12-2018

PriBioAuth: Privacy-preserving biometric-based remote user authentication

Yangguang TIAN
Singapore Management University, ygtian@smu.edu.sg

Yingjiu LI
Singapore Management University, yjli@smu.edu.sg

Ximeng LIU
Singapore Management University, xmliu@smu.edu.sg

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Binanda SENGUPTA
Singapore Management University, binandas@smu.edu.sg

Published in 2018 IEEE Conference on Dependable and Secure Computing DSC: Kaohsiung, Taiwan, December 10-13: Proceedings, 112-132.
https://doi.org/10.1109/DESEC.2018.8625169

Accepted version

PriBioAuth: Privacy-Preserving Biometric-Based Remote User Authentication

Yangguang Tian, Yingjiu Li, *Member, IEEE,* Ximeng Liu, *Member, IEEE,* Robert H. Deng, *Fellow, IEEE,*
Binanda Sengupta
School of Information Systems, Singapore Management University
Email: {ygtian,yjli,xmliu,robertdeng,binandas}@smu.edu.sg

***Abstract*—Biometric-based remote user authentication (BRUA) is a useful primitive that allows an authorized user to remotely authenticate to a cloud server using biometrics. However, the existing BRUA solutions in the client-server setting lack certain privacy considerations. For example, authorized user's multiple sessions should not be linked while his identity remains anonymous to cloud server. In this work, we introduce an identity-concealed and unlinkable biometric-based remote user authentication framework, such that authorized users authenticate to an honest-but-curious server in an anonymous and unlinkable manner. In particular, we employ two non-colluding cloud servers to perform the complex biometrics matching. We formalize two new security models, including biometrics privacy and user privacy, for our proposed framework, and prove the security of the proposed framework in the standard model.**

***Index Terms*—Remote User Authentication, Unlinkability, Biometrics Matching, User Privacy.**

I. Introduction

Biometric-based user authentication has been widely used in many real-life applications, such as mobile security, financial transactions and identification checks [1]. There are some attractive features using biometrics over conventional password. For example, people need to remember many secure passwords for many different accounts and update passwords frequently for security reasons. By contrast, biometrics is permanently and uniquely associated with an individual, so the individual can use biometrics for user authentication.

Biometric-based user authentication also leads to some security and privacy concerns. First, biometrics is not revocable. If biometrics is compromised, then the user may lose its security forever, especially for the single-factor biometric-based user authentication. Second, authorized users may concern the privacy of biometrics stored on the authentication server. Therefore, no biometrics should be stored in plaintext, because biometrics may contain a wealth of personal information (e.g., DNA).

To protect biometrics information, there are mainly three methods in the literature: non-invertible transform [2], fuzzy extractors [3] and homomorphic cryptosystem [4]. The non-invertible transform relies on a static secret key. Essentially, it is a two-factor (biometrics plus secret key) user authentication and not scalable for cross-platform setting[1], because the secret key must be available at the time of authentication to transform the requested biometrics for subsequent user authentication. The fuzzy extractors based user authentication [5], [6] is a single-factor user authentication. However, deriving a secret key from biometrics and other noisy data with high stability and entropy simultaneously is a non-trivial task.

Using homomorphic encryption [4] to protect biometrics information is a promising approach when designing biometric-based user authentication. In particular, the authentication server in cloud can perform complex mathematical computations (i.e., biometrics matching) in the encrypted format, since cloud computing provides ubiquitous, dynamic, scalable and on-demand services. That is, the cloud-based biometrics can facilitate efficient biometrics matching for user authentication. In this work, we focus on biometric-based remote user authentication (BRUA) using homomorphic encryption, where authorized users wish to remotely authenticate to an authentication server using encrypted biometrics.

The privacy should be preserved not only for biometrics information, but also for non-biometrics information (such as identity, behaviour and interaction history). Identity-concealment is an important privacy property and is mandated or recommended by some widely standardized and deployed cryptographic protocols, such as TLS1.3 and QUIC [7]. Identity-concealment means that the transcript of protocol execution should not leak authorized user's identity information. Moreover, unlinkability is also desired, such that multiple sessions of the same authorized user cannot be linked by the authentication server. The main *goal* of this work is to design an identity-concealed and unlinkable BRUA using homomorphic encryption.

Homomorphic encryption can be used to encrypt identity information of authorized users during the protocol execution. However, if the same anonymous user authenticates twice to an authentication server, then authentication server can still link the anonymous authenticated user to a specific record in his database which stores all enrolled user's records. Note that such kind of unlinkability between authorized user and database record is an important feature for sensitive IT infrastructure such as personal record management systems [8].

Since biometrics matching of BRUA may handle various kinds of distance calculations (e.g., Euclidean distance, Hamming distance or Chebyshev distance), a suitable homomorphic encryption primitive is critical to the success of user authentication. Full homomorphic encryption can easily

[1]In practice, users may own several devices (e.g., smart-phone, pad and tablet) and access to the same service provider from various platforms.

support all aforementioned distance calculations. Specifically, it enables addition and multiplication simultaneously (on encrypted biometrics) when performing biometrics matching. However, it is not practical in real-world environment (such as resource-limited devices) due to its computational cost and system complexity [9], [10].

Instead of full homomorphic encryption, we rely on partial homomorphic encryption such as Paillier cryptosystem. However, Paillier cryptosystem is limited to additive operations over encrypted biometrics. Sometimes the multiplicative operations are mandatory when Euclidean distance based biometrics matching is applied. Therefore, how to exploit the Paillier cryptosystem to support complex mathematical operations for biometrics matching is our first challenge task.

Furthermore, biometrics are typically encrypted under user's own public keys and stored in the authentication server. Since biometrics matching takes different ciphertexts under the same public key as input, the authentication server must transform the ciphertexts under different public keys into the ciphertexts under the same public key. Such transformation is easy when authorized users are identified. However, this contradicts to the user privacy we desired. Hence, achieving an anonymous and unlinkable user authentication is a rather challenging task.

A. This Work

In this work, we introduce the notion of privacy-preserving biometric-based remote user authentication (PriBioAuth in short), allowing authorized users to remotely authenticate to an authentication server using encrypted biometrics. Our proposed solution employs two (non-colluding) honest-but-curious cloud servers in the system ([11], [12]), one acts as authentication server, while the other one acts as a dedicated computational server which works with authentication server to assist certain biometrics matching.

As for anonymous and unlinkable PriBioAuth, we first propose an anonymous key transformation (AKeyTrans) protocol, such that authentication server performs the key transformation in an anonymous manner. Meanwhile, inspired by the concept of oblivious access control [13], [14], we allow authenticated users to authenticate an authentication server in an oblivious manner. Based on the anonymous key transformation and oblivious access control, the proposed PriBioAuth can achieve the claimed user privacy. Our overall contributions can be summarized as follows.

