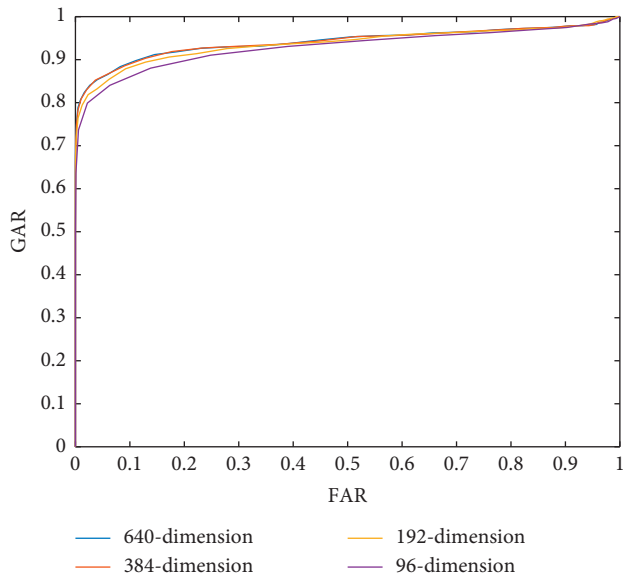

Figure 4: ROC curves of different feature dimensions.

error rate (EER). The EER is the error rate when the false acceptance rate (FAR) and the false rejection rate (FRR) are equal, which can reflect the overall accuracy of the fingerprint recognition system and the acceptability of the query user. In order to test the impact of the number of features in the FingerCode template on performance, we generated a total of 4 different configurations, corresponding to 4 sets of FingerCode vectors with a length ranging from 640 features (original configuration) to 96 features. Table 2 lists the detailed parameters of each configuration (N_d, N_f, N_b, b, N_k represent the dimension of features, the number of filters, the number of centric bands, the width of each band, and the number of sectors in each band, respectively).

The experimental results in Figure 3 and Table 3 prove that the protection method proposed in this paper has no effect on the performance of the fingerprint authentication system and fully demonstrates that the template protection scheme proposed in this paper guarantees the recognition performance and feasibility of the fingerprint recognition system.

It can be seen from Figure 4 that the performance between different feature quantity configurations is very close, but compared with the original 640-dimensional feature configuration, the performance of the other two configurations (192-dimension and 96-dimension) is slightly worse, which shows the fingerprint template feature reduction will influence the system accuracy.

8. Conclusion

In this paper, a privacy-preserving fingerprint authentication scheme is proposed. We utilize the secret sharing technology to store keys to reduce the risk of key leakage and exploit the D-H key exchange to conceal the real identity of the users and generate various fingerprint templates to prevent cross-matching attack over bilinear groups. And in order to protect privacy and confidentiality of all fingerprint templates, the matcher matches the templates in ciphertext format without destroying authentication accuracy. The designed framework maintains user anonymity, diversity, revocability, non-invertibility, and indistinguishability against chosen-plaintext attacks. Through security and experimental analysis, we demonstrate the security strength and the performance of the proposed system.

Since the user's original fingerprint template is stored in ciphertext format in the template storage center, it requires a user to reenroll if the template storage center is attacked and the templates in it are compromised, which is a limitation of the proposed method.

Recently, some novel fingerprint representations demonstrated superior authentication accuracy, which lay a great foundation for the future development of fingerprint template protection methods, for example, the Minutia Cylinder-Code (MCC) [32], a representation based on 3D data structures (called cylinders) and the DeepPrint [33], a fixed-length fingerprint representation of only 200 bytes. Built upon this novel fingerprint representation, we will investigate template protection for fingerprint biometric systems. In addition, multiple-biometric template protection and artificial intelligence (AI) based template data protection are worthy of further study.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This article is supported in part by the National Key R&D Program of China under Projects 2020YFB1006003 and 2020YFB1006004, the National Natural Science Foundation of China under Projects 61772150, 61862012, and 61962012, the Guangdong Key R&D Program under Project 2020B0101090002, the Guangxi Natural Science Foundation under Grants 2018GXNSFDA281054, 2019GXNSFFA245015, 2019GXNSFGA245004, and AD19245048, the Peng Cheng Laboratory Project of Guangdong Province PCL2018KP004, and the Innovation Project of GUET Graduate Education 2021YCXS115 and 2021YCXS116.

References

[1] Y. J. Lee, K. Bae, S. J. Lee, K. R. Park, and J. Kim, "Biometric key binding: fuzzy vault based on iris images," in *Proceedings of the International Conference on Biometrics*, pp. 800–808, Springer, Seoul, South Korea, August 2007.

[2] C. Rathgeb and A. Uhl, "A survey on biometric cryptosystems and cancelable biometrics," *EURASIP Journal on Information Security*, vol. 2011, no. 1, pp. 1–25, 2011.

[3] M. A. Acquah, N. Chen, J.-S. Pan, H.-M. Yang, and B. Yan, "Securing fingerprint template using blockchain and distributed storage system," *Symmetry*, vol. 12, no. 6, p. 951, 2020.

[4] L. James, A. K. Jain, D. Maltoni, and D. Maio, *Biometric Systems: Technology, Design and Performance Evaluation*, Springer Science & Business Media, Berlin, Germany, 2005.

[5] P. Elena and A. Mitrokotsa, "Privacy-preserving biometric authentication: challenges and directions," *Security and Communication Networks*, vol. 2017, Article ID 7129505, 9 pages, 2017.

