IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 13, 2019, accepted July 26, 2019, date of publication July 30, 2019, date of current version August 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931881

An Efficient Biometric Identification in Cloud Computing With Enhanced Privacy Security

CHUN LIU[1,2], XUEXIAN HU[1,2], QIHUI ZHANG[1,2], JIANGHONG WEI[1,2], AND WENFEN LIU[3]

[1]5th Department, PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China
[2]State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
[3]Department of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China

Corresponding author: Xuexian Hu (xuexian_hu@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702549, Grant 61862011, and Grant 61872449, in part by the Guangxi Natural Science Foundation under Grant 2018GXNSFAA138116, in part by the Guangxi Key Laboratory of Cryptography and Information Security under Grant GCIS201704, and in part by the Foundation of Science and Technology on Information Assurance Laboratory under Grant KJ-17-001.

ABSTRACT Biometric identification has developed rapidly in recent years because of its convenience and reliability. Due to the sensitivity of biometric data, many privacy-preserving biometric identification schemes have been put forward, exploiting either homomorphic encryption or matrix-transformation. However, existing schemes based on homomorphic encryption generally suffer from low computational efficiency, and existing matrix-transformation-based schemes are insufficiently secure. In this paper, we demonstrate that the matrix-transformation-based privacy-preserving biometric identification scheme recently proposed by Zhu et al. is vulnerable to a known-plaintext attack (KPA). To remedy this security flaw, we propose a new privacy-preserving biometric identification scheme, in which the property of the orthogonal matrix and additional randomness are utilized. Security analysis and comparisons indicate that our scheme can resist not only the KPA attack but also the more powerful chosen-plaintext attack (CPA), which is a reasonable attack in practice. Moreover, our scheme enjoys higher computational efficiency than other similar schemes, which implies our scheme can better support a huge database for practical biometric identification, and it also enhances privacy security of sensitive biometric data.

INDEX TERMS Biometric identification, privacy-preserving, CPA attack, cloud computing.

I. INTRODUCTION

Biometric identification provides a promising method in access control to authenticate users by their biometric traits, *i.e.* physiological traits (*e.g.* fingerprints, iris, face) and behavioral traits (*e.g.* voice, gait, typing rhythms). Biometric traits can not be lost, stolen, or forgotten like passwords, since they are naturally bound up with individuals [1], [2]. Just as Schneier has said: ''You are your key'' [3]. Due to such a strong bond, biometric identification is a much more reliable and convenient approach than passwords, the most traditional authentication method. With the boom of smartphones, the integration of biometric sensors into mobile phones has boosted the adoption of biometric technologies. For instance, more online banks seek to incorporate biometric identification into their systems [4]. In addition to the online payment, there are many other access control scenarios, which will lead to a growing demand for biometric technologies, such as industrial IoT deployment [5], telecare medical information system [6], smart city [7] and other applications [8]–[10]. According to Research and Markets Ltd., the largest market research store, the global biometric market will reach $42.4 billion by 2021 [11].

Despite the positive prospect and trend in biometric identification, there are still many challenges concerned with the privacy, security, and efficiency since biometric data is highly sensitive and impossible to be revoked and replaced once leaked [2]. For example, if a person's fingerprint is compromised, he/she can not change it like traditional passwords and will no longer rely on it as a security mechanism. Moreover, the biometric data may also reveal sensitive personal information such as genetic information and some information about users' diseases [12], [13]. Therefore, appropriate security and privacy protection scheme should be proposed to resist the disclosure and misuse of biometric data(*i.e. biometric template*).

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaofan He.

A. RELATED WORK

The increasing demand for a reliable and convenient authentication promotes the development of biometric identification. However, many data breaches raise increasing concerns on privacy issues recently. Various solutions on privacy-preserving biometric identification have been recommended.

Directly encrypting biometric template plaintexts and matching template ciphertexts bit by bit seem to be the most robust method to protect sensitive template data from disclosure [2]. However, since there are inherent noises in biometric feature extracting process, direct encrypting method tends to amplify small differences, generally resulting in the failure of identification. To improve the fault tolerance of template ciphertexts matching, Jin *et al.* [14] proposed a two-factor authentication scheme based on iterated inner products between the template and tokenized pseudo-random numbers, well-known as BioHashing. But the scheme's performance is not as good as claimed when an attacker steals the tokenized pseudo-random numbers [15]. Later, by resorting to error correcting codes, Juels and Sudan [16] presented a fuzzy vault scheme, but it is extremely vulnerable to record-multiplicity attacks, in which an attacker can recover a particular template from a collection of multiple enrollment template encodings [17].

Instead of bitwise matching of template ciphertexts, calculating the Euclidean distance between two templates is another way to determine whether they are from the same user. However, the computation of distance is usually conducted by the cloud server, who holds only the ciphertexts of the biometric templates for the sake of privacy protection. This leads to a challenge in how the cloud server computes the distance of template plaintexts through operations on the corresponding ciphertexts. Therefore, the promising homomorphic encryption is introduced to this area. Barni *et al.* [18] proposed a privacy-preserving fingerprint authentication scheme based on an additively homomorphic encryption called Pallier scheme [19]. However, due to the low performance, their scheme is limited by the size of database and number of concurrent requests. Later, Catalano and Fiore [20] presented a method to boost additively homomorphic encryption to a more complicated cryptosystem supporting degree-2 computation on encrypted data. Based on Dario et al.'s work and Paillier cryptosystem, Im *et al.* [21] implemented a palm print authentication. But the efficiency is not yet satisfactory. Further, Zhu *et al.* [22] designed a more efficient system model by utilizing BGN cryptosystem [23], a somewhat homomorphic encryption which is able to evaluate 2-DNF formulas on ciphertexts. Nevertheless, their experimental results are performed on a small dataset named FVC2006 which contains only 150 fingers, so it seems that their scheme can hardly support a huge database for practical usage.

To achieve higher performance and scalability of biometric identification, privacy-preserving schemes based on matrix transformation were proposed [24]–[27] as alternatives to those schemes based on homomorphic encryption. Yuan and Yu [24] presented the first efficient privacy-preserving biometric identification scheme based on matrix transformation. However, Zhu *et al.* [25] pointed out that their system can be destroyed by a collusion attack launched by malicious users and the cloud. To remedy the deficiency of [24], two improved protocols were put forward, in which additional randomness is introduced [26], [27]. Nevertheless, the computational efficiency of Hu et al.'s scheme [27] is not suitable for deployment in practical scenarios, which is even lower than [24]. Moreover, we find that both Zhu et al.'s scheme [26] and Hu et al.'s scheme [27] are still insufficiently secure as they are vulnerable to *known-plaintext attack* (KPA) under their security assumption.

B. OUR CONTRIBUTIONS

To further exploit the high performance of matrix transformation [28] and remedy the security flaws in previous schemes, we propose an efficient biometric identification with enhanced privacy security to resist not only the KPA attack but also the *chosen-plaintext attack*(CPA), which is reasonable in practice. Specifically, our main contributions can be summarized as follows:

- Based on the typical security assumption mentioned in Zhu et al.'s work [26], we demonstrate their scheme is not KPA-secure as they claimed, in which an attacker can recover any template by the collusion of cloud server with malicious users.
- We consider a more adversarial setting — CPA attack, which has been used as a de facto standard for checking the security of cryptographic schemes in classical cryptology [29]. Besides the typical security assumption of biometric identification [26], we find the CPA attack is also very reasonable in practice. We formally present a more powerful threat model by extending the typical security model with the reasonable CPA attack.
- We propose a new biometric identification scheme with enhanced privacy security. The security analysis shows our scheme achieves a higher level of privacy protection, in the sense that our proposed scheme can defend against not only attacks mentioned in [26] but also the CPA attack.
- We present a detailed implementation of the proposed protocol built with Python. Performance comparisons show that our proposed scheme provides higher computational efficiency than existing biometric identification protocols.

