8-2018

Privacy-preserving biometric-based remote user authentication with leakage resilience

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

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

Rongmao CHEN

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

Bing CHANG
Singapore Management University, bingchang@smu.edu.sg

See next page for additional authors

Citation

TIAN, Yangguang; LI, Yingjiu; CHEN, Rongmao; LIU, Ximeng; CHANG, Bing; and YU, Xingjie. Privacy-preserving biometric-based remote user authentication with leakage resilience. (2018). *Security and Privacy in Communication Networks: 14th International Conference, SecureComm 2018, Singapore, August 8-10, Proceedings*. 254, 112-132.

Available at: https://ink.library.smu.edu.sg/sis_research/4391

Author

Yangguang TIAN, Yingjiu LI, Rongmao CHEN, Ximeng LIU, Bing CHANG, and Xingjie YU

Published in Security and Privacy in Communication Networks: 14th International Conference, SecureComm 2018, Singapore, August 8-10, Proceedings, Pages 112-132.
https://doi.org/10.1007/978-3-030-01701-9_7

Privacy-Preserving Biometric-Based Remote User Authentication with Leakage Resilience

Yangguang Tian[1](✉), Yingjiu Li[1], Rongmao Chen[2], Nan Li[3], Ximeng Liu[1],
Bing Chang[1], and Xingjie Yu[1]

[1] School of Information Systems, Singapore Management University,
Singapore, Singapore
{ygtian,yjli,xmliu,bingchang}@smu.edu.sg, stefanie.yxj@hotmail.com
[2] College of Computer, National University of Defense Technology, Changsha, China
chromao@nudt.edu.cn
[3] School of Electrical Engineering and Computing, University of Newcastle,
Callaghan, Australia
nan.li@newcastle.edu.au

Abstract. Biometric-based remote user authentication is a useful primitive that allows an authorized user to authenticate to a remote server using his biometrics. Leakage attacks, such as side-channel attacks, allow an attacker to learn partial knowledge of secrets (e.g., biometrics) stored on any physical medium. Leakage attacks can be potentially launched to any existing biometric-based remote user authentication systems. Furthermore, applying plain biometrics is an efficient and straightforward approach when designing remote user authentication schemes. However, this approach jeopardises user's biometrics privacy. To address these issues, we propose a novel leakage-resilient and privacy-preserving biometric-based remote user authentication framework, such that registered users securely and privately authenticate to an honest-but-curious remote server in the cloud. In particular, the proposed generic framework provides optimal efficiency using lightweight symmetric-key cryptography, and it remains secure under leakage attacks. We formalize several new security models, including leakage-resilient user authenticity and leakage-resilient biometrics privacy, for biometric-based remote user authentication, and prove the security of proposed framework under standard assumptions.

Keywords: Remote user authentication · Leakage-resilient
Biometrics privacy · Generic framework

1 Introduction

User authentication is the first line of defense in most information systems. While password-based user authentication is still pervasive, it triggers increasing concerns over security (e.g., password leakage and correlated passwords) and usability (e.g., many passwords for each user to remember and frequent update of passwords). To address these concerns, biometrics based user authentication has become increasingly popular in practice in recent years. We focus on biometric-based remote user authentication in this work.

Biometrics (such as face, fingerprint, iris and voice) based remote user authentication may be vulnerable to some leakage attacks in the real world, such as "side channel attacks" on computation time, power consumption, radiation/noise/heat emission. An attacker is able to obtain some imperfect information of the secrets (e.g., biometrics) stored at either user or remote server's side. Specifically, if an impersonator is able to obtain imperfect/partial knowledge of one user's biometrics stored in cloud, then user's authenticity may be compromised. To capture such leakage attacks in biometrics-based remote user authentication setting is the main motivation of this work.

Furthermore, we consider user's biometrics as a secret value in this work. One may argue that biometrics is public information [2,7,28] such as face or fingerprint, but certain liveness detection systems in the literature [24,32] confirmed that biometrics acts as a secret key for (remote) user authentication. In particular, we consider biometrics privacy against an honest-but-curious remote cloud server.

The proposed leakage-resilient and privacy-preserving biometric-based remote user authentication framework has the following properties: (1) user's secret biometrics is hidden to the public; (2) user relies on encryption technique to protect biometrics, the encryption key is permanently stored locally and user's encrypted biometrics is stored in remote cloud; (3) user's encryption key and encrypted biometrics remain secure under certain leakage attacks.

The proposed biometrics-based and privacy-preserving remote user authentication framework is significantly useful in many real-world applications. We take mobile device users enrolling/logging in a service provider in cloud as an example, where they have their respective roles (i.e., client and server). The user authenticity of proposed framework assists in ensuring that a registered user and the remote service provider are performing authentication successfully using encrypted biometrics that are stored in cloud. In other words, user authenticity aims to capture impersonation attacks performed by outsider attackers. The biometrics privacy prevents the honest-but-curious remote service provider from revealing the registered user's secret biometrics. Furthermore, these aforementioned attacks will not be successful under the leakage of secret values.

1.1 This Work

In this work, we introduce the notion of leakage-resilient and privacy-preserving biometric-based remote user authentication (LR-BUA), allowing registered users authenticate to an honest-but-curious remote server using biometrics, and at the same time ensuring leakage resilience to any secrets stored on physical medium and privacy protection on biometrics. Our contributions can be summarized as follows.

- We present the formal security definitions for biometrics-based and privacy-preserving remote user authentication schemes. In particular, we propose a user authenticity model to capture impersonation attacks, and a biometrics privacy model to address an honest-but-curious remote server.
- We present the *first* leakage-resilient user authenticity security model and biometrics privacy model to capture the computationally hard-to-invert leakage attacks on all secret values in the auxiliary inputs model.
- We present the *first* generic construction on leakage-resilient and privacy-preserving biometric-based remote user authentication, and prove that the proposed LR-BUA generic construction can achieve leakage-resilient user authenticity and biometrics privacy under standard assumptions.
- We show the instantiations of all the building blocks. In particular, we present a lightweight biometrics-based remote user authentication scheme and its overall performance analysis.

Table 1. A comparative summary of biometrics-based user authentication.

Function/scheme	[2]	[29]	[18]	[28]	[15]	[23]	Ours
Biometrics privacy	✓	✓	✓	×	✓	✓	✓
†-Factor authentication[a]	Two	One	One	One	Three	One	Two
Lightweight cryptography[b]	✓	✓	×	×	×	✓	✓
Remote user authentication	✓	×	×	✓	✓	×	✓
Leakage-resilient w.r.t user	×	×	×	×	×	×	✓
Leakage-resilient w.r.t server	×	×	×	×	×	×	✓

[a] † denotes number of factors for authentication/identification.
[b] Lightweight Cryptography means symmetric key cryptography (e.g., symmetric key encryption [29]) rather than public key cryptography (e.g., homomorphic encryption [18,27]).

1.2 Related Work

Biometric-based Authentication. Atallah et al. [2] proposed the first lightweight biometrics-based authentication using cryptographic hash functions, and formally defined security requirements for biometrics-based authentication including confidentiality, integrity and availability. Notice that some research work in the literature [2,7,28] assume that the biometrics is a public value (such as fingerprint and face), and their *privacy* concern is the relationship between a biometric information and user's real identity.

