Theoretical Computer Science 814 (2020) 223–233

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Leakage-resilient biometric-based remote user authentication with fuzzy extractors

Yangguang Tian [a,*], Yingjiu Li [b], Binanda Sengupta [c,1], Nan Li [d], Chunhua Su [e]

[a] *School of Information Systems, Singapore Management University, Singapore*
[b] *Computer and Information Science Department, University of Oregon, USA*
[c] *Institute for Infocomm Research (I2R), Singapore*
[d] *School of Electrical Engineering and Computing, University of Newcastle, Australia*
[e] *Division of Computer Science, University of Aizu, Japan*

ARTICLE INFO

Article history:
Received 23 July 2018
Received in revised form 24 December 2019
Accepted 1 February 2020
Available online 4 February 2020
Communicated by G. Persiano

Keywords:
Leakage resilient
Remote user authentication
Cross platform setting

ABSTRACT

Fuzzy extractors convert biometrics and other noisy data into a cryptographic key for security applications such as remote user authentication. Leakage attacks, such as side channel attacks, have been extensively modelled and studied in the literature. However, to the best of our knowledge, leakage attacks to biometric-based remote user authentication with fuzzy extractors have never been studied rigorously. In this paper, we propose a generic framework of leakage-resilient and privacy-preserving biometric-based remote user authentication that allows an authorized user to securely authenticate herself to a remote authentication server using her biometrics. In particular, the authorized user relies only on her secret biometrics to perform a valid authentication — which is suitable for user authentications in a cross-platform setting.

1. 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 many attractive features using biometrics over conventional passwords. For example, people need to remember many secure passwords for various accounts and update passwords frequently for security reasons. By contrast, biometrics is permanently and uniquely associated to an individual, so it is convenient to use biometrics for user authentication.

Biometric-based user authentication also leads to some security and usability issues. First, biometrics is not revocable. If biometrics is compromised, then user may loss its security forever, especially for the single-factor-based user authentication [2–4]. Second, authorized users may concern the security of biometrics stored on the authentication server. Therefore, no biometrics should be stored in plaintext at the server side, because biometrics may contain a wealth of personal information (e.g., DNA).

There are mainly three methods to protect biometrics information: non-invertible transform [5], homomorphic cryptosystems [6], and fuzzy extractors [7]. The non-invertible transform relies on a secret key. Specifically, it is a two-factor

[*] Corresponding author.
E-mail addresses: ygtian@smu.edu.sg (Y. Tian), yingjiul@uoregon.edu (Y. Li), binanda_sengupta@i2r.a-star.edu.sg (B. Sengupta), nan.li@newcastle.edu.au (N. Li), chsu@u-aizu.ac.jp (C. Su).
[1] Work done while at Singapore Management University, Singapore.

https://doi.org/10.1016/j.tcs.2020.02.001
0304-3975/© 2020 Elsevier B.V. All rights reserved.

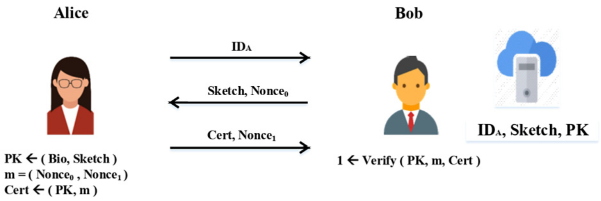

Fig. 1. Overview of biometrics-based user authentications. Server Bob maintains a database to store the enrolled information from all enrolled users, which includes user Alice with identity ID_A. A secure sketch (i.e., an "encrypted" form of biometrics) is used to recover the original biometrics from a nearby biometrics.

(biometrics plus secret key) user authentication and not scalable for a cross-platform setting (see the example below). This is because the secret key must be available at the time of authentication to transform the requested biometrics for subsequent user authentication. Homomorphic encryption is a straightforward approach to protect biometrics, where a biometric-matching (i.e., searches user's enrolled information from the backend database and makes a decision) is performed by the server in ciphertext. As a result, the server can obtain an authentication result without revealing the user's encrypted biometrics. However, it is not practical in a real-world environment (e.g., resource-constrained devices) due to its computational cost and system complexity [8].

In this work, we focus on biometric-based remote user authentication (BRUA) using fuzzy extractors that enables an authorized user Alice to authenticate herself to a remote server Bob using her biometrics. Specifically, Alice relies on her enrolled sketch stored at the server side in order to derive a public key for generating a certificate associated to a message (e.g., nonce). Bob then verifies the certificate based on Alice's enrolled public key (see Fig. 1). We emphasize that no information is stored locally at the user side, and no biometrics is stored in plaintext at the server side (i.e., biometric-privacy).

The BRUA with fuzzy extractors may be vulnerable to leakage attacks in the real world such as side channel attacks on computation time, power consumption, radiation, noise and heat emission. An attacker is able to obtain some imperfect information of the secret (e.g., biometrics) stored at either the user side or the server side. If an attacker is able to obtain imperfect/partial knowledge of a user's biometrics, then the user's security could be compromised, because the user may use the same biometrics multiple times across different authentication sessions. The main goal of this work is to design leakage-resilient biometric-based remote user authentication (LR-BRUA) with fuzzy extractors.

The number of authentication factors plays a critical role in the usability of user authentications. It is desirable that Alice treats her biometrics as the only secret information and uses it for user authentications. Specifically, the success of remote user authentication solely relies on her biometrics, which is suitable for user authentication where a user holds several devices (e.g., smart-phone and tablet) and accesses the same service provider from various platforms in a cross-platform setting. The advantage is that Alice does not need to store any other secret information (e.g., a secret key) in each of these devices. Therefore, another goal is to design a single-factor LR-BRUA using fuzzy extractors.

The single-factor LR-BRUA scheme with fuzzy extractors is significantly useful in real-world applications. We take mobile-device users enrolling/logging in a cloud service provider as an example. User authenticity prevents impersonation attacks from any third parties, and biometric-privacy prevents an honest-but-curious service provider from revealing the authorized user's secret biometrics. In particular, these attacks will not be successful under the leakage of secret information.

This work We introduce the concept of leakage-resilient and privacy-preserving biometric-based remote user authentication (LR-BRUA), such that an authorized user can authenticate herself to an honest-but-curious server using her biometrics. It ensures leakage resilience for any secrets involved in the system, while the biometric-privacy is held as well. Our contributions can be summarized as follows.

- *Generic Framework.* We introduce the *first* generic framework of LR-BRUA using fuzzy extractors in a single-factor setting, and prove that the proposed construction achieves leakage-resilient user authenticity and biometric-privacy.
- *Ease of Use.* The LR-BRUA is a single-factor user authentication (i.e., no storage for secret information is required at the user side), which is suitable for user authentication in a cross-platform setting.

1.1. Related work

Biometric-based authentication Biometric-privacy is a basic security requirement for biometric-based user authentication/identification [9–14], and the definitions of privacy are various. We notice that some works [9–11] claimed that the biometric is stored in plaintext at server's database, and the privacy concern is the relationship between a user's biometric and the user's real identity (or pseudonym). However, the three-factor authentication (such as smart card, password and biometrics) formed an opposite research direction [15,16]. The proposed three-factor solutions provide an enhanced security to user authentication, because it takes biometrics as an additional secret for user authentication. One three-factor authentication was done by Fan and Lin [15], in which an efficient three-factor authentication with privacy protection on biometrics was proposed and formally proven in Bellare and Rogaway's model [17]. Specifically, the biometrics matching is performed by an authentication server who is not able to access user's plain biometrics.

Table 1
A comparative summary of biometric-based user authentication. Biometric-privacy means no biometrics is stored in plaintext at the server side. † denotes the number of factors for authentication. Lightweight Cryptography means the construction does not rely on homomorphic cryptography (e.g., Paillier encryption [21]).