The rest of this paper is organized as follows. In Section II, we introduce preliminary knowledge of privacy-preserving biometric identification. We review Zhu et al.'s scheme in Section III and demonstrate its insecurity under a KPA attack in Section IV. In Section V, we propose our efficient biometric identification scheme, followed by security analysis and performance analysis in Section VI and VII. Finally, we conclude this paper in Section VIII.

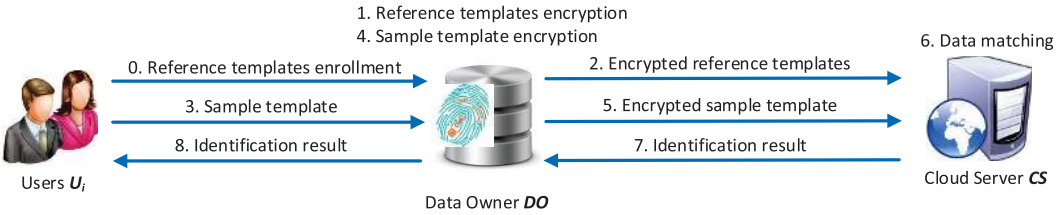


FIGURE 1. System model for biometric identification.

TABLE 1. Definitions and notations in this paper.

Symbol	Definition
b_i	The i-th reference template enrolled in the data owner
b_c	The sample template for identification
C_i	The ciphertext of the i-th reference template b_i
C_c	The ciphertext of the sample template b_c
m	The size of the reference template database
n	The dimension of the biometric template
τ	The presupposed threshold
sk	The secret key of the data owner, represented as the matrix
$Enc(sk, \cdot)$	Encrypt certain data with secret key sk
λ	The security parameter, representing the length of the secret key

II. PRELIMINARIES

In this section, we first introduce the system model and how to represent biometric templates. Then, we present the formalized threat model and security definition for biometric identification. In the description of threat model, we also analyze rationality of the CPA attack in practice. For the sake of clarity, we list main notations used throughout the paper in Table 1.

A. SYSTEM MODEL

The model in this paper follows the typical model introduced by Zhu et al's scheme [26]. As shown in Fig. 1, there exist three entities: data owner, cloud server and users. The data owner holds a database containing numerous biometric data $\langle b_i \rangle_{i=1}^{m}$ (*i.e.* reference template) which has been enrolled by users in the system. Based on this model, a biometric identification scheme generally includes three stages: preparation stage, request stage and identification stage. In the preparation stage, the data owner encrypts reference templates b_i and outsources ciphertexts $C_i \leftarrow Enc(sk, b_i)$ to the cloud server for storage. In the request stage, when a user requests for identification and sends his/her biometric trait b_c (*i.e.* sample template) to the data owner, the data owner encrypts the sample template plaintext b_c and sends the ciphertext $C_c \leftarrow Enc(sk, b_c)$ to the cloud server. In the identification stage, upon receipt of the request from the data owner, the cloud server performs operations on ciphertexts to figure out whether there is a matched reference template. Finally, the identification result will be sent to the data owner and user successively.

From the description of the system model, we can find the security of biometric templates depends on the operation $Enc(sk, \cdot)$. Additionally without loss of generality, we adopt FingerCode [30] to represent biometric templates similar to the existing work [24], [26], [27].

B. BIOMETRIC TEMPLATE REPRESENTATION

We apply FingerCode which is got by a filter-based algorithm [30] to represent biometric templates. Given a fingerprint image, the filter-based algorithm uses a bank of Gabor filters to extract features in the fingerprint and then outputs a compact fixed length (generally set as 640) vector, *i.e.* FingerCode.

The sample template $b_c = [b_{c1}, b_{c2}, \cdots, b_{cn}]$ and reference template $b_i = [b_{i1}, b_{i2}, \cdots, b_{in}]$ are considered from the same individual if and only if the Euclidean distance between them is below the presupposed threshold τ, *i.e.*

$$||b_i - b_c|| = \sqrt{\sum_{j=1}^{n} (b_{ij} - b_{cj})^2} < \tau.$$

C. THREAT MODEL

According to the typical threat model [26], the cloud server is assumed "honest-but-curious" or "semi-honest", which means the cloud server strictly executes the designed protocol, but tries to analyze the received messages to learn additional information about the honest users' biometric templates or the data owner's secret key. The semi-honest cloud server may even collude with malicious users further to attack the biometric identification system. On the basis of attack abilities, attackers are classified into three levels [26], with respect to ciphertext-only attack, known-candidate attack and known-plaintext attack respectively.

In addition, we further introduce a new level through considering the more powerful CPA attack, which is reasonable in practice.

Remark : As indicated in the survey of security and privacy issues for biometric-based remote authentication in cloud [31], malicious users have the ability to forge biometric templates during enrollment, which means the attacker can get any plaintext of fake reference template. Then in the preparation step, the reference template plaintext will be encrypted by the data owner and the ciphertext will be sent to the semi-honest cloud server. For the reason that the semi-honest cloud server can collude with malicious users, the attacker will get arbitrary plaintext and corresponding

ciphertext of fake reference templates. Therefore, CPA attack for biometric identification is reasonable in practice.

As described above, the CPA attack is also reasonable in practice. So we extend the typical threat model [26] by adding the CPA attack as level-4. To better formalize the strength of attackers, we classify attackers as follows:

- Level 1: Attackers can only observe the encrypted template in the cloud. This follows the well-known ciphertext-only attack model [29].
- Level 2: In addition to the encrypted templates in the cloud, attackers can access to some template plaintexts but do not know the corresponding ciphertexts in the encrypted database, similar to the known-candidate attack model [32].
- Level 3: Besides all the abilities in level-2, attackers are able to get a set of template plaintexts and know the corresponding ciphertexts. This level follows the known-plaintext attack (KPA) model [29].
- Level 4: With enhanced ability, malicious users can forge templates during biometric database enrollment and collude with the semi-honest cloud server [31]. So, attackers can get any template plaintext and the corresponding ciphertext, which follows the chosen-plaintext attack (CPA) model [29].

D. SECURITY DEFINITION

For the threat model, a higher-level attack is more powerful than a lower-level one. If a scheme can defend against a higher-level attack, it can resist a lower-level one as well. So we define security resisting above threat model based on level-4 attack, *i.e.* CPA attack.

We first define the CPA indistinguishability(IND-CPA) experiment for the biometric identification in Experiment 1.

Experiment 1 IND-CPA Experiment $\mathrm{CPA}_{\mathcal{A}}(\lambda)$

1: Given the security parameter λ, the data owner, *i.e.* challenger $\mathcal{C}$, generates a secret key sk.
2: The attacker $\mathcal{A}$ is given oracle access to $Enc(sk, \cdot)$ and outputs a pair of biometric template plaintexts b_0 and b_1 of the same length to the data owner.
3: A random bit $i \leftarrow \{0, 1\}$ is chosen uniformly by the data owner, and then a ciphertext $Enc(sk, b_i)$ is computed and given to the attacker.
4: The attacker continues to have oracle access to $Enc(sk, \cdot)$, and outputs a bit i'.
5: If $i = i'$, the output of the experiment is 1. Otherwise, it is 0.

On the basis of the IND-CPA experiment $\mathrm{CPA}_{\mathcal{A}}(\lambda)$, we define the CPA security for biometric identification protocols as Definition 1.

Definition 1: We say that a biometric identification protocol is CPA-secure, if there exists a negligible function $negl$ such that, for all polynomial-time attackers $\mathcal{A}$, the probability

$$\left|\Pr(\mathrm{CPA}_{\mathcal{A}}(\lambda) = 1) - \frac{1}{2}\right| \leq negl(\lambda)$$

III. REVIEW OF ZHU ET AL.'S SCHEME

In this section, we will review Zhu et al.'s scheme [26] in detail.

In addition to the three stages presented in system model, [26] also includes a retrieval stage because the cloud server only gets an index of the most probably matched template in the identification stage. In the retrieval stage, after receiving this index from cloud server, the data owner retrieves the most probably matched template plaintext according to the index, calculates the Euclidean distance between it and the sample template plaintext to decide whether the user is legitimate. The detailed protocol is described as follows.