However, three-factor [15,17] and multi-factor [16] authentication (such as smart card, password and biometrics) in the literature formed an opposite research direction, such that biometrics acts as a secret key for (remote) user authentication, and the proposed three/multi-factor solutions are able to provide enhanced security on user authentication. Meanwhile, another research line [24,32] also confirmed this assumption. One well-known 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 [4] model. Specifically, they require user's biometrics is not sharing with remote server, and the biometrics matching is performed by remote server.

Moreover, some research work focused on privacy-preserving (remote) user biometrics authentication/identification, and a few novel solutions [18,23,27,29] are mainly for biometrics *identification* in the cloud. For instance, Schoenmakers and Tuyls [27] proposed to use a homomorphic encryption scheme for efficient biometric authentication by employing multi-party computation techniques. Wang et al. [29] used invertible matrices as symmetric-key secrets to encrypt biometrics and the exact biometrics matching are executed in the transformed (i.e., encrypted) domain, namely, transformation-based cancellable biometrics [22]. In Table 1, we compare our proposed solution with typical works on biometric-based authentication/identification to highlights our distinctions: it shows that our proposed solution is the first lightweight biometrics based remote user authentication with leakage-resilient and biometrics privacy.

"Fast Identity Online" (FIDO) alliance [1] is an industry consortium to address the lack of interoperability between authentication devices and user authentication experiences. Specifically, FIDO is used to enhance user authentication security (e.g., using biometrics) on local devices, while we focus on remote biometric-based user authentication in this work.

Modelling Leakage Attacks. Biometrics and secret values used in biometrics-based user authentication may be subject to leakage attacks. Micali and Reyzin [25] 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 user's secret key. 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 [26] 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. [12] proposed a more general model: "auxiliary inputs". Instead of min-entropy requirement on secret key x, they only require 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 any upper bound that defined in the bounded/noisy leakage models, and the chosen functions f must "hard-to-invert". Notice that leakage-resilient cryptography (e.g., [10,30,31]) has been extensively studied in the auxiliary inputs model. However, all the previous leakage-resilient works didn't address the leakages on secret biometrics in the (remote) user authentication systems, such as the secret (encrypted) biometrics stored in the remote server. Furthermore, the leakage attacks on secret biometrics become more challenging as those encrypted biometrics is a key to the authentication success, and the adversarial capability has not been formally captured by the existing leakage models.

Fuzzy Extractor. Fuzzy extractor is one of the building blocks for constructing biometric-based remote user authentication in this work. Juels and Wattenberg [21] introduced a new type of cryptography primitive "fuzzy commitment scheme". It is particularly useful for biometric authentication systems because error-correcting property within a suitable metric. Juels and Sudan [20] proposed another novel construction "fuzzy vault scheme". It is based on set distance rather than hamming distance used in [21]. 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. [14] formally introduced the notion of secure sketches and fuzzy extractors, and use biometrics to derive a cryptographic key for various cryptographic applications, such as password-based authentication.

Recently, Li et al. [23] proposed the first fuzzy extractor based biometric identification protocol using a newly built fuzzy extractor, which is focusing on real number strings with Chebyshev distance. In particular, the proposed fuzzy extractor is suitable for efficient user identification, but its drawback is less error-tolerance than hamming distance or edit distance. In order to achieve fast remote user authentication on-line, we implement this succinct fuzzy extractor in our proposed instantiation scheme.

With regard to specific attacks on fuzzy extractor, Boyen et al. [6] introduced a notion called "robust sketches", and provided a generic conversion to prevent an active attack, such that adversary can modify the public helper data so as to compromise the security of secure sketches and fuzzy extractors. Later on, Canetti et al. [8] presented another notion, namely "reusable fuzzy extractor" (the prior work is [5]). It addressed an issue that user has multiple sketches from the same sketch scheme, and his (low-entropy) biometrics information may be leaked.

2 Security Model

In this section, we firstly present the system model for biometric-based remote user authentication, then we present the security models for LR-BUA.

Notation. We define a system with n users. We denote the i-th session established by a user as Π_U^i, and identities of all the users recognised 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 the user U. Specifically, $\mathsf{sid}_U^i = \{m_j\}_{j=1}^n$, where $m_j \in \{0,1\}^*$ is the message transcript among 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 initialisation the oracle is marked as used

Table 2. Summary of notations

Notation	Definition
$\mathtt{pk}_i/\mathtt{sk}_i$	User i' public key/private 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)
$\mathcal{B}$	Biometrics information
$\mathcal{C}$	Encrypted biometrics information
T_{Enc}	One-way transformation-based encryption scheme
$\mathsf{Ext}(x,r)$	Strong extractor

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 finalise 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 having accepted. In addition, we present the commonly used notations (see Table 2) in this paper.

2.1 System Model

In this work, we present a biometric-based remote user authentication system involving two entities: user and cloud server. We then define a biometric-based remote user authentication framework which consists of the following algorithms:

- Registration. This is an algorithm that executed between a user and a cloud server $\widehat{S}$ in a secure channel. User registers his identity ID along with a reference biometric information $\mathcal{B}$[1] to cloud server $\widehat{S}$.
- Authentication. This is an interactive algorithm between a registered user and a cloud server $\widehat{S}$ in a public channel. User sends his identity ID and specific information associates with a candidate biometric information $\mathcal{B}'$ to cloud server $\widehat{S}$, while $\widehat{S}$ accept it if and only if $t' = \mathsf{dist}(\mathcal{B}', \mathcal{B}) \leq t$.

2.2 Security Model

We define a formal user authenticity model to capture the impersonation attacks performed by outsider adversaries, and a formal biometrics privacy

[1] Reference biometrics can be interpreted as either encrypted biometrics [9] or plain biometrics.

model to capture an honest-but-curious server for biometric-based authentication/identification protocols. Furthermore, we extend both user authenticity and biometrics privacy models to the leakage-resilient against auxiliary inputs models for tackling leakage attacks, such as side-channel attacks.

Authenticity. Informally, an adversary $\mathcal{A}$ attempts to impersonate a registered user and authenticate to a cloud server. We then define a formal authenticity game between a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ and a simulator $\mathcal{S}$ (i.e., challenger) as follows.

- Setup. $\mathcal{S}$ first generates identity/static key pair $(ID_i, \mathtt{sk}_i)$ for n users and an identity $ID_{\widehat{S}}$ for cloud server in the system, where $\mathtt{sk}_i$ denotes the secret key of user i. In addition, $\mathcal{S}$ honestly generates user's reference biometric information $\{\mathcal{B}_i\}$. Eventually, $\mathcal{S}$ sends user/cloud server's identities $(\{ID_i\}, ID_{\widehat{S}})$ 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.
  - Biometric Reveal: If $\mathcal{A}$ issues a biometric reveal query to user i, then $\mathcal{S}$ returns user i's reference biometric information $\mathcal{B}_i$ to $\mathcal{A}$.
  - Static Key Reveal: If $\mathcal{A}$ issues a static key reveal (or corrupt, for short) query to user i, then $\mathcal{S}$ returns user i's static secret key $\mathtt{sk}_i$ (e.g., static key stored in ROM) to $\mathcal{A}$.
  - State Reveal: If $\mathcal{A}$ issues a state reveal query to (possibly unaccepted) instance oracle $\Pi_{U_i}^j$ $(j \neq i)$, then $\mathcal{S}$ will return all internal state values (e.g., ephemeral key stored in RAM) contained in $\Pi_{U_i}^j$ at the moment the query is asked.
- Attack. $\mathcal{A}$ wins the game if all of the following conditions hold.
  - $\mathcal{S}$ accept user i; It implies $\mathsf{sid}_{\widehat{S}}^s$ exists.
  - $\mathcal{A}$ did *not* issue Biometric Reveal query with regard to user i;
  - $m_i \in \mathsf{sid}_{\widehat{S}}^s$, *but* there exists *no* $\Pi_{U_i}^s$ which has sent m_i (m_i denotes the message transcript from user i)[2].

