

Received September 18, 2017, accepted October 17, 2017, date of publication October 20, 2017, date of current version December 5, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2764913

On the Design of Provably Secure Lightweight Remote User Authentication Scheme for Mobile Cloud Computing Services

SANDIP ROY¹, SANTANU CHATTERJEE², ASHOK KUMAR DAS^{ID}³, (Member, IEEE), SAMIRAN CHATTOPADHYAY⁴, NEERAJ KUMAR⁵, (Senior Member, IEEE), AND ATHANASIOS V. VASILAKOS⁶

¹Department of Computer Science and Engineering, Asansol Engineering College, Asansol 713 305, India

²Research Center Imarat, Defence Research and Development Organization, Hyderabad 500 069, India

³Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India

⁴Department of Information Technology, Jadavpur University, Kolkata 700 098, India

⁵Department of Computer Science and Engineering, Thapar University, Patiala 147 004, India

⁶Lab of Networks and Cybersecurity, Innopolis University, 420500 Innopolis, Russia

Corresponding author: Ashok Kumar Das (iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in)

ABSTRACT Secure and efficient lightweight user authentication protocol for mobile cloud computing becomes a paramount concern due to the data sharing using Internet among the end users and mobile devices. Mutual authentication of a mobile user and cloud service provider is necessary for accessing of any cloud services. However, resource constraint nature of mobile devices makes this task more challenging. In this paper, we propose a new secure and lightweight mobile user authentication scheme for mobile cloud computing, based on cryptographic hash, bitwise XOR, and fuzzy extractor functions. Through informal security analysis and rigorous formal security analysis using random oracle model, it has been demonstrated that the proposed scheme is secure against possible well-known passive and active attacks and also provides user anonymity. Moreover, we provide formal security verification through ProVerif 1.93 simulation for the proposed scheme. Also, we have done authentication proof of our proposed scheme using the Burrows-Abadi-Needham logic. Since the proposed scheme does not exploit any resource constrained cryptosystem, it has the lowest computation cost in compare to existing related schemes. Furthermore, the proposed scheme does not involve registration center in the authentication process, for which it is having lowest communication cost compared with existing related schemes.

INDEX TERMS Remote mobile user authentication, distributed mobile cloud computing, user anonymity, user biometrics, random oracle, BAN logic, ProVerif simulation.

I. INTRODUCTION

Mobile Cloud Computing (MCC) provides cloud resources through on-demand basis by integrating cloud computing into mobile environment [1], [2]. Nowadays, both in industry as well as academia, mobile cloud computing has drawn much attention. A recent analysis done by Heavy Reading estimates that, by the end of 2017, mobile cloud computing market will generate around 68 U.S. billion dollars of direct revenue [3]. Reports from different sources like ABI research [4] predict that, number of worldwide mobile cloud computing users have exploded rapidly, from 42.8 million users on 2008 to 998 million users in 2014 [2].

IT organizations are now increasingly using various cloud computing softwares, infrastructures (like Gmail,

Facebook etc.) and frameworks (like Google AppEngine, Amazon web service etc.). Cloud computing are getting popular to IT developers and users day by day. On the other side, worldwide deployment and development of various smartphone applications are also increasing exponentially. Rapid development and implementation of many IT services in mobile cloud computing necessitates extensive research on security issues [5]–[9].

An architecture for distributed mobile cloud computing is represented in Fig. 1. To access a mobile cloud computing service, a mobile user MU_i requests the cloud service through an installed mobile App or web browser. After that a mutual authentication between MU_i and the cloud service provider CS_j is done by the user mobile App or web browser [10].

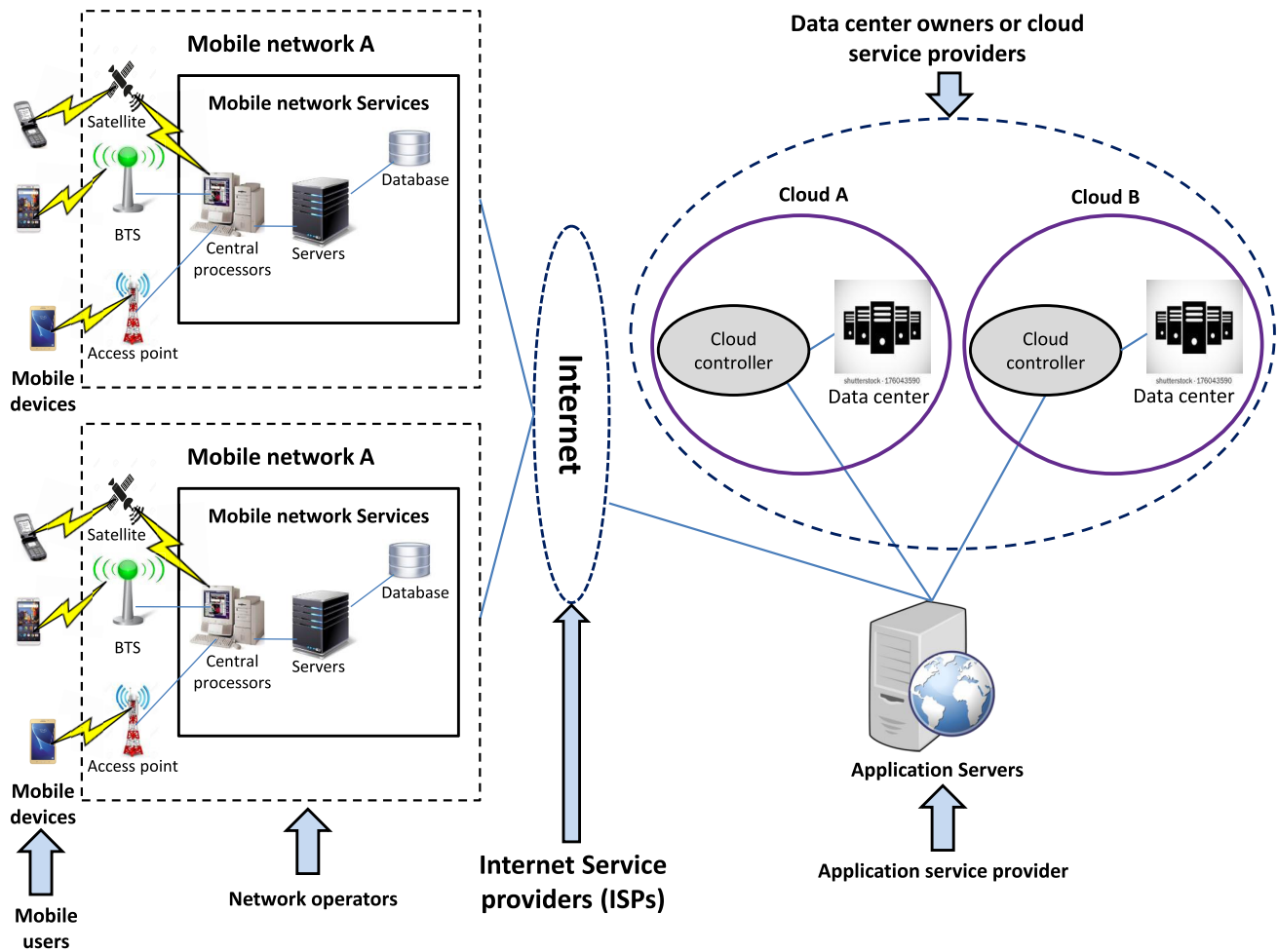


FIGURE 1. An architecture for distributed mobile cloud computing.

Both MU_i and CS_j need to go through a secure mutual authentication process that should support some basic requirements. These include computation efficiency, user anonymity, session key security etc. in order to prevent various threats over insecure channel.

Intrinsically, mobile cloud computing services are quite distributed and heterogeneous in nature. Thus, registering separately for each cloud service provider by maintaining respective user account is almost an impossible task. To be precise, MU_i requires to access several cloud services from CS_j with the help of single registered user account. But, traditional two-party single server authentication scheme cannot be directly applicable for a multiple mobile cloud servers environment. Single sign-on (SSO) scheme requires single credential and single registration for accessing multiple mobile cloud computing services [11], [12]. Three separate parties, namely, mobile user, cloud service provider/cloud server and trusted identity provider (IdP) (registration center (RC) or smart card generator (SCG)) are involved in SSO mechanism. However, SSO authentication can be done both

with involving IdP/SCG/RC [13], [14] and without involving IdP/SCG/RC [15], [16].

Currently, many internet service providers and websites are using OpenId to implement distributed SSO techniques. In this case, both user and service provider need to register to IdP in advance. During login, mobile user sends the adopted OpenId to cloud service provider, which in turn redirects it to IdP for verification of user authenticity. This technique has two major issues. First, over involvement of IdP might turn into a bottleneck for the conventional SSO scheme. Second, OpenId scheme requires message communication through SSL network connection. Unfortunately, SSL based schemes need high computation costs as SSL technique is primarily based on public key cryptosystem like RSA [10], [17].

Wang and Wang [18] first attempted to explore the underlying rationales for preserving user privacy property in two-factor authentication schemes. They pointed out that without any public-key techniques, it is quite impossible to come up with a privacy-preserving scheme using only lightweight cryptographic primitives, such as one-way

cryptographic hash functions. Wang and Wang [19] also presented some security failures of previous user authentication schemes. Three important suggestions were made by them [19] while analyzing those user authentication schemes: 1) user anonymity preservation with the help of public key techniques, 2) application of fuzzy-verifier to have trade-off between usability and security, and 3) privileged insider attack protection using salt values (i.e., random nonces). Wang *et al.* [20] suggested some evaluation metrics for designing anonymous two-factor user authentication and also pointed out how to make an acceptable trade-off among usability, security and privacy. Huang *et al.* [21] also suggested that design of password-based authentication schemes using smart cards requires the importance to elaborate security models along with the formal security analysis. Huang *et al.* [22] also proposed a generic multi-factor authentication scheme that uses user password, smart-card and biometrics as three factors, and observed that a stand-alone authentication in which a user can be authenticated correctly, even if the connection to the remote server is down. Ma *et al.* [23] highlighted three principles for designing more robust user authentication schemes. Huang *et al.* [24] investigated a systematic approach for authenticating clients using three factors: 1) password, 2) smart card and 3) biometrics. They proposed a generic and secure model to upgrade a two-factor authentication scheme to a three-factor authentication scheme. It is interesting to observe that their conversion not only significantly improved the information assurance at low cost but also protected client privacy in distributed systems. Wang and Wang [25] presented a proposal of a new authentication scheme. It meets simplicity, practicability, and strong notions. In addition, they provided an adversary model and criteria set which can provide a benchmark for the evaluation of current and future two-factor authentication schemes. Moreover, Wang *et al.* [26] pointed out that there are at least seven different attacking scenarios and those attack scenarios may lead to the failure of an authentication scheme in arriving truly two-factor security. They also conducted a large-scale comparative evaluation of 26 representative two-factor schemes, and their results synopsis the request for better measurement when assessing new authentication schemes.