A. PREPARATION STAGE

The data owner holds a database containing numerous reference templates $< b_i >_{i=1}^{m}$, where b_i represents the i-th reference template vector derived from fingerprint image using FingerCodes algorithm [30]. In more detail, b_i is a n-dimensional vector *i.e.* $b_i = [b_{i1}, b_{i2}, \cdots, b_{in}]$, with l bits of each element where generally $n = 640$ and $l = 8$. Data owner first extends b_i as $B_i = [b_{i1}, b_{i2}, \cdots, b_{i(n+1)}]$ by adding a $(n+1)$-th element, where $b_{i(n+1)} = -0.5(b_{i1}^2 + b_{i2}^2 + \cdots + b_{in}^2)$.

Then, the data owner randomly generates $(n+1) \times (n+1)$ matrices M_1, M_2 and a $(n+1)$ vector H as secret key, where M_1 and M_2 are invertible. And the data owner further performs the following encryption operations:

$$C_i = B_i \times M_1$$
$$C_h = M_2^{-1} \times H^T$$

Subsequently, the data owner uploads encrypted database (C_i, C_h, I_i) to cloud, in which I_i is an index associated with each biometric template B_i.

B. REQUEST STAGE

When a user wishing to access the system inputs a sample template $b_c = [b_{c1}, b_{c2}, \cdots, b_{cn}]$, the data owner extends b_c to B_c by adding a $(n+1)$-th element equaling to 1.

The data owner then generates a $(n+1) \times (n+1)$-dimensional matrix E, where the i-th row vector is represented as $E_i = [E_{i1}, E_{i2}, \cdots, E_{i(n+1)}]$. Additionally, the first n elements of E_i are random and the $(n+1)$-th element is set as $E_{i(n+1)} = (1 - \sum_{j=1}^{n} E_{ij} * H_j)/H_{n+1}$. After that, the data owner performs following operation to hide B_c:

$$F_c = [E_1^T \times b_{c1}, E_2^T \times b_{c2}, \cdots, E_{(n+1)}^T \times b_{c(n+1)}]^T$$

For security of F_c during transmission to cloud, the data owner further encrypts F_c with the secret key and a random integer $r(r > 0)$ as follows:

$$C_f = M_1^{-1} \times r \times F_c \times M_2$$

Note that the value of r is fixed for this request.

Then, C_f is sent from data owner to cloud server for identification.

C. IDENTIFICATION STAGE

After receiving C_f, the cloud searches the encrypted database to find the template having the minimum Euclidean distance with B_c, and responds with its index to data owner.

The cloud computes a relative distance P_i between a reference template b_i and the sample template b_c:

$$\begin{aligned} P_i &= C_i \times C_f \times C_h \\ &= B_i \times M_1 \times M_1^{-1} \times r \times F_c \times M_2 \times M_2^{-1} \times H^T \\ &= B_i \times r \times F_c \times H^T \\ &= r \times B_i \times [b_{c1} \times H \times E_1^T, \cdots, b_{c(n+1)} \times H \times E_{n+1}^T]^T \\ &= r \times B_i \times B_c^T \\ &= \sum_j^{n+1} r \times b_{ij} \times b_{cj} \end{aligned}$$

To compare the Euclidean distance between sample template b_c and any other two reference templates b_x, b_y where $1 \le x, y \le m, x \ne y$, the cloud should also perform following operation:

$$\begin{aligned} P_x - P_y &= \sum_{j=1}^{n+1} r \times b_{xj} \times b_{cj} - \sum_{j=1}^{n+1} r \times b_{yj} \times b_{cj} \\ &= (\sum_{j=1}^{n} r \times b_{xj} \times b_{cj} - 0.5 \sum_{j=1}^{n} r \times b_{xj}^2) \\ &\quad -(\sum_{j=1}^{n} r \times b_{yj} \times b_{cj} - 0.5 \sum_{j=1}^{n} r \times b_{yj}^2) \\ &= 0.5r(dist_{yc}^2 - dist_{xc}^2) \end{aligned}$$

where $dist_{xc}$ (*resp.* $dist_{yc}$) is the Euclidean distance between b_x and b_c (*resp.* b_y and b_c). If $P_x - P_y > 0$, b_x matches b_c better. Otherwise b_y does.

After iteration of comparisons, the ciphertext of template b_i will be found which has the minimum Euclidean distance with b_c. The corresponding index I_i will be transmitted from cloud to data owner.

D. RETRIEVAL STAGE

On receipt of the index I_i, the data owner searches the corresponding reference template b_i and calculates the Euclidean distance $dist_{ic}$ between b_i with b_c. Finally, the data owner compares $dist_{ic}$ with the presupposed threshold τ. If $dist_{ic} < \tau$, the user is authenticated to access the system. Otherwise, the identification fails.

IV. KPA ATTACK ON ZHU ET AL.'S SCHEME

In this section, based on the typical security assumption of Zhu et al.'s scheme [26], we present a KPA attack against [26] to recover the secret key M_1 (*i.e.*, achieve a total break) and obtain arbitrary honest user's biometric template, which implies [26] doesn't guarantee the privacy-preserving requirement as they claimed.

A. RECOVER THE SAMPLE TEMPLATE

According to the KPA attack for biometric identification, the attacker can get a set of plaintexts b_i^* enrolled in the data owner and corresponding ciphertexts C_i^*.

In the preparation stage, these leaked templates $b_i^* = [b_{i1}^*, b_{i2}^*, \cdots, b_{n1}^*]$ are extended to $B_i^* = [b_{i1}^*, b_{i2}^*, \cdots, b_{in}^*, b_{i(n+1)}^*]$, where $b_{i(n+1)}^* = -0.5(b_{i1}^{*2} + b_{i2}^{*2} + \cdots + b_{in}^{*2})$. And then, the data owner encrypts these templates and outsources the ciphertexts C_i^*, C_h to the cloud server. Assuming $(n+1)$ plaintext-ciphertext pairs of leaked templates are known to the attacker.

In the request stage, an honest user wants to be identified and submits his/her template plaintext $b_c = [b_{c1}, b_{c2}, \cdots, b_{cn}]$ to the data owner. At first, this sample template b_c will be extended to $B_c = [b_{c1}, b_{c2}, \cdots, b_{cn}, b_{c(n+1)}]$ where $b_{c(n+1)} = 1$. Then, the data owner will encrypt the sample template and send the ciphertext $C_f = M_1^{-1} \times r \times F_c \times M_2$ to the cloud server. Recall that value of the random number r is fixed for this request.

In the identification stage, when the request arrives, the cloud server calculates the relative distance $P_i^* = C_i^* \times C_f \times C_h = \sum_j^{n+1} r \times b_{ij}^* \times b_{cj}$. Due to the collusion of the malicious users with the cloud server, values of b_i^* and P_i^* are known by the attacker. And then, the attacker can get equations as follows:

$$\begin{cases} P_1^* = r \times (\sum_{j=1}^{n} b_{1j}^* b_{cj} - 0.5 \sum_{j=1}^{n} b_{1j}^{*2}) \\ P_2^* = r \times (\sum_{j=1}^{n} b_{2j}^* b_{cj} - 0.5 \sum_{j=1}^{n} b_{2j}^{*2}) \\ \vdots \\ P_{n+1}^* = r \times (\sum_{j=1}^{n} b_{(n+1)j}^* b_{cj} - 0.5 \sum_{j=1}^{n} b_{(n+1)j}^{*2}) \end{cases}$$

The first equation can be transformed as $r = \frac{P_1^*}{(\sum_{j=1}^{n} b_{1j}^* b_{cj} - 0.5 \sum_{j=1}^{n} b_{1j}^{*2})}$. After substituting the r into the other n equations, the attacker will get:

$$\begin{cases} \sum_{j=1}^{n} (P_2^* b_{1j}^* - P_1^* b_{2j}^*) b_{cj} = 0.5 \sum_{j=1}^{n} (P_2^* b_{1j}^{*2} - P_1^* b_{2j}^{*2}) \\ \sum_{j=1}^{n} (P_3^* b_{1j}^* - P_1^* b_{3j}^*) b_{cj} = 0.5 \sum_{j=1}^{n} (P_3^* b_{1j}^{*2} - P_1^* b_{3j}^{*2}) \\ \vdots \\ \sum_{j=1}^{n} (P_{(n+1)}^* b_{1j}^* - P_1^* b_{(n+1)j}^*) b_{cj} \\ \qquad = 0.5 \sum_{j=1}^{n} (P_{(n+1)}^* b_{1j}^{*2} - P_1^* b_{(n+1)j}^{*2}) \end{cases}$$

As there are n unknowns and n linear equations, the attacker can work out these equations and recover the sample template b_c.