- *Security and Privacy Guarantee:* We provide the formal security requirements for privacy-preserving biometric-based remote user authentication protocols. We formalize two formal security models which include various kinds of security and privacy properties, such as biometrics privacy, obliviousness of access control, identity-concealment (anonymity) and unlinkability;
- *Practical Construction:* In order to enable the authentication server to perform efficient biometrics matching, we present a practical solution for biometric-based remote user authentication using two non-colluding could servers, an authentication server and a computational server;
- *Secure Biometrics Matching:* The authentication server in conjunction with the computational server can perform various kinds of mathematical computations for biometrics matching. We provide a set of secure multi-party computation (SMC) sub-protocols to guarantee the success of biometric-based remote user authentication, including less than, equivalent testing and multiplicative computation protocols. In particular, no user interaction is required for biometrics matching;
- *Scalability of Use:* It is easy to employ our solution in a cross-platform setting. Because the proposed PriBioAuth solution is a single-factor user authentication without generating extra secret keys at the time of authentication.

B. Related Work

Biometric-based User Authentication/Identification. Privacy-preserving was the main focus of designing biometric-based user authentication and identification in the literature [15], [16], [17], [18], [19], but the definition on privacy are various. For example, some works [15], [16] assume that biometrics template is a public information (e.g., fingerprint and face). Specifically, they assume an authentication server (or service provider) and a non-colluding database in the system. In particular, the plain biometrics template is stored in database, and the privacy concern is about the relationship between biometrics template and identity (or pseudonym). However, we assume biometrics is a secret information in this work.

Homomorphic encryption (see below) is a suitable cryptographic tool to protect biometrics instead of non-invertible transform and fuzzy extractors. In particular, it supports the secure multi-party computations (SMC) on encrypted biometrics for biometrics matching. Note that some well-known works [20], [17], [18], [19] have used the Paillier cryptosystem as encryption primitive to protect user's biometrics. For example, Huang et al. [19] proposed a flexible biometric-based identification framework. They used the garbled circuit [21] to efficiently and obliviously perform biometrics matching and retrieve the outcome of results. However, the authentication server should interact with authorized user to finalize the biometrics matching.

Bringer et al. [15] proposed a biometric-based user authentication protocol using Goldwasser-Micali (GM) cryptosystem [22]. Note that the GM cryptosystem takes the binary string (such as Iris [23]) as input. To allow Paillier cryptosystem process the binary input, Schoenmakers and Tuyls [24] proposed a generic framework, such that the underlying Paillier cryptosystem [25] can process binary string for biometrics matching. That is, the Paillier crytosystem can handle bits strings using their proposed binary conversion.

Homomorphic Encryption. Homomorphic encryption (HE) is a well-known approach for privacy-preserving secure multi-party computation. There are mainly two types of HE system in the literature: one is full FE, and the other is partial HE. The latter type consists of additive homomorphic encryption and multiplicative homomorphic encryption separately, while the former type can support both addition and multiplication over ciphertext simultaneously. We omit the somewhat HE for simplicity.

Gentry [4] proposed the first full HE scheme based on lattice-based cryptography. While a number of following

works (e.g., [26], [9], [10]) have been proposed afterwards, it is still not practical to implement in real-life applications. The partial homomorphic encryption is often considered as a suitable alternative in practice. For example, Paillier cryptosystem [25] is supporting addition over ciphertext, while ElGamal cryotosystem [27] is supporting multiplication over ciphertext.

Based on the practical Paillier cryptosystem, Peter et al. [11] proposed an efficient outsourcing SMC protocol which is proven to be secure in honest-but-curious model. In particular, their proposed method can be used for privacy-preserving face authentication. Later on, Liu et al. [12] proposed an efficient outsourcing toolkits for SMC protocols. To support various computations (e.g., multiplication, less than and division) in cloud, Liu et al. proposed a new cryptographic primitive: distributed two trapdoors public key cryptosystem (DT-PKC) (which is an extension from [28]).

This work aims to exploit some inherent features of DT-PKC for remote user authentication. In particular, we discover that such kind of homomorphic cryptosystems [28], [11], [12] have desired "key privacy" [29] property, which will be formally defined and analyzed in III-C.

The remainder of this paper is structured as follows: In the next Section, we formalize the system model and the threat models (namely, biometrics privacy and user privacy). In Section 3, we describe some preliminaries which will be used in our proposed constructions, and present the proposed authentication framework. We then present our security analysis and performance analysis in Section 4 and 5 respectively. The paper is concluded in Section 6.

II. Security Model

In this section, we present the corresponding models for privacy-preserving biometric-based remote user authentication (PriBioAuth) protocols. As mentioned in the introduction, a PriBioAuth should achieve several security and privacy goals: biometrics privacy and user privacy. We first present a notation Table I below.

TABLE I
SUMMARY OF NOTATIONS

Notation	Definition
$\mathtt{pk}_i/\mathtt{sk}_i$:	User i's public/secret key
ID_i:	Identity of user i
$\mathtt{dist}(x, y)$:	Distance between vector x and vector y
$t \in \mathbb{R}^+$:	Threshold value (positive real number)
$\mathcal{B}$:	Plain biometrics
$\mathcal{C}$:	Reference biometrics
$\mathbb{N}$:	Dimension of biometrics
$\mathbb{Z}$:	Finite field
n:	Number of users
k:	Number of secret credentials
$[\![x]\!]$ (i.e., $[\![x]\!]_{\mathtt{pk}}$):	Encryption on x under the public key $\mathtt{pk}$
(N, g):	Public paramaters in DT-PKC
S:	Splitting technique in DT-PKC
Enc:	Encryption algorithm in DT-PKC
Dec:	Decryption algorithm in DT-PKC
PD$(1/2)$:	Partial decryption algorithm in DT-PKC

A. System Model

We present a biometric-based remote user authentication system involving three types of entities: key generation center (KGC), requested user (RU) and authentication server CP (which may consist of an additional computational cloud server (CSP)). We then define a biometric-based remote user authentication which consists of the following algorithms:

- Setup: The KGC takes the security parameter $\mathfrak{D}$ as input, outputs a master public/secret key pair $(\mathtt{mpk}, \mathtt{msk})$. In addition, KGC outputs a set of credentials $\{\mathtt{msk}^{(i)}\}^k$, and distributes them to respective CP and CSP_i through a secure channel.
- KeyGen. User takes master public key $\mathtt{mpk}$ as input, outputs a public/secret key pair $(\mathtt{pk}, \mathtt{sk})$.
- Registration. User enrolls his/her identity ID along with a reference biometrics $\mathcal{C}$ to CP[2]. There may exist an interactive algorithm between the CP and a CSP_i in cloud. User becomes a RU after registration.
- Authentication. RU sends his/her identity ID and a candidate biometrics $\mathcal{C}'$ to the cloud server CP, then CP accept it if and only if $\mathtt{dist}(\mathcal{C}', \mathcal{C}) \leq t$. There may exist an interactive algorithm between CP and CSP_i in cloud.