In a distributed mobile cloud computing environment, user mobile devices are quite resource constrained in nature and SSO technique does not provide a practical solution. Recently, Odelu *et al.* [27], and Gope and Das [28] proposed secure authentication schemes for mobile cloud computing. Odelu *et al.* [27] exploits elliptic curve cryptography (ECC) point multiplication and asymmetric bilinear pairing operation for authentication and key establishment phase, while Gope-Das's scheme [28] is based on hash chain method. In Gope-Das's scheme [28], a mobile subscriber is allowed to obtain the ubiquitous services only up to a specific time-period (say n -times), where the access duration strictly depends on the principle that the cloud user has paid for services. In this paper, we aim to design a novel secure and

efficient scheme based on only lightweight computations, such as cryptographic hash function computation and bit wise XOR operation, while the biometric verification is done by the mobile user using the widely-used fuzzy extractor method.

In short, a mobile user authentication scheme for distributed mobile cloud computing environment should have the following properties:

- 1) A trusted third party (i.e., registration center (RC) or identity provider (IdP) or smart card generator (SCG)) should not be involved during user login process. However, during registration phase, both MU_i and CS_j should register to trusted third party only once.
- 2) A mobile user must avoid multiple credentials, such as identity and/or password for accessing different mobile cloud services.
- 3) The authentication process should avoid computationally costly operation in user mobile device. Also, storage requirement in user mobile should be less.
- 4) Mutual authentication between a mobile user and a cloud server CS_j should use lightweight cryptographic operations.

A. MOTIVATION

The following basic motivating factors are behind the proposal of our scheme in this paper:

- 1) As user's mobile device generally operates through battery limited equipments, mobile user authentication mechanism should consume minimum possible computation, communication and storage costs. However, existing authentication schemes for mobile cloud computing environments, mostly based on resource consuming cryptosystems, such as bilinear pairing [10], [29] and ECC [30]–[32]. This necessitates the design of an efficient mobile user authentication scheme that could avoid such cryptosystems without degrading overall security of the system.
- 2) A careful study on the existing authentication schemes under mobile cloud computing environment reveals most of those schemes have security flaws. Hence, design of more secure authentication scheme is needed in this domain.
- 3) Further, several schemes, such as the schemes of Shen *et al.* [31], Yoon and Yoo [30], and He and Wang [32] involve the RC or IdP or SCG in mobile user login process which results in more communication and computation costs.

B. THREAT MODEL

In this paper, we follow the widely-accepted Dolev-Yao threat model (DY model) [33], which accepts the following basic assumptions:

- The messages are communicated over a public insecure channel.

- The public channel messages are susceptible to eavesdropping, deletion or modification, which are executed by adversary \mathcal{A} .
- If, by any means, the adversary \mathcal{A} obtains legal user's smart card or mobile device, he/she can execute power analysis attack and also extract all stored information from the device [34].

C. RESEARCH CONTRIBUTIONS

The following contributions are made in this paper:

- 1) The proposed scheme provides mobile user authentication in distributed mobile cloud computing environment, which supports secure key exchange, and user anonymity and untraceability properties.
- 2) Compared with the related existing authentication schemes proposed in the mobile cloud computing environment, the proposed scheme has the lowest computation and storage requirements. This is primarily due to usage of efficient one way cryptographic hash function, bitwise XOR operation and fuzzy extractor operation only.
- 3) No trusted third party, like IdP, SCG or RC, is involved in user login and authentication phases. This reduces overall communication and computation time of the proposed scheme.
- 4) The proposed scheme is lightweight in nature, and meanwhile, it also removes the security and functionality drawbacks of the earlier existing schemes.
- 5) The proposed scheme has the ability to resist various known attacks, which are evident through the rigorous formal security proof through random oracle model and BAN logic, the formal security verification using the ProVerif 1.93 simulation tool as well as through informal security analysis.

D. PAPER ORGANIZATION

Section II provides the mathematical preliminaries in brief, which are necessary to describe and analyze the proposed scheme. Section III presents the proposed mobile user authentication scheme for the distributed mobile cloud computing environment. Section IV provides the detailed formal and informal security analysis of the proposed scheme, while Section V presents formal security verification using ProVerif 1.93 simulation tool. Section VI presents the performance comparison of the proposed scheme with other related existing schemes. At last, Section VII provides the conclusion of the paper.

II. MATHEMATICAL PRELIMINARIES

To design the proposed scheme, we use cryptographic hash function [35], bitwise XOR operation and fuzzy extractor technique over biometrics input. This section presents a brief introduction about these basic mathematical preliminaries.

A. COLLISION-RESISTANT ONE-WAY HASH FUNCTION

A one-way hash function $H: \{0, 1\}^* \rightarrow \{0, 1\}^l$ receives a binary string of variable length, say $x \in \{0, 1\}^*$ and then produces a binary string $H(x) \in \{0, 1\}^l$ as an output of fixed length, say l bits. The hash function can be formally defined as follows [36].

Definition 1: An adversary \mathcal{A} 's advantage in finding collision with the execution time et is denoted and defined by $Adv_{\mathcal{A}}^{HASH}(et) = Pr[(u, v) \leftarrow_R \mathcal{A}: u \neq v \text{ and } H(u) = H(v)]$, where $Pr[X]$ is an event X 's probability, and $(u, v) \leftarrow_R \mathcal{A}$ indicates that the pair (u, v) is randomly selected by \mathcal{A} . By an (ϵ, et) -adversary \mathcal{A} attacking the collision resistance of $H(\cdot)$, it means that the execution time of \mathcal{A} is at most et and that $Adv_{\mathcal{A}}^{HASH}(et) \leq \epsilon$.

Examples of a one-way hash function include the Secure Hash Standard (SHA-1) hashing algorithm and the stronger SHA-256 hashing algorithm [37].

B. FUZZY EXTRACTOR AND BIOMETRICS

Nowadays, biometric keys are popularly used in various authentication process. The basic features and advantages of biometric keys can be found in [38]–[40].

Fuzzy extractor is a technique that generates the same output string, even if the input biometric differs from recorded biometric sample upto a permissible error tolerance threshold limit. In general, a fuzzy extractor exploits two functions: 1) *Generation*(\cdot) and 2) *Reproduction*(\cdot), which are probabilistic and deterministic in nature, respectively.

Definition 2: Considering biometric key of length l bits generated from the given biometrics \mathcal{B} and $S = \{0, 1\}^v$ being a metric space of finite dimensional biometric data points, the following functions are defined as follows.

- *Generation: It takes $\mathcal{B} \in S$ as input and produces a pair (θ, ϕ) , where $\theta \in \{0, 1\}^l$ is the biometric key and ϕ is the public reproduction parameter.*
- *Reproduction: It recovers the original biometric key $\theta \in \{0, 1\}^l$ on biometrics \mathcal{B}' provided that the Hamming distance between \mathcal{B}' and the original biometrics \mathcal{B} at the time of registration by the same user is less than or equal to t , where t is a pre-defined error tolerance threshold parameter; that is, $\theta = \text{Reproduction}(\mathcal{B}', \phi)$.*

More discussion on fuzzy extractor method is available in [41]. Also, in recent years, the fuzzy extractor becomes popular as it is applied for biometric authentication purpose in several authentication protocols [41]–[44].

III. THE PROPOSED SCHEME

In this section, we describe various phases related to the proposed scheme.

The proposed scheme is based on the basic assumption that the distributed mobile cloud computing environment has three basic entities: 1) mobile users, 2) cloud server or cloud service provider, and 3) trusted registration center (RC). The system contains a set of m legal mobile users, $M = \{MU_i | i = 1, 2, \dots, m\}$, a set of n cloud servers,

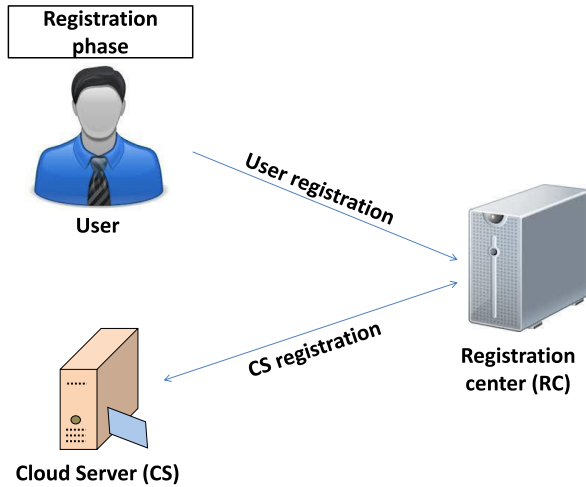


FIGURE 2. Framework of the proposed scheme (registration phase).