Remark: With respect to Hu et al.'s scheme [27], it is easy to note that their relative distance $T_i = \sum_{j=1}^{n} r_c b_{ij} b_{cj} - 0.5 \sum_{j=1}^{n} r_c b_{ij}^2 + r_c'$ has a similar drawback to the definition of P_i presented above. Therefore, one can demonstrate that a KPA attacker could recover any sample template similar to the above attack, thus damage the security of Hu et al.'s scheme.

B. TOTAL BREAK BY KPA ATTACK

As described above, the attacker is able to get $(n+1)$ pairs of B_i^* and C_i^*. The ciphertext $C_i^* = B_i^* \times M_1 = [c_{i1}^*, c_{i2}^*, \cdots, c_{i(n+1)}^*]$, where M_1 can be represented as

$$M_1 = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1(n+1)} \\ m_{21} & m_{22} & \cdots & m_{2(n+1)} \\ \vdots & \vdots & \ddots & \vdots \\ m_{(n+1)1} & m_{(n+1)2} & \cdots & m_{(n+1)(n+1)} \end{bmatrix}$$

Take the 1-st column of M_1 as an example. Given B_i^* and C_i^*, the attacker can get equations as follows:

$$\begin{cases} \sum_{j=1}^{n+1} b_{1j}^* m_{j1} = c_{11}^* \\ \sum_{j=1}^{n+1} b_{2j}^* m_{j1} = c_{21}^* \\ \vdots \\ \sum_{j=1}^{n+1} b_{(n+1)j}^* m_{j1} = c_{(n+1)1}^* \end{cases}$$

As there are $(n+1)$ unknowns and $(n+1)$ linear equations, the attacker can work out the 1-st column vector of M_1, *i.e.* $[m_{11}, m_{21}, \cdots, m_{(n+1)1}]^T$. Other column vectors of M_1 can be worked out similarly. Therefore, the secret key M_1 is recovered.

C. RECOVER THE REFERENCE TEMPLATE

After working out M_1, all the reference templates b_i of honest users can be recovered from the corresponding ciphertext C_i.

In summary, Zhu et al.'s scheme [26] will be broken by the KPA attack for biometric identification(*i.e.* level-3 attack), revealing the secret key M_1, all reference templates b_i and sample templates b_c.

Remark : Since that the CPA attack is more powerful than KPA attack, Zhu et al.'s scheme [26] is also vulnerable to CPA attack (*i.e.* level-4 attack).

V. OUR PROPOSED SCHEME

In this section, we present the detailed description of our biometric identification scheme.

Our proposed scheme, as an enhancement of Zhu et al.'s scheme [26], follows a similar designing paradigm as [26]. Nevertheless, there are two main differences. Firstly, we adopt a novel method to extend vectors of reference and sample template plaintexts by introducing additional randomness, which makes our scheme able to resist not only the KPA attack but also the CPA attack. Secondly, we use random orthogonal matrices instead of general random matrices. Owing to the property of orthogonal matrices, the computational efficiency of identification stage is increased by 98.95% (see details in Section VII). In addition, the identification result will be obtained directly by cloud server without the redundant retrieval stage, which makes our scheme suitable for more application scenarios.

A. PREPARATION STAGE

In this stage, the i-th reference template vector b_i is extracted from users using FingerCode [30] and enrolled in the data owner. It is extended to a $(n+5)$-dimensional vector as $b_i' = [\alpha_i b_{i1}, \alpha_i b_{i2}, \cdots, \alpha_i b_{in}, -\frac{1}{2}\alpha_i \sum_{j=1}^{n} b_{ij}^2, \alpha_i, \frac{1}{2}\alpha_i \tau^2, r_i, 0]$, where α_i is a random positive real number, r_i is a random real number named security factor, and τ is the presupposed threshold.

Then, the data owner generates a random $(n+5)\times(n+5)$-dimensional orthogonal matrix M as a secret key and encrypts the reference templates by computing $C_i = b_i' M$.

Finally, the data owner outsources all C_i to the cloud server.

B. REQUEST STAGE

On receiving an identification request b_c from a user, data owner extends it as $b_c' = [\beta_c b_{c1}, \beta_c b_{c2}, \cdots, \beta_c b_{cn}, \beta_c, -\frac{1}{2}\beta_c \sum_{j=1}^{n} b_{cj}^2, \beta_c, 0, r_c]$, in which β_c is a random positive real number and r_c is another security factor.

Then the data owner encrypts b_c' with the secret key M and gets $C_c = b_c' M$.

Finally, the C_c is transmitted to the cloud server.

C. IDENTIFICATION STAGE

When identification request arrives at cloud server, the cloud server performs inner product of C_i and C_c, *i.e.* relative distance $R_i = C_i \cdot C_c$.

If $R_i > 0$, the user is identified and the result will be sent to the data owner, permitting his/her access to the system. Otherwise, the request is denied.

D. CORRECTNESS ANALYSIS

The transformation $b \mapsto bM$ is called an orthogonal transformation in linear algebra, since M is an orthogonal matrix. Based on the property of this transformation, the inner product of two vectors is maintained.

In other words, there are two vectors a and b, the inner product of them can be represented as $a \cdot b = a \times b^T$. Given an orthogonal matrix M, there exists following property:

$$\begin{aligned} aM \cdot bM &= aM \times (bM)^T \\ &= aMM^T b^T \end{aligned}$$

$$= a \times b^T$$
$$= a \cdot b$$

Therefore, the relative distance R_i can be figured out as follows:

$$\begin{aligned} R_i &= C_i \cdot C_c \\ &= b_i' M \cdot b_c' M \\ &= b_i' \cdot b_c' \\ &= \alpha_i \beta_c (\sum_{j=1}^{n} b_{ij} b_{cj} - \frac{1}{2}\sum_{j=1}^{n} b_{ij}^2 - \frac{1}{2}\sum_{j=1}^{n} b_{cj}^2 + \frac{1}{2}\tau^2) \\ &= \frac{1}{2}\alpha_i \beta_c [\tau^2 - \sum_{j=1}^{n} (b_{ij} - b_{cj})^2] \end{aligned}$$

If $R_i > 0$, $\sum_{j=1}^{n} (b_{ij} - b_{cj})^2 < \tau^2$, *i.e.* the relative distance R_i meets the identification conditions.

VI. SECURITY ANALYSIS

As indicated in Section II, a CPA-secure protocol is capable of resisting the proposed threat model. In this section, we will analyze the CPA attack indistinguishability of our biometric identification scheme by considering the advantage of attackers. However, indistinguishability of ciphertexts is not enough to resist CPA attacks in the real situation, where the goal of attackers is to recover the secret key (*i.e.*, achieve a total break) and obtain honest users' templates. Therefore, we further analyze two real situations: (1) CPA attackers try to recover the secret key M to get honest users' templates, and (2) CPA attackers try to recover honest users' templates by exploiting relative distances R_i. The analysis shows that a CPA attacker might recover honest users' templates with a negligible probability. Moreover, we analyze some other security features considered in matrix-transformation-based biometric identification [33], [34], *i.e.* signature linking attack (SLA) and modified signature linking attack (MSLA).

A. IND-CPA SECURITY

It is sufficient to prove our protocol is CPA-secure under the situation where the attacker $\mathcal{A}$ submits one pair of templates b_0 and b_1 to the data owner (*i.e.* challenger $\mathcal{C}$), because it has been proven in [35] that any CPA-secure private-key encryption scheme is also CPA-secure for multiple encryptions. Therefore, we are supposed to prove that given oracle access to $Enc(sk, \cdot)$, the attacker $\mathcal{A}$ can't distinguish $Enc(sk, b_0)$ and $Enc(sk, b_1)$.