Function/Scheme	[2]	[34]	[19]	[14]	[11]	[15]	[4]	[35]	Ours
Biometric-Privacy	✓	✓	✓	✓	×	✓	✓	✓	✓
Leakage-Resilient	×	×	×	×	×	×	×	✓	✓
†-factor Auth	One	Two	One	One	Two	Three	One	Two	One
Lightweight-crypto	✓	✓	✓	×	×	×	✓	✓	✓

Biometric-based authentication/identification with biometric-privacy is extensively studied in the literature [14,18,19,4,35]. For example, some well-known works [20,12–14] used the Paillier cryptosystem [21] as encryption primitive to protect user's biometrics, and the biometrics matching is executed in the encrypted domain. Huang et al. [14] proposed a flexible biometric-based identification framework using Paillier encryption. They used garbled circuit [22] to efficiently and obliviously perform biometrics matching and retrieve the outcome of an authentication. Wang et al. [19] used invertible matrices as secret key to encrypt biometrics and the exact biometrics matching is executed in the transformed domain. However, authenticated users need to store those invertible matrices as secret keys, and these secret keys must be available at the time of authentication to transform the requested biometrics for user authentication. As we have discussed earlier, this is not suitable for a cross-platform setting.

Modelling leakage attacks Secret biometrics used in biometric-based user authentication is subject to leakage attacks. Micali and Reyzin [23] firstly introduced a leakage-resilient cryptography model to capture various side channel attacks. Specifically, an adversary is allowed to access a leakage oracle: adversary can query a polynomial time computable function f, and receive the output of $f(x)$, where x is a user's secret value. They also put some restrictions on $f(x)$ such that the adversary is not able to recover the secret key x completely through the chosen function f, and the amount of leakage $f(x)$ must be less than $|x|$. Later on, Naor and Segev [24] relaxed the restriction on $f(x)$, and stated that the lower bound of leaked bits is confined to the minimal entropy of secret key x, namely, "noisy leakage" model.

Dodis et al. [25] proposed a general model: "auxiliary inputs". Instead of a min-entropy requirement on the secret key x, it only requires the chosen leakage functions to be computationally hard to compute x given $f(x)$. The adversary is allowed to obtain the leakage bits larger than the upper bound defined in the bounded/noisy leakage models, and the chosen functions f must be "hard-to-invert". We note that leakage-resilient cryptography has been extensively studied in the auxiliary inputs model [25–28]. However, any leakage-resilient work did not address leakage attacks on the biometrics used in user authentications. We stress that leakage attacks on secret biometrics need to be investigated, as the biometrics could affect the success of user authentication.

Fuzzy extractor Fuzzy extractor (FE) is one of the promising approaches to construct a biometric-based (remote) user authentication scheme. Juels and Wattenberg [29] introduced a type of cryptography primitive called "fuzzy commitment scheme". It is particularly useful for biometric-based authentication systems, because its error-correcting technique can correct certain errors within a suitable metric (Hamming distance). Juels and Sudan [30] proposed another construction: "fuzzy vault scheme". It is based on a Set distance rather than the Hamming distance used in [29]. Specifically, the fuzzy vault scheme randomly creates a secret k-degree polynomial $p(x)$ during the sketch generation procedure. Given valid biometric information, a user can reproduce the polynomial and recover x.

Dodis et al. [31] formally introduced the notion of secure sketches and fuzzy extractors. They used biometrics to derive a cryptographic key for various cryptographic applications such as password-based authentication. They also provided concrete constructions of secure sketches and fuzzy extractors in three metrics: Hamming distance, Set difference and Edit distance. Li et al. [4] proposed an efficient fuzzy-extractor-based biometric identification protocol using a new fuzzy extractor. The new fuzzy extractor is Chebyshev distance based, which takes a real number string as input. Nevertheless, it is less error-tolerant than Hamming/Edit distance.

With regard to specific attacks on FE, Boyen et al. [3] introduced a notion called "robust sketches". They also provided a generic conversion to tackle active attacks such that adversary modifies the public sketch (or helper data) and compromises the security of fuzzy extractors. Meanwhile, Boyen [2] presented another notion, namely, "reusable fuzzy extractor". It states that user may produce multiple secret string and public sketch pairs using the same biometrics: $(R_i, P_i) \leftarrow \mathsf{Gen}(Bio)$ (see Section 3.1). Later, Canetti et al. [32] proposed the first reusable (and robust) fuzzy extractor from some low-entropy distributions. In particular, their refined security of "reusable fuzzy extractor" states that secret string R_i remains secure even if all other secret strings R_j $(j \neq i)$ are revealed. In Table 1, we compare our proposed solution with the existing closely-related works. It shows that our proposed solution is the first biometric-based remote user authentication with leakage-resilience and biometrics-privacy in a single-factor setting. We stress that the concern of false acceptance or false rejection is not the main focus in this work. This is because false acceptance/rejection is likely to occur due to various reasons. The accuracy of biometric extraction can significantly influence the authentication result (e.g., face recognition algorithms find difficulties in distinguishing twins). We note that these issues can be resolved by using multiple types of biometrics, such as finger-

Table 2
Summary of notations.

Notation	Definition
$(\mathtt{pk}_i, \mathtt{sk}_i)$	User i' public key/secret key
$(ID_i, ID_{\widehat{S}})$	Identity of user i/server $\widehat{S}$
$\mathsf{dist}(x, y)$	Distance between vector x and vector y
$t \in \mathbb{R}^+$	Threshold value (positive real number)
$\mathcal{B}/\mathcal{C}$	Biometrics/Encrypted biometrics
$\mathsf{SS}(x, r)$	Secure sketch
(R, P)	Secret/Public string pair
$\mathsf{Ext}(x, r)$	Strong extractor
$\mathtt{H}$	Collision-resistant hash function

print and iris [33]. For similar reasons, the existing works mentioned in this paper also do not take into account false acceptance/rejection for biometric-based authentication.

2. Security model

In this section, we present the system model and security model for biometric-based remote user authentication framework.

Notation We denote the i-th session established by a user as Π_U^i, and identities of all the users recognized by Π_U^i during the execution of that session by partner identifier pid_U^i. We define sid_U^i as the unique session identifier belonging to the session i established by a user U. Specifically, $\mathsf{sid}_U^i = \{m_j\}_{j=1}^n$, where $m_j \in \{0,1\}^*$ is the message transcript exchanged between users.

We say an oracle Π_U^i may be used or unused. The oracle is considered as unused if it has never been initialized. The oracle is initialized as soon as it becomes part of a group. After the initialization, the oracle is marked as used and turns into the stand-by state where it waits for an invocation to execute a protocol operation. Upon receiving such invocation the oracle Π_U^i learns its partner identifier pid_U^i and turns into a processing state where it sends, receives and processes messages according to the description of the protocol. During that phase, the internal state information $state_U^i$ is maintained by the oracle. The oracle Π_U^i remains in the processing state until it collects enough information to finalize the user authentication. As soon as the authentication is accomplished, Π_U^i accepts and terminates the protocol execution meaning that it would not send or receive further messages. If the protocol execution fails then Π_U^i terminates without being accepted. In addition, we present the commonly used notations (see Table 2) in this paper.

2.1. System model

A biometric-based remote user authentication framework consists of the following algorithms:

- Enrollment. This is a non-interactive protocol between a user and an authentication server over a secure channel. The user enrolls her identity ID, a reference biometric $\mathcal{C}$ ($\mathcal{C} \leftarrow \mathsf{SS}(\mathcal{B}, r)$), and a public key $\mathtt{pk}$ to the authentication server. The randomness r is chosen at random by the enrolled user, and it is erased after enrollment. The enrolled user becomes authorized one after the enrollment.
- Authentication. This is an interactive protocol between an authorized user and an authentication server over a public channel. The user sends her identity ID and a certificate associated to a candidate biometric $\mathcal{B}'$ (i.e., $\mathsf{dist}(\mathcal{B}', \mathcal{B}) \leq t$) to the authentication server. The authentication server accepts it if and only if the certificate is verified as valid under the enrolled public key $\mathtt{pk}$ corresponding to ID.