[6] W. Qing, H. Zhu, R. Lu, and H. Li, "Achieve efficient and privacy-preserving online fingerprint authentication over encrypted outsourced data," in *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Paris, France, May 2017.

[7] S. Pan, Y. Shen, and W.-T. Zhu, *Security Analysis on Privacy-Preserving Cloud Aided Biometric Identification Schemes*, Springer-Verlag, Berlin, Germany, 2016.

[8] X. Dong, Z. Jin, A. B. Jin, M. Tistarelli, and K. S. Wong, "On the security risk of cancelable biometrics," 2019, https://arxiv.org/abs/1910.07770.

[9] A. B. J. Teoh and D. C. L. Ngo, "Cancellable biometerics featuring with tokenised random number," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1454–1460, 2005.

[10] P. Tuyls, B. Škoric, and T. Kevenaar, *Security with Noisy Data: On Private Biometrics, Secure Key Storage and Anti-counterfeiting*, Springer Science & Business Media, Berlin, Germany, 2007.

[11] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer Science & Business Media, Berlin, Germany, 2009.

[12] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," *EURASIP Journal on Applied Signal Processing*, vol. 2008, pp. 1–17, 2008.

[13] M. Shahzad, S. Wang, G. Deng, and W. Yang, "Alignment-free cancelable fingerprint templates with dual protection," *Pattern Recognition*, vol. 111, Article ID 107735, 2021.

[14] R. Cappelli, D. Maio, A. Lumini, and D. Maltoni, "Fingerprint image reconstruction from standard templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1489–1503, 2007.

[15] A. Anil Kumar and K. S. A. Kumar, "Security and performance enhancement of fingerprint biometric template using symmetric hashing," *Computers & Security*, vol. 90, Article ID 101714, 2020.

[16] A. Siswanto, N. Katuk, and K. R. Ku-Mahamud, "Chaotic-based encryption algorithm using henon and logistic maps for fingerprint template protection," *International Journal of Communication Networks and Information Security*, vol. 12, no. 1, pp. 1–9, 2020.

[17] S. Syed, V. Singh Baghel, I. I. Ganapathi, and S. Prakash, "Robust biometric authentication system with a secure user template," *Image and Vision Computing*, vol. 104, Article ID 104004, 2020.

[18] J. B. Kho, J. Kim, I.-J. Kim, and A. B. J. Teoh, "Cancelable fingerprint template design with randomized non-negative least squares," *Pattern Recognition*, vol. 91, pp. 245–260, 2019.

[19] A. Muhammed, N. C. Mhala, and A. R. Pais, "A novel fingerprint template protection and fingerprint authentication scheme using visual secret sharing and super-resolution," *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 10255–10284, 2021.

[20] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.

[21] A. Ahmed, A. El-Latif, M. S. Hossain, and N. Wang, "Score level multibiometrics fusion approach for healthcare," *Cluster Computing*, vol. 22, no. 1, pp. 2425–2436, 2019.

[22] S. Ahmed, H. Mohamed, A. El-Latif et al., "Deep learning modalities for biometric alteration detection in 5g networks-based secure smart cities," *IEEE Access*, vol. 9, pp. 94780–94788, 2021.

[23] J. Peng, B. Yang, B. B. Gupta, and A. A. El-Latif, "A biometric cryptosystem scheme based on random projection and neural network," *Soft Computing*, vol. 25, no. 11, pp. 7657–7670, 2021.

[24] L. Zhu, C. Zhang, C. Xu, X. Liu, and C. Huang, "An efficient and privacy-preserving biometric identification scheme in cloud computing," *IEEE Access*, vol. 6, pp. 19025–19033, 2018.

[25] C. Liu, X. Hu, Q. Zhang, J. Wei, and W. Liu, "An efficient biometric identification in cloud computing with enhanced privacy security," *IEEE Access*, vol. 7, pp. 105363–105375, 2019.

[26] H. Zhu, Q. Wei, X. Yang, R. Lu, and H. Li, "Efficient and privacy-preserving online fingerprint authentication scheme over outsourced data," *IEEE Transactions on Cloud Computing*, vol. 9, no. 2, pp. 576–586, 2021.

[27] A. K. Trivedi, D. M. Thounaojam, and S. Pal, "Non-invertible cancellable fingerprint template for fingerprint biometric," *Computers & Security*, vol. 90, Article ID 101690, 2020.

[28] H. Kaur and P. Khanna, "Privacy preserving remote multi-server biometric authentication using cancelable biometrics and secret sharing," *Future Generation Computer Systems*, vol. 102, no. 30–41, 2020.

[29] A. K. Jain, S. Prabhakar, H. Lin, and S. Pankanti, "Fingercode: a filterbank for fingerprint representation and matching," in *Proceedings of the. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 2, pp. 187–193, IEEE, Fort Collins, CO, USA, June 1999.

[30] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, 2000.

[31] M. Barni, T. Bianchi, D. Catalano et al., "A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates," in *Proceedings of the 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–7, IEEE, Washington, DC, USA, Sep 2010.

[32] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: a new representation and matching technique for fingerprint recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2128–2141, 2010.

[33] J. J. Engelsma, K. Cao, and A. K. Jain, "Learning a fixed-length fingerprint representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 1981–1997, 2021.