Consider the attacker $\mathcal{A}$ that outputs a pair of template plaintexts (b_0, b_1) and then receives ciphertext $C_i \leftarrow Enc(sk, b_i)$ from the data owner $\mathcal{C}$. Since $\mathcal{A}$ has oracle access to $Enc(sk, \cdot)$, $\mathcal{A}$ can request this oracle to encrypt the template plaintexts b_0 and b_1, thereby $\mathcal{A}$ can obtain $C_0 \leftarrow Enc(sk, b_0)$ and $C_1 \leftarrow Enc(sk, b_1)$.

As described in the IND-CPA experiment, the data owner $\mathcal{C}$ chooses uniformly a random bit $i \leftarrow \{0, 1\}$ and sends the "challenge ciphertext" C_i to the attacker $\mathcal{A}$. And then $\mathcal{A}$ decides which one of b_0 and b_1 is the corresponding plaintext of C_i. The attacker $\mathcal{A}$ might compare C_0 and C_1 to the challenge ciphertext C_i; if $C_0 = C_i$, $\mathcal{A}$ outputs $i' = 0$, and if $C_0 = C_i$, $\mathcal{A}$ outputs $i' = 1$. However, this method doesn't work, because the encryption of our scheme is probabilistic, not deterministic [35].

We take the challenge ciphertext C_i into consideration. Its corresponding plaintext $b_i = [b_{i1}, b_{i2}, \cdots, b_{in}]$ is first extended to a $(n+5)$-dimensional vector

$$b_i' = [\alpha_i b_{i1}, \alpha_i b_{i2}, \cdots, \alpha_i b_{in}, -\frac{1}{2}\alpha_i \sum_{j=1}^{n} b_{ij}^2, \alpha_i, \frac{1}{2}\alpha_i \tau^2, r_i, 0]$$

where α_i and r_i are one-time random numbers. And then, we get the ciphertext $C_0 = b_0' M$, of which the element can be represented as

$$c_{ij} = \alpha_i \sum_{k=1}^{n} b_{ik} m_{kj} - \frac{1}{2}\alpha_i m_{(n+1)j} \sum_{k=1}^{n} b_{ik}^2 + \alpha_i m_{(n+2)j} + \frac{1}{2}\alpha_i \tau^2 m_{(n+3)j} + r_i m_{(n+4)j}$$

Similarly, we can represent the element of $C_0 \leftarrow Enc(sk, b_0)$ and $C_1 \leftarrow Enc(sk, b_1)$ as follows:

$$\begin{aligned} c_{0j} &= \alpha_0 \sum_{k=1}^{n} b_{0k} m_{kj} - \frac{1}{2}\alpha_0 m_{(n+1)j} \sum_{k=1}^{n} b_{0k}^2 + \alpha_0 m_{(n+2)j} \\ &\quad + \frac{1}{2}\alpha_0 \tau^2 m_{(n+3)j} + r_0 m_{(n+4)j} \\ c_{1j} &= \alpha_1 \sum_{k=1}^{n} b_{1k} m_{kj} - \frac{1}{2}\alpha_1 m_{(n+1)j} \sum_{k=1}^{n} b_{1k}^2 + \alpha_1 m_{(n+2)j} \\ &\quad + \frac{1}{2}\alpha_1 \tau^2 m_{(n+3)j} + r_1 m_{(n+4)j} \end{aligned}$$

Note that α_i and r_i are one-time random numbers. They might be equal to α_0, r_0 or α_1, r_1 with probability $2 \times (2^{-\lambda})^2 = 2^{-2\lambda+1}$, in which the security parameter λ represents the length of the secret key's element. This probability is also the advantage Adv of the attacker $\mathcal{A}$, due to $\mathcal{A}$ could distinguish the challenge ciphertext C_i only if (α_i, r_i) equals to (α_0, r_0) or (α_1, r_1). The Adv is negligible if the security parameter λ is large enough, *e.g.* in our scheme, $\lambda = 128$, the advantage $Adv = 2^{-255}$. Therefore we get

$$\left| \Pr(\mathrm{CPA}_{\mathcal{A}}(\lambda) = 1) - \frac{1}{2} \right| \leq negl(\lambda)$$

which implies our scheme is CPA-secure. Further, we can enhance the security by extending template with more security factors r_i, whose overhead is tolerable as described in Section VII.

B. SECURITY AGAINST TOTAL BREAK

To intuitively prove that the secret key M won't be recovered by an attacker, we first introduce the *degree of freedom*, represented as D_{dof}. To generate a random matrix, D_{dof} is the number of parameters that may vary independently in this matrix.

1) D_{dof} OF ORTHOGONAL MATRIX

A square matrix can represent any linear vector transformation. Sometimes we want to constrain elements of the matrix so that it represents a pure rotation, *i.e.* orthogonal transformation. Therefore, the matrix representation of an orthogonal transformation contains redundant information. For example, a 3×3 orthogonal matrix contains 9 elements, while the D_{dof} of it only equals 3.

For a $n \times n$ orthogonal matrix with column vectors $v_1, v_2, \cdots, v_n$, there are following constraint conditions:

$$\begin{cases} v_i \cdot v_j = 0, (1 \le i, j \le n, i \ne j) \\ v_i \cdot v_i = 1, (1 \le i \le n) \end{cases}$$

The first condition has $\frac{n(n-1)}{2}$ unique equations, and the second has n constraints.

The matrix has n^2 elements, so the $D_{dof} = n^2 - \frac{n(n-1)}{2} - n = \frac{n(n-1)}{2}$, *i.e.* there are $\frac{n(n-1)}{2}$ independent variables of $(n \times n)$-dimensional orthogonal matrix.

2) BRUTE-FORCE ATTACK

For the $(n+5) \times (n+5)$ orthogonal matrix M, the $D_{dof} = \frac{(n+5)(n+4)}{2}$. By brute-force attack, the probability to work out the secret key M is $2^{-\frac{\lambda(n+5)(n+4)}{2}}$, where λ is the security parameter which means the length of the secret key's element. Due to $\lambda > 1$ and $n = 640$ generally, the success probability is $2^{-207690\lambda}$, which implys the attacker might work out the secret key M by brute-force attack with a negligible probability.

3) SECURITY AGAINST TOTAL BREAK BY CPA ATTACK

The CPA attack is another possible way to attack the secret key M. As malicious users have the ability to produce fake biometrics during enrollment and collude with the semi-honest cloud server, the attacker can get any fake template's plaintext $\widetilde{b}_i$ and corresponding ciphertext $\widetilde{C}_i$. In the $\widetilde{C}_i$, there are $(n+5)$ elements which can be represented as

$$\widetilde{c_{ij}} = \alpha_i \sum_{k=1}^{n} \widetilde{b_{ik}} m_{kj} - \frac{1}{2}\alpha_i m_{(n+1)j} \sum_{k=1}^{n} \widetilde{b_{ik}}^2 + \alpha_i m_{(n+2)j} + \frac{1}{2}\alpha_i \tau^2 m_{(n+3)j} + r_i m_{(n+4)j}$$

Each ciphertext $\widetilde{C}_i$ can provide $(n+5)$ constraints. So in general, the attacker needs at least $\frac{(n+4)}{2}$ fake templates to work out the M. However, with one more fake template used to get $(n+5)$ more constraints, there are 2 more independent randomness α_i and r_i introduced to prevent the attacker get simultaneous equations with any other fake template, unless the attacker gets values of all randomness. All the randomness α_i and r_i are known only to the data owner, so the attacker can only guess the randomness by brute-force attack. The probability of geting correct values of all randomness is $(2^{-2\lambda})^{\frac{n+4}{2}} = 2^{-\lambda(n+4)}$. Due to $\lambda > 1$ and $n = 640$ generally, this probability is $2^{-644\lambda}$, which implies the attacker might get values of all randomness with a negligible probability. This is also the case with encryption of fake sample template. Therefore, the total break by CPA attack is infeasible, and all honest users' templates can not be recovered from their ciphertexts.