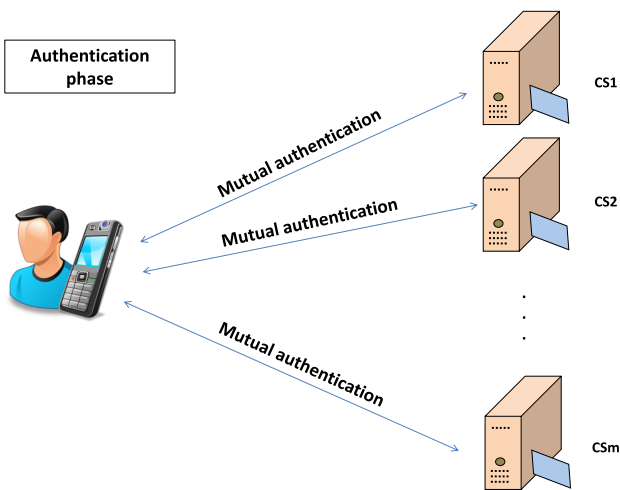


FIGURE 3. Framework of the proposed scheme (authentication phase).

$N = \{CS_j | j = 1, 2, \dots, n\}$ and the trusted RC . A legal user or an unregistered external person may execute malicious activities in the system, called as an adversary \mathcal{A} . From different cloud service providers, a mobile user can access multiple mobile cloud computing services. The RC needs not to involve in the login and authentication processes. Fig. 2 and 3 present the framework of the proposed scheme.

The proposed scheme is composed of five phases, namely, 1) registration, 2) login, 3) authentication and key establishment, 4) password change, and 5) revocation of mobile device. The registration phase is composed of mobile user registration phase as well as cloud server registration phase. During the registration phase, the mobile users and cloud servers register to the RC independently. The RC generates the master secret keys randomly for the registered servers, and also generates mutual secret keys between respective mobile users and cloud servers. The login phase receives the user’s credentials and verifies his/her authenticity. During the authentication phase, the mobile user and cloud server

TABLE 1. Notations used in the proposed scheme.

Symbol	Description
RC	Registration center
MU_i	i^{th} mobile user
CS_j	j^{th} cloud service provider
ID_i	Identity of MU_i
ID_{S_j}	Identity of CS_j
r_{ij}	1024-bit random number selected by RC for MU_i and CS_j
b	128-bit random number chosen by MU_i
X_j	1024-bit master secret key of server CS_j
SN_i	Serial number of MU_i 's mobile device
$H(\cdot)$	One-way cryptographic hash function
\parallel, \oplus	Concatenation, bitwise XOR operations
TS_i	Timestamp generated by MU_i
TS_j	Timestamp generated by CS_j
RN_i	128-bit MU_i 's random number
RN_j	128-bit CS_j 's random number
$A \xrightarrow{\langle M \rangle} B$	Message (M) transmission from A to B
ΔT	Maximum transmission delay

authenticate each other and mutually generate the secret shared session key. The password change phase gives the flexibility to MU_i in order to locally update old password into new password at any time for security reasons.

The basic notations listed in Table 1 are used to design the proposed scheme. The proposed scheme makes use of the current system timestamps along with the random nonces to protect strong replay attacks. To achieve this goal, we assume that all the entities (mobile users, cloud service providers and the RC) in the network are synchronized with their clocks. This is a reasonable assumption as it is also applied in designing many authentication protocols proposed recently [27], [41]–[46].

A. REGISTRATION PHASE

In this phase, both mobile users and cloud servers register to the registration center independently. This phase is composed of two sub-phases: 1. mobile user registration phase and 2. cloud server registration phase. Both the phases are executed only once and messages are communicated through secure channel (for example, in person).

1) MOBILE USER REGISTRATION PHASE

A mobile user MU_i registers to the RC through the following steps:

- 1) MU_i chooses his/her own identity ID_i , password PW_i , biometrics \mathcal{B}_i , two 128-bit random numbers b and k .
- 2) MU_i produces $(\theta_i, \phi_i) = \text{Generation}(\mathcal{B}_i)$ and computes the masked password $RPWB_i = H(ID_i \parallel H(PW_i \parallel \theta_i \parallel b))$. MU_i then submits the registration request message $\{ID_i, (RPWB_i \oplus k)\}$ to the RC via secure channel.
- 3) The RC selects an 1024-bit master secret key X_j for server CS_j . RC also selects an 1024-bit random number r_{ij} for each MU_i and CS_j pair. Further, RC computes

$A_{ij} = H(H(ID_i \oplus r_{ij}) || X_j)$, $V_{ij} = A_{ij} \oplus RPWB_i$ and the pseudo-identity of CS_j as $RID_{S_j} = H(ID_{S_j} || X_j)$.

- 4) In order to maintain user anonymity, instead of actual identity ID_i , RC selects a unique and random temporary identity TID_i for MU_i .
- 5) The RC saves n server key-plus-id combinations $\{TID_i, (ID_{S_j}, V_{ij}, RID_{S_j}) \mid 1 \leq j \leq n\}$ in mobile device of MU_i and delivers the mobile device to MU_i securely.
- 6) MU_i computes $D_i^1 = H(PW_i || \theta_i) \oplus b$ and $D_i^2 = H(ID_i || PW_i || \theta_i || b)$, and $V'_{ij} = V_{ij} \oplus k = A_{ij} \oplus H(ID_i || H(PW_i || \theta_i || b))$, $RID'_{ij} = TID_i \oplus H(ID_i || V'_{ij})$ and $RID'_{S_j} = RID_{S_j} \oplus H(\theta_i || b)$ for $1 \leq j \leq n$. Finally, MU_i also stores $\phi_i, D_i^1, D_i^2, V'_{ij}$'s, RID'_{ij} 's and RID'_{S_j} 's into his/her own mobile device, and deletes V_{ij} 's, TID_i and RID_{S_j} 's from the mobile device.

2) CLOUD SERVER REGISTRATION PHASE

In order to register to the RC , a new cloud server must execute the following steps:

- 1) The cloud server (cloud service provider) CS_j sends its identity ID_{S_j} to the RC through a secure channel.
- 2) The RC provides the master secret key X_j to each CS_j .
- 3) For all MU_i 's, the RC saves the credentials $\{TID_i, (ID_i, r_{ij})\}$ in database of CS_j .
- 4) The RC also stores $\{ID_{S_j}, X_j\}$ in the database of CS_j .

Finally, the RC also saves pair (ID_i, SN_i) in its own database, where SN_i is the serial number of MU_i 's mobile device.

Figure 4 shows the fundamental steps of user and server registration phase.

Remark 1: In the proposed scheme, a mobile user MU_i needs to store all the credentials in his/her mobile device. For example, if the mobile user MU_i wants to access 100 cloud service providers CS_j 's, his/her mobile device needs to store 100 credentials (i.e., keys). Note that in the proposed scheme, instead of using a smart card, a mobile device is used. Since the mobile device is resource-rich device as compared to resource-constrained smart device, the storage space in MU_i 's mobile device is not an issue. Hence, storing more credentials in MU_i 's mobile device is not a problem in the proposed scheme.

B. LOGIN PHASE

This phase describes how a legal mobile user MU_i logs into the CS_j . Fig. 5 shows the basic steps of login and authentication-key establishment phases. The following steps are essential to complete the login phase:

- 1) MU_i inputs his/her identity ID_i , password PW_i and personal biometrics \mathcal{B}'_i into his/her own mobile device. Using the fuzzy extractor reproduction procedure and stored ϕ_i , MU_i computes $\theta_i = \text{Reproduction}(\mathcal{B}'_i, \phi_i)$. Moreover, using the stored parameter D_i^1 , MU_i generates $b' = D_i^1 \oplus H(PW_i || \theta_i)$.
- 2) MU_i then computes $H(ID_i || PW_i || \theta_i || b')$ and checks if $D_i^2 = H(ID_i || PW_i || \theta_i || b')$ is true or not. MU_i proceeds to the next step only if this verification holds.

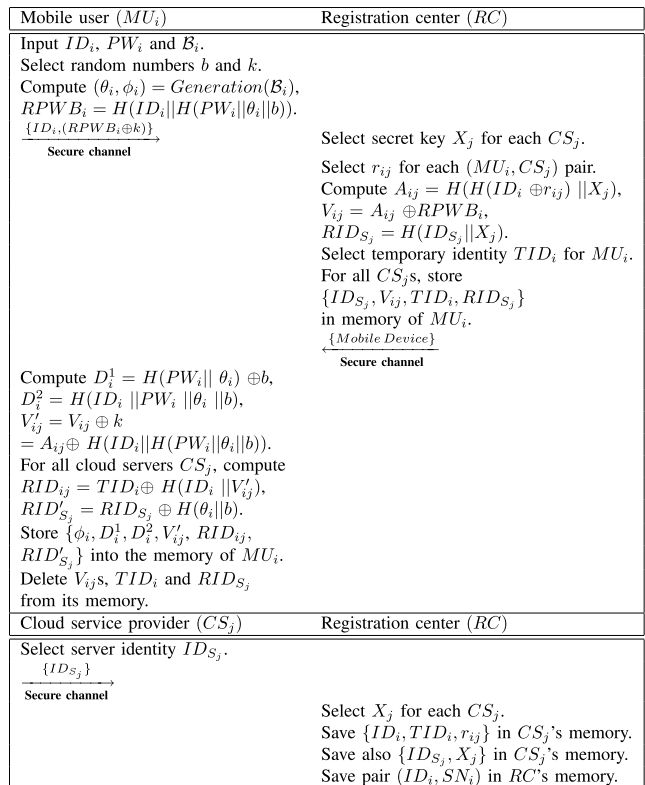


FIGURE 4. User and server registration phases of the proposed scheme.

- 3) MU_i calculates $RPWB_i = H(ID_i || H(PW_i || \theta_i || b'))$. Using the mobile device parameter V'_{ij} , MU_i also generates $A_{ij} = V'_{ij} \oplus RPWB_i$. In addition, MU_i selects an 128 bit random number RN_i , generates the current timestamp TS_i , and then computes

$$C_1 = A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j}) \\ = H(H(ID_i \oplus r_{ij}) || X_j) \oplus RN_i \\ \oplus TS_i \oplus H(ID_{S_j}),$$

$$H_1 = H(ID_i || C_1 || RN_i || TS_i),$$

$$TID_i = RID'_{ij} \oplus H(ID_i || V'_{ij}),$$

$$RID_{S_j} = RID'_{S_j} \oplus H(\theta_i || b'),$$

$$TID_i^* = TID_i \oplus H(RID_{S_j} || TS_i).$$

- 4) Finally, MU_i sends login request $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ to CS_j via a public channel.