  We define the advantage of an adversary $\mathcal{A}$ in the above game as

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

Definition 1. *We say a biometric-based remote user authentication (BUA) scheme has* authenticity *if for any PPT $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A}}^{BUA}(\lambda)$ is a* negligible *function of the security parameter λ.*

[2] We do not consider the collude attack between an impersonator and a curious server in this work.

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

- Setup: $\mathcal{S}$ first generates the identity/static key pair $(ID_i, \mathtt{sk}_i)$ for n user in the system, where $\mathtt{sk}_i$ denote the secret key of user i. In addition, $\mathcal{S}$ honestly generates user's reference biometric information $\{\mathcal{C}_i\}$[3]. Eventually, $\mathcal{S}$ sends user's identities $\{ID_i\}$ to $\mathcal{A}$. We denote the original n users set as $\mathcal{U}$.
- Training: $\mathcal{A}$ is allowed to issue Send, Biometric reveal, State reveal and at most n-1 Static key reveal queries to $\mathcal{S}$. We denote the honest (i.e., uncorrupted) user set as $\mathcal{U}'$.
- Challenge: $\mathcal{S}$ randomly selects a reference biometrics information $\mathcal{C}_i$ $(ID_i \in \mathcal{U}')$ as challenge candidate, and send it to $\mathcal{A}$. $\mathcal{A}$ wins the game if $\mathcal{B}_i \leftarrow \mathcal{A}(\mathcal{C}_i)$. We then define the advantage of an adversary $\mathcal{A}$ in the above game as

$$\mathtt{Adv}_{\mathcal{A}}^{BUA}(\lambda) = |\Pr[\mathcal{A}\ wins]|. \tag{1}$$

Definition 2. *We say a BUA scheme has* biometrics privacy *if for any PPT $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A}}^{BUA}(\lambda)$ is a* negligible *function of the security parameter λ.*

Authenticity Against Auxiliary Inputs. To model the leakage on both the biometric information and the static key with respect to auxiliary inputs, we first define a set of admissible functions $\mathcal{H}$. According to the work of Dodis et al. [12], we 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 $h : \{0,1\}^{|bio|} \rightarrow \{0,1\}^*$, 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 $\mathcal{H}_{ow}(\epsilon_{sta})$ be the class of all the polynomial-time computable functions $h : \{0,1\}^{|sta|} \rightarrow \{0,1\}^*$, such that given $h(sta)$ (for a randomly generated static key sta), no PPT adversary can find sta with probability $\geq \epsilon_{sta}$. The function $h(sta)$ can be viewed as a composition of $q_{sta} \in \mathbb{N}^+$ functions, i.e., $h(sta) = (h_1(sta), \cdots, h_{q_{sta}}(sta))$ where for all $i \in \{1, \cdots, q_{sta}\}, h_i \in \mathcal{H}_{ow}(sta)$.

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

[3] The secret key is used to protect biometrics, such as $\mathcal{C}_i \leftarrow F(\mathtt{sk}_i, \mathcal{B}_i)$, where F denotes a one-way function.

- Static Key Leakage: If $\mathcal{A}$ issues a static key leakage query to user i (i.e., $\mathcal{O}_{sta}(i)$), then $\mathcal{S}$ returns $f_{Sta}(Sta_i)$ to $\mathcal{A}$, where $f_{Sta} \in \mathcal{H}_{ow}(\epsilon_{sta})$, and Sta_i denotes the static key of user i.

A General Trivial Attack. Consider an adversary is allowed to reveal user's secret key Sta in the LR-BUA model, she then can launch a trivial attack by encoding the reference derivation function into the leakage function of f_{Sta}, hence obtains biometrics information $\mathcal{B}_i$ and wins the leakage-resilient user authenticity game. Similarly, an adversary can launch another trivial attack by encoding the static key derivation function into the leakage function of f_{Bio} if user's reference biometrics is revealed, which is corresponding to the leakage-resilient biometrics privacy game below.

Our Treatment. In our proposed leakage-resilient biometric-based user authenticity model, we ask the adversary to submit two leakage function sets $\mathcal{F}_{Bio} \subseteq \mathcal{H}_{ow}(\epsilon_{bio}), \mathcal{F}_{Sta} \subseteq \mathcal{H}_{ow}(\epsilon_{sta})$, where both $\mathcal{F}_{Bio}$ and $\mathcal{F}_{Sta}$ are polynomial in the security parameter λ, prior to game Setup which is observed in [10]. During the LR-BUA security game, $\mathcal{A}$ is allowed to adaptively access both biometric leakage oracle f_{Bio} and static key leakage oracle f_{Sta}. We require that $f_{Bio} \in \mathcal{F}_{Bio}, f_{Sta} \in \mathcal{F}_{Sta}$ and $\mathcal{A}$ is not allowed to leak reference biometric information $\mathcal{B}_i$ entirely. We define the advantage of an adversary $\mathcal{A}$ in the LR-BUA game as

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

Definition 3. *We say a BUA scheme has* leakage-resilient authenticity *if for any PPT $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A}}^{LR-BUA}(\lambda)$ is a* negligible *function of the security parameter λ.*

Biometrics Privacy Against Auxiliary Inputs. In this extended biometrics privacy against auxiliary inputs model, $\mathcal{A}$ is additionally allowed to access challenge user's Static Key Leakage oracle $\mathcal{O}_{sta}(i)$, and $\mathcal{A}$ is not allowed to leak static secret key $\mathtt{sk}_i$ entirely. We follow the same treatment described above and define the advantage of an adversary $\mathcal{A}$ in the biometrics privacy game as

$$\mathtt{Adv}_{\mathcal{A}}^{LR-BUA}(\lambda) = |\Pr[\mathcal{A}\ wins]|. \quad (2)$$

Definition 4. *We say a BUA scheme has* leakage-resilient biometrics privacy *if for any PPT $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A}}^{LR-BUA}(\lambda)$ is a* negligible *function of the security parameter λ.*

3 Our Construction

In this section, we present the proposed generic fuzzy extractor that will be used in the proposed generic construction, and present our proposed LR-BUA generic framework and security analysis respectively.

3.1 Generic Fuzzy Extractor

We present a generic fuzzy extractor with hard-to-invert auxiliary inputs, which is built on top of a (robust)[4] secure sketch [14] and a (δ, ϵ)-strong extractor with hard-to-invert auxiliary inputs [10,12,31].