Remark. Note that the reference and candidate biometrics are in encrypted format, more specifically, they are encrypted under user's own public key.

B. Threat Model

1) Biometrics Privacy: Informally, an adversary attempts to learn user's plain biometrics. Below is the biometrics privacy game between an adversary $\mathcal{A}$ and a simulator $\mathcal{S}$ as follows.

- Setup: $\mathcal{S}$ first generates public/secret key pairs $(\mathtt{pk}_i, \mathtt{sk}_i)$ $(i \in [1, n])$ for n users and m servers respectively in the system. In addition, $\mathcal{S}$ generates a set of secret credentials $\{\mathtt{sk}^{(j)}\}_{j=1}^k$ for k $(k \leq m)$ servers. $\mathcal{S}$ also generates user's plain biometrics $\{\mathcal{B}_i\}$ and their corresponding reference biometrics $\{\mathcal{C}_i\}$, and returns all reference biometrics to $\mathcal{A}$. $\mathcal{S}$ eventually tosses a random coin b which will be used later in the game.
- Training: $\mathcal{A}$ can make the following queries in arbitrary sequence to $\mathcal{S}$.
  - Send: If $\mathcal{A}$ issues a send query in the form of (ID, i, msg) (resp. (CP, i, msg)) to simulate a network message for the i-th session of user ID (resp. server CP), then $\mathcal{S}$ would simulate the reaction of instance oracle Π^i_{ID} (resp. Π^i_{CP})[3] upon receiving message msg, and return to $\mathcal{A}$ the response that Π^i_{ID} (Π^i_{CP}) would generate. If $\mathcal{A}$ issues a Send query in the form of $(\mathrm{ID}', \text{`}start'\text{)}$ (resp. $(\mathrm{CP}', \text{`}start'\text{)}$), then $\mathcal{S}$ creates a new instance oracle $\Pi^i_{\mathrm{ID}'}$ (resp. $\Pi^i_{\mathrm{CP}'}$) and returns to $\mathcal{A}$ the first protocol message.
  - Secret Key Reveal: If $\mathcal{A}$ issues a Secret Key Reveal (or corrupt, for short) query to user i, then $\mathcal{S}$ returns user i's secret key $\mathtt{sk}_i$ to $\mathcal{A}$. Note that $\mathcal{A}$ is allowed to issue at most n-1 Secret Key Reveal queries to $\mathcal{S}$. We denote the honest (i.e., uncorrupted) user set as $\mathcal{U}'$.
  - Secret Credential Reveal: If $\mathcal{A}$ issues a credential reveal query to the CP, then $\mathcal{S}$ returns CP's secret credential $\mathtt{sk}^{(j)}$ to $\mathcal{A}$.

[2]Note that the binding between user identity ID and his/her public key $\mathtt{pk}$ is authenticated by a certificate $\mathtt{cert}$ issued by KGC.

[3]We denote the i-th session established by user ID as instance oracle Π^i_{ID}.

- Challenge: $\mathcal{A}$ randomly chooses two challenge biometrics $(\mathcal{B}_0, \mathcal{B}_1)(\notin \{\mathcal{B}_i\})$ of a challenge user $\mathrm{ID}_i \in \mathcal{U}'$, and sends the challenge biometrics to $\mathcal{S}$. $\mathcal{S}$ simulates the reference biometrics of user U_i by either $\mathcal{C}_b^* = \mathsf{F}(\mathtt{pk}_i, \mathcal{B}_0)$ if $b = 0$ or $\mathcal{C}_b^* = \mathsf{F}(\mathtt{pk}_i, \mathcal{B}_1)$ if $b = 1$.
  Note that $\mathcal{A}$ is allowed to reveal k-1 secret credentials (by corrupting servers), and F denotes a probabilistic algorithm. Finally, $\mathcal{A}$ outputs b' as its guess for b. If $b' = b$, then $\mathcal{S}$ outputs 1; otherwise, $\mathcal{S}$ outputs 0. We define the advantage of an adversary $\mathcal{A}$ in the above game as

$$\mathtt{Adv}_{\mathcal{A}}(\mathfrak{O}, k) = |\Pr[\mathcal{S} \to 1] - 1/2|.$$

Definition 2.1: We say that a PriBioAuth scheme has *biometrics privacy* if for any probabilistic polynomial-time (PPT) $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A}}(\mathfrak{O}, k)$ is a *negligible* function of the security parameter $\mathfrak{O}$.

2) User Privacy: Informally, an adversary attempts to identify the users involved in a biometric-based remote user authentication protocol. Below is the user privacy game between an adversary $\mathcal{A}$ and a simulator $\mathcal{S}$ as follows.

- Setup: $\mathcal{S}$ first generates public/secret key pairs $(\mathtt{pk}_i, \mathtt{sk}_i)$ $(i \in [1, n])$ for n users and m servers respectively in the system. In addition, $\mathcal{S}$ generates a set of secret credentials $\{\mathtt{sk}^{(j)}\}_{j=1}^{k}$ for k $(k \leq m)$ servers. $\mathcal{S}$ also generates user's plain biometrics $\{\mathcal{B}_i\}$ and their corresponding reference biometrics $\{\mathcal{C}_i\}$, and returns all public information (including $\{\mathcal{C}_i\}$) to $\mathcal{A}$. $\mathcal{S}$ eventually tosses a random coin b which will be used later in the game.
- Training: $\mathcal{A}$ is allowed to issue Send query, at most n-2 Secret Key Reveal and k-1 Secret Credential Reveal queries to $\mathcal{S}$. We denote the honest (i.e., uncorrupted) user set as $\mathcal{U}'$.
- Challenge: $\mathcal{A}$ randomly selects two users $\mathrm{ID}_i, \mathrm{ID}_j \in \mathcal{U}'$ as challenge candidates, then $\mathcal{S}$ removes them from $\mathcal{U}'$ and simulates ID_b^* to $\mathcal{A}$ by either $\mathrm{ID}_b^* = \mathrm{ID}_i$ if $b = 0$ or $\mathrm{ID}_b^* = U_j$ if $b = 1$.

$$\mathcal{A} \Leftrightarrow \mathrm{ID}_b^* = \begin{cases} \mathrm{ID}_i & b = 0 \\ \mathrm{ID}_j & b = 1 \end{cases}$$

  Let $\mathcal{A}$ interact with ID_b^*. Finally, $\mathcal{A}$ outputs b' as its guess for b. If $b' = b$, then $\mathcal{S}$ outputs 1; otherwise, $\mathcal{S}$ outputs 0. We define the advantage of an adversary $\mathcal{A}$ in the above game as

$$\mathtt{Adv}_{\mathcal{A}}(\mathfrak{O}, k) = |\Pr[\mathcal{S} \to 1] - 1/2|.$$

Definition 2.2: We say that a PriBioAuth scheme has *user privacy* if for any PPT $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A}}(\mathfrak{O}, k)$ is a *negligible* function of the security parameter $\mathfrak{O}$.