Remark 2: The current timestamp TS_i in a particular session is used to make TID_i^ as dynamic, because $TID_i^* = TID_i \oplus H(RID_{S_j} || TS_i)$. In addition, even if an adversary A eavesdrops $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ and gets TID_i^* , it is computationally infeasible problem for A to know TID_i from TID_i^* without having the permanent secret RID_{S_j} ($= H(ID_{S_j} || X_j)$) as it involves the secret key X_j of CS_j . Suppose the same mobile user MU_i sends the login message $Msg'_1 = \{TID_i^{**}, C'_1, H'_1, TS'_1\}$ to CS_j in another session, where $TID_i^{**} = TID_i \oplus H(RID_{S_j} || TS'_1)$ and TS'_1 is the current timestamp generated by MU_i in that session. In this case,*

Mobile user (MU_i)	Cloud service provider (CS_j)
Login phase	
Input ID_i, PW_i and \mathcal{B}'_i . Compute $\theta_i = \text{Reproduction}(\mathcal{B}'_i, \phi_i)$, $b' = D_i^1 \oplus H(PW_i \theta_i)$. Verifies if $D_i^2 = H(ID_i PW_i \theta_i b')$? If verification holds, compute $RPWB_i = H(ID_i H(PW_i \theta_i b'))$, $A_{ij} = V'_{ij} \oplus RPWB_i$. Generate RN_i . Compute $C_1 = A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j})$, $H_1 = H(ID_i C_1 RN_i TS_i)$, $TID_i = RID_{ij} \oplus H(ID_i V'_{ij})$, $RID_{S_j} = RID'_{S_j} \oplus H(\theta_i b')$, $TID_i^* = TID_i \oplus H(RID_{S_j} TS_i)$. $\{TID_i^*, C_1, H_1, TS_i\}$ $\xrightarrow{\text{public channel}}$	
Authentication phase	
Verify if $ TS_i^* - TS_j \leq \Delta T$? Compute $RID_{S_j} = H(ID_{S_j} X_j)$, $TID_i = TID_i^* \oplus H(RID_{S_j} TS_i)$. Find the record $\langle ID_i, TID_i, r_{ij} \rangle$ from its database. Compute $B_{ji} = H(H(ID_i \oplus r_{ij}) X_j)$. $M_1 = C_1 \oplus TS_i \oplus H(ID_{S_j}) \oplus B_{ji}$. $= RN_i$, as $A_{ij} = B_{ji} = H(H(ID_i \oplus r_{ij}) X_j)$, $H_2 = H(ID_i C_1 M_1 TS_i)$. Verify if $H_2 = H_1$? If verification holds, generate RN_j . Compute $C_2 = B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i$, $SK_{CS_j, MU_i} = H(ID_i ID_{S_j} B_{ji} M_1 RN_j TS_i TS_j)$, $H_3 = H(ID_i M_1 RN_j TS_i TS_j SK_{CS_j, MU_i})$. $\{C_2, H_3, TS_j\}$ $\xrightarrow{\text{public channel}}$	Verify if $ TS_i^* - TS_j \leq \Delta T$? Compute $RID_{S_j} = H(ID_{S_j} X_j)$, $TID_i = TID_i^* \oplus H(RID_{S_j} TS_i)$. Find the record $\langle ID_i, TID_i, r_{ij} \rangle$ from its database. Compute $B_{ji} = H(H(ID_i \oplus r_{ij}) X_j)$. $M_1 = C_1 \oplus TS_i \oplus H(ID_{S_j}) \oplus B_{ji}$. $= RN_i$, as $A_{ij} = B_{ji} = H(H(ID_i \oplus r_{ij}) X_j)$, $H_2 = H(ID_i C_1 M_1 TS_i)$. Verify if $H_2 = H_1$? If verification holds, generate RN_j . Compute $C_2 = B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i$, $SK_{CS_j, MU_i} = H(ID_i ID_{S_j} B_{ji} M_1 RN_j TS_i TS_j)$, $H_3 = H(ID_i M_1 RN_j TS_i TS_j SK_{CS_j, MU_i})$. $\{C_2, H_3, TS_j\}$ $\xrightarrow{\text{public channel}}$
Verify if $ TS_j^* - TS_j \leq \Delta T$? Compute $M_2 = C_2 \oplus TS_j \oplus ID_i \oplus A_{ij}$, $= RN_j$, as $A_{ij} = B_{ji} = H(H(ID_i \oplus r_{ij}) X_j)$. $SK_{MU_i, CS_j} = H(ID_i ID_{S_j} A_{ij} RN_i M_2 TS_i TS_j)$, $H_4 = H(ID_i RN_i M_2 TS_i TS_j SK_{MU_i, CS_j})$. Verify if $H_4 = H_3$? If verification holds, store session key $SK_{MU_i, CS_j} (= SK_{CS_j, MU_i})$.	Store session key $SK_{CS_j, MU_i} (= SK_{MU_i, CS_j})$.

FIGURE 5. Login and authentication phases of the proposed scheme.

A can not also derive TID_i from TID_i^{**} without having the permanent secret RID_{S_j} of CS_j . It is also observed that both TID_i^* and TID_i^{**} are distinct due to involvement of RID_{S_j} and current timestamps. This clearly shows that the user anonymity is completely preserved in the proposed scheme as the real identity ID_i as well as the temporary identity TID_i of MU_i are not revealed to the adversary \mathcal{A} .

C. AUTHENTICATION AND KEY ESTABLISHMENT PHASE

In this phase, CS_j and MU_i mutually authenticate each other. After successful authentication, MU_i and CS_j establish a secret session key for data communication in the current session. This phase involves the following steps:

- 1) After receiving the login request message $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ from MU_i , CS_j verifies if $|TS_i^* - TS_j| \leq \Delta T$, where TS_i^* is the actual received time of the message Msg_1 and ΔT is the maximum transmission delay. If this verification fails, CS_j rejects the

login request immediately; otherwise, CS_j proceeds to the next step.

- 2) CS_j computes $RID_{S_j} = H(ID_{S_j} || X_j)$ and then extracts $TID_i = TID_i^* \oplus H(RID_{S_j} || TS_i)$ and then finds the record $\langle ID_i, r_{ij} \rangle$ from its database corresponding to TID_i . Using ID_{S_j} and the master key X_j , CS_j computes

$$\begin{aligned}
 B_{ji} &= H(H(ID_i \oplus r_{ij}) || X_j), \\
 M_1 &= C_1 \oplus TS_i \oplus H(ID_{S_j}) \oplus B_{ji} \\
 &= A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j}) \oplus TS_i \\
 &\quad \oplus H(ID_{S_j}) \oplus B_{ji} \\
 &= RN_i,
 \end{aligned}$$

as $A_{ij} = B_{ji} = H(H(ID_i \oplus r_{ij}) || X_j)$.

- 3) Using computed parameter M_1 and received parameters $\{C_1, TS_i\}$, CS_j generates the hash value $H_2 = H(ID_i || C_1 || M_1 || TS_i)$. CS_j then verifies whether computed hash value $H_2 \stackrel{?}{=} H_1$. Failure of the condition

terminates the current session. Otherwise, CS_j accepts the login request and proceeds to the next step. CS_j also saves record $\langle ID_i, RN_i, TS_i \rangle$ in its database to resist strong replay attack. For instance, if CS_j receives another login request message, say $Msg'_1 = \{TID_i^*, C'_1, H'_1, TS'_1\}$ next time, it first checks the validity of TS'_1 . If it is valid, CS_j further verifies if the extracted $RN'_i = C'_1 \oplus TS'_1 \oplus H(ID_{S_j}) \oplus B_{ji}$ matches with the stored RN_i in its database corresponding to ID_i . If it is present, Msg'_1 is treated as a replay message. Thus, in the proposed scheme, to protect replay attack strongly we verify both timestamp as well as random nonce embedded in the login request message. Note that the schemes based on only random nonces do not provide strong replay attack as demonstrated by several researchers in the literature [47], [48].

- 4) CS_j then selects an 128 bit random number RN_j and computes $C_2 = B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i$, where TS_j is the current timestamp generated by CS_j . Further, CS_j computes the secret session key shared with MU_i as $SK_{S_j, MU_i} = H(ID_i || ID_{S_j} || B_{ji} || M_1 || RN_j || TS_i || TS_j)$, which is used for future message communication with MU_i . Finally, CS_j generates a hash value $H_3 = H(ID_i || M_1 || RN_j || TS_i || TS_j || SK_{CS_j, MU_i})$ and sends the authentication request message $Msg_2 = \{C_2, H_3, TS_j\}$ to MU_i via a public channel.
- 5) After MU_i receiving the authentication request message Msg_2 at time TS_j^* from CS_j , the condition $|TS_j^* - TS_j| \leq \Delta T$ is verified. If message transmission delay is within allowable limit, MU_i computes

$$\begin{aligned} M_2 &= C_2 \oplus TS_j \oplus ID_i \oplus A_{ij} \\ &= B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i \oplus TS_j \oplus ID_i \oplus A_{ij} \\ &= RN_j, \end{aligned}$$

as $A_{ij} = B_{ji} = H(H(ID_i \oplus r_{ij}) || X_j)$.

- 6) Using the received timestamp TS_j and computed parameter M_2 , MU_i generates the session key shared with CS_j as $SK_{MU_i, CS_j} = H(ID_i || ID_{S_j} || A_{ij} || RN_i || M_2 || TS_i || TS_j)$. Further, MU_i generates a hash value $H_4 = H(ID_i || RN_i || M_2 || TS_i || TS_j || SK_{MU_i, CS_j})$ and verifies whether $H_4 \stackrel{?}{=} H_3$. If the verification succeeds, MU_i assumes that the shared secret session key SK_{MU_i, CS_j} ($= SK_{CS_j, MU_i}$) is mutually verified and established. This is used for future message communication with CS_j in the current session.