2.2. Security model

We review the security model defined in [35], including user authenticity and biometric-privacy with leakage-resilience against auxiliary input models, which were used to capture an impersonator and an honest-but-curious server, respectively, with the help of leakage attacks such as side channel attacks.

2.2.1. Authenticity

The authenticity game between a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ and a simulator (i.e., challenger) is defined $\mathcal{S}$ as follows.

- Setup. $\mathcal{S}$ first generates user identities $\{ID_i\}$ ($i \in [1, n]$) and server identities $\{ID_{\widehat{S}_j}\}$ ($j \in [1, m]$) in the system. $\mathcal{S}$ also generates biometrics information $\{\mathcal{B}_i\}$ for n users. In particular, $\mathcal{S}$ generates a set of secret/public key pairs $\{\mathtt{sk}_i^j, \mathtt{pk}_i^j\}_{j=1}^m$ for user i with respect to m servers. Eventually, $\mathcal{S}$ sends all enrolled users' identities $\{ID_i, ID_{\widehat{S}_j}\}$ to $\mathcal{A}$.

- 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 (U, i, m) to simulate a network message for the i-th session of user U, then $\mathcal{S}$ would simulate the reaction of instance oracle Π_U^i upon receiving message m, and return to $\mathcal{A}$ the response that Π_U^i would generate; If $\mathcal{A}$ issues a send query in the form of $(U'$, '$start$'), then $\mathcal{S}$ creates a new instance oracle $\Pi_{U'}^i$ and returns to $\mathcal{A}$ the first protocol message.
  - Biometrics Reveal: If $\mathcal{A}$ issues a biometrics reveal (or corrupt, for short) query to user i, then $\mathcal{S}$ returns user i's biometric information $\mathcal{B}_i$ to $\mathcal{A}$.
  - Secret Key Reveal: If $\mathcal{A}$ issues a secret key reveal query to user i with respect to server j, then $\mathcal{S}$ returns the secret key $\mathtt{sk}_i^j$ to $\mathcal{A}$.
- Attack. $\mathcal{A}$ wins the game if all of the following conditions hold.
  1. $\widehat{S}_j$ accepts user i $(j \neq i)$; It implies $\mathsf{pid}_{\widehat{S}_j}^s$ and $\mathsf{sid}_{\widehat{S}_j}^s$ exist.
  2. $\mathcal{A}$ did not issue Biometrics Reveal query to user i;
  3. $\mathcal{A}$ did not issue Secret Key Reveal query to user i with respect to $\widehat{S}_j$;
  4. $m \in \mathsf{sid}_{\widehat{S}_j}^s$, but there exists no $\Pi_{U_i}^s$ which has sent m (m denotes the message transcript from user i).

  $\mathcal{A}$ is allowed to reveal user i's secret keys $\{\mathtt{sk}_i^w\}$ associated to w $(w \neq j)$ servers. We define the advantage of an adversary $\mathcal{A}$ in the above game as

$$\mathtt{Adv}_{\mathcal{A}}^{\mathsf{BRUA}}(\lambda) = |\Pr[\mathcal{A}\ wins]|.$$

Definition 2.1. *We say a BRUA scheme has* authenticity *if for any PPT $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A}}^{BRUA}(\lambda)$ is a* negligible *function of the security parameter λ.*

2.2.2. Authenticity against auxiliary inputs

To model the leakage on biometrics and secret key against auxiliary inputs, we first define two classes of auxiliary input leakage functions below.

- Let $\mathcal{H}_{ow}(\epsilon_{bio})$ be the class of all the polynomial-time computable functions $h : \{0,1\}^{|bio|} \to \{0,1\}^*$ such that given $h(bio)$ (for a randomly generated biometric information bio), no PPT adversary can find bio with probability $\geq \epsilon_{bio}$. The function $h(bio)$ can be viewed as a composition of $q_{bio} \in \mathbb{N}^+$ functions, i.e., $h(bio) = (h_1(bio), \cdots, h_{q_{bio}}(bio))$ where for all $i \in \{1, \cdots, q_{bio}\}, h_i \in \mathcal{H}_{ow}(bio)$.
- Let $\mathcal{H}_{ow}(\epsilon_{sk})$ be the class of all the polynomial-time computable functions $h : \{0,1\}^{|sk|} \to \{0,1\}^*$ such that given $h(sk)$ (for a randomly generated secret key sk), no PPT adversary can find sk with probability $\geq \epsilon_{sk}$. The function $h(sk)$ can be viewed as a composition of $q_{sk} \in \mathbb{N}^+$ functions, i.e., $h(sk) = (h_1(sk), \cdots, h_{q_{sk}}(sk))$ where for all $i \in \{1, \cdots, q_{sk}\}, h_i \in \mathcal{H}_{ow}(sk)$.

We then present the leakage-resilient biometric-based user authenticity model (LR-BRUA), which is an extension of previous authenticity model. Specifically, we provide two leakage queries for $\mathcal{A}$ in the LR-BRUA model.

- Biometric Leakage: If $\mathcal{A}$ issues a biometric leakage query to user i (i.e., $\mathcal{O}_{bio}(i)$), then $\mathcal{S}$ returns $f_{Bio}(\mathcal{B}_i)$ to $\mathcal{A}$, where $f_{Bio} \in \mathcal{H}_{ow}(\epsilon_{bio})$, and $\mathcal{B}_i$ denotes the biometric information of user i.
- Secret Key Leakage: If $\mathcal{A}$ issues a secret key leakage query to user i (i.e., $\mathcal{O}_{sk}(i)$), then $\mathcal{S}$ returns $f_{Sk}(sk_i)$ to $\mathcal{A}$, where $f_{Sk} \in \mathcal{H}_{ow}(\epsilon_{sk})$, and sk_i denotes the secret key of user i.

In the proposed leakage-resilient biometric-based user authenticity model, we let the adversary submit two leakage function sets $\mathcal{F}_{Bio} \subseteq \mathcal{H}_{ow}(\epsilon_{bio}), \mathcal{F}_{Sk} \subseteq \mathcal{H}_{ow}(\epsilon_{sk})$, where both $\mathcal{F}_{Bio}$ and $\mathcal{F}_{Sk}$ are polynomials in the security parameter λ, prior to Setup stage. During the LR-BRUA security game, $\mathcal{A}$ is allowed to access both biometrics leakage oracle f_{Bio} and secret key leakage oracle f_{Sk} adaptively. We require that $f_{Bio} \in \mathcal{F}_{Bio}$, $f_{Sk} \in \mathcal{F}_{Sk}$ and we define the advantage of an adversary $\mathcal{A}$ in the LR-BRUA game as

$$\mathtt{Adv}_{\mathcal{A}}^{\mathsf{LR\text{-}BRUA}}(\lambda) = |\Pr[\mathcal{A}\ wins]|.$$

Definition 2.2. *We say a BRUA scheme has* leakage-resilient authenticity *if for any PPT $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A}}^{LR\text{-}BRUA}(\lambda)$ is a* negligible *function of the security parameter λ.*

2.2.3. Biometric-privacy

The biometric-privacy game between an adversary $\mathcal{A}$ and a simulator $\mathcal{S}$ is defined as follows.

- Setup: $\mathcal{S}$ first generates user identities $\{ID_i\}$ $(i \in [1, n])$ and server identities $\{ID_{\widehat{S}_j}\}$ $(j \in [1, m])$ in the system. $\mathcal{S}$ also generates biometrics information $\{\mathcal{B}_i\}$ for n users. In particular, $\mathcal{S}$ generates a set of secret/public key pairs

$\{\mathtt{sk}_i^j, \mathtt{pk}_i^j\}_{j=1}^m$ for individual user i with respect to m servers. Eventually, $\mathcal{S}$ sends all enrolled users' identities $\{ID_i, ID_{\widehat{S}_j}\}$ and user's secret keys $\mathtt{sk}_i^j$ to $\mathcal{A}$.