Remark. We assume a passive adversary, who is able to monitor or eavesdropping (*except* modifying or tampering) all transcripts send on the network. We consider an honest-but-curious model in this work, which is formalized by some existing works (e.g., [20], [19], [11]). Specifically, the request user and the authentication server are assumed to execute the protocol as specified, just try to learn additional information from the transcript and intermediate results during protocol execution.

III. Proposed Construction

A. Preliminary

We briefly present some secure computation protocols described in [12], which will be used in our proposed user authentication framework. Note that we just mention their functionality for simplicity.

- Secure Less Than Protocol (SLT). We assume two encrypted integers $[\![x]\!]$ and $[\![y]\!]$, the SLT protocol will provide an encrypted results $[\![u]\!]$, which can be used to determine the relationship between the plaintexts of two encrypted integers (i.e., $x > y$ or $x \leq y$). As a result, $u = 0$ indicates $x > y$, and $u = 1$ indicates $x \leq y$.
- Secure Equivalent Testing Protocol (SEQ). Given two encrypted integers $[\![x]\!]$ and $[\![y]\!]$, SEQ will provide the encrypted results $[\![f]\!]$ to determine whether the plaintext of the two encrypted integers are equivalent (i.e., $x \stackrel{?}{=} y$). As a result, $f = 1$ indicates $x = y$, and $f = 0$ indicates $x \neq y$.
- Secure Multiplicative Computation Protocol (SMT). Given two encrypted integers $[\![x]\!]$ and $[\![y]\!]$ as input, the SMT can generate the result $[\![x \cdot y]\!]$ by using two non-colluding cloud servers CP and CSP.

B. Secure Euclidean Distance Computation Protocol (SEDC)

We present the proposed secure Euclidean distance computation protocol. We use Fingerprints as the candidate of biometrics, which is represented by *FingerCode*. The *FingerCode* [30] is typically a $\mathbb{N}$-dimensional (e.g., $\mathbb{N}$=640) feature vector, and each entry is a 8-bit integer. The Euclidean distance $d = \mathsf{dist}(\mathcal{B}, \mathcal{B}')$ between reference biometrics $\mathcal{B} = (v_1, \cdots, v_{\mathbb{N}})$ and candidate biometrics $\mathcal{B}' = (v'_1, \cdots, v'_{\mathbb{N}})$ is calculated as.

$$\begin{aligned} d &= \Sigma_{j=1}^{\mathbb{N}}(v_j - v'_j)^2 \\ &= (v_1 - v'_1)^2 + (v_2 - v_2)^2 + \cdots + (v_{\mathbb{N}} - v'_{\mathbb{N}})^2 \\ &= \Sigma_{j=1}^{\mathbb{N}} v_j^2 + \Sigma_{j=1}^{\mathbb{N}}(-2v_j \cdot v'_j) + \Sigma_{j=1}^{\mathbb{N}} v_j'^2. \end{aligned}$$

Note that the CP and CSP perform the biometrics matching between encrypted vectors $[\![\mathcal{B}]\!] = \{[\![v_j]\!]\}_{j=1}^{\mathbb{N}}$ and $[\![\mathcal{B}']\!] = \{[\![v'_j]\!]\}_{j=1}^{\mathbb{N}}$ as shown in Figure 1.

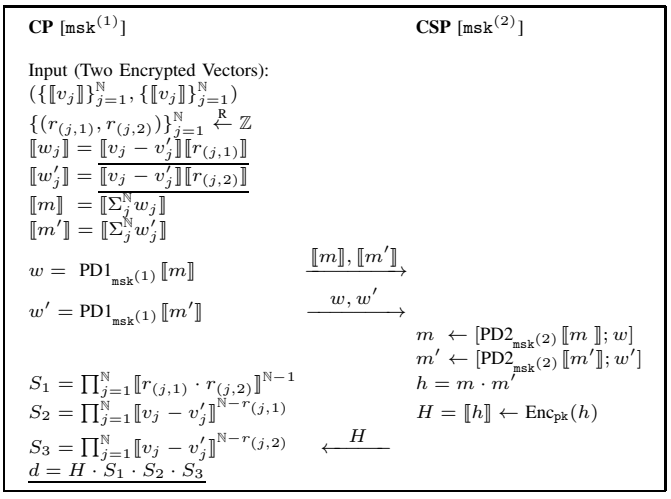


Fig. 1. Secure Euclidean Distance Computation Protocol (SEDC).

Correctness of d. CP performs the following calculation, and $h = m \cdot m' = \Sigma_{j=1}^{\mathbb{N}}[(v_j - v'_j) \cdot r_{(j,1)}][(v_j - v'_j) \cdot r_{(j,2)}]$.

$$\begin{aligned} d &= H \cdot S_1 \cdot S_2 \cdot S_3 \\ &= [\![\Sigma_{j=1}^{\mathbb{N}}[(v_j - v'_j) \cdot r_{(j,1)}][(v_j - v'_j) \cdot r_{(j,2)}] \\ &\quad -\Sigma_{j=1}^{\mathbb{N}}[r_{(j,1)} \cdot (v_j - v'_j) + r_{(j,2)} \cdot (v_j - v'_j) \\ &\quad +r_{(j,1)} \cdot r_{(j,2)}]] = [\![\Sigma_{j=1}^{\mathbb{N}}(v_j - v'_j)^2]\!]. \end{aligned}$$

C. Another look of DT-PKC

The underlying DT-PKC is the main building block of the proposed PriBioAuth framework. We discover that the DT-PKC has an inherent feature: "key privacy", which is introduced by Bellare et al. [29]. It means that an adversary in possession of a ciphertext cannot tell which specific key, out of a set of known public keys, is the one under which the ciphtertext was created. In particular, they formalized a new model: "indistinguishability of keys" (IK). We formally prove the DT-PKC cryptosystem is secure in the IK-CPA model, in addition to its IND-CPA security [12]. We believe that both BCP [28] and its variant DT-PKC cryptosystem have such implicit property.