The summary of the authentication phase of the proposed scheme is also provided in Fig. 5.

Remark 3: To speed up the searching of the credentials $\{ID_i, r_{ij}\}$ corresponding to the computed TID_i in the database of CS_j , the following procedure can be adapted. The credentials $\{TID_i, (ID_i, r_{ij})\}$ in the database of CS_j can be sorted in ascending order according to the key value TID_i by the RC during the cloud server registration phase in Section III-A.2. Then, we can perform the binary search algorithm to find the credentials $\{ID_i, r_{ij}\}$ corresponding to TID_i , which is executed

Mobile user (MU_i)	Mobile device
Input ID_i, PW_i and B'_i . $\{ID_i, PW_i, B'_i\}$	Compute $\theta_i = \text{Reproduction}(B'_i, \phi_i)$, $b' = D_i^1 \oplus H(PW_i \theta_i)$. Verify if $D_i^2 = H(ID_i PW_i \theta_i b')$? If verification holds, request to supply new password.
Input new password PW'_i . $\{PW'_i\}$	Compute $D_i^{1*} = D_i^1 \oplus H(PW_i \theta_i) \oplus H(PW'_i \theta_i)$, $D_i^{2*} = H(ID_i PW'_i \theta_i b')$, $V_{ij}^* = V_{ij}^1 \oplus H(ID_i H(PW_i \theta_i b'))$ $\oplus H(ID_i H(PW'_i \theta_i b'))$, $TID_i = RID_{ij} \oplus H(ID_i V_{ij}^*)$. For all servers CS_j s, compute $RID_{ij}^* = TID_i \oplus H(ID_i V_{ij}^*)$. Update $D_i^1 \leftarrow D_i^{1*}$, $D_2 \leftarrow D_i^{2*}$, $V_{ij}^1 \leftarrow V_{ij}^*$ and $RID_{ij} \leftarrow RID_{ij}^*$.

FIGURE 6. Password change phase of the proposed scheme.

in $O(\log_2(n))$ time complexity where n is number of CS_j s for each user MU_i . Hence, it is clear that the time for searching $\{ID_i, r_{ij}\}$ corresponding to TID_i by CS_j is not heavy even if n is large while implementing the proposed scheme for practical application, because CS_j is not resource-limited entity in the network.

D. PASSWORD CHANGE PHASE

The password change phase causes MU_i to update original password PW_i by the new password PW'_i . Note that this phase does not involve the RC or any CS_j and it is completely done locally. Figure 6 tabulates the basic steps of the password change phase. This phase requires the following steps:

- 1) MU_i enters his/her old password PW_i along with identity ID_i and biometrics B'_i .
- 2) Using the fuzzy extractor reproduction procedure and stored ϕ_i , MU_i computes $\theta_i = \text{Reproduction}(B'_i, \phi_i)$. Moreover, using the stored parameter D_i^1 , MU_i generates $b' = D_i^1 \oplus H(PW_i || \theta_i)$.
- 3) MU_i then computes $H(ID_i || PW_i || \theta_i || b')$ and checks if $D_i^2 = H(ID_i || PW_i || \theta_i || b')$ is true or not. MU_i proceeds to the next step only if this verification holds.
- 4) MU_i enters new password PW'_i and computes $D_i^{1*} = D_i^1 \oplus H(PW_i || \theta_i) \oplus H(PW'_i || \theta_i)$. Further, MU_i computes $D_i^{2*} = H(ID_i || PW'_i || \theta_i || b')$, $V_{ij}^* = V_{ij}^1 \oplus H(ID_i || H(PW_i || \theta_i || b')) \oplus H(ID_i || H(PW'_i || \theta_i || b'))$, $TID_i = RID_{ij} \oplus H(ID_i || V_{ij}^*)$ and $RID_{ij}^* = TID_i \oplus H(ID_i || V_{ij}^*)$ for $1 \leq j \leq n$.
- 5) The user mobile device updates D_i^1 with D_i^{1*} , D_2 with D_i^{2*} , V_{ij}^1 with V_{ij}^* and RID_{ij} with RID_{ij}^* in its memory.

Finally, the password change phase of the proposed scheme is summarized in Fig. 6.

E. MOBILE DEVICE REVOCATION PHASE

If a legal mobile user MU_i 's device is lost or stolen, it is necessary to ensure that in-spite of accessing the stored

information, an adversary \mathcal{A} can not make a login to the cloud server. It is required to revoke the lost mobile device and allow MU_i to login using new mobile device. For this purpose, the following steps are executed:

- 1) MU_i enters ID_i , PW_i and biometrics \mathcal{B}_i , and also generates two new 128 bit random number b' and k' .
- 2) MU_i produces $(\theta_i, \phi_i) = \text{Generation}(\mathcal{B}_i)$ and computes the masked password $RPW_{B_i} = H(ID_i || H(PW_i || \theta_i || b'))$. MU_i then submits the registration request message $\langle ID_i, (RPW_{B_i} \oplus k') \rangle$ to the RC via secure channel.
- 3) The RC selects an 1024-bit random number r'_{ij} for each MU_i and CS_j pair. Further, RC computes $A_{ij} = H(H(ID_i \oplus r'_{ij}) || X_j)$ and $V_{ij} = A_{ij} \oplus RPW_{B_i}$.
- 4) In order to maintain user anonymity, instead of actual identity ID_i , RC selects a unique and random temporary identity TID'_i for MU_i .
- 5) The RC saves n server key-plus-id combinations $\{(ID_{S_j}, V_{ij}, TID'_i) \mid 1 \leq j \leq n\}$ in mobile device of MU_i and delivers the mobile device to MU_i securely.
- 6) The RC verifies authenticity of MU_i by checking his/her other credentials, such as date of birth (DOB) and registered id number. MU_i computes $D_i^1 = H(PW_i || \theta_i) \oplus b'$ and $D_i^2 = H(ID_i || PW_i || \theta_i || b')$, and $V'_{ij} = V_{ij} \oplus k' = A_{ij} \oplus H(ID_i || H(PW_i || \theta_i || b'))$ and $RID_{ij} = TID'_i \oplus H(ID_i || V'_{ij})$ for $1 \leq j \leq n$. Finally, MU_i also stores ϕ_i , D_i^1 , D_i^2 , V'_{ij} s, RID_{ij} s and RID'_{S_j} s ($RID'_{S_j} = RID_{S_j} \oplus H(\theta_i || b')$) into his/her own mobile device, and deletes V_{ij} s and TID'_i from the mobile device.
- 7) All servers CS_j s also update (ID_i, TID_i, r_{ij}) with (ID_i, TID'_i, r'_{ij}) in their databases after receiving the update request message securely from the RC .

IV. SECURITY ANALYSIS

This section provides formal security analysis of the proposed scheme. Moreover, this section also puts forward discussion on how the proposed scheme resists various other security attacks.

A. FORMAL SECURITY USING ROR MODEL

The formal security of the proposed mobile user authentication protocol, say \mathcal{P} , is done using the widely-accepted Real-Or-Random (ROR) model [49]–[51].

1) OUTLINE OF ROR MODEL

According to the ROR model, various queries, that simulate the real attack, are executed by an adversary \mathcal{A} [52], [53].

Table 2 contains various notations and brief descriptions of various oracle queries that are used in this proof. We assume that \mathcal{A} interacts with \mathcal{P}^t , the t^{th} instance of an executing participant (MU_i or CS_j).

Definition 3 (Semantic Security): Let $\text{Adv}_{\mathcal{P}}^{\text{MUAP}}$ denotes the advantage of \mathcal{A} running in polynomial time in breaking the semantic security of proposed mobile user authentication protocol (MUAP), referred as \mathcal{P} . Then, $\text{Adv}_{\mathcal{P}}^{\text{MUAP}} = |2 \Pr[b' = b] - 1|$, where b' is the guessed bit.

TABLE 2. Different oracle queries and their descriptions.

Query	Description/purpose
$\text{Send}(\mathcal{P}^t, m)$	It enables \mathcal{A} to send request message m to \mathcal{P}^t and \mathcal{P}^t replies accordingly
$\text{Corrupt}(MU_i, a)$	Depending on a , \mathcal{A} can obtain biometric and password of MU_i
$\text{Test}(\mathcal{P}^t)$	\mathcal{A} requests \mathcal{P}^t for the session key SK , \mathcal{P}^t replies probabilistically on outcome of a flipped coin b
$\text{Execute}(MU_i, CS_j)$	It enables \mathcal{A} to eavesdrop the messages communicated between MU_i and CS_j
$\text{Reveal}(\mathcal{P}^t)$	It enables \mathcal{A} to obtain the session key SK generated between \mathcal{P}^t and its partner

TABLE 3. Symbols used in the real-or-random (ROR) model.

Symbol	Description
q_H	Total number of hash H oracle queries execution
q_s	Total number of Send oracle queries execution
q_e	Total number of Execute oracle queries execution
l_H	Length of hash output string
l_r	Length of random number string
l_b	Length of user biometric key
ϵ_{bm}	Probability of false positive in biometrics [54]
\mathcal{D}	Password space with its frequency distribution follows a Zipf's law [55]
C', s'	Zipf parameters [55]
L_H	List that stores output of hash H oracle query
L_A	List that records random oracle outputs
L_T	List that records message transcripts between MU_i and CS_j

Definition 4: The proposed protocol \mathcal{P} is semantically secure if the advantage function $\text{Adv}_{\mathcal{P}}^{\text{MUAP}}$ is only negligibly larger than $\max\{C' \cdot q_s^s, q_s(\frac{1}{2^{l_b}}, \epsilon_{bm})\}$, where q_s , l_b , C' and s' denote their usual meanings as tabulated in Table 3.