Definition 5. *A generic fuzzy extractor with ϵ-hard-to-invert auxiliary inputs consists of two randomised procedures* $(\mathsf{Gen}, \mathsf{Rep})$ *with the following properties.*

- Gen*: Let* SS *be a secure sketch and* Ext *be a strong extractor with ϵ-hard-to-invert auxiliary inputs. Given an input x,* $\mathsf{Gen}(x; r_1, r_2) \rightarrow (P, R)$*, such that*
$$P = (\mathsf{SS}(x; r_1), r_2), \;\; R = \mathsf{Ext}(x; r_2).$$
- Rep*: Given an noisy input x' and P, recover the original input* $x = \mathsf{Rec}(x', \mathsf{SS}(x; r_1))$*, then compute* $R = \mathsf{Ext}(x; r_2)$*.*

Theorem 1. *The proposed generic fuzzy extractor with ϵ-hard-to-invert auxiliary inputs is secure if the (robust) secure sketch is secure and the (δ, ϵ)-strong extractor with hard-to-invert auxiliary inputs is secure.*

The security of proposed generic fuzzy extractor is based on the statistical indistinguishability of two distributions below.

$$\begin{aligned} &|\Pr[\mathcal{A}(r_2, f(x), \mathsf{SS}(x; r_1), \mathsf{Ext}(x; r_2)) = 1]| \\ &\quad -|\Pr[\mathcal{A}(r_2, f(x), \mathsf{SS}(x; r_1), u) = 1]| < \delta \end{aligned}$$

Where $x, r_1 \in_R \{0,1\}^{l_1}, r_2 \in_R \{0,1\}^{l_2}, u \in_R \{0,1\}^m$ and $f \in \mathcal{H}_{ow}(\epsilon)$.

Proof. We use (δ, ϵ)-strong extractor with hard-to-invert auxiliary inputs to derive the strong extractor Ext from the proposed generic fuzzy extractor. The (δ, ϵ)-strong extractor with hard-to-invert auxiliary inputs can guarantee the security of such (leakage-resilient) strong extractor of proposed generic fuzzy extractor. In other words, the output string $\mathsf{Ext}(x; r_2)$ is statistically indistinguishable with a string u which is generated uniformly at random, even if a leakage function f is provided. Furthermore, the secure sketch $\mathsf{SS}(x; r_1)$ is secure due to the fact that adversary can recover x with a negligible advantage [14]. Therefore, the proposed generic fuzzy extractor with ϵ-hard-to-invert auxiliary inputs is secure.

Remark. The proposed fuzzy extractor with ϵ-hard-to-invert auxiliary inputs is a *stronger* assumption than a generic fuzzy extractor defined in [14], which allows adversary to access a leakage function f (adaptively). We stress that

[4] It can detect the modification of helper data P_i over public channel (secure in the random oracle model), please refer to [6,13,23] for detailed generic construction of *robust secure sketch*.

the proposed fuzzy extractor with ϵ-hard-to-invert auxiliary inputs is a generic construction (i.e., without concrete construction). To this end, Dodis et al. [12] constructed the first reusable (and robust) extractor with hard-to-invert auxiliary inputs at the non-fuzzy case (i.e., without helper data and Rep algorithm, or when $x = x'$, where x' denotes a noisy input). Meanwhile, as stated by Canetti et al. [8], most constructions of fuzzy extractor are not reusable (except [5,8]), and adding error-correcting codes to a strong extractor with hard-to-invert auxiliary inputs at the fuzzy case (i.e., when $x \neq x'$) is a challenging task.

3.2 Generic Framework

High-level Description. User submits his/her reference biometrics to a remote server during registration phase; Remote server then acknowledges user's authenticity if and only if user's candidate biometrics is statistically "close" to his/her reference biometrics during authentication phase. We define a collision-resistant hash function as $\mathtt{H} : \{0,1\}^* \rightarrow \mathbb{Z}_q$, a strong extractor with ϵ_2-hard-to-invert auxiliary inputs $\mathsf{Ext}_2 : \{0,1\}^{l_1'(\lambda)} \times \{0,1\}^{l_2'(\lambda)} \rightarrow \{0,1\}^{m_2(\lambda)}$ and a generic fuzzy extractor with ϵ_1-hard-to-invert auxiliary inputs ($\mathsf{Ext}_1 : \{0,1\}^{l_1(\lambda)} \times \{0,1\}^{l_2(\lambda)} \rightarrow \{0,1\}^{m_1(\lambda)}$) in the system.

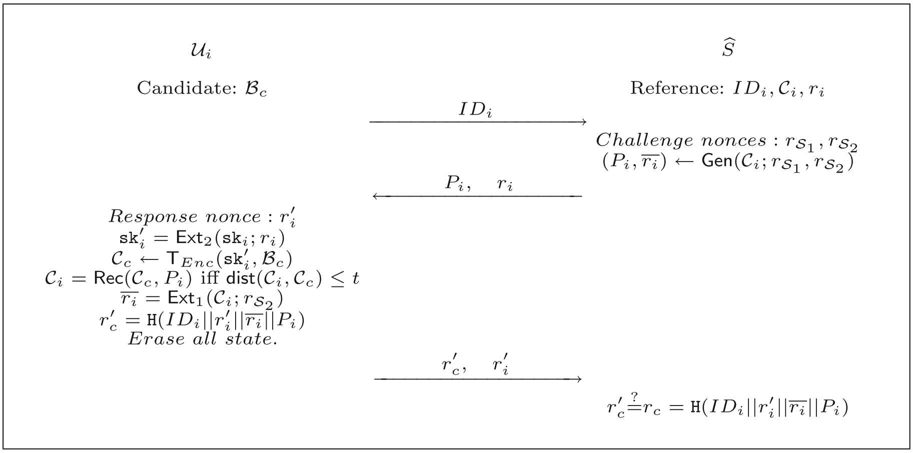

$\mathcal{U}_i$
$\widehat{S}$
Candidate: $\mathcal{B}_c$
Reference: $ID_i, \mathcal{C}_i, r_i$
ID_i
$Challenge\ nonces : r_{\mathcal{S}_1}, r_{\mathcal{S}_2}$
$(P_i, \overline{r_i}) \leftarrow \mathsf{Gen}(\mathcal{C}_i; r_{\mathcal{S}_1}, r_{\mathcal{S}_2})$
$P_i, \quad r_i$
$Response\ nonce : r_i'$
$\mathtt{sk}_i' = \mathsf{Ext}_2(\mathtt{sk}_i; r_i)$
$\mathcal{C}_c \leftarrow \mathsf{T}_{Enc}(\mathtt{sk}_i', \mathcal{B}_c)$
$\mathcal{C}_i = \mathsf{Rec}(\mathcal{C}_c, P_i)$ iff $\mathsf{dist}(\mathcal{C}_i, \mathcal{C}_c) \leq t$
$\overline{r_i} = \mathsf{Ext}_1(\mathcal{C}_i; r_{\mathcal{S}_2})$
$r_c' = \mathtt{H}(ID_i || r_i' || \overline{r_i} || P_i)$
$Erase\ all\ state.$
$r_c', \quad r_i'$
$r_c' \overset{?}{=} r_c = \mathtt{H}(ID_i || r_i' || \overline{r_i} || P_i)$

Fig. 1. Authentication. (public channel)

- $\mathsf{Registration}$. A user i performs below.
  1. Generate a biometric information $\mathcal{B}_i$, and a secret key $\mathtt{sk}_i$ along with a public randomness r_i; Note that user i takes $\mathtt{sk}_i$ as a secret key and stores it locally.
  2. Compute an encryption key $\mathtt{sk}_i' = \mathsf{Ext}_2(\mathtt{sk}_i; r_i)$ using fuzzy extractor with ϵ_2-hard-to-invert auxiliary inputs Ext_2;

3. Compute the reference biometrics $\mathcal{C}_i = \mathsf{T}_{Enc}(\mathtt{sk}_i', \mathcal{B}_i)$, and sends $(ID_i, \mathcal{C}_i, r_i)$ to a cloud server $\widehat{S}$.

Note that cloud server $\widehat{S}$ takes/stores reference biometrics $\mathcal{C}_i$ as a shared secret key with user i, and the registered user erases $\mathtt{sk}_i'$ after the generation of reference biometrics.