1) Security model of key privacy:

Definition 3.1: The IK-CPA experiment between an adversary $\mathcal{A}$ and a simulator $\mathcal{S}$ is defined below [29].

$$\begin{aligned} &\text{Experiment } \mathsf{Exp}_{\mathsf{PE}}^{\text{IK-CPA}}(\mathfrak{O}) \\ &(\mathtt{pk}_0, \mathtt{sk}_0), (\mathtt{pk}_1, \mathtt{sk}_1) \leftarrow \mathsf{KeyGen}(1^{\mathfrak{O}}) \\ &(msg^*, st) \leftarrow \mathcal{A}(\text{find}, \mathtt{pk}_0, \mathtt{pk}_1) \\ &C^* \leftarrow \mathsf{Enc}_{\mathtt{pk}_b}(msg^*) \\ &b' = \mathcal{A}(\text{guess}, st, C^*) \\ &If\ b' = b, return\ 1; else, return\ 0. \end{aligned}$$

Note that st denotes some state information. We define the advantage of the adversary as

$$\mathtt{Adv}_{\mathcal{A}}^{\text{IK-CPA}}(\mathfrak{O}) = |\Pr[\mathcal{S} \rightarrow 1] - 1/2|. \quad (1)$$

Definition 3.2: An encryption scheme $(\mathsf{PE}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be IK-CPA secure if $\mathtt{Adv}_{\mathcal{A}}^{\text{IK-CPA}}(\mathfrak{O})$ is negligible in $\mathfrak{O}$ for any PPT adversary $\mathcal{A}$.

2) Security of DT-PKC: We prove the DT-PKC is IK-CPA secure if the underlying DDH assumption holds in group $\mathbb{Z}_{N^2}^*$ [28]. In particular, we assume the factorization of the modulus N is hard, or the DDH assumption over $\mathbb{Z}_{N^2}^*$ turns out to be easy (refer to Theorem 4 in [28] for detailed relations).

Theorem 3.3: The DT-PKC achieves IK-CPA security if the DDH assumption holds in $\mathbb{Z}_{N^2}^*$.

The detailed proof is deferred to the full version of this work due to page limitation.

D. Proposed Framework

We now present our privacy-preserving biometric-based remote user authentication (PriBioAuth) framework. KGC first generates two secret credentials and distributes them to CP and CSP respectively. A RU encrypts ID and biometrics using his/her own public key, and sends them to CP for Registration. As for Authentication, RU sends encrypted ID and candidate biometrics to CP, while CP accept RU iff the candidate biometrics is "close enough" to RU's reference biometrics. In particular, we assume that CP stores a set of encrypted identities and biometrics information after Registration.

Problem Statement. In the Authentication stage, the candidate biometrics should be compared with reference biometrics in database. The obvious problem is that a set of enrolled biometrics are not under *same* public key, while the underlying DT-PKC requires homomorphic operations under the same public key. Another problem is that CP should perform biometrics matching between one record in database and candidate identity/biometrics. In other words, CP can explicitly *link* the anonymous authenticated RU to a specific record in database.

High-level Description. To address the above problems, we first need an additional procedure to fix these "various" encrypted data prior to the actual biometrics matching between CP and CSP. Specifically, CP partially decrypts reference data using distributed secret credential, and sends them to CSP for full decryption on reference data. Then CSP randomly chooses a "dummy" public key $\mathtt{pk}^*$ such that $\mathtt{pk}^* \neq \{\mathtt{pk}_i, \mathtt{pk}_{\text{CP}}\}$, and re-encrypts data using $\mathtt{pk}^*$[4].

After anonymous key transformation (AKeyTrans) during Authentication, CP and CSP run the corresponding SLT and SEQ protocols on candidate identity and reference identity. Consequently, CP and CSP run the SEDC and SLT protocols to obtain the relationship between candidate biometrics and reference biometrics. If both SEQ protocol and SLT protocol output "$[\![1]\!]_*$" (encryption under public key $\mathtt{pk}^*$), then CP authenticates a requested user RU.

To achieve the claimed user privacy, CP will go through all records in database when authenticating a RU. More precisely, CP obtains a set of individual encrypted results $\{[\![0]\!]_*, [\![1]\!]_*, \cdots, [\![0]\!]_*\}$ after going through the entire database; then CP can obtain the encrypted final results $[\![1]\!]_* (= [\![0]\!]_* \cdot [\![1]\!]_* \cdots [\![0]\!]_*)$. After interacting with CSP, CP outputs the plain authentication results "1" iff the candidate identity/biometrics is matching one of records in database. We present the detailed PriBioAuth framework below.

- Setup: KGC takes the security parameter as input, outputs master public/secret key pair $(\mathtt{mpk}, \mathtt{msk})$. In addition, KGC outputs two secret credentials $(\mathtt{msk}^{(1)}, \mathtt{msk}^{(2)}) \leftarrow \text{S}(\mathtt{msk})$, and distributes them to CP and CSP respectively.
- KeyGen: User takes master public key $\mathtt{mpk}$ as input, outputs a public/secret key pair $(\mathtt{pk}, \mathtt{sk})$.
- Registration: User randomly chooses a nonce r first; then computes reference identity $[\![\text{ID}]\!]$, biometrics $[\![\mathcal{B}]\!]$ (i.e., $\mathsf{Enc}_{\mathtt{pk}}(\mathcal{B})$), and two encrypted nonces $[\![r]\!], [\![r]\!]_*$ (the second one is using public key $\mathtt{pk}^*$). Eventually, user sends his/her identity ID and all encrypted values to CP. In particular, CP and CSP perform the AKeyTrans protocol as described in Fig. 2.

  Note that $\mathcal{B} = (v_1, \cdots, v_{\mathbb{N}}) = \{v_j\}_{j=1}^{\mathbb{N}}$, and CP holds a set of transformed reference identity/biometrics $\{(\text{ID}_i, [\![\text{ID}_i]\!]_*, [\![\mathcal{B}_i]\!]_*)\}$ under public key $\mathtt{pk}^*$.

[4]The secret key $\mathtt{sk}^*$ is unknown to all RU, CP and CSP, while $\mathtt{pk}^*$ is known to all users.

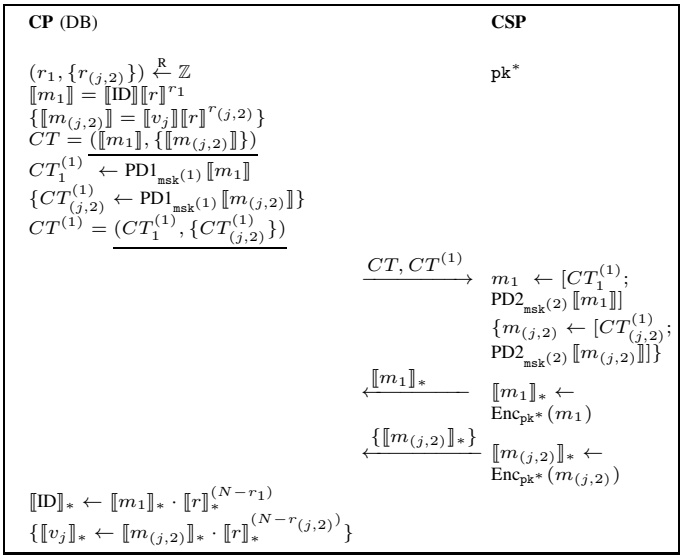


Fig. 2. AKeyTrans Protocol under Public Key $\mathtt{pk}^*$.