2) SECURITY PROOF

We use the notations listed in Table 3 for the formal security proof. Recent research has shown that user-chosen passwords (also termed as “weak secrets”) follow the Zipf’s law [55], which is a vastly different distribution from the uniform distribution. Actually, the size $|\mathcal{D}|$ of password dictionary \mathcal{D} is generally much constrained in the sense that the users will not use the whole space of passwords, but rather a small space of the allowed characters space [55]. On the other hand, even if we consider only trawling guessing attacks, actually the advantage of an adversary will be over 0.5 when $q_s = 10^7$ or 10^8 [55], [56]. When further considering targeted guessing attacks in which the adversary can make use of the target user’s personal information, the advantage of the adversary will be over 0.5 when $q_s \leq 10^6$ [57].

Theorem 1: Suppose $\text{Adv}_{\mathcal{P}}^{\text{MUAP}}$ denotes the advantage function of an adversary \mathcal{A} in breaking the semantic security of the proposed scheme \mathcal{P} as defined in Definition 4. Then,

$$\text{Adv}_{\mathcal{P}}^{\text{MUAP}} \leq \frac{q_H^2 + 18q_H}{2^{l_H}} + \frac{(q_s + q_e)^2 + 4q_s}{2^{l_r}} + 2 \max\{C' \cdot q_s^s, q_s(\frac{1}{2^{l_b}}, \epsilon_{bm})\},$$

where $q_H, q_s, q_e, l_H, l_r, l_b, \varepsilon_{bm}, C'$ and s' have their usual meanings as tabulated in Table 3.

Proof: We follow the similar proof as in [41] and [45]. The proof is composed of five games Gm_i ($i = 0, 1, 2, 3, 4$). In a game Gm_i , an adversary \mathcal{A} tries to guess a correct bit b through the *Test* query. This event is defined as S_i and the corresponding probability is denoted by $Pr[S_i]$.

- **Game Gm_0 :** The initial game Gm_0 is considered to be identical with the actual protocol executing under the ROR model. Hence, we have,

$$Adv_{\mathcal{A}}^{MUAP} = |2Pr[S_0] - 1|. \quad (1)$$

- **Game Gm_1 :** This game considers simulation of *Send*, *Test*, *Execute*, *Reveal*, and *Corrupt* queries with respect to the proposed scheme. Table 4 describes the working procedure of *Execute* and *Send* query. Further, this game considers lists L_H, L_A , and L_T for storing results of various oracle queries. Due to indistinguishability of Gm_0 and Gm_1 , we obtain,

$$Pr[S_1] = Pr[S_0]. \quad (2)$$

- **Game Gm_2 :** The collision probability of random oracle query and hash (H) oracle query are considered in this game for all the communicated messages between MU_i and CS_j . In $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ and $Msg_2 = \{C_2, H_3, TS_j\}$, MU_i and CS_j use random numbers RN_i and RN_j , and also the current timestamps TS_i and TS_j , respectively. This causes collision probability at most $\frac{(q_s + q_e)^2}{2^{l_r + 1}}$. Moreover, based on the birthday paradox, use of H oracle query results in collision probability of $\frac{q_H^2}{2^{l_H + 1}}$. Overall, we obtain,

$$|Pr[S_2] - Pr[S_1]| \leq \frac{(q_s + q_e)^2}{2^{l_r + 1}} + \frac{q_H^2}{2^{l_H + 1}}. \quad (3)$$

- **Game Gm_3 :** Since H hash oracle query is already considered in the game Gm_2 , we need to calculate collision probability from all other remaining oracle queries.

Case 1: After executing $Send(CS_j, Msg_1)$ query on $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$, it is noted that $H_1 = H(ID_i || C_1 || RN_i || TS_i) \in L_A$ has collision probability at most $\frac{q_H}{2^{l_H}}$. To launch attack successfully, $H(ID_i || \theta_i) \oplus b$ of D_i^1 , $H(ID_i || PW_i || \theta_i || b)$ of D_i^2 and $A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j})$ of C_1 should be revealed to \mathcal{A} . This results in the total collision probability up to $\frac{4q_H}{2^{l_H}}$. Moreover, as transcript message Msg_1 contains RN_i , $Msg_1 \in L_T$ must hold with probability up to $\frac{q_s}{2^{l_r}}$.

Case 2: Considering \mathcal{A} executes the query $Send(MU_i, Msg_2)$ and $H_3 \in L_A$ holds, the calculated probability becomes $\frac{q_H}{2^{l_H}}$. Furthermore, CS_j computes $H(H(ID_i \oplus r_{ij}) || X_j)$ for B_{ji} , $H(ID_{S_j})$ in M_1 , verifies $H(ID_i || C_1 || M_1 || TS_i)$ with H_1 , and finally, it computes $SK_{CS_j, MU_i} = H(ID_i || ID_{S_j} || B_{ji} || M_1 || RN_j || TS_i || TS_j)$. Hence the total probability for this part is $\frac{5q_H}{2^{l_H}}$. Due to the message

TABLE 4. Simulation of execute and send oracle queries.

<p>Simulation of $Execute(MU_i, CS_j)$ query occurs in succession with simulation of <i>Send</i> queries as given below. Compute C_1 and H_1 as given in Fig. 5. MU_i sends the message $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ to CS_j. Compute C_2, H_3 and SK_{CS_j, MU_i} as given in Fig. 5. CS_j sends authentication message $Msg_2 = \{C_2, H_3, TS_j\}$ to MU_i. Note that $\langle TID_i^*, C_1, H_1, TS_i \rangle \leftarrow Send(MU_i, \mathbf{start})$, $\langle C_2, H_3, TS_j \rangle \leftarrow Send(CS_j, \langle TID_i^*, C_1, H_1, TS_i \rangle)$. Finally, Msg_1 and Msg_2 are returned.</p>
<p><i>Send</i> query simulation is done as per the proposed scheme: (a) On $Send(MU_i, \mathbf{start})$ query, MU_i responds as follows. Compute TID_i^*, C_1, H_1, TS_i as in Fig. 5. Output $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$.</p>
<p>(b) Over $Send(CS_j, \langle TID_i^*, C_1, H_1, TS_i \rangle)$ query, CS_j responds as follows. Test if $TS_i^* - TS_i \leq \Delta T$ and then generates B_{ji} and M_1. Also, verify the parameter H_1. Terminate the current session if verification fails. Moreover, CS_j computes C_2, SK_{CS_j, MU_i} and H_3, and output $Msg_2 = \{C_2, H_3, TS_j\}$.</p>
<p>(c) MU_i answers $Send(MU_i, \langle C_2, H_3, TS_j \rangle)$ query mentioned below. Check if $TS_j^* - TS_j \leq \Delta T$. If verification passes, compute M_2 and SK_{MU_i, CS_j}. Finally, verify H_3. If verification fails, terminate the current session. Otherwise, compute and accept SK_{MU_i, CS_j} as the session key as depicted in Fig. 5. On establishment of the shared session key, both MU_i & CS_j terminate the session.</p>

transcript $Msg_2 \in L_T$, we obtain $\frac{q_s}{2^{l_r}}$ as the collision probability. As a whole, we obtain,

$$|Pr[S_3] - Pr[S_2]| \leq \frac{2q_s}{2^{l_r}} + \frac{9q_H}{2^{l_H}}. \quad (4)$$

- **Game Gm_4 :** In game Gm_4 , by exploiting *Corrupt* query, the adversary \mathcal{A} tries to guess user's private credentials like password and biometric in online as well as offline modes. The guessing of biometric has maximum probability up to $\max\{q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}$ [41], [45], and that for password is $C' \cdot q_s^s$ [55]. Since the games Gm_3 and Gm_4 are identical when these guessing attacks are absent, we have,

$$|Pr[S_4] - Pr[S_3]| \leq \max\{C' \cdot q_s^s, q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}. \quad (5)$$

After executing all the games, \mathcal{A} is only left in guessing the correct bit b . It is then clear that

$$Pr[S_4] = \frac{1}{2}. \quad (6)$$

Applying the law of triangular inequality, we have,

$$\begin{aligned} |Pr[S_0] - \frac{1}{2}| &= |Pr[S_1] - Pr[S_4]| \\ &\leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_4]| \\ &\leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_3]| \\ &\quad + |Pr[S_3] - Pr[S_4]|. \end{aligned} \quad (7)$$

TABLE 5. Symbols and their descriptions in BAN logic.

Symbol	Description
$Q \equiv S$	Q believes that the statement S is true
$Q \triangleleft S$	Q can see the statement S
$\#(S)$	Formula S is considered as fresh
$Q \mid\sim S$	Q said the statement S once
$Q \Rightarrow S$	Q keeps jurisdiction over the statement S
$\langle S \rangle_T$	Formula S is combined with the formula T
$Q \xrightarrow{K} R$	Only Q and R know the value of the key K and it is used for communication between them
$Q \stackrel{S}{\rightleftharpoons} R$	Only Q and R know the secret statement S . Principals trusted by Q & R may know S
SK	Current session key

Using Equations (1)-(7), we obtain,

$$\begin{aligned} \frac{1}{2} Adv_{\mathcal{P}}^{MUAP} &= |Pr[S_0] - \frac{1}{2}| \\ &\leq \frac{(q_s + q_e)^2}{2^{l_r+1}} + \frac{q_H^2}{2^{l_H+1}} + \frac{2q_s}{2^{l_r}} + \frac{9q_H}{2^{l_H}} \\ &\quad + \max\{C' \cdot q_s', q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}. \end{aligned} \quad (8)$$

Multiplying both sides of Equation (8) by a factor of 2 and then rearranging the terms, we finally obtain

$$\begin{aligned} Adv_{\mathcal{P}}^{MUAP} &\leq \frac{q_H^2 + 18q_H}{2^{l_H}} + \frac{(q_s + q_e)^2 + 4q_s}{2^{l_r}} \\ &\quad + 2 \max\{C' \cdot q_s', q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}. \end{aligned}$$

Hence, the theorem is proved. \square

B. AUTHENTICATION PROOF USING BAN LOGIC

BAN logic is used to mutual authentication between two communicating parties in a network [58]. Using the broadly-used BAN logic, we show that the proposed scheme achieves authentication goals discussed below. The basic BAN logic notations and their meanings are tabulated in Table 5.