- Authentication. The interaction between a registered user and cloud server performs as follows (see Fig. 1).
  - Upon receiving a request ID_i from user i, cloud server $\widehat{S}$ performs below.
    1. Compute the challenge nonces $r_{\mathcal{S}_1}, r_{\mathcal{S}_2}$;
    2. Run the generic fuzzy extractor with ϵ_1-hard-to-invert auxiliary inputs to obtain $(P_i, \overline{r_i}) \leftarrow \mathsf{Gen}(\mathcal{C}_i; r_{\mathcal{S}_1}, r_{\mathcal{S}_2})$, where $P_i = (\mathsf{SS}(\mathcal{C}_i; r_{\mathcal{S}_1}), r_{\mathcal{S}_2}), \overline{r_i} = \mathsf{Ext}_1(\mathcal{C}_i; r_{\mathcal{S}_2})$;
    3. Send (P_i, r_i) to user i.
  - Then user i performs below.
    1. Generate a candidate biometric information $\mathcal{B}_c$ and compute $\mathcal{C}_c = \mathsf{T}_{Enc}(\mathtt{sk}_i', \mathcal{B}_c)$, where encryption key $\mathtt{sk}_i' = \mathsf{Ext}_2(\mathtt{sk}_i; r_i)$ is computed using locally stored secret key $\mathtt{sk}_i$ and public randomness r_i;
    2. Run the generic fuzzy extractor with ϵ_1-hard-to-invert auxiliary inputs to obtain $\mathcal{C}_i = \mathsf{Rec}(\mathcal{C}_c, P_i)$ $(P_i = (\mathsf{SS}(\mathcal{C}_i; r_{\mathcal{S}_1}), r_{\mathcal{S}_2})$ if and only if $\mathsf{dist}(\mathcal{C}_i, \mathcal{C}_c) \leq t$, and compute $\overline{r_i} = \mathsf{Ext}_1(\mathcal{C}_i; r_{\mathcal{S}_2})$;
    3. Choose a response nonce r_i' and compute the token $r_c' = \mathtt{H}(ID_i || r_i' || \overline{r_i} || P_i)$;
    4. Erase all state and send (r_c', r_i') to cloud server $\widehat{S}$.
  - Eventually, cloud server $\widehat{S}$ computes the token $r_c = \mathtt{H}(ID_i || r_i' || \overline{r_i} || P_i)$ and checks $r_c' \overset{?}{=} r_c$. If it does hold, accept; Otherwise, reject.

3.3 Security Analysis

Theorem 2. *The proposed LR-BUA achieves leakage-resilient authenticity (*Definition 3*) in the random oracle model if the generic fuzzy extractor with ϵ_1-hard-to-invert auxiliary inputs is secure, where ϵ_1 is negligible.*

High-Level Discussion. Before we present detailed security proof, we clarify the motivation of each game for leakage-resilient user authenticity security. Game $\mathbb{G}_1$ is used to prevent replay attacks; Game $\mathbb{G}_2$ is used to capture an adversary, who is allowed to reveal the static key of user i, aims to impersonate *corrupted* user i to authenticate to a remote server $\widehat{S}$.

Proof. We define a sequence of games $\{\mathbb{G}_i\}$ and let $\mathtt{Adv}_i^{LR-BUA}$ 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 `abort` if challenge/response nonce (i.e., $r_{\mathcal{S}_\in}, r'_i$) is used twice by the server/user in two different sessions. Therefore, we have

$$\left|\mathtt{Adv}_0^{LR-BUA} - \mathtt{Adv}_1^{LR-BUA}\right| \leq m^2/2^\lambda \tag{3}$$

- $\mathbb{G}_2$: This game is identical to game $\mathbb{G}_1$ except that in the "Attack"session, $\mathcal{S}$ replaces the real value $\overline{r_i}$ by a random value $R \in \{0,1\}^{m_1(\lambda)}$ with regard to instance oracle $\Pi^i_{U_i}$. 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_1-hard-to-invert auxiliary inputs is secure.

  Let $\mathcal{S}$ denote an adversary, who is given $(r, f_1(\mathcal{C}_i), \cdots, f_{q_{Bio}}(\mathcal{C}_i), \mathsf{SS}$ $(C_i; r_1), T_b)$, aims to break the generic fuzzy 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 with the corresponding identity, secret key and public randomness $\{ID_i, \mathtt{sk}_i, r_i\}$. $\mathcal{S}$ randomly selects an index i and guesses that the "Attack" event will happen with regard to user i. In addition, $\mathcal{S}$ honestly generates rest user's biometrics information $\{\mathcal{B}_j\}^n_{j\neq i}$ and their corresponding reference biometrics $\{\mathcal{C}_j\}$. It is obvious that $\mathcal{S}$ can answer all the queries made by $\mathcal{A}$ except user i (w.r.t. reference biometrics $\mathcal{C}_i$). Below we mainly focus on the simulation of user i only.
  - 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.s.t instance oracle $\Pi^i_{U_i}$, $\mathcal{S}$ forwards it to his challenger and obtains a helper data P_i (where $P_i = (\mathsf{SS}(\mathcal{C}_i; r_1), r)$, and (r_1, r) are chosen by his challenger), and returns (P_i, r_i) to $\mathcal{A}$ as the query response. Note that r_i is the public randomness chosen by $\mathcal{S}$.

      If $\mathcal{A}$ issues a send query in the form of (P_i, r_i) to $\mathcal{S}$, $\mathcal{S}$ randomly chooses a response nonce r'_i and sets $\overline{r_i} = T_b$; $\mathcal{S}$ then computes the token $r'_c = \mathtt{H}(ID_i||r_i||\overline{r_i}||P_i)$ and returns (r'_c, r'_i) to $\mathcal{A}$. Note that T_b can be either $T_0 = \mathsf{Ext}_1(\mathcal{C}_i; r)$ or $T_1 \in_R \{0,1\}^{m_1(\lambda)}$.
    - If $\mathcal{A}$ issues a static key leakage query to user i, then $\mathcal{S}$ randomly chooses a leakage function $f_{Sta} \in \mathcal{F}_{Sta} \subseteq \mathcal{H}_{ow}(\epsilon_2)$ and returns $f_{Sta}(\mathtt{sk}_i)$ to $\mathcal{A}$ as the leakage query outputs. Note that $\mathcal{A}$ is allowed to reveal $\mathtt{sk}_i$ entirely.
    - If $\mathcal{A}$ issues a biometric leakage query to user i, then $\mathcal{S}$ returns $f_1(\mathcal{C}_i)$, $\cdots, f_{q_{Bio}}(\mathcal{C}_i)$ as the leakage query outputs.
    - If $\mathcal{A}$ issues a state reveal query to an instance oracle $\Pi^i_{U_i}$, then $\mathcal{S}$ returns (r'_i, r) to $\mathcal{A}$.