C. SECURITY OF THE RELATIVE DISTANCE

As for relative distance result $R_i = C_i \cdot C_c = \frac{1}{2}\alpha_i\beta_c[\tau^2 - \sum_{j=1}^{n}(b_{ij} - b_{cj})^2]$, if the attacker forge a fake refernce template $\widetilde{b}_i$, the result $\widetilde{R}_i = \frac{1}{2}\alpha_i\beta_c[\tau^2 - \sum_{j=1}^{n}(\widetilde{b_{ij}} - b_{cj})^2]$. Without considering the randomness α_i, there are $(n+1)$ unknowns b_{cj} and β_c, which means the attacker has to produce at least $(n+1)$ fake templates to get enough equations to recover the sample template b_c. However, with one more fake template used to establish an equation, there is one more new unknown random element α_i introduced. The attacker might work out the sample template b_c only by getting values of these randomness α_i, whereas this probability is $(2^{-\lambda})^{n+1} = 2^{-\lambda(n+1)}$. Due to $\lambda > 1$ and $n = 640$ generally, this probability is $2^{-641\lambda}$, which implies the attacker might get values of these randomness α_i with a negligible probability. This is also the case when the attacker wishes to figure out reference templates by producing fake sample templates $\widetilde{b}_c$. So, the attacker can't recover any reference template b_i or sample template b_c from the relative distance R_i.

D. SECURITY AGAINST SLA AND MSLA ATTACK

According to [33], an attacker might "upgrade" level-2 knowledge to level-3 using signature linking attack (SLA). Later, a more flexible SLA attack called modified-SLA (MSLA) is presented in [34]. While our proposed scheme can resist this "upgrade".

Prior to describing SLA/MSLA attack, we first introduce the definition of distance-recoverable encryption (DRE) into biometric identification.

Definition 2: Given an encryption function *Enc* and a secret key sk, let $C_i = Enc(b_i, sk)$ be the encrypted value of a template b_i in the biometric database $\langle b_i \rangle_{i=1}^{m}$. *Enc* is distance-recoverable if and only if there exists a computational procedure f such that $\forall b_i, b_j, sk, f(C_i, C_j) = d(i,j)$, where $d(i,j)$ is the Euclidean distance between b_i and b_j.

1) SLA ATTACK

For a DRE scheme, a level-2 attacker holds an ordered set of template plaintexts $B = \{b_1, b_2, \cdots, b_{|B|}\}$ and constructs a signature $sig(B)$ of B by pairwise distances between every two templates in B, *i.e.* $sig(B) = \{d(1,2), d(1,3), \cdots, d(1,|B|), d(2,3), \cdots, d(|B|-1,|B|)\}$. As malicious users have the ability to collude with the cloud server, the attacker can get the encrypted biometric database $\langle C_i \rangle_{i=1}^{m}$. Then, the attacker tries to find an ordered subset C in $\langle C_i \rangle_{i=1}^{m}$, such that $|C| = |B|$ and C has the same signature as B. Let $C = \{C_1, C_2, \cdots, C_{|B|}\}$, the signature $sig(C) = \{f(C_1, C_2), f(C_1, C_3), \cdots, f(C_1, C_{|B|}), f(C_2, C_3), \cdots,$

TABLE 2. Security comparison between our proposed scheme and [26], [27].

	Zhu et al.'s scheme [26]	Hu et al.'s scheme [27]	Our proposed scheme
Security of sample templates under level-3 attack	No	No	Yes
Security of reference templates under level-3 attack	No	Yes	Yes
Security of sample templates under level-4 attack	No	No	Yes
Security of reference templates under level-4 attack	No	Yes	Yes

$f(C_{|B|-1}, C_{|B|})\}$. Due to the signature collision is generally very small [33], if there is only one set C with a matching signature, the attacker can conclude that C_i in the C is the corresponding ciphertext of b_i in the B. However our scheme doesn't belong to DRE schemes, so our proposed scheme can resist the SLA attack.

2) MSLA ATTACK

The MSLA attack is a more flexible SLA attack, in which the signature doesn't only depend on the Euclidean distance, any recoverable value got by performing computation on ciphertexts can be used to represent the signature. In our scheme, values got by performing computation on ciphertexts are as follows:

$$R_i = C_i \cdot C_c = \frac{1}{2}\alpha_i\beta_c[\tau^2 - \sum_{j=1}^{n}(b_{ij} - b_{cj})^2]$$

$$R'_{ij} = C_i \cdot C_j = b'_i M \cdot b'_j M = b'_i \cdot b'_j$$
$$= \alpha_i\alpha_j(\sum_{k=1}^{n} b_{ik}b_{jk} + \frac{1}{4}\sum_{k=1}^{n} b_{ik}^2 \sum_{k=1}^{n} b_{jk}^2 + \frac{1}{4}\tau^4 + 1) + r_i r_j$$

$$R'_c = C_c \cdot C_{c'} = b'_c M \cdot b'_{c'} M = b'_c \cdot b'_{c'}$$
$$= \beta_c\beta_{c'}(\sum_{k=1}^{n} b_{ck}b_{c'k} + \frac{1}{4}\sum_{k=1}^{n} b_{ck}^2 \sum_{k=1}^{n} b_{c'k}^2 + 2) + r_c r_{c'}$$

which represent the relative distance, inner product of reference template ciphertexts and inner product of sample template ciphertexts respectively. For the relative distance R_i, the attacker can only get the value of $\tau^2 - \sum_{j=1}^{n}(b_{ij} - b_{cj})^2$, but randomness α_i and β_c are unknown to the attacker. This is similar to R'_{ij} and R'_c, the attacker can get $\sum_{k=1}^{n} b_{ik}b_{jk} + \frac{1}{4}\sum_{k=1}^{n} b_{ik}^2 \sum_{k=1}^{n} b_{jk}^2 + \frac{1}{4}\tau^4 + 1$ and $\sum_{k=1}^{n} b_{ck}b_{c'k} + \frac{1}{4}\sum_{k=1}^{n} b_{ck}^2 \sum_{k=1}^{n} b_{c'k}^2 + 2$, but can't get values of the randomness. Therefore, values got by performing computation on ciphertexts are unrecoverable and our proposed scheme can resist the MSLA attack.

Moreover, we compare security features of our scheme with the schemes proposed in [26] and [27]. According to the Table 2, other schemes have some weaknesses, while our scheme is secure under the reasonable threat model presented in Section II.

VII. PERFORMANCE ANALYSIS

To evaluate the performance of our scheme, we compare both computational and communication complexity with existing works, and then we fully implemented them to evaluate the practicality numerically. The analysis shows our scheme may downgrade the efficiency in the preparation stage, but achieves a significant improvement of performance in the identification stage. Due to database outsourcing is only a one-off process and identification is the most frequent operation, our scheme is more suitable for practical applications.

A. COMPLEXITY ANALYSIS

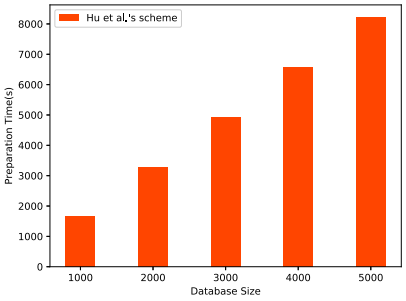
As described in Section V, our scheme can be decomposed into three stages. In the 1^{st} preparation stage, to outsource the whole database, the owner should encrypt every record in the database by performing vector-matrix multiplication, whose computational complexity is $O(n^2)$. Therefore, the total complexity of this stage is $O(mn^2)$, where m is the total number of the FingerCode records in the database. In the 2^{nd} request stage, a sample template is sent to data owner for identification. Then the data owner will encrypt it similar to what has been done in the 1^{st} stage. So, the encryption also costs $O(n^2)$. In the 3^{rd} identification stage, the encrypted sample template is submitted to the cloud server. All the ciphertexts are vectors. Thus, the total time complexity of the inner product is $O(mn)$. For communication complexity, besides the one-off outsource cost of $O(mn)$ for the 1^{st} stage, the request costs $O(n)$ and the cost of identification response is $O(1)$.