The main logical postulates of the BAN logic are defined by a set of laws (rules) as listed below [58], [59].

- Law 1 (Message Meaning Law (MML)).

$$\frac{Q \equiv R \stackrel{K}{\rightleftharpoons} Q, Q \triangleleft \langle S \rangle_K}{Q \equiv R \mid\sim S}.$$

- Law 2 (Nonce Verification Law (NVL)).

$$\frac{Q \mid\sim \#(S), Q \mid\sim R \mid\sim S}{Q \mid\sim R \equiv S}.$$

- Law 3 (Freshness Concatenation Law (FCL)).

$$\frac{Q \mid\sim \#(S)}{Q \mid\sim \#(S, T)}.$$

- Law 4 (Jurisdiction Law (JL)).

$$\frac{Q \mid\sim R \Rightarrow S, Q \mid\sim R \equiv S}{Q \mid\sim S}.$$

- Law 5 (Additional Laws (AL)).

$$\frac{Q \mid\sim \langle S, T \rangle}{Q \mid\sim S}, \frac{Q \triangleleft \langle S, T \rangle}{Q \triangleleft S}, \frac{Q \mid\sim R \sim \langle S, T \rangle}{Q \mid\sim R \sim S}.$$

To complete the authentication proof, the proposed scheme must meet the following two goals:

$$\text{Goal 1. } MU_i \mid\equiv (MU_i \xrightarrow{SK} CS_j).$$

$$\text{Goal 2. } CS_j \mid\equiv (MU_i \xrightarrow{SK} CS_j).$$

The generic types of the messages in the proposed scheme are given below.

Message 1. $MU_i \rightarrow CS_j: \{TID_i^*, H(H(ID_i \oplus r_{ij}) || X_j) \oplus RN_i \oplus TS_j \oplus H(ID_{S_j}), TS_i, H_1\}$.

Message 2. $CS_j \rightarrow MU_i: \{B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i, TS_j, H_3\}$.

The idealized forms of the above generic messages are provided mentioned below.

Message 1. $MU_i \rightarrow CS_j: \{TID_i, TS_i, \langle ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j}) \rangle_{X_j}, H_1\}$.

Message 2. $CS_j \rightarrow A: \{TS_j, \langle RN_j, TS_j, ID_i \rangle_{X_j}, H_3\}$.

The authentication proof of the proposed scheme starts with the following basic assumptions:

$$\text{A.1: } MU_i \mid\equiv \#(TS_j);$$

$$\text{A.2: } CS_j \mid\equiv \#(TS_i);$$

$$\text{A.3: } MU_i \mid\equiv (MU_i \stackrel{A_{ij}}{\rightleftharpoons} CS_j);$$

$$\text{A.4: } CS_j \mid\equiv (MU_i \stackrel{A_{ij}}{\rightleftharpoons} CS_j);$$

$$\text{A.5: } MU_i \mid\equiv CS_j \Rightarrow (ID_{S_j}, RN_j, TS_j);$$

$$\text{A.6: } CS_j \mid\equiv MU_i \Rightarrow (ID_i, RN_i, TS_i);$$

$$\text{A.7: } MU_i \mid\equiv TS_j;$$

$$\text{A.8: } MU_i \mid\equiv RN_i;$$

$$\text{A.9: } MU_i \mid\equiv ID_i;$$

$$\text{A.10: } MU_i \mid\equiv ID_{S_j};$$

$$\text{A.11: } CS_j \mid\equiv TS_j;$$

$$\text{A.12: } CS_j \mid\equiv RN_j;$$

$$\text{A.13: } CS_j \mid\equiv ID_{S_j}.$$

Considering these basic assumptions, idealized forms and fundamental logical postulates, in the following we show the achievement of both the goals **Goal 1** and **Goal 2**.

According to the message 1, we obtain,

$$\bullet S_1: CS_j \triangleleft \{ID_i, TS_i, \langle ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j}) \rangle_{X_j}, H_1\}.$$

$$\bullet S_2: \text{According to the law AL, we obtain, } CS_j \triangleleft \{ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j})\}_{X_j}.$$

$$\bullet S_3: \text{According to A.4 and MML, we obtain, } CS_j \mid\equiv MU_i \mid\sim (ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j})).$$

$$\bullet S_4: \text{According to A.2 and FCL, we get, } CS_j \mid\equiv \#(ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j})).$$

$$\bullet S_5: \text{According to NVL, we have, } CS_j \mid\equiv MU_i \mid\equiv (ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j})).$$

$$\bullet S_6: \text{Using A.6 and JL, we get, } CS_j \mid\equiv (ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j})).$$

$$\bullet S_7: \text{From } S_6 \text{ and AL, we obtain, } CS_j \mid\equiv RN_i, CS_j \mid\equiv TS_i, CS_j \mid\equiv ID_i.$$

$$\bullet S_8: \text{According to A.11, A.12, A.13, we get, } CS_j \mid\equiv ID_{S_j}, CS_j \mid\equiv TS_j \text{ and } CS_j \mid\equiv RN_j.$$

$$\bullet S_9: \text{Since } SK_{CS_j, MU_i} = H(ID_i || ID_{S_j} || B_{ji} || M_1 || RN_j || TS_i || TS_j) \text{ and the results in Steps } S_7 \text{ and } S_8 \text{ give}$$

$$CS_j \mid\equiv (MU_i \xrightarrow{SK_{CS_j, MU_i}} CS_j). \quad (\text{Goal 2})$$

$$\bullet S_{10}: \text{Using the message 2 and AL, we obtain, } MU_i \triangleleft \langle RN_j, TS_j \rangle_{X_j}.$$

$$\bullet S_{11}: \text{According to A.3 and MML, we get, } MU_i \mid\equiv CS_j \mid\sim (RN_j, TS_j).$$

$$\bullet S_{12}: \text{Using A.1 and FCL, we obtain, } MU_i \mid\equiv \#(RN_j, TS_j).$$

$$\bullet S_{13}: \text{Using NVL, we obtain, } MU_i \mid\equiv CS_j \mid\equiv (RN_j, TS_j).$$

- S_{14} : A.5 and JL give $MU_i \equiv (RN_j, TS_j)$.
- S_{15} : According to S_{14} and AL, we have, $MU_i \equiv RN_j$, $MU_i \equiv TS_j$.
- S_{16} : According to A.7-A.10, we obtain, $MU_i \equiv ID_i$, $MU_i \equiv ID_{S_j}$, $MU_i \equiv TS_i$, $MU_i \equiv RN_j$.
- S_{17} : The results of Steps S_{15} and S_{16} give $MU_i \equiv (MU_i \xleftrightarrow{SK_{CS_j, MU_i}} CS_j)$. **(Goal 1)**

As a result, **Goal 1** and **Goal 2** ensure that both MU_i and CS_j mutually authenticate each other.

C. DISCUSSION ON OTHER ATTACKS

This section also informally analyzes the security of the proposed scheme to show that it can defend the following other known attacks.

1) REPLAY ATTACK

According to the proposed scheme, the login and authentication phases require two message communications. In login phase, MU_i sends $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ to CS_j , whereas in authentication phase, CS_j sends $Msg_2 = \{C_2, H_3, TS_j\}$ to MU_i . CS_j does not accept Msg_1 if $|TS_i^* - TS_i| > \Delta T$. To resist replay attack, CS_j further computes $H(ID_i || C_1 || M_1 || TS_i)$ and verifies it with received hash value as $H_1 \stackrel{?}{=} H(ID_i || C_1 || M_1 || TS_i)$. The cloud server CS_j rejects login request if this verification fails. As explained in Step 6 of authentication phase in Section III-C, an attacker also fails to replay the authentication message Msg_2 . In addition, CS_j also stores parameters (ID_i, RN_i, TS_i) in its database to resist strong replay attack. If CS_j receives another login request message, say $Msg_1' = \{TID_i^*, C_1', H_1', TS_i'\}$ next time, it first checks the validity of TS_i' . If it is valid, CS_j further verifies if the extracted $RN_i' = C_1' \oplus TS_i' \oplus H(ID_{S_j}) \oplus B_{ji}$ matches with the stored RN_i in its database corresponding to ID_i . If it is present, Msg_1' is treated as a replay message. As a whole, the proposed scheme protects strong replay attack because both the current timestamp and random nonce are applied.

2) MAN-IN-THE-MIDDLE ATTACK

An adversary \mathcal{A} may try to launch man-in-the-middle attack in order to set up a third party independent connection with both MU_i and CS_j for a particular session. Moreover, \mathcal{A} might intend to modify public message parameters to invalidate a login request of a legal user. The proposed scheme uses hash function, random nonce and bitwise XOR operation in both message Msg_1 and Msg_2 . However, \mathcal{A} can not modify any message as these need the credentials, such as A_{ij} , ID_{S_j} , V_{ij}' and RID_{ij} . This causes the proposed scheme to resist the man-in-the-middle attack.

3) STOLEN/LOST MOBILE DEVICE ATTACK

The user mobile device contains $D_i^1 = H(PW_i || \theta_i) \oplus b$ and $D_i^2 = H(ID_i || PW_i || \theta_i || b)$. As guessing of ID_i , PW_i and biometrics \mathcal{B}_i from D_i^1 and D_i^2 is computationally infeasible,

\mathcal{A} can not obtain these credentials from user mobile device. Further, user mobile device contains $V_{ij}' = A_{ij} \oplus RPWB_i$, where $A_{ij} = H(H(ID_i \oplus r_{ij}) || X_j)$ and $RPWB_i = H(ID_i || H(PW_i || \theta_i || b))$. Since r_{ij} and b are random numbers and $H(\cdot)$ is a collision-resistant, it is computationally infeasible problem to obtain ID_i , PW_i and θ_i from V_{ij}' and A_{ij}' in polynomial time. Hence, the proposed scheme resists this attack.