- Authentication: RU generates the candidate request using the same method described above, and sends message $([\![\mathrm{ID}]\!], [\![\mathcal{B}']\!], [\![r_{\mathrm{RU}}]\!], [\![r_{\mathrm{RU}}]\!]_*)$ as authentication Request to CP. Then CP and CSP take one record in database $([\![\mathrm{ID}_i]\!]_*, [\![\mathcal{B}_i]\!]_*)$ as reference input, and perform user authentication as specified in Fig. 3. Eventually, CP accept RU if the final results is "1"; otherwise, CP outputs "$\perp$".

Remark 1. In order to prevent replay attacks, we can use the time-stamp since the proposed framework requires no user interaction. More specifically, RU generates an encrypted time-stamp $[\![TS']\!]_*$ and sends it to CP. The rest procedure will follow the protocol specification in Fig. 3. Another countermeasure is to let CP to store all seen requested values (in a certain time-window encoded in an additional nonce value) from RU in order to detect and reject repeated requests with the same value and nonce.

Remark 2. We can use the *packing* technique [17], [19] to save both computation and bandwidth between RU and CP. We assume RU sends encrypted biometrics $[\![\mathcal{B}]\!] = \{[\![v_{j+\mathtt{K}}]\!]\}$ to CP ($\mathtt{K}$ denote the number of single entry which is "packed" into one ciphertext). According to packing implementation in [17], [19], we know that $\mathtt{K} = 20$ if a 1024-bit modulus is used in Paillier cryptosystem. Given the "packed" ciphertext, CP and CSP run the SEDC protocol afterwards.

However, there may exist a serious problem. If one bit of candidate "packed" biometrics does not match the reference "packed" biometrics, then the result of Euclidean distance may easily beyond threshold t. To let CP and CSP perform SEDC protocol successfully, we can use secure multi-bit extraction (MBE) protocol and secure ciphertext partition (SCP) protocol in [31] to extract the correct (sliced) ciphertext with respect to single integer.

IV. Security Analysis

Theorem 4.1: The proposed PriBioAuth framework has biometrics privacy if the underlying DT-PKC is semantically (IND-CPA) secure.

The proof of biometrics privacy is obvious, because if an attacker can break the biometrics privacy security, then we can construct an efficient algorithm to break the IND-CPA security of underlying DT-PKC.

Theorem 4.2: The proposed PriBioAuth framework has user privacy if the underlying DT-PKC is IK-CPA secure.

Proof 1: We define a sequence of games $\mathbb{G}_i$, $i = 0, \cdots, 3$ and let $\mathtt{Adv}_i^{\mathrm{PriBioAuth}}$ denote the advantage of the adversary in game $\mathbb{G}_i$.

- $\mathbb{G}_0$ This is the original game for user privacy.
- $\mathbb{G}_1$ This game is identical to game $\mathbb{G}_0$ except that at the challenge stage, we replace the component $[\![\mathrm{ID}_i]\!]$ of first message by $[\![\mathrm{ID}_i]\!]_R$, where R is a random public key. Below we show the difference between $\mathbb{G}_0$ and $\mathbb{G}_1$ is negligible under the assumption that the DT-PKC is IK-CPA secure. Let $\mathcal{S}$ denote an attacker against DT-PKC, who is given challenge public keys $(\mathtt{pk}_0, \mathtt{pk}_1)$, aims to break the IK-CPA security of DT-PKC. $\mathcal{S}$ simulates the game for $\mathcal{A}$ as follows.
  - Setup: $\mathcal{S}$ first generates public/secret key pair $(\mathtt{pk}_j, \mathtt{sk}_j)$ for n-1 users and two servers (CP and CSP). In addition, $\mathcal{S}$ generates a public/secret key pair $(\mathtt{pk}^*, \mathtt{sk}^*)$ for anonymous key-transformation. $\mathcal{S}$ also honestly generates two secret credentials $\mathtt{sk}^{(1)}, \mathtt{sk}^{(2)}$ for CP and CSP. $\mathcal{S}$ generates user's plain biometrics $\{\mathcal{B}_j\}$ and their corresponding reference biometrics $\{[\![\mathcal{B}_j]\!]_{\mathtt{pk}_j}\}$. $\mathcal{S}$ sets public key of user i ($i \neq j$) as $\mathtt{pk}_0$. It is obvious that $\mathcal{S}$ can answer all the queries made by $\mathcal{A}$ except user i. Below we mainly focus on the simulation of user i.
  - Training: $\mathcal{S}$ answers $\mathcal{A}$'s queries as follows.
    - If $\mathcal{A}$ issues a send query in the form of $(\mathrm{ID}', `start')$ to $\mathcal{S}$, then $\mathcal{S}$ will return $([\![\mathrm{ID}']\!]_{\mathtt{pk}'}, [\![\mathcal{B}']\!]_{\mathtt{pk}'})$ to $\mathcal{A}$. Note that ID' and $\mathcal{B}'(\notin \{\mathcal{B}_i\})$ are encrypted using ID'.
      If $\mathcal{A}$ issues a send query in the form of (CSP, i, msg) to $\mathcal{S}$, then $\mathcal{S}$ decrypts msg (using secret credential) and returns the ciphertext which is encrypted using the public key $\mathtt{pk}^*$. msg denotes the partially decrypted randomized ciphertext. In particular, if the randomized ciphertext denote as $[\![z + r]\!]_{\mathtt{pk}^*}$, where r is randomly chosen by $\mathcal{A}$, then $\mathcal{S}$ obtains the message $z + r$ and returns the ciphertext which is encrypted using $\mathtt{pk}_{\mathrm{CP}}$. Note that $\mathcal{S}$ uses the same method to simulate transmitted message if $\mathcal{A}$ issues a send query in the form of (CP, i, msg). Also note that the send query is mainly used to simulate the transmitted messages of all corresponding subprotocols (such as SEQ, SEDC, SLT, etc) between CP and CSP.
    - If $\mathcal{A}$ issues secret key reveal query to user i, then $\mathcal{S}$ $\mathtt{abort}$. Recall that $\mathcal{A}$ has the following restrictions: 1) $\mathcal{A}$ can corrupt at most n-2 users; 2) $\mathcal{A}$ can corrupt (secret credential reveal) either CP or CSP; 3) $\mathcal{A}$ cannot corrupt key pair $(\mathtt{pk}^*, \mathtt{sk}^*)$.
  - Challenge: $\mathcal{S}$ first follows the user privacy game to select ID_b. If the challenge user ID_b is not user i, then $\mathtt{abort}$; otherwise, $\mathcal{S}$ sets the challenge message in its IK-CPA game as $m = \mathrm{ID}_i$ and receives a challenge ciphertext C^* from its own challenger. Eventually, $\mathcal{S}$ generates the complete transcript as $(C^*, [\![\mathcal{B}_i]\!], [\![r_i]\!], [\![r_i]\!]_{\mathtt{pk}^*})$ (where