- Training: $\mathcal{A}$ is allowed to issue Send queries and Biometrics Reveal queries to $\mathcal{S}$ with arbitrary sequence.
- Challenge: $\mathcal{A}$ randomly chooses two challenge biometrics $(\mathcal{B}_0, \mathcal{B}_1)$ of a user ID_i (possibly corrupted), with the condition that $\mathsf{dist}(\mathcal{B}_0, \mathcal{B}_1) \leq t$, and sends them to $\mathcal{S}$. $\mathcal{S}$ simulates the reference biometrics of user ID_i by either $\mathcal{C}_b^* \leftarrow \mathsf{F}(\mathcal{B}_0)$ if $b = 0$ or $\mathcal{C}_b^* \leftarrow \mathsf{F}(\mathcal{B}_1)$ if $b = 1$.

  F denotes a (public) probabilistic algorithm, and $\mathcal{A}$ is not allowed to issue Biometrics Reveal query on reference biometrics $\mathcal{C}_b^*$. 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

$$\mathtt{Adv}_{\mathcal{A}}^{\mathsf{BRUA}}(\lambda) = |\Pr[\mathcal{S} \to 1] - 1/2|.$$

Definition 2.3. *We say a BRUA scheme has* biometric-privacy *if for any PPT $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A}}^{BRUA}(\lambda)$ is a* negligible *function of the security parameter λ.*

2.2.4. Biometric-privacy against auxiliary inputs

If the adversary $\mathcal{A}$ reveals the randomness r used in the reference biometrics, then she can encrypt the two challenge biometrics $\mathcal{B}_0$ and $\mathcal{B}_1$ by herself using r and compare whether they are equal to the challenge ciphertext, thus win the biometric-privacy game trivially. To model the leakage on randomness r against post-challenge auxiliary inputs, we define a class of auxiliary input leakage functions and a leakage query for $\mathcal{A}$ respectively.

- Let $\mathcal{H}_{ow}(\epsilon_r)$ be the class of all the polynomial-time computable functions $h : \{0,1\}^{|r|} \to \{0,1\}^*$ such that given $h(r)$ (for a randomly generated randomness r), no PPT adversary can find r with probability $\geq \epsilon_r$. The function $h(r)$ can be viewed as a composition of $q_r \in \mathbb{N}^+$ functions, i.e., $h(r) = (h_1(r), \cdots, h_{q_r}(r))$ where for all $i \in \{1, \cdots, q_r\}, h_i \in \mathcal{H}_{ow}(r)$.
- Randomness Leakage: If $\mathcal{A}$ issues a randomness leakage query to user i (i.e., $\mathcal{O}_r(i)$), then $\mathcal{S}$ returns $f_r(r_i)$ to $\mathcal{A}$, where $f_r \in \mathcal{H}_{ow}(\epsilon_r)$, and r_i denotes the randomness of user i.

In the biometric-privacy game against post-challenge auxiliary inputs model, $\mathcal{A}$ is additionally allowed to access user's Randomness Leakage oracle w.r.t. the randomness used in reference biometrics $\mathcal{C}_b^*$, and we define the advantage of an adversary $\mathcal{A}$ in the biometric-privacy game as

$$\mathtt{Adv}_{\mathcal{A}}^{\mathsf{LR\text{-}BRUA}}(\lambda) = |\Pr[\mathcal{S} \to 1] - 1/2|.$$

Definition 2.4. *We say a BRUA scheme has* leakage-resilient biometric-privacy *if for any PPT $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A}}^{LR\text{-}BRUA}(\lambda)$ is a* negligible *function of the security parameter λ.*

3. Preliminaries

In this section, we review the preliminaries and the building blocks that will be used in the proposed LR-BRUA generic framework.

3.1. Fuzzy extractor

A fuzzy extractor converts non-uniform data to uniformly random strings which can be used in cryptographic applications. A typical application of fuzzy extractor is to extract reproducible string from biometric information. The extracted string then is considered as a secret for user authentication.

Secure sketches and fuzzy extractors Secure sketch is a building block of fuzzy extractors. A secure sketch scheme takes noisy information x as input, outputs a sketch s which is an auxiliary string. Note that secure sketches and fuzzy extractors are applicable to various noisy data other than biometric information. Secure sketch schemes normally use error correcting techniques to recover x under s if and only if the given input x' is statistically close to x. The sketch s can be published since it does not reveal much information about x. Let $\mathcal{M}$ be a metric space on N points with distance function $\mathsf{dist} : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+ = [0, \infty)$, where N$= |\mathcal{M}|$.

Definition 3.1. *A secure sketch consists of two randomized procedures* (SS, Rec) *with the following properties.*

- *The sketch* SS *takes $x \in \mathcal{M}$ as input, and outputs a sketch $s \in \{0,1\}^*$.*
- *The function* Rec *takes an element $x' \in \mathcal{M}$ and a sketch $s \in \{0,1\}^*$ as input, and outputs x if* $\mathsf{dist}(x, x') \leq t$.

Fuzzy extractors extract some randomness from a noisy input $x \in \mathcal{M}$. Then, it can also be recovered from a given input x' if x and x' are statistically close. The difference is that fuzzy extractors return a uniform string, but secure sketch returns a non-uniform string.

Definition 3.2. *A fuzzy extractor consists of two randomized procedures* $(\mathsf{Gen}, \mathsf{Rep})$ *with the following properties.*

- *The generation function* Gen *takes* $x \in \mathcal{M}$ *as input, and outputs a string* $R \in \{0,1\}^{\ell}$ *and helper data* $P \in \{0,1\}^{*}$ *such that*

$$(R, P) \leftarrow \mathsf{Gen}(x).$$

- *The reproduction procedure* Rep *takes an element* $x' \in \mathcal{M}$ *and helper data* $P \in \{0,1\}^{*}$ *as input, and outputs* R *such that*

$$R \leftarrow \mathsf{Rep}(x', P) \text{ iff } \mathsf{dist}(x, x') \leq t.$$

Since secure sketch can reconstruct the original input from some given noisy data, it can be used to construct fuzzy extractor schemes. Generally speaking, a fuzzy extractor can be derived by using a secure sketch with a strong randomness extractor. We now review a generic fuzzy extractor construction from a secure sketch.

- Gen*: Let* SS *be a secure sketch and* Ext *be a strong extractor. Given an input* x*,* $(P, R) \leftarrow \mathsf{Gen}(x; r_1, r_2)$ *such that*

$$P \leftarrow (\mathsf{SS}(x; r_1), r_2), \ \ R \leftarrow \mathsf{Ext}(x; r_2).$$

  Note that r_1 *and* r_2 *are secret and public randomness respectively.*
- Rep*: Given an noisy input* x' *and* P*, recover the original input* $x \leftarrow \mathsf{Rec}(x', \mathsf{SS}(x; r_1))$*, then compute* $R \leftarrow \mathsf{Ext}(x; r_2)$*.*

3.2. Strong extractor with hard-to-invert auxiliary inputs

Definition 3.3. (δ, ϵ)***-Strong Extractor (Ext) with Hard-to-Invert Auxiliary Inputs [27]****. Let* $\mathsf{Ext} : \{0,1\}^{l_1} \times \{0,1\}^{l_2} \to \{0,1\}^{m}$*, where* l_1, l_2, m *are polynomials in the security parameter* λ*.* Ext *is said to be a strong extractor with* ϵ*-Hard-to-Invert auxiliary inputs, if for all pairs* (x, f) *such that* $x \in \{0,1\}^{l_1}$ *and* $f \in \mathcal{H}_{ow}(\epsilon)$ *(see Section 2.2.2), we have*

$$|\Pr[\mathcal{A}(r_2, f(x), \mathsf{Ext}(x, r_2)) = 1]| - |\Pr[\mathcal{A}(r_2, f(x), u) = 1]| < \delta,$$

where $r_2 \in_R \{0,1\}^{l_2}, u \in_R \{0,1\}^{m}$*.*

The leakage function $f \in \mathcal{H}_{ow}(\epsilon)$ can be interpreted as a composition of q functions $f_1, f_2, \cdots, f_q$, where $q \in \mathbb{N}^{+}$ and $f_i \in \mathcal{H}_{ow}(\epsilon)$. And the following Lemma is obtained from [27].