To compare our protocol with previous schemes [26], [27] intuitively, we illustrate the complexities in Table 3. As shown in Table 3, for computation complexity, in the preparation stage, our scheme has similar cost $O(mn^2)$ as Zhu et al.'s scheme [26] does, lower than the cost $O(mn^3)$ of Hu et al.'s scheme [27]. In the identification stage, the data owner of our scheme need only to encrypt the sample template with overhead $O(n^2)$. Besides higher cost $O(n^3)$ in template encryption, the data owner of the other two works still has to compute the Euclidean distance between the sample template and the reference template corresponding to the retrieved index. The cloud server of our protocol will spend only $O(mn)$ working out whether the identification is successful.

TABLE 3. Comparison of complexity between our proposed scheme, [26] and [27]. *m* denotes the number of templates in database and *n* denotes the dimension of the template.

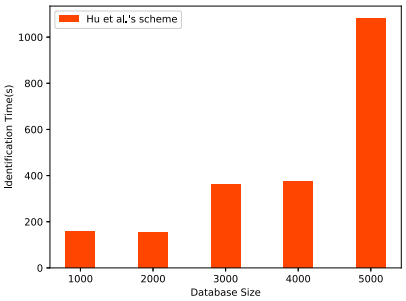
	Participant	Stage	Zhu et al.'s scheme [26]	Hu et al.'s scheme [27]	Our proposed scheme
Computation	data owner	preparation	$O(mn^2)$	$O(mn^3)$	$O(mn^2)$
		request	$O(n^3)$	$O(n^3)$	$O(n^2)$
		retrieval	$O(n)$	$O(n)$	/
	cloud server	identification	$O(mn^2)$	$O(mn^3)$	$O(mn)$
Communication	data owner	preparation	$O(mn)$	$O(mn^2)$	$O(mn)$
		request	$O(n^2)$	$O(n^2)$	$O(n)$
		retrieval/result	$O(1)$	$O(1)$	$O(1)$
	cloud server	identification	$O(1)$	$O(1)$	$O(1)$

(a) Time costs in preparation stage of our scheme and [26]



(b) Time costs in preparation stage of [27]

(c) Time costs in identification stage of our scheme and [26]



(d) Time costs in identification stage of [27]

FIGURE 2. Comparison of time cost in different stages between our proposed scheme, [26] and [27].

TABLE 4. Comparison of communication cost in different stages between our proposed scheme and [26], [27].

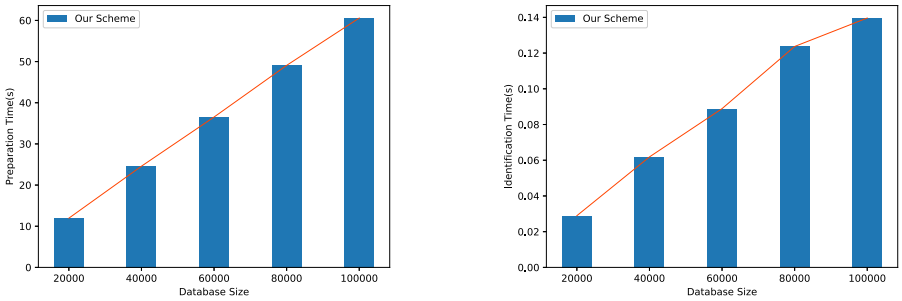
	Zhu et al.'s scheme [26]	Hu et al.'s scheme [27]	Our proposed scheme
preparation	$5.14(m+1)$KB	$6.29m$MB	$5.17m$KB
identification	3.14MB	6.29MB	5.17KB

While the cloud server of [26] and [27] will afford heavier cost $O(mn^2)$ and $O(mn^3)$ respectively to calculate the relative distance P_i and find the most closely matched template's index. So the identification efficiency of our scheme is much better than the other two works. For communication cost, on the data owner side, in the preparation stage, our scheme and [26] transmit encrypted vectors to the cloud, so the complexity is $O(mn)$. While [27] transfers matrix ciphertexts with cost $O(mn^2)$. In the identification stage, our scheme costs $O(n)$ to send a request containing a vector to the cloud. However, [26] and [27] transmit request consisting of matrices with complexity $O(n^2)$. At last, all schemes cost $O(1)$ to return the identification result to user. On the cloud server side, all schemes will send the cloud computing result (identification result or the most closely matched index) to the data owner with communication overhead $O(1)$.

B. EXPERIMENTAL EVALUATION

1) EXPERIMENT SETUP

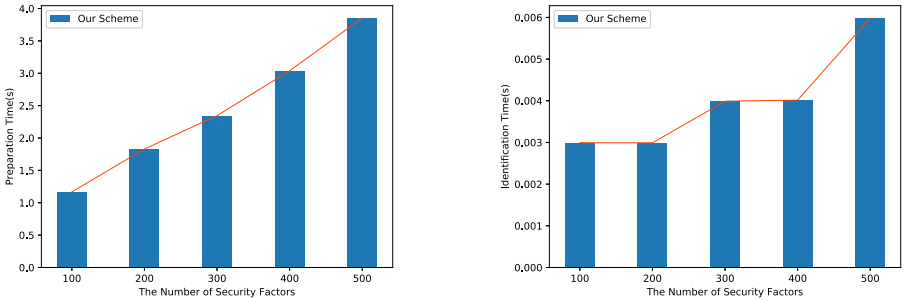
To evaluate the improvement of our scheme, compared with the existing schemes [26], [27], we implement these systems using Python as the programming language. All experiments are conducted on a Windows 10 machine with



(a) Time costs in preparation stage of our scheme

(b) Time costs in identification stage of our scheme

FIGURE 3. Time cost for lager databases in different stages of our proposed scheme.



(a) Time costs in preparation stage of our scheme

(b) Time costs in identification stage of our scheme

FIGURE 4. Time cost for higher security in different stages of our proposed scheme.

8-core 3.40GHz Intel i7 CPU and 24GB RAM, to simulate the process on data owner, cloud server and users. In addition, we use a synthetic datasets consisting of randomly generated 640-dimensional vectors to represent the FingerCodes, as [26] and [27] does.

2) RESULTS OVER SYNTHETIC DATA

For a better performance evaluation, we first compare all schemes with the size of database m varying from 1000 records to 5000 records. And then we test the performance of our proposed scheme with larger databases. Further, we will show the impact of extending templates for higher security with more security factors.

From Fig. 2, we can see the efficiency of our scheme and Zhu et al.'s scheme [26] is much better than Hu et al.'s scheme [27]. When the database has 5000 records, Fig. 2(b) shows the time cost of preparation stage for [27] is 2.28 hours, much slower than our scheme's 2.94s and 2.43s of [26] shown in Fig. 2(a). Because template encryption of our scheme is more complicated than [26], our scheme has a tiny delay. Note that this stage is an one-off process, it also proves that our scheme has a comparable performance in this stage. Refer to Fig. 2(c) and Fig. 2(d), our scheme costs only 0.008s, much more efficient than 18.01 minutes of [27]. And compared to 0.76s of [26], our scheme saves 98.95% time. The identification is the most process executed, so our scheme has the best performance and can be applied to a much larger size of biometric database.

The communication cost is described in Table 4, where m is the number of database records.

Moreover, we test our scheme using larger databases, with the size varying from 20000 to 100000. The result is described in Fig. 3(a) and Fig. 3(b). It shows the performance of our scheme is practical.

Further, we test the impact of extending templates with more security factors for higher security, with the size of the database set 1000. The result is given in Fig. 4(a) and Fig. 4(b). It is revealed that more security factors will affect the preparation stage but have less influence on the identification. So, the increasing overhead for higher security is tolerable.