  If the challenge of $\mathcal{S}$ is $T_0 = \mathsf{Ext}_1(\mathcal{C}_i; r)$, 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_1-hard-to-invert auxiliary inputs. Therefore we have

$$\left|\mathtt{Adv}_1^{LR-BUA} - \mathtt{Adv}_2^{LR-BUA}\right| \leq n \cdot m \cdot \mathtt{Adv}_{\mathcal{S}}^{\mathsf{Ext}_1}(\lambda) \tag{4}$$

- $\mathbb{G}_3$ This game is identical to game $\mathbb{G}_2$ except that in the "Attack" session, we replace the token r'_c by a random value R. 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., $\mathbb{G}_2$) did not happen, then we have

$$\mathtt{Adv}_2^{LR-BUA} = \mathtt{Adv}_3^{LR-BUA}$$

It is easy to see that in game $\mathbb{G}_3$, $\mathcal{A}$ has no advantage, i.e.,

$$\mathtt{Adv}_3^{LR-BUA} = 0 \tag{5}$$

Combining the above results together, we have

$$\mathtt{Adv}_{\mathcal{A}}^{LR-BUA}(\lambda) \leq m^2/2^\lambda + n \cdot m \cdot \mathtt{Adv}_{\mathcal{S}}^{\mathsf{Ext}_1}(\lambda)$$

Theorem 3. *The proposed LR-BUA achieves leakage-resilient biometrics privacy (*Definition 4*) if* Ext_2 *is a strong extractor with* ϵ_2*-hard-to-invert auxiliary inputs, where* ϵ_2 *is negligible.*

Proof. Let $\mathcal{S}$ denote an adversary, who is given $(r, f_1(\mathtt{sk}_i), \cdots, f_{q_{Sta}}(\mathtt{sk}_i), T_b)$, 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 with the corresponding identity/biometric $\{ID_i, \mathcal{B}_i\}$. $\mathcal{S}$ randomly selects an index i and guesses that the challenge reference biometrics $\mathcal{C}^*$ will happen with regard to user i. In addition, $\mathcal{S}$ honestly generates rest user's secret key and public randomness pair $\{\mathtt{sk}_j, r_j\}_{j \neq i}^n$ and their corresponding reference biometrics $\{\mathcal{C}_j\}$. Eventually, $\mathcal{S}$ sends all the reference biometrics (include $\mathcal{C}^*$) to $\mathcal{A}$. It is obvious that $\mathcal{S}$ can answer all static secret reveal queries made by $\mathcal{A}$ except user i. Below we mainly focus on the simulation of user i only.
- Training. $\mathcal{S}$ answers $\mathcal{A}$'s queries as follows.
  - If $\mathcal{A}$ issues a send query in the form of (P_i, r) to $\mathcal{S}$, then $\mathcal{S}$ performs the simulation as follows. Firstly, $\mathcal{S}$ chooses the response randomness r'_i, and computes the challenge reference biometrics $\mathcal{C}^* = \mathsf{T}_{Enc}(T_b, \mathcal{B}_i)$; Secondly, $\mathcal{S}$ runs the generation of generic fuzzy extractor to obtain $(P_i, \overline{r_i}) \leftarrow \mathsf{Gen}(\mathcal{C}_i; r_{i1}, r_{i2})$, where P_i denotes a helper date and $\overline{r_i} = \mathsf{Ext}_1(\mathcal{C}_i; r)$, and (r_{i1}, r_{i2}) are randomly chosen by $\mathcal{S}$; Eventually, $\mathcal{S}$ computes the token $r'_c = \mathtt{H}(ID_i || r'_i || \overline{r_i} || P_i)$ and sends (r'_c, r'_i) to $\mathcal{A}$ as the query response. Note that T_b can be either $T_0 = \mathsf{Ext}_2(\mathtt{sk}_i; r)$ or $T_1 \in_R \{0,1\}^{m_2(\lambda)}$.

    We assume user i may use same $\mathtt{sk}_i, \mathcal{B}_i$ with different public randomness $r^* \neq r_i$ at most $n(\lambda)$ times (where n is a polynomial in the security parameter λ) for generating different references during registration. For instance, $\mathcal{C}_i^* = \mathsf{T}_{Enc}(\mathtt{sk}_i^*, \mathcal{B}_i), \mathtt{sk}_i^* = \mathsf{Ext}_2(\mathtt{sk}_i; r^*)$.
  - If $\mathcal{A}$ issues a static key leakage query to user i, then $\mathcal{S}$ returns $f_1(\mathtt{sk}_i), \cdots, f_{q_{Sta}}(\mathtt{sk}_i)$ as the leakage query outputs.

- If $\mathcal{A}$ issues a state reveal query to an instance oracle $\Pi_{U_i}^{i}$, then $\mathcal{S}$ returns (r_i', r_{i2}) to $\mathcal{A}$.

Finally, $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs. If $\mathcal{A}$ guesses the random bit correctly, then $\mathcal{S}$ can break the strong extractor with ϵ_2-hard-to-invert auxiliary inputs. Hence, we have

$$\mathtt{Adv}_{\mathcal{A}}^{LR-BUA}(\lambda) \leq n(\lambda) \cdot \mathtt{Adv}_{\mathcal{S}}^{\mathsf{Ext}_2}(\lambda) \tag{6}$$

4 Instantiation

In this section, we first present a lightweight biometric-based remote user authentication scheme using an efficient fuzzy extractor proposed in [23]. We then present the performance analysis and efficiency analysis respectively. Note that the work in [31] showed that a strong extractor with auxiliary inputs can be constructed from the modified Goldreich-Levin theorem (refer to [31] for detailed instantiation).

4.1 The Lightweight Biometric-Based Remote User Authentication Scheme

We present a lightweight and efficient biometric-based remote user authentication scheme below.

- Registration. A user i performs below.
  1. Generate a biometric information vector $\mathcal{B}_i = [b_{i1}, b_{i2}, \cdots, b_{in}]$ $(b_i \in \mathbb{Z}_q)$;
  2. Choose an encryption key $\mathtt{sk}_i' \in_R \{0,1\}^{n|q|}$;
  3. Compute the reference biometric information $\mathcal{C}_i = \mathtt{sk}_i' \oplus \mathcal{B}_i$ and send $(ID_i, \mathcal{C}_i)$ to cloud server $\widehat{S}$.
- Authentication. The interaction between a user and the cloud server performs as follows.
  - Upon receiving a request ID_i from user i, cloud server $\widehat{S}$ performs below.
    1. Compute a challenge nonce $r_{\mathcal{S}} \in \mathbb{Z}_p$;
    2. Run the fuzzy extractor in [23]to obtain $(P_i, \overline{r_i}) \leftarrow \mathsf{Gen}(\mathcal{C}_i; r_{\mathcal{S}})$, where $P_i = (\mathsf{SS}(\mathcal{C}_i), r_{\mathcal{S}}), \overline{r_i} = \mathsf{Ext}(\mathcal{C}_i; r_{\mathcal{S}})$;
    3. Send P_i to user i.
  - Then user i performs below.
    1. Generate a candidate biometric information vector $\mathcal{B}_c = [b_{c1}, b_{c2}, \cdots, b_{cn}]$, and computes the candidate biometrics $\mathcal{C}_c = \mathtt{sk}_i' \oplus \mathcal{B}$;
    2. Run the fuzzy extractor in [23] to obtain $\mathcal{C}_i = \mathsf{Rec}(\mathcal{C}_c, P_i)$ $(P_i = (\mathsf{SS}(\mathcal{C}_i), r_{\mathcal{S}})$ if and only if $\mathsf{dist}(\mathcal{C}_i, \mathcal{C}_c) \leq t$, and compute $\overline{r_i} = \mathsf{Ext}_1(\mathcal{C}_i; r_{\mathcal{S}})$;
    3. Choose a random nonce $r_i' \in \mathbb{Z}_p$ and computes the token $r_c' = \mathtt{H}(ID_i || r_i' || \overline{r_i} || P_i)$;

Candidates:	$\|q\|$	n
Fingerprint [3, 29]	4-8	16-640
Face [33]	4-8	1024 - 16384

Fig. 2. General parameters.