4) OFFLINE PASSWORD GUESSING ATTACK

According to user registration phase described in Section III-A.1, mobile device of MU_i contains $\langle D_i^1, D_i^2, \phi_i, \{(ID_{S_j}, V_{ij}', RID_{ij}, RID_{S_j}') \mid 1 \leq j \leq n\} \rangle$. As discussed in Section IV-C.3, \mathcal{A} can not guess password from any stored parameters like D_i^1 , D_i^2 and V_{ij}' as PW_i is masked with θ_i and random secret b . To obtain PW_i , \mathcal{A} needs to guess these parameters simultaneously, which have negligible probability. So, this kind of attack is resisted by the proposed scheme.

5) FORWARD SECRECY

Forward secrecy (also known as known key secrecy) ensures that a compromised session key does not help an adversary to compute past session keys. According to the proposed scheme, the session key is mutually computed as $SK_{MU_i, CS_j} = SK_{CS_j, MU_i} = H(ID_i || ID_{S_j} || A_{ij} || RN_i || RN_j || TS_i || TS_j)$ where $A_{ij} = B_{ji} = H(H(ID_i \oplus r_{ij}) || X_j)$. Due to the use of RN_i , TS_i , RN_j , and TS_j , for every new login session, SK_{MU_i, CS_j} ($= SK_{CS_j, MU_i}$) is generated in random but in a unique way. As a consequence, compromise of the current session key provides no crucial information to the adversary that helps him/her to compute previous session keys.

6) ANONYMITY AND UNTRACEABILITY

Generally, the mobile users intend to access cloud services in an anonymous way. As defined in the threat model in Section I-B, \mathcal{A} is able to eavesdrop public messages transmitted between MU_i and CS_j . During login time, MU_i sends $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ to CS_j . Note that MU_i does not send the original identity in login message. Rather, it sends the temporary identity TID_i embedded in $TID_i^* = TID_i \oplus H(RID_{S_j} || TS_i)$ (see Remark 2). Moreover, from $C_1 = A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j})$ and $H_1 = H(ID_i || C_1 || RN_i || TS_i)$, it is not possible to obtain ID_i . During authentication phase, CS_j sends $Msg_2 = \{C_2, H_3, TS_j\}$ to MU_i , where $C_2 = B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i$ and $H_3 = H(ID_i || M_1 || RN_j || TS_i || TS_j || SK_{CS_j, MU_i})$. As hash function $H(\cdot)$ is considered to be collision resistant, from these eavesdropped messages, it is computationally infeasible for an attacker to compute ID_i . Thus, the proposed scheme provides user anonymity property.

The messages $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ and $Msg_2 = \{C_2, H_3, TS_j\}$ are unique and dynamic in nature in each session, because each component in these messages use either timestamp or random nonce. Hence, the proposed scheme also preserves the untraceability property as an attacker can not trace the same user in different session.

(* ——— channels ——— *)
free pch: channel. (* public channel *)
free sch: channel [private]. (* private channel *)
(* ——— shared keys ——— *)
free SKus:bitstring [private].(* the session key of user *)
free SKsu:bitstring [private]. (* the session key of server *)
(* ——— Servers secret key ——— *)
free Xj:bitstring [private].
free rij:bitstring [private].
(* ——— constants ——— *)
free IDSj:bitstring [private].
free ID:bitstring [private].
free TID:bitstring [private].
free PW:bitstring [private].
const Bi:bitstring [private].
(* ——— functions and equations ——— *)
fun h(bitstring):bitstring. (* hash function *)
fun FE(bitstring):bitstring. (* Fuzzy extractor function *)
fun xor(bitstring,bitstring):bitstring. (* XOR operation *)
fun con(bitstring,bitstring):bitstring. (* string concatenation *)
equation forall x:bitstring,y:bitstring; xor(xor(x,y),y) = x.
(* ——— aims for verification ——— *)
query attacker(SKus).
query attacker(SKsu).
query id:bitstring; inj-event(UserAuth(id)) ==> inj-event(UserStart(id)).
(* ——— event ——— *)
event UserStart(bitstring). (* User starts authentication *)
event UserAuth(bitstring). (* User is authenticated *)

FIGURE 7. Declaration of channels, keys, constants, functions, equations, queries and events.

```

let User=
new b:bitstring;
let alpha = FE(Bi) in
let RPWB = h(con(ID,h(con(PW,con(alpha,b)))) in
out(sch,(ID,RPWB));
in(sch,(rVij:bitstring));
let D1 = xor(h(con(PW,alpha)),b) in
let D2 = h(con(ID,con(PW,con(alpha,b)))) in
!
(
event UserStart(ID);
let b1 = xor(D1,h(con(PW,alpha))) in
let D21 = h(con(ID,con(PW,con(alpha,b1)))) in
if D2 = D21 then
new RNi:bitstring;
new TSi:bitstring;
let Aij1 = xor(rVij,RPWB) in
let C1 = xor(Aij1,xor(RNi,xor(TSi,h(IDSj)))) in
let H1 = h(con(ID,con(C1,con(RNi,TSi)))) in
out(pch,(TID,C1,TSi,H1));
in(pch,(rC2:bitstring,rTSj:bitstring,rH3:bitstring));
let M2 = xor(rC2,xor(rTSj,xor(ID,Aij1))) in
let SKus = h(con(ID,con(IDSj,con(Aij1,con(RNi,con(M2,con(TSi,rTSj)))))) in
let H4 = h(con(ID,con(RNi,con(M2,con(TSi,con(rTSj,SKus)))))) in
if H4 = rH3 then
0
).

```

FIGURE 8. ProVerif code for the process of mobile user MU_i .

7) SESSION KEY SECURITY

According to the proposed scheme, both MU_i and CS_j mutually establish a common session key SK_{MU_i,CS_j} ($= SK_{CS_j,MU_i}$) for future communication. Note that the session key is calculated as

$$\begin{aligned}
SK_{MU_i,CS_j} &= H(ID_i || ID_{S_j} || A_{ij} || RN_i || M_2 || TS_i || TS_j) \\
&= H(ID_i || ID_{S_j} || B_{ji} || RN_i || M_2 || TS_i || TS_j) \\
&= H(ID_i || ID_{S_j} || B_{ji} || M_1 || M_2 || TS_i || TS_j)
\end{aligned}$$

```

let SReg =
in(sch,(sID:bitstring,sRPWB:bitstring));
let Aij = h(con(h(xor(sID,rij)),Xj)) in
let Vij = xor(Aij,sRPWB) in
out(sch,(Vij)).

let SAAuth =
in(pch,(xID:bitstring,xC1:bitstring,xTSi:bitstring,xH1:bitstring));
let Bji = h(con(h(xor(xID,rij)),Xj)) in
let M1 = xor(Bji,xor(xC1,xor(xTSi,h(IDSj)))) in
let H2 = h(con(xID,con(xC1,con(M1,xTSi)))) in
if H2 = xH1 then
event UserAuth(xID);
new RNj:bitstring;
new TSj:bitstring;
let C2 = xor(Bji,xor(RNj,xor(TSj,xID))) in
let SKsu = h(con(xID,con(IDSj,con(Bji,con(M1,con(RNj,con(xTSi,TSj)))))) in
let H3 = h(con(xID,con(M1,con(RNj,con(xTSi,con(TSj,SKsu)))))) in
out(pch,(C2,TSj,H3)).
let S = SReg — SAAuth.
process !User — !S

```

FIGURE 9. ProVerif code for the process of the cloud server CS_j .

```

File "/tmpfiles/40438219/inpProt.pv", line 54, character 5 - line 54, character 9:
Warning: identifier SKus rebound
File "/tmpfiles/40438219/inpProt.pv", line 75, character 5 - line 75, character 9:
Completing equations...
Completing equations...
- Query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
Completing...
200 rules inserted. The rule base contains 200 rules. 28 rules in the queue.
Starting query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
RESULT inj-event(UserAuth(id)) ==> inj-event(UserStart(id)) is true.
- Query not attacker(SKsu[])
Completing...
200 rules inserted. The rule base contains 200 rules. 22 rules in the queue.
Starting query not attacker(SKsu[])
RESULT not attacker(SKsu[]) is true.
- Query not attacker(SKus[])
Completing...
200 rules inserted. The rule base contains 200 rules. 22 rules in the queue.
Starting query not attacker(SKus[])
RESULT not attacker(SKus[]) is true.

```

FIGURE 10. Analysis of the simulation results.

$$\begin{aligned}
&= H(ID_i || ID_{S_j} || B_{ji} || M_1 || RN_j || TS_i || TS_j) \\
&= SK_{CS_j,MU_i}.
\end{aligned}$$

To establish the session key, both MU_i and CS_j mutually authenticate each other. Furthermore, to derive the session key an attacker need to have the credentials ID_i , ID_{S_j} , A_{ij} ($= B_{ji}$). Hence, the session key is secure.

8) PARALLEL SESSION AND REFLECTION ATTACKS

An adversary \mathcal{A} can masquerade as a genuine user and then try to initiate a new parallel session with CS_j if the credentials belonging to a legal user are obtained. On the other side, as already explained in Sections IV-C.1, IV-C.2 and IV-C.5, \mathcal{A} can not obtain mobile user credentials through offline guessing attack or with any eavesdropped messages. As a result, the proposed scheme resists parallel session as well as reflection attacks.

9) EPHEMERAL SECRET LEAKAGE ATTACK

Under this attack, the exposor of ephemeral (temporary) secrets (e.g., random numbers) of a session may harm the secrecy of a session key. After execution of the

TABLE 6. Security and functionality comparison with the recent authentication schemes.

Security attributes	He-Wang [32]	Yoon-Yoo[30]	Shen et al. [31]	Tsai-Lo [10]	Tseng et al. [29]	Our
Stolen mobile device/smart card attack	✓	X	✓	✓	NA	✓
Strong replay attack	X	✓	✓	✓	✓	✓
Password guessing attack (online)	✓	✓	✓	✓	✓	✓
Password guessing attack (offline)	✓	✓	✓	✓	✓	✓
Privileged insider attack	✓	X	✓	✓	✓	✓
DoS attack	X	✓	✓	✓	✓	✓
Known session key secrecy	✓	✓	✓	✓	✓	✓
Strong user anonymity provision	X	X	X	✓	X	✓
Forward