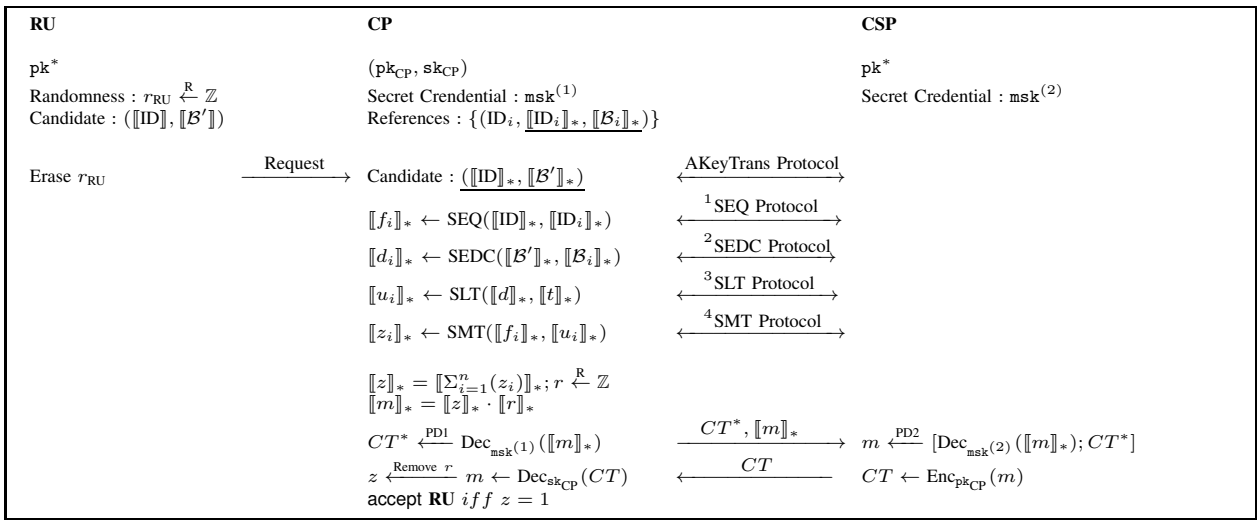


Fig. 3. Authentication with Corresponding Sub-computational Protocols.

randomness r_i is chosen by $\mathcal{S}$.) and sends it to $\mathcal{A}$ as the transmitted message from RU to CP.

If the public key used to generate C^* is $\mathtt{pk}_0$, then the simulation is consistent with $\mathbb{G}_0$; otherwise, the simulation is consistent with $\mathbb{G}_1$. Therefore, if the advantage of $\mathcal{A}$ is significantly different in $\mathbb{G}_0$ and $\mathbb{G}_1$, $\mathcal{S}$ can break the IK-CPA security of DT-PKC. Hence, we have

$$\left|\mathtt{Adv}_0^{\text{PriBioAuth}} - \mathtt{Adv}_1^{\text{PriBioAuth}}\right| \leq n \cdot \mathtt{Adv}_{\mathcal{S}}^{\text{IK-CPA}}(\mathfrak{O}). \quad (2)$$

- $\mathbb{G}_2$ This game is identical to game $\mathbb{G}_0$ except that at the challenge stage, we replace the component $[\![\mathcal{B}_i]\!]$ of first message by $[\![\mathcal{B}_i]\!]_R$, where R is a random public key. By following the same analysis as above, we have

$$\left|\mathtt{Adv}_1^{\text{PriBioAuth}} - \mathtt{Adv}_2^{\text{PriBioAuth}}\right| \leq n \cdot \mathtt{Adv}_{\mathcal{S}}^{\text{IK-CPA}}(\mathfrak{O}). \quad (3)$$

  We assume that there is a sequence of sub-games $\mathbb{G}_{2,i}$ $(1 \leq i \leq \mathbb{N})$ in game $\mathbb{G}_2$. The actual number of sub-games depends on both the dimensional of biometrics $\mathcal{B}_i$ and if the packing technique is used or not.
- $\mathbb{G}_3$ This game is identical to game $\mathbb{G}_0$ except that at the challenge stage, we replace the component $[\![r_i]\!]$ of first message by $[\![r_i]\!]_R$, where R is a random public key. By following the same analysis as above, we have

$$\left|\mathtt{Adv}_2^{\text{PriBioAuth}} - \mathtt{Adv}_3^{\text{PriBioAuth}}\right| \leq n \cdot \mathtt{Adv}_{\mathcal{S}}^{\text{IK-CPA}}(\mathfrak{O}). \quad (4)$$

By combing the above results together, we have.

$$\mathtt{Adv}_{\mathcal{A}}^{\text{PriBioAuth}}(\mathfrak{O}) \leq 3 \cdot n \cdot \mathtt{Adv}_{\mathcal{S}}^{\text{IK-CPA}}(\mathfrak{O}).$$

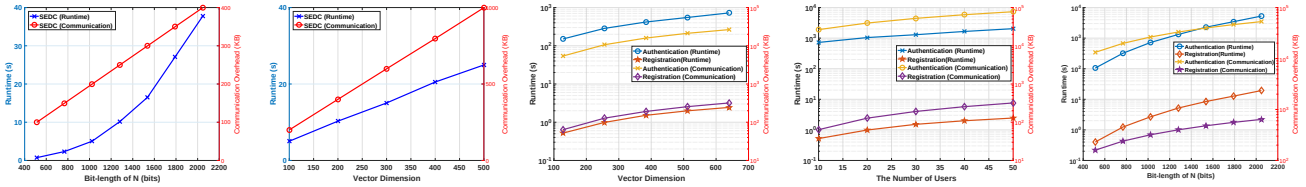
V. Performance Analysis

This experiment was run on virtual machines (3.6 GHz single-core processor and 6 GB RAM memory). The experiment assumes that user's biometric information has been converted into the format needed, because the representation (depends on the feature extraction algorithms) of biometric data may vary. The running time and communication cost mainly depend on the bit length of N. Two extra factors are also needed to be considered, one is the vector dimension $\mathbb{N}$, and the other one is the number of users n when evaluating the proposed PriBioAuth framework. The comprehensive performance analysis is presented below.