VIII. CONCLUSION

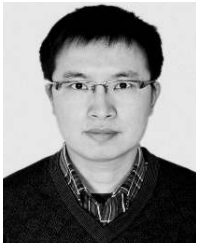
In this paper, we proposed an efficient privacy-preserving biometric identification scheme based on matrix transformation. Compared to the existing matrix-transformation-based scheme put forward by Zhu et al. recently, we improve the security of biometric identification by introducing additional randomness. Further, we reduce the computational complexity by exploiting orthogonal matrix, which means our scheme makes the biometric identification more practical for a large-scale database of templates in an actual

situation. Our scheme may also benefit other areas, such as privacy-preserving cloud computing.

REFERENCES

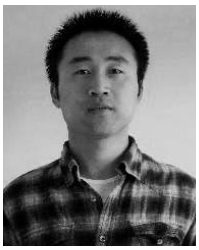
[1] Q. Feng, D. He, S. Zeadally, and H. Wang, "Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment," *Future Gener. Comput. Syst.*, vol. 84, pp. 239–251, Jul. 2018.
[2] E. Pagnin and A. Mitrokotsa, "Privacy-preserving biometric authentication: Challenges and directions," *Secur. Commun. Netw.*, vol. 2017, Sep. 2017, Art. no. 7129505.
[3] B. Schneier, "Biometrics: Uses and abuses," *Commun. ACM*, vol. 42, no. 8, p. 58, 1999.
[4] Wells Fargo Bank. (2019). *Convenient Access to Your Accounts*. [Online]. Available: https://www.wellsfargo.com/online-banking/biometric/
[5] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4900–4913, Dec. 2018.
[6] Q. Jiang, Z. Chen, B. Li, J. Shen, L. Yang, and J. Ma, "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 4, pp. 1061–1073, 2018.
[7] X. Li, J. Niu, S. Kumari, F. Wu, and K.-K. R. Choo, "A robust biometrics based three-factor authentication scheme for global mobility networks in smart city," *Future Gener. Comput. Syst.*, vol. 83, pp. 607–618, Jun. 2018.
[8] D. He, Y. Zhang, and J. Chen, "Robust biometric-based user authentication scheme for wireless sensor networks," *Adhoc Sensor Wireless Netw.*, vol. 25, no. 3, pp. 309–321, 2012.
[9] S. N. Syed, A. Z. Shaikh, and S. Naqvi, "A novel hybrid biometric electronic voting system: Integrating finger print and face recognition," 2018, *arXiv:1801.02430*. [Online]. Available: https://arxiv.org/abs/1801.02430
[10] C.-A. Toli and B. Preneel, "Privacy-preserving biometric authentication model for e-finance applications," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 353–360.
[11] M. S. Obaidat, I. Traore, and I. Woungang, *Biometric-Based Physical and Cybersecurity Systems*. Springer, 2019.
[12] P. Tuyls and J. Goseling, "Capacity and examples of template-protecting biometric authentication systems," in *Proc. Int. Workshop Biometric Authentication*. Berlin, Germany: Springer, 2004, pp. 158–170.
[13] P. Tuyls, E. Verbitskiy, J. Goseling, and D. Denteneer, "Privacy protecting biometric authentication systems: An overview," in *Proc. 12th Eur. Signal Process. Conf.*, Sep. 2004, pp. 1397–1400.
[14] A. T. B. Jin, D. N. C. Ling, and A. Goh, "BioHashing: Two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognit.*, vol. 37, no. 11, pp. 2245–2255, Apr. 2004.
[15] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You, "An analysis of biohashing and its variants," *Pattern Recognit.*, vol. 39, no. 7, pp. 1359–1368, Jul. 2006.
[16] A. Juels and M. Sudan, "A fuzzy vault scheme," *Des., Codes Cryptogr.*, vol. 38, no. 2, pp. 237–257, Feb. 2006.
[17] W. J. Scheirer and T. E. Boult, "Cracking fuzzy vaults and biometric encryption," in *Proc. Biometrics Symp.*, Sep. 2007, pp. 1–6.
[18] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. D. Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, F. Scotti, and A. Piva, "Privacy-preserving fingercode authentication," in *Proc. 12th ACM Workshop Multimedia Secur.*, 2010, pp. 231–240.
[19] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1999, pp. 223–238.
[20] D. Catalano and D. Fiore, "Using linearly-homomorphic encryption to evaluate degree-2 functions on encrypted data," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1518–1529.
[21] J.-H. Im, J. Choi, D. Nyang, and M.-K. Lee, "Privacy-preserving palm print authentication using homomorphic encryption," in *Proc. IEEE DASC/PICom/DataCom/CyberSciTec*, Aug. 2016, pp. 878–881.
[22] H. Zhu, Q. Wei, X. Yang, R. Lu, and H. Li, "Efficient and privacy-preserving online fingerprint authentication scheme over outsourced data," *IEEE Trans. Cloud Comput.*, to be published. doi: 10.1109/TCC.2018.2866405.
[23] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2005, pp. 325–341.
[24] J. Yuan and S. Yu, "Efficient privacy-preserving biometric identification in cloud computing," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2652–2660.
[25] Y. Zhu, T. Takagi, and R. Hu, "Security analysis of collusion-resistant nearest neighbor query scheme on encrypted cloud data," *IEICE Trans. Inf. Syst.*, vol. 97, no. 2, pp. 326–330, 2014.
[26] 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.
[27] S. Hu, M. Li, Q. Wang, S. S. M. Chow, and M. Du, "Outsourced biometric identification with privacy," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2448–2463, Oct. 2018.
[28] Z. Shan, K. Ren, M. Blanton, and C. Wang, "Practical secure computation outsourcing: A survey," *ACM Comput. Surv.*, vol. 51, no. 2, 2018, Art. no. 31.
[29] H. Delfs and H. Knebl, *Introduction to Cryptography*, vol. 2. Springer, 2002.
[30] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 846–859, May 2000.
[31] T. Bhattasali, K. Saeed, N. Chaki, and R. Chaki, "A survey of security and privacy issues for biometrics based remote authentication in cloud," in *Proc. IFIP Int. Conf. Comput. Inf. Syst. Ind. Manage.* Berlin, Germany: Springer, 2015, pp. 112–121.
[32] K. Liu, C. Giannella, and H. Kargupta, "An attacker's view of distance preserving maps for privacy preserving data mining," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*. Berlin, Germany: Springer, 2006, pp. 297–308.
[33] W. K. Wong, D. W.-I. Cheung, B. Kao, and N. Mamoulis, "Secure KNN computation on encrypted databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2009, pp. 139–152.
[34] S. Pan, S. Yan, and W.-T. Zhu, "Security analysis on privacy-preserving cloud aided biometric identification schemes," in *Proc. Australas. Conf. Inf. Secur. Privacy*. Cham, Switzerland: Springer, 2016, pp. 446–453.
[35] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Boca Raton, FL, USA: CRC Press, 2014.

CHUN LIU received the B.E. degree from Zhejiang University, Hangzhou, China, in 2017. He is currently pursuing the master's degree with PLA Strategic Support Force Information Engineering University, Zhengzhou, China. His research interests include applied cryptography and big data security.



XUEXIAN HU received the Ph.D. degree in information security from the Institute of Information Engineering, Zhengzhou, China, in 2010. He is currently an Associate Professor with PLA Strategic Support Force Information Engineering University. His current research interests include applied cryptography and big data security.

QIHUI ZHANG received the master's degree in information security from the Institute of Information Engineering, Zhengzhou, China, in 2007. She is currently pursuing the Ph.D. degree with PLA Strategic Support Force Information Engineering University, where she is currently a Lecturer. Her main research interest includes big data security.



JIANGHONG WEI received the Ph.D. degree in information security from the Institute of Information Engineering, Zhengzhou, China, in 2016. He is currently a Lecturer with PLA Strategic Support Force Information Engineering University, Zhengzhou. His research interests include applied cryptography and cloud computing security.

WENFEN LIU received the Ph.D. degree in mathematics from the Institute of Information Engineering, Zhengzhou, China, in 1998. She is currently a Full Professor with the Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China, and serves as the Head of probability statistics. Her research interests include probability statistics, network communications, and information security.