4. Erase all state and send (r'_c, r'_i) to cloud server $\widehat{S}$.

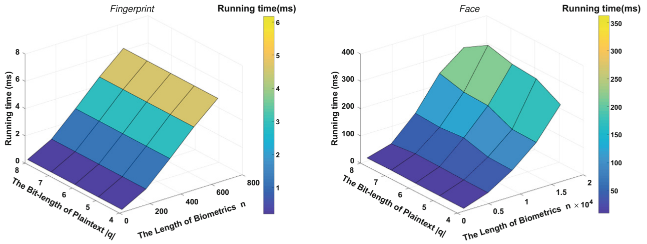
- Eventually, cloud server $\widehat{S}$ computes the token $r_c = \mathtt{H}(ID_i||r'_i||\overline{r_i}||P_i)$ and checks $r'_c \overset{?}{=} r_c$. If it does hold, accept; Otherwise, reject.

A Trivial Attack. We notice that both user authenticity and biometric privacy may suffer to brute force attacks. For instance, an adversary may choose a random candidate biometrics $\mathcal{C}^* \in \{0,1\}^{n|q|}$ for remote user authentication. More formally, adversary wins the user authenticity game with probability $(C^0_{n|q|} + C^1_{n|q|} + \cdots C^t_{n|q|})/2^{n|q|}$ (C denote the combinatorial number system in the form of $C^n_m = m!/n!(m-n)!$), which is negligible in terms of security parameters.

4.2 Performance Analysis

This experiment was run on virtual machines (3.6 GHz single-core processor and 6 GB RAM memory). In this experiment, we use Fingerprint and Face as candidates biometrics to initialize biometric-based remote user authentication scheme (BUA) (see Fig. 2). The experiment assumes that user biometric data has been converted into the format needed (we focus on real number strings here because the input requirement of fuzzy extractor [23]), because the representation (depends on the feature extraction algorithms) of biometric data could be vary. Without loss of generality, we use simulated data which is independent from various type of biometrics. We analyze the BUA in terms of computation cost and communication overhead, and we assume an identity has 256-bit size, a hash function SHA-256 has 256-bit output size, and the helper data of fuzzy extractor includes a secure sketch with $n \cdot log(k \cdot a + 1)$-bit output size (Refer to [23] for detailed description of parameters, such as t, k, a).

- Fingerprint 3a: Typically, the bit length of FingerCode (Refer to [19]) is ranging from 64 bits to 5120 bits. Specifically, a proper fingerprint has the following parameters: 2–5 concentric bands, 4–16 sectors, 2–8 Gabor filters, quantised with 4–8 bits and stored with five different orientations [3, 19]. Note that there are two main factors that affect the computation cost: (1) Length of b_i (4–8 bits); (2) Dimension of FingerCode n (16–640).
  From Fig. 3a, we can see that the running time increases linearly with whole size of bit length because the computational cost of fuzzy extractor and XOR operation are relying on the actual size of biometrics. Furthermore, we take $b_i = 4$ and $n = 640$ as a sample FingerCode, it requires about 6.16 ms for efficient computation (w.r.t. authentication) on-line. If we assume $p = 256$, then server and user has 703-bit and 768-bit communication overhead respectively. Note that the output size of secure sketch is $447 \approx 640 \cdot log(4+1)$ bits.



(a) Running Time (Fingerprint) (b) Running Time (Face)

Fig. 3. Evaluation findings

- Face 3b: An image pixel is usually quantised to store from 4-bit to 8-bit length, and the size of image is ranging from 32 × 32 to 128 × 128 with respect to grayscale image. Note that 32×32 is the minimal recognised value of a grayscale image, and 128×128 is a most used image size according to the experimental results (see Table 3 in [33]). From Fig. 3b, the running time also increases linearly with whole size of bit length (the same reason as explained above). We then take $b_i = 4$ and $n = 16384$ as a sample of face recognition, it requires about 277.46 ms for efficient computation (w.r.t. authentication) on-line. Furthermore, server and user has 11707-bit and 768-bit communication overhead respectively, and in particular, the user' communication overhead is a constant value. Note that the output size of secure sketch is $11451 \approx 16384 \cdot log(4+1)$ bits.

Remark. Note that some types of biometric data such as iris or an audio recording of a voice, are typically quantised in the binary format [11,32] which can also be processed using above fuzzy extractor. The reason is that, the input of fuzzy extractor [23] is actually a ciphertext, which means any specified format (such as binary, integer, vector and matrix) will be transformed into a real random string using XOR operation (recall that $b_i \in \mathbb{Z}_q$).

4.3 Efficiency Analysis

We then present an efficiency comparison among relevant lightweight biometric-based and fuzzy-extractor based user authentication and identification schemes in terms of storage costs and computational costs. We consider a two-party (namely, user and server) setting only for fair comparison.

- Storage cost: Let $\mathcal{L}_{\mathcal{B}}$ denote the length of biometrics $\mathcal{B}$ (e.g., $|q|n$); $\mathcal{L}_{\mathbb{Z}_q}$ denote the length of element in $\mathbb{Z}_q$. In Table 3, user's storage cost (such as encryption key or randomness) in our proposed solution is less than [23] since cloud server stores the encrypted biometrics and the corresponding helper data, and user does not need to run Gen algorithm during authentication phase. As for the basic scheme in [29], it requires more storage due to two diagonal matrixes are replying on flexible dimension of biometrics. Thus our proposed generic construction has less storage cost than [23,29] from user's perspective.

Table 3. Storage costs in various schemes.

Schemes	Public/secret key (user)	Stored info (server)
[29]	$(\mathcal{L}_{\mathcal{B}} + 2)^2$	$(\mathcal{L}_{\mathcal{B}} + 2)^2$
[23]	$\mathcal{L}_{\mathcal{B}} + \mathcal{L}_{\mathbb{Z}_q}$	$\mathcal{L}_{\mathcal{B}}+2\mathcal{L}_{\mathbb{Z}_q}$
LR-BUA	$2\mathcal{L}_{\mathbb{Z}_q}$	$\mathcal{L}_{\mathcal{B}}+ 3\mathcal{L}_{\mathbb{Z}_q}$

- Computational cost: Let T_{Mul} denote the multiplication operation; T_{Ext} denote the fuzzy extractor; $T_{\mathsf{Ext}'}$ denote the strong extractor (non-fuzzy case); T_{KG} denote the key generation algorithm; T_{Enc} denote the encryption scheme; T_{Sign} denote the digital signature scheme; $T_{\mathtt{H}}$ denote the hash function. In Table 4, user's computational cost of our proposed construction at registration phase is larger than [23] since additional encryption Enc algorithm is required for biometrics privacy and Ext' is required for preventing leakage attacks. However, user has less computational cost than [23] during authentication phase. Specifically, user may perform lightweight Enc algorithm as above instantiation described, when it compared to the Sign algorithm in [23]. Furthermore, the computational cost of our proposed construction and [23] are linear, while [29] requires cubic growth of computational cost which is relying on the dimensional of biometrics. According to the performance analysis, we can infer that the computational cost in [29] is more efficient than [23] and LR-BUA at low-dimensional (of biometrics) case, but it performs worse compared to [23] and LR-BUA at high-dimensional case.

Table 4. Computational costs in various schemes.