1) SEDC sub-protocol. The SEDC sub-protocol is essential for the efficiency of our proposed framework. We analyze its performance at Fig. 4a and Fig. 4b respectively, and we observe that both the running time (see left coordinate) and communication cost (see right coordinate) increase with bit length N and vector dimension $\mathbb{N}$. In particular, the vector dimension of biometrics here ranges from 100 to 500, and each vector is a 8-bit integer. Note that the efficiency of SEDC sub-protocol will grow linearly with the dimension of extracted feature vectors $\mathbb{N}$.
2) PriBioAuth framework. From Fig. 4c to Fig. 4e, we observe that the running time and communication cost will increase with vector dimension $\mathbb{N}$, number of users n (10-50) and bit length N. We also observe that the PriBioAuth framework is linear with these factors. In particular, CP and CSP in Authentication stage perform more cryptographic operations than Registration stage, because the corresponding computational sub-protocols are required.
3) Time-Complexity (see Table II). The time-complexity relies on the size of public parameter N, the number of records in database n, the number of addition, multiplication and exponentiation operations. Let $\mathcal{O}(N)$ be a linear time algorithm, $\mathcal{O}(N^{\alpha})$ denotes a polynomial time algorithm for constant α and sets $\alpha = 3$ w.r.t. exponentiation. Note that the Retrieval means that CP retrieves the outcome of authentication from CSP. We stress that the action of RU (e.g., a resource-limited device without storing any secret keys) is just Pallier encryption on ID and plain biometrics, while CP and CSP in cloud collaboratively run the corresponding sub-protocols without interacting with RU.

VI. Conclusion

In this paper, we proposed a framework of privacy-preserving biometric-based remote user authentication using



(a) Running time and Communication cost (vary with bit length of N)

(b) Running time and Communication cost (vary with vector dimension $\mathbb{N}$, set N=1024)

(c) Running time and Communication cost (vary with vector dimension $\mathbb{N}$, set n=20)

(d) Running time and Communication cost (vary with number of users n, set N=1024)

(e) Running time and Communication cost (vary with bit length of N)

Fig. 4. Evaluation findings of PriBioAuth and its corresponding sub-protocols.

TABLE II
THE COMPLEXITY COST OF PRIBIOAUTH.

Stages	PriBioAuth
Reg	$\mathcal{O}(N^3)$ on RU
AKeyTrans	$\mathcal{O}(N^3)$ on CP and CSP
SEQ	$n \cdot \mathcal{O}(N^3)$ on CP and CSP
SEDC	$n \cdot \mathcal{O}(N^3)$ on CP and CSP
SLT	$n \cdot \mathcal{O}(N^3)$ on CP and CSP
SMT	$n \cdot \mathcal{O}(N^3)$ on CP and CSP
Retrieval	$\mathcal{O}(N^3)$ on CP and CSP

homomorphic encryption. We also defined the new formal security models for biometrics privacy and user privacy, and proved the security of the proposed framework in the standard model. We leave the construction of biometric-based remote user authentication without going through the whole database as our future work, such that the time-complexity is not linear in the number of enrolled users.

ACKNOWLEDGEMENTS

The work is supported by the Singapore National Research Foundation under NCR Award Number NRF2014NCR-NCR001-012. It is also supported by AXA Research Fund.

REFERENCES

[1] A. K. Jain, K. Nandakumar, and A. Ross, "50 years of biometric research: Accomplishments, challenges, and opportunities," *Pattern Recognition Letters*, vol. 79, pp. 80–105, 2016.

[2] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," *EURASIP Journal on advances in signal processing*, vol. 2008, p. 113, 2008.

[3] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *EUROCRYPT*, 2004, pp. 523–540.

[4] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *STOC*, 2009, pp. 169–169.

[5] X. Boyen, "Reusable cryptographic fuzzy extractors," in *ACM CCS*, 2004, pp. 82–91.

[6] N. Li, F. Guo, Y. Mu, W. Susilo, and S. Nepal, "Fuzzy extractors for biometric identification," in *ICDCS*, 2017, pp. 667–677.

[7] Y. Zhao, "Identity-concealed authenticated encryption and key exchange," in *ACM CCS*, 2016, pp. 1464–1479.

[8] M. Maffei, G. Malavolta, M. Reinert, and D. Schröder, "Privacy and access control for outsourced personal records," in *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 2015, pp. 341–358.

[9] S. Halevi and V. Shoup, "Helib-an implementation of homomorphic encryption," *Cryptology ePrint Archive, Report 2014/039*, 2014.

[10] X. Liu, R. Deng, K.-K. R. Choo, Y. Yang, and H. Pang, "Privacy-preserving outsourced calculation toolkit in the cloud," *IEEE TDSC*, 2018.

[11] A. Peter, E. Tews, and S. Katzenbeisser, "Efficiently outsourcing multiparty computation under multiple keys," *IEEE TIFS*, vol. 8, no. 12, pp. 2046–2058, 2013.

[12] X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE TIFS*, vol. 11, no. 11, pp. 2401–2414, 2016.

[13] J. Camenisch, M. Dubovitskaya, and G. Neven, "Oblivious transfer with access control," in *ACM CCS*, 2009, pp. 131–140.

[14] J. Han, W. Susilo, Y. Mu, M. H. Au, and J. Cao, "AAC-OT: accountable oblivious transfer with access control," *IEEE TIFS*, vol. 10, no. 12, pp. 2502–2514, 2015.

[15] J. Bringer, H. Chabanne, M. Izabachène, D. Pointcheval, Q. Tang, and S. Zimmer, "An application of the goldwasser-micali cryptosystem to biometric authentication," in *ACISP*, 2007, pp. 96–106.

[16] Q. Tang, J. Bringer, H. Chabanne, and D. Pointcheval, "A formal study of the privacy concerns in biometric-based remote authentication schemes," in *ISPEC*, 2008, pp. 56–70.

[17] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *ICISC*, 2009, pp. 229–244.

[18] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. Donida Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, F. Scotti *et al.*, "Privacy-preserving fingercode authentication," in *Proceedings of the 12th ACM workshop on Multimedia and security*, 2010, pp. 231–240.

[19] Y. Huang, L. Malka, D. Evans, and J. Katz, "Efficient privacy-preserving biometric identification," in *NDSS*, 2011.

[20] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *International Symposium on PET*, 2009, pp. 235–253.

[21] A. C.-C. Yao, "How to generate and exchange secrets," in *FOCS*, 1986, pp. 162–167.

[22] S. Goldwasser and S. Micali, "Probabilistic encryption and how to play mental poker keeping secret all private information," in *Proceedings 14th ACM Symposium on the Theory of Computing*, vol. 4.

[23] J. Daugman, "How iris recognition works," *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 21–30, 2004.

[24] B. Schoenmakers and P. Tuyls, "Efficient binary conversion for paillier encrypted values," in *EUROCRYPT*, 2006, pp. 522–537.

[25] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT*, 1999, pp. 223–238.

[26] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *CRYPTO*, 2013, pp. 75–92.

[27] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[28] E. Bresson, D. Catalano, and D. Pointcheval, "A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications," in *ASIACRYPT*, 2003, pp. 37–54.

[29] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval, "Key-privacy in public-key encryption," in *ASIACRYPT*, 2001, pp. 566–582.

[30] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Fingercode: a filterbank for fingerprint representation and matching," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, 1999, pp. 187–193.

[31] Y. Yang, X. Liu, R. H. Deng, and J. Weng, "Flexible wildcard searchable encryption system," *IEEE TSC*, 2017.