Lemma 3.1. *Let* $r_2 \in_R \{0,1\}^{l_2}$ *be chosen uniformly at random. For any pair* (x, f) *where* $x \in \{0,1\}^{l_1}$ *and* $f \in \mathcal{H}_{ow}(\epsilon)$*, no PPT adversary can recover* x *with probability* $\geq 2 \cdot \delta$ *given* $(r_2, f, \mathsf{Ext}(x, r_2))$*, provided that* $\mathsf{Ext}(x, r_2)$ *is a strong extractor with* ϵ*-hard-to-invert auxiliary inputs.*

3.3. Generic fuzzy extractor with hard-to-invert auxiliary inputs

Definition 3.4. *A generic fuzzy extractor with* ϵ*-hard-to-invert auxiliary inputs consists of two randomized procedures* $(\mathsf{Gen}, \mathsf{Rep})$ *with the following properties.*

- Gen*: Let* SS *be a (reusable) secure sketch and* Ext *be a strong extractor with* ϵ*-hard-to-invert auxiliary inputs. Given an input* x*,* $(P, R) \leftarrow \mathsf{Gen}(x; r_1, r_2)$ *such that*

$$P \leftarrow (\mathsf{SS}(x; r_1), r_2), \ \ R \leftarrow \mathsf{Ext}(x; r_2).$$

- Rep*: Given an noisy input* x' *and* P*, recover the original input* $x \leftarrow \mathsf{Rec}(x', \mathsf{SS}(x; r_1))$*, then compute* $R \leftarrow \mathsf{Ext}(x; r_2)$*.*

The generic fuzzy extractor with hard-to-invert auxiliary inputs is derived from reusable secure sketch and (δ, ϵ)-strong extractor with hard-to-invert auxiliary inputs. We refer to [35] for the following theorem and its security analysis.

Theorem 3.2. *The generic fuzzy extractor with* ϵ*-hard-to-invert auxiliary inputs is information theoretically secure if the (reusable) secure sketch is secure and the* (δ, ϵ)*-strong extractor with hard-to-invert auxiliary inputs is secure.*

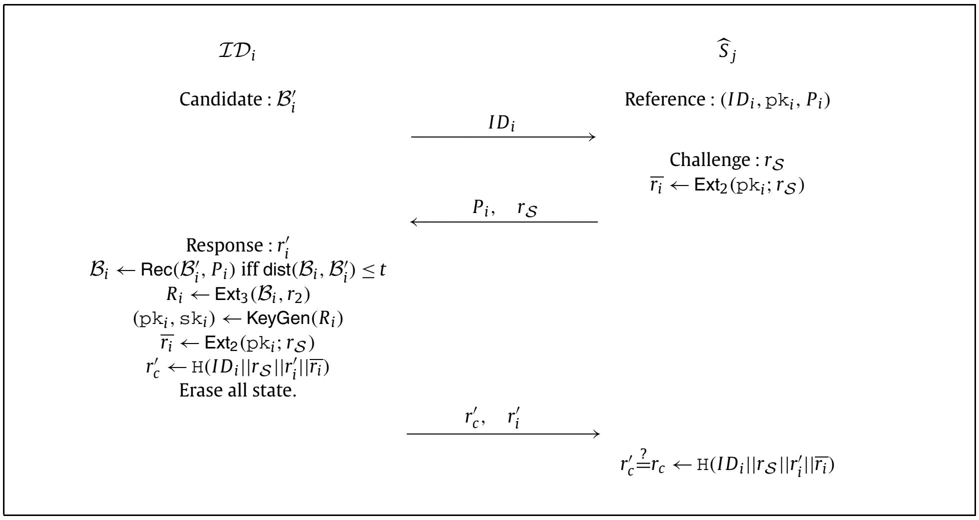
$\mathcal{ID}_i$
$\widehat{S}_j$
Candidate : $\mathcal{B}'_i$
Reference : $(ID_i, \mathtt{pk}_i, P_i)$
ID_i
Challenge : $r_{\mathcal{S}}$
$\overline{r_i} \leftarrow \mathsf{Ext}_2(\mathtt{pk}_i; r_{\mathcal{S}})$
$P_i, \quad r_{\mathcal{S}}$
Response : r'_i
$\mathcal{B}_i \leftarrow \mathsf{Rec}(\mathcal{B}'_i, P_i)$ iff $\mathsf{dist}(\mathcal{B}_i, \mathcal{B}'_i) \le t$
$R_i \leftarrow \mathsf{Ext}_3(\mathcal{B}_i, r_2)$
$(\mathtt{pk}_i, \mathtt{sk}_i) \leftarrow \mathsf{KeyGen}(R_i)$
$\overline{r_i} \leftarrow \mathsf{Ext}_2(\mathtt{pk}_i; r_{\mathcal{S}})$
$r'_c \leftarrow \mathtt{H}(ID_i || r_{\mathcal{S}} || r'_i || \overline{r_i})$
Erase all state.
$r'_c, \quad r'_i$
$r'_c \overset{?}{=} r_c \leftarrow \mathtt{H}(ID_i || r_{\mathcal{S}} || r'_i || \overline{r_i})$

Fig. 2. Authentication (public channel).

4. Proposed construction

Overview On a high level, a user extracts a secret/public string pair (R, P) (P is also called helper data) using the Gen algorithm of generic fuzzy extractor on a biometrics $\mathcal{B}$, and derives a public key $\mathtt{pk}$ using a key generation algorithm. Then, the user sends $(ID, \mathtt{pk}, P)$ to an authentication server for enrollment. During user authentication, upon receiving a helper data P from the authentication server, the authorized user obtains the original biometrics $\mathcal{B}$ iff $\mathsf{dist}(\mathcal{B}', \mathcal{B}) \le t$ by running the Rec algorithm of generic fuzzy extractor, which can be used to perform the subsequent user authentications. We stress that the public key $\mathtt{pk}$ can be regarded as the key shared between a user and a server in order to ensure the authenticity of the user. We define two strong extractors with ϵ_1-hard-to-invert auxiliary inputs $\mathsf{Ext}_1 : \{0,1\}^{l_1(\lambda)} \times \{0,1\}^{l_2(\lambda)} \to \{0,1\}^{m_1(\lambda)}$ and ϵ_3-hard-to-invert auxiliary inputs $\mathsf{Ext}_2 : \{0,1\}^{l'_1(\lambda)} \times \{0,1\}^{l'_2(\lambda)} \to \{0,1\}^{m_2(\lambda)}$ respectively, and a generic fuzzy extractor with ϵ_3-hard-to-invert auxiliary inputs ($\mathsf{Ext}_3 : \{0,1\}^{l''_1(\lambda)} \times \{0,1\}^{l_2(\lambda)} \to \{0,1\}^{m_3(\lambda)}$) in the system. Let $\mathtt{H} : \{0,1\}^* \to \mathbb{Z}_q$ be a collision-resistant hash function.

- $\mathsf{Enrollment}$. An enrolled user i performs the following steps.
  1. Generate a biometric information $\mathcal{B}_i \leftarrow \{0,1\}^{l''_1(\lambda)}$ and a secret/public randomness pair $(r'_1, r_2) \leftarrow \{0,1\}^{l_1(\lambda)} \times \{0,1\}^{l_2(\lambda)}$;
  2. Compute the secret randomness $r_1 \leftarrow \mathsf{Ext}_1(r'_1; r_2)$;
  3. Run the generic fuzzy extractor with ϵ_3-hard-to-invert auxiliary inputs to obtain $(P_i, R_i) \leftarrow \mathsf{Gen}(\mathcal{B}_i; r_1, r_2)$, where $P_i \leftarrow (\mathsf{SS}(\mathcal{B}_i; r_1), r_2)$ and $R_i \leftarrow \mathsf{Ext}_3(\mathcal{B}_i; r_2)$;
  4. Generate a public/secret key pair $(\mathtt{pk}_i, \mathtt{sk}_i)$ by running the $\mathsf{KeyGen}(R_i)$ algorithm, which takes the secret string R_i as input;
  5. Send $(ID_i, \mathtt{pk}_i, P_i)$ to an authentication server $\widehat{S}_j$.

  The user i derives a public key $\mathtt{pk}_i \in \{0,1\}^{l'_1(\lambda)}$ from the secret string R_i (step 4). Accordingly, the authentication server $\widehat{S}_j$ takes public key $\mathtt{pk}_i$ as a shared secret key with user i.
- $\mathsf{Authentication}$. The interaction between an authorized user i and an authentication server $\widehat{S}_j$ takes place as follows (see Fig. 2).
  - Upon receiving a request ID_i from user i, the authentication server $\widehat{S}_j$ chooses a challenge nonce $r_{\mathcal{S}} \leftarrow \{0,1\}^{l'_2(\lambda)}$ first, then computes a value $\overline{r_i} \leftarrow \mathsf{Ext}_2(\mathtt{pk}_i; r_{\mathcal{S}})$ and sends $(P_i, r_{\mathcal{S}})$ to user i.
  - Then user i performs the following steps.
    1. Generate a candidate biometric information $\mathcal{B}'_i \leftarrow \{0,1\}^{l''_1(\lambda)}$;
    2. Run the generic fuzzy extractor with ϵ_3-hard-to-invert auxiliary inputs to obtain $\mathcal{B}_i \leftarrow \mathsf{Rec}(\mathcal{B}'_i, P_i)$ ($P_i \leftarrow (\mathsf{SS}(\mathcal{B}_i; r_1), r_2)$ if and only if $\mathsf{dist}(\mathcal{B}_i, \mathcal{B}'_i) \le t$ and $R_i \leftarrow \mathsf{Ext}_3(\mathcal{B}_i; r_2)$;
    3. Compute the public/secret key pair $(\mathtt{pk}_i, \mathtt{sk}_i) \leftarrow \mathsf{KeyGen}(R_i)$;
    4. Compute the value $\overline{r_i} \leftarrow \mathsf{Ext}_2(\mathtt{pk}_i; r_{\mathcal{S}})$;
    5. Choose a response nonce $r'_i \leftarrow \mathbb{Z}_q$ and compute a certificate $r'_c \leftarrow \mathtt{H}(ID_i || r_{\mathcal{S}} || r'_i || \overline{r_i})$;
    6. Erase all state and send (r'_c, r'_i) to $\widehat{S}$.
  - $\widehat{S}_j$ computes a certificate $r_c \leftarrow \mathtt{H}(ID_i || r_{\mathcal{S}} || r'_i || \overline{r_i})$ and checks $r'_c \overset{?}{=} r_c$. If it does hold, $\widehat{S}_j$ accepts user i; otherwise, $\widehat{S}_j$ rejects her.

4.1. Security analysis

Theorem 4.1. *The proposed LR-BRUA achieves leakage-resilient authenticity (Definition 2.2) in the random oracle model if ϵ_2-hard-to-invert auxiliary inputs is secure, and the generic fuzzy extractor with ϵ_3-hard-to-invert auxiliary inputs is secure, where ϵ_2 and ϵ_3 are negligible.*

High-level discussion We clarify the motivation of each game before detailed proof. Game $\mathbb{G}_1$ is to prevent replay attacks; Game $\mathbb{G}_2$ is to capture an adversary, who is not allowed to reveal the biometrics of user i, aims to impersonate user i and authenticate to a server; Game $\mathbb{G}_3$ is to capture an adversary, who is not allowed to reveal the secret key of user i w.r.t. server $\widehat{S}_j$, aims to impersonate user i and authenticate to server $\widehat{S}_j$.

Proof. We define a sequence of games $\{\mathbb{G}_i\}$ and let $\mathtt{Adv}_i^{\text{LR-BRUA}}$ denote the advantage of the adversary in game $\mathbb{G}_i$. Assume that $\mathcal{A}$ activates at most m sessions in each game.

- $\mathbb{G}_0$: This is the original game for leakage-resilient authenticity security.
- $\mathbb{G}_1$: This game is identical to game $\mathbb{G}_0$ except that $\mathcal{S}$ will $\mathtt{abort}$ if a challenge/response nonce (i.e., $r_{\mathcal{S}}, r_i'$) is used twice by the server/user in two different sessions. Therefore, we have

$$\left|\mathtt{Adv}_0^{\text{LR-BRUA}} - \mathtt{Adv}_1^{\text{LR-BRUA}}\right| \leq m^2/2^{\lambda} \tag{1}$$

- $\mathbb{G}_2$: This game is identical to game $\mathbb{G}_1$ except that in the Attack session, $\mathcal{S}$ replaces the real value R_i by a random value $R \in \{0,1\}^{m_3(\lambda)}$ with regard to instance oracle $\Pi_{U_i}^i$. Below we show the difference between $\mathbb{G}_1$ and $\mathbb{G}_2$ is negligible under the assumption that the generic fuzzy extractor with ϵ_3-hard-to-invert auxiliary inputs is secure.
  Let $\mathcal{S}$ denote an adversary, who is given $(r, f_1(\mathcal{B}_i), \cdots, f_{q_{Bio}}(\mathcal{B}_i), \mathsf{SS}(\mathcal{B}_i; r_1), T_b)$, aims to break the generic fuzzy extractor with ϵ_3-hard-to-invert auxiliary inputs. $\mathcal{S}$ simulates the game for $\mathcal{A}$ as follows.
  - Setup. $\mathcal{S}$ sets up the game for $\mathcal{A}$ by creating n users and m servers with the corresponding identities $\{ID_i, ID_j\}$, where $i \in [1,n], j \in [1,m]$. $\mathcal{S}$ randomly selects an index i and guesses that if the "Attack" event will happen with regard to user i at server j. $\mathcal{S}$ sets the challenge reference biometrics as $(ID_i, \mathtt{pk}_i, P_i) = [ID_i, \mathsf{KeyGen}(T_b), (\mathsf{SS}(\mathcal{B}_i; r_1), r)]$ (where (r_1, r) are chosen by his challenger). In addition, $\mathcal{S}$ honestly generates rest user's biometrics and their corresponding reference $\{(\mathtt{pk}_l, P_l)\}$ $(l \neq i)$. Eventually, $\mathcal{S}$ sends all the references (include $(ID_i, \mathtt{pk}_i, P_i)$) to $\mathcal{A}$. It is obvious that $\mathcal{S}$ can answer all queries made by $\mathcal{A}$ except user i. Below we mainly focus on the simulation of user i only. Note that T_b can be either $T_0 = \mathsf{Ext}_3(\mathcal{B}_i; r)$ or $T_1 \in_R \{0,1\}^{m_3(\lambda)}$, and secret key pair is defined as $(\mathtt{pk}_i, \mathtt{sk}_i) \leftarrow \mathsf{KeyGen}(T_b)$ with respect to server j.
  - Training. $\mathcal{S}$ answers $\mathcal{A}$'s queries as follows.
    - ∗ If $\mathcal{A}$ issues a send query in the form of $(P_i, r_{\mathcal{S}})$ to $\mathcal{S}$, then $\mathcal{S}$ firstly chooses a response nonce r_i'; $\mathcal{S}$ then computes $\overline{r_i} = \mathsf{Ext}_2(\mathtt{pk}_i; r_{\mathcal{S}})$ and a certificate $r_c' = \mathtt{H}(ID_i||r_{\mathcal{S}}||r_i'||\overline{r_i})$, and returns (r_c', r_i') to $\mathcal{A}$.
    - ∗ If $\mathcal{A}$ issues a biometric leakage query to user i, then $\mathcal{S}$ returns $f_1(\mathcal{B}_i), \cdots, f_{q_{Bio}}(\mathcal{B}_i)$ as the leakage query outputs.
    - ∗ If $\mathcal{A}$ issues a secret key query to an instance oracle $\Pi_{U_i}^i$ w.r.t. server j, then $\mathcal{S}$ returns $\mathtt{sk}_i$ to $\mathcal{A}$.

    If the challenge of $\mathcal{S}$ is $T_0 \leftarrow \mathsf{Ext}_3(\mathcal{B}; r)$, then the simulation is consistent with $\mathbb{G}_1$; otherwise, the simulation is consistent with $\mathbb{G}_2$. If the advantage of $\mathcal{A}$ is significantly different in $\mathbb{G}_1$ and $\mathbb{G}_2$, then $\mathcal{S}$ can break the generic fuzzy extractor with ϵ_3-hard-to-invert auxiliary inputs. We assume a user i uses biometrics $\mathcal{B}$ at most $n(\lambda)$ times for generating different references w.r.t. m servers, hence we have

$$\left|\mathtt{Adv}_1^{\text{LR-BRUA}} - \mathtt{Adv}_2^{\text{LR-BRUA}}\right| \leq n \cdot n(\lambda) \cdot \mathtt{Adv}_{\mathcal{S}}^{\text{Ext}_3}(\lambda)$$

- $\mathbb{G}_3$: This game is identical to game $\mathbb{G}_2$ except that in the "Attack" session, $\mathcal{S}$ replaces the real value $\overline{r_i}$ by a random value $R \in \{0,1\}^{m_2(\lambda)}$ with regard to instance oracle $\Pi_{U_i}^i$. Below we show the difference between $\mathbb{G}_2$ and $\mathbb{G}_3$ is negligible under the assumption that the strong extractor with ϵ_2-hard-to-invert auxiliary inputs is secure.
  Let $\mathcal{S}$ denote an adversary, who is given $(r, f_1(\mathtt{pk}^*), \cdots, f_{q_{sk}}(\mathtt{pk}^*), T_b)$, aims to break the strong extractor with ϵ_2-hard-to-invert auxiliary inputs. $\mathcal{S}$ simulates the game for $\mathcal{A}$ as follows.
  - Setup. $\mathcal{S}$ sets up the game for $\mathcal{A}$ by creating n users and m servers with the corresponding identities $\{ID_i, ID_j\}$, where $i \in [1,n], j \in [1,m]$. $\mathcal{S}$ randomly selects an index i and guesses that the "Attack" event will happen to user i with respect to a server j $(j \neq i)$. $\mathcal{S}$ sets the secret public key of user i at server j as $\mathtt{pk}^*$. In addition, $\mathcal{S}$ honestly generates rest user's biometrics and their corresponding references $\{(\mathtt{pk}_l, P_l)\}(l \neq i)$ according to the protocol specification. It is obvious that $\mathcal{S}$ can answer all the queries made by $\mathcal{A}$ except user i at server j. Below we mainly focus on the simulation of such case. Note that $\mathcal{S}$ generates independent public/secret key pairs to simulate user i's secret keys associates to other servers.
  - Training. $\mathcal{S}$ answers $\mathcal{A}$'s queries as follows.
    - ∗ If $\mathcal{A}$ issues a send query in the form of ID_i to $\mathcal{S}$ w.r.t. instance oracle $\Pi_{U_i}^i$, $\mathcal{S}$ firstly simulates a helper data as $P_i = (\{0,1\}^*, r)$ (where r is chosen by his challenger); $\mathcal{S}$ then chooses a challenge nonce $r_{\mathcal{S}}$ and returns $(P_i, r_{\mathcal{S}})$ to

$\mathcal{A}$ as the query response. Recall that $\mathcal{A}$ is not allowed to reveal biometrics $\mathcal{B}_i$ of user i, thus the simulation of such query is perfect.

If $\mathcal{A}$ issues a send query in the form of $(P_i, r_{\mathcal{S}})$ to $\mathcal{S}$, $\mathcal{S}$ firstly chooses a response nonce r'_i; Secondly, $\mathcal{S}$ sets $\overline{r_i} = T_b$ and computes a certificate $r'_c \leftarrow \mathtt{H}(ID_i||r_{\mathcal{S}}||r'_i||\overline{r_i})$; Eventually, $\mathcal{S}$ returns (r'_c, r'_i) to $\mathcal{A}$ as the query response. Note that T_b can be either $T_0 = \mathsf{Ext}_2(\mathtt{pk}^*; r)$ or $T_1 \in_R \{0,1\}^{m_2(\lambda)}$.

- $*$ If $\mathcal{A}$ issues a secret leakage query to user i, then $\mathcal{S}$ returns $f_1(\mathtt{pk}^*), \cdots, f_{q_{sk}}(\mathtt{pk}^*)$ as the leakage query outputs.
- $*$ If $\mathcal{A}$ issues a biometric leakage query to user i, then $\mathcal{S}$ `abort`.

If the challenge of $\mathcal{S}$ is $T_0 \leftarrow \mathsf{Ext}_2(\mathtt{pk}^*; r)$, then the simulation is consistent with $\mathbb{G}_2$; otherwise, the simulation is consistent with $\mathbb{G}_3$. If the advantage of $\mathcal{A}$ is significantly different in $\mathbb{G}_2$ and $\mathbb{G}_3$, then $\mathcal{S}$ can break the strong extractor with ϵ_2-hard-to-invert auxiliary inputs. Therefore we have

$$\left|\mathtt{Adv}_2^{\text{LR-BRUA}} - \mathtt{Adv}_3^{\text{LR-BRUA}}\right| \leq n \cdot m \cdot \mathtt{Adv}_{\mathcal{S}}^{\mathsf{Ext}_2}(\lambda)$$

- $\mathbb{G}_4$ This game is identical to game $\mathbb{G}_3$ except that in the "Attack" session, we replace the certificate r'_c by a random value. Since we model $\mathtt{H}$ as a random oracle, if the replay attacks (w.r.t., $\mathbb{G}_1$) and impersonation attacks (w.r.t., $\mathbb{G}_2, \mathbb{G}_3$) did not happen, then we have

$$\mathtt{Adv}_3^{\text{LR-BRUA}} = \mathtt{Adv}_4^{\text{LR-BRUA}} \tag{2}$$

It is easy to see that in game $\mathbb{G}_4$, $\mathcal{A}$ has no advantage, i.e.,

$$\mathtt{Adv}_4^{\text{LR-BRUA}} = 0 \tag{3}$$

Combining the above results together, we have

$$\mathtt{Adv}_{\mathcal{A}}^{\text{LR-BRUA}}(\lambda) \leq \quad m^2/2^{\lambda} + n \cdot [n(\lambda) \cdot \mathtt{Adv}_{\mathcal{S}}^{\mathsf{Ext}_3}(\lambda) + m \cdot \mathtt{Adv}_{\mathcal{S}}^{\mathsf{Ext}_2}(\lambda)] \quad \square$$

Theorem 4.2. *The proposed LR-BRUA achieves leakage-resilient biometric-privacy (Definition 2.4) if the strong extractor ϵ_1-hard-to-invert auxiliary inputs is secure, where ϵ_1 is negligible.*

Proof. Let $\mathcal{S}$ denote an adversary who given $(r_2, f_1(r'_1), \cdots, f_{q_r}(r'_1), T_b)$, aims to break the strong extractor with ϵ_1-hard-to-invert auxiliary inputs. $\mathcal{S}$ simulates the game for $\mathcal{A}$ as follows.

- Setup. $\mathcal{S}$ sets up the game for $\mathcal{A}$ by creating n users and m servers with the corresponding identities $\{ID_i, ID_j\}$, where $i \in [1,n]$, $j \in [1,m]$. $\mathcal{S}$ honestly generates n user's biometrics and their corresponding reference biometrics according to the protocol specification of Enrollment. In addition, $\mathcal{S}$ generates user's secret keys with respect to m servers. Note that $\mathcal{S}$ can answer all the queries made by $\mathcal{A}$ using self-generated biometrics and associated secret keys during Training stage.
- Challenge. Upon receiving the challenge request (i.e., $\mathcal{B}_0, \mathcal{B}_1$ of a user i) from $\mathcal{A}$, $\mathcal{S}$ replaces the previously generated reference biometrics of user i by $P_b^* \leftarrow (\mathsf{SS}(\mathcal{B}_b, T_b), r_2)$ (r_2 is chosen by his challenger) and returns it to $\mathcal{A}$. Note that T_b can be either $T_0 = \mathsf{Ext}_1(r'_1; r_2)$ or $T_1 \in_R \{0,1\}^{m_1(\lambda)}$, and $\mathtt{pk}_b$ is perfectly simulated using the enrolled biometrics $\mathcal{B}_b$. In particular, if $\mathcal{A}$ issues a randomness leakage query to user i, then $\mathcal{S}$ returns $f_1(r'_1), \cdots, f_{q_r}(r'_1)$ as the randomness leakage query outputs. In the end, $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs. $\square$

5. Conclusion

In this work, we have proposed a generic framework of leakage-resilient privacy-preserving biometric-based remote user authentication with fuzzy extractors. To the best of our knowledge, this is the first attempt to address leakage attacks on privacy-preserving biometric-based remote user authentication using fuzzy extractors in a single-factor setting. We have defined the formal security models for leakage-resilient user authenticity and biometric-privacy, and we have proved the security of the proposed generic framework under standard assumptions in the random oracle model. We leave the concrete construction of leakage-resilient biometric-based remote user authentication in a single-factor setting as our future work.

Declaration of competing interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Acknowledgements

This work is supported by the Singapore National Research Foundation under NCR Award Number NRF2014NCR-NCR001-012, and is partially sponsored by the National Science Foundation of China under Grant No. 61872264.

References

[1] A.K. Jain, K. Nandakumar, A. Ross, 50 years of biometric research: accomplishments, challenges, and opportunities, Pattern Recognit. Lett. 79 (2016) 80–105.
[2] X. Boyen, Reusable cryptographic fuzzy extractors, in: ACM CCS, 2004, pp. 82–91.
[3] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, A.D. Smith, Secure remote authentication using biometric data, in: EUROCRYPT, vol. 3494, 2005, pp. 147–163.
[4] N. Li, F. Guo, Y. Mu, W. Susilo, S. Nepal, Fuzzy extractors for biometric identification, in: ICDCS, 2017, pp. 667–677.
[5] A.K. Jain, K. Nandakumar, A. Nagar, Biometric template security, EURASIP J. Adv. Signal Process. 2008 (2008) 113.
[6] C. Gentry, Fully homomorphic encryption using ideal lattices, in: STOC, 2009, p. 169.
[7] Y. Dodis, L. Reyzin, A. Smith, Fuzzy extractors: how to generate strong keys from biometrics and other noisy data, in: EUROCRYPT, 2004, pp. 523–540.
[8] S. Halevi, V. Shoup, Helib-an implementation of homomorphic encryption, Cryptology ePrint Archive, Report 2014/039.
[9] J. Bringer, H. Chabanne, M. Izabachène, D. Pointcheval, Q. Tang, S. Zimmer, An application of the goldwasser-micali cryptosystem to biometric authentication, in: ACISP, 2007, pp. 96–106.
[10] J. Bringer, H. Chabanne, D. Pointcheval, Q. Tang, Extended private information retrieval and its application in biometrics authentications, in: International Conference on Cryptology and Network Security, 2007, pp. 175–193.
[11] Q. Tang, J. Bringer, H. Chabanne, D. Pointcheval, A formal study of the privacy concerns in biometric-based remote authentication schemes, in: ISPEC, 2008, pp. 56–70.
[12] A.-R. Sadeghi, T. Schneider, I. Wehrenberg, Efficient privacy-preserving face recognition, in: ICISC, 2009, pp. 229–244.
[13] 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.
[14] Y. Huang, L. Malka, D. Evans, J. Katz, Efficient privacy-preserving biometric identification, in: NDSS, 2011.
[15] C. Fan, Y. Lin, Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics, IEEE Trans. Inf. Forensics Secur. 4 (4) (2009) 933–945.
[16] X. Huang, Y. Xiang, E. Bertino, J. Zhou, L. Xu, Robust multi-factor authentication for fragile communications, IEEE Trans. Dependable Secure Comput. 11 (6) (2014) 568–581.
[17] M. Bellare, P. Rogaway, Entity authentication and key distribution, in: Advances in Cryptology - CRYPTO '93, 1993, pp. 232–249.
[18] B. Schoenmakers, P. Tuyls, Efficient binary conversion for paillier encrypted values, in: EUROCRYPT, 2006, pp. 522–537.
[19] Q. Wang, S. Hu, K. Ren, M. He, M. Du, Z. Wang, Cloudbi: practical privacy-preserving outsourcing of biometric identification in the cloud, in: ESORICS, 2015, pp. 186–205.
[20] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, T. Toft, Privacy-preserving face recognition, in: International Symposium on Privacy Enhancing Technologies Symposium, 2009, pp. 235–253.
[21] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: EUROCRYPT, 1999, pp. 223–238.
[22] A.C.-C. Yao, How to generate and exchange secrets, in: 27th Annual Symposium on Foundations of Computer Science, 1986, pp. 162–167.
[23] S. Micali, L. Reyzin, Physically observable cryptography, in: Theory of Cryptography Conference, 2004, pp. 278–296.
[24] M. Naor, G. Segev, Public-key cryptosystems resilient to key leakage, in: CRYPTO, 2009, pp. 18–35.
[25] Y. Dodis, Y.T. Kalai, S. Lovett, On cryptography with auxiliary input, in: STOC, 2009, pp. 621–630.
[26] G. Yang, Y. Mu, W. Susilo, D.S. Wong, Leakage resilient authenticated key exchange secure in the auxiliary input model, in: ISPEC, 2013, pp. 204–217.
[27] T.H. Yuen, Y. Zhang, S. Yiu, J.K. Liu, Identity-based encryption with post-challenge auxiliary inputs for secure cloud applications and sensor networks, in: ESORICS, 2014, pp. 130–147.
[28] R. Chen, Y. Mu, G. Yang, W. Susilo, F. Guo, Strongly leakage-resilient authenticated key exchange, in: CT-RSA, 2016, pp. 19–36.
[29] A. Juels, M. Wattenberg, A fuzzy commitment scheme, in: ACM CCS, 1999, pp. 28–36.
[30] A. Juels, M. Sudan, A fuzzy vault scheme, Des. Codes Cryptogr. 38 (2) (2006) 237–257.
[31] Y. Dodis, R. Ostrovsky, L. Reyzin, A.D. Smith, Fuzzy extractors: how to generate strong keys from biometrics and other noisy data, SIAM J. Comput. 38 (1) (2008) 97–139.
[32] R. Canetti, B. Fuller, O. Paneth, L. Reyzin, A.D. Smith, Reusable fuzzy extractors for low-entropy distributions, in: EUROCRYPT, 2016, pp. 117–146.
[33] S.K.S. Modak, V.K. Jha, Multibiometric fusion strategy and its applications: a review, Inf. Fusion 49 (2019) 174–204.
[34] M.J. Atallah, K.B. Frikken, M.T. Goodrich, R. Tamassia, Secure biometric authentication for weak computational devices, in: FC, 2005, pp. 357–371.
[35] Y. Tian, Y. Li, R. Chen, N. Li, X. Liu, B. Chang, X. Yu, Privacy-preserving biometric-based remote user authentication with leakage resilience, in: SecureComm, 2018, pp. 112–132.