Schemes	Registration	Authentication
[29]	$\mathcal{O}(\mathcal{B}^3)[T_{Mul}]$	$\mathcal{O}(\mathcal{B}^3)[T_{Mul}]$
[23]	$\mathcal{O}(\mathcal{B})[T_{\mathsf{Ext}} + T_{\mathsf{KG}}]$	$\mathcal{O}(\mathcal{B})[T_{\mathsf{Ext}} + T_{\mathsf{KG}} + T_{\mathsf{Sign}}]$
LR-BUA	$\mathcal{O}(\mathcal{B})[T_{\mathsf{Ext}'} + T_{\mathsf{Enc}}]$	$\mathcal{O}(\mathcal{B})[T_{\mathsf{Ext}} + T_{\mathsf{Ext}'} + T_{\mathsf{Enc}} + T_{\mathtt{H}}]$

5 Conclusion

In this paper, we proposed a notion of leakage-resilient biometric-based remote user authentication and its generic framework, and a lightweight instantiation with overall efficiency analysis. We also defined the new formal security models for leakage-resilient user authenticity and biometrics privacy, and proved the security of the proposed generic construction under standard assumptions. We leave the construction of leakage-resilient and privacy-preserving biometric-based user authentication against impersonation attacks from multiple remote servers as our future work.

Acknowledgements. This work is supported by the Singapore National Research Foundation under NCR Award Number NRF2014NCR-NCR001-012, the National Natural Science Foundation of China (Grant No. 61702541,61702105), the Young Elite Scientists Sponsorship Program by CAST (Grant No. 2017QNRC001) and the Science Research Plan Program by NUDT (Grant No. ZK17-03-46).

References

1. Fido alliance (2017). https://fidoalliance.org
2. Atallah, M.J., Frikken, K.B., Goodrich, M.T., Tamassia, R.: Secure biometric authentication for weak computational devices. In: Patrick, A.S., Yung, M. (eds.) FC 2005. LNCS, vol. 3570, pp. 357–371. Springer, Heidelberg (2005). https://doi.org/10.1007/11507840_32
3. Barni, M., et al.: Privacy-preserving fingercode authentication. In: Proceedings of the 12th ACM Workshop on Multimedia and Security, pp. 231–240 (2010)
4. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48329-2_21
5. Boyen, X.: Reusable cryptographic fuzzy extractors. In: ACM CCS, pp. 82–91 (2004)
6. Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., Smith, A.: Secure remote authentication using biometric data. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 147–163. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_9
7. Bringer, J., Chabanne, H., Izabachène, M., Pointcheval, D., Tang, Q., Zimmer, S.: An application of the goldwasser-micali cryptosystem to biometric authentication. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) ACISP 2007. LNCS, vol. 4586, pp. 96–106. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73458-1_8
8. Canetti, R., Fuller, B., Paneth, O., Reyzin, L., Smith, A.: Reusable fuzzy extractors for low-entropy distributions. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 117–146. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_5
9. Castiglione, A., Choo, K.-K.R., Nappi, M., Narducci, F.: Biometrics in the cloud: challenges and research opportunities. IEEE Cloud Comput. **4**(4), 12–17 (2017)
10. Chen, R., Mu, Y., Yang, G., Susilo, W., Guo, F.: Strongly leakage-resilient authenticated key exchange. In: Sako, K. (ed.) CT-RSA 2016. LNCS, vol. 9610, pp. 19–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29485-8_2
11. Daugman, J.: How iris recognition works. In: The Essential Guide to Image Processing, pp. 715–739 (2009)
12. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: STOC, pp. 621–630 (2009)
13. Dodis, Y., Kanukurthi, B., Katz, J., Reyzin, L., Smith, A.: Robust fuzzy extractors and authenticated key agreement from close secrets. IEEE Trans. Inf. Theory **58**(9), 6207–6222 (2012)

14. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008)
15. Fan, C., Lin, Y.: Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics. IEEE Trans. Inf. Forensics Secur. **4**(4), 933–945 (2009)
16. Huang, X., Xiang, Y., Bertino, E., Zhou, J., Xu, L.: Robust multi-factor authentication for fragile communications. IEEE Trans. Dependable Secur. Comput. **11**(6), 568–581 (2014)
17. Huang, X., Xiang, Y., Chonka, A., Zhou, J., Deng, R.H.: A generic framework for three-factor authentication: preserving security and privacy in distributed systems. IEEE Trans. Parallel Distrib. Syst. **22**(8), 1390–1397 (2011)
18. Huang, Y., Malka, L., Evans, D., Katz, J.: Efficient privacy-preserving biometric identification. In: NDSS (2011)
19. Jain, A.K., Prabhakar, S., Hong, L., Pankanti, S.: Fingercode: a filterbank for fingerprint representation and matching. In: 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 187–193 (1999)
20. Juels, A., Sudan, M.: A fuzzy vault scheme. Des. Codes Cryptogr. **38**(2), 237–257 (2006)
21. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: ACM CCS, pp. 28–36 (1999)
22. Kanade, S.G., Petrovska-Delacrétaz, D., Dorizzi, B.: Enhancing Information Security and Privacy by Combining Biometrics with Cryptography. Synthesis Lectures on Information Security, Privacy, and Trust. Morgan & Claypool Publishers, San Rafael (2012)
23. Li, N., Guo, F., Mu, Y., Susilo, W., Nepal, S.: Fuzzy extractors for biometric identification. In: ICDCS, pp. 667–677 (2017)
24. Li, Y., Li, Y., Yan, Q., Kong, H., Deng, R.H.: Seeing your face is not enough: an inertial sensor-based vileness detection for face authentication. In: ACM CCS, pp. 1558–1569 (2015)
25. Micali, S., Reyzin, L.: Physically observable cryptography. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24638-1_16
26. Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_2
27. Schoenmakers, B., Tuyls, P.: Efficient binary conversion for paillier encrypted values. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 522–537. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_31
28. Tang, Q., Bringer, J., Chabanne, H., Pointcheval, D.: A formal study of the privacy concerns in biometric-based remote authentication schemes. In: Chen, L., Mu, Y., Susilo, W. (eds.) ISPEC 2008. LNCS, vol. 4991, pp. 56–70. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79104-1_5
29. Wang, Q., Hu, S., Ren, K., He, M., Du, M., Wang, Z.: CloudBI: practical privacy-preserving outsourcing of biometric identification in the cloud. In: Pernul, G., Ryan, P.Y.A., Weippl, E. (eds.) ESORICS 2015. LNCS, vol. 9327, pp. 186–205. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24177-7_10
30. Yang, G., Mu, Y., Susilo, W., Wong, D.S.: Leakage resilient authenticated key exchange secure in the auxiliary input model. In: Deng, R.H., Feng, T. (eds.) ISPEC 2013. LNCS, vol. 7863, pp. 204–217. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38033-4_15

31. Yuen, T.H., Zhang, Y., Yiu, S.M., Liu, J.K.: Identity-based encryption with post-challenge auxiliary inputs for secure cloud applications and sensor networks. In: Kutyłowski, M., Vaidya, J. (eds.) ESORICS 2014. LNCS, vol. 8712, pp. 130–147. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11203-9_8
32. Zhang, L., Tan, S., Yang, J., Chen, Y.: Voicelive: a phoneme localization based liveness detection for voice authentication on smartphones. In: ACM CCS, pp. 1080–1091 (2016)
33. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: a literature survey. ACM Comput. Surv. (CSUR) **35**(4), 399–458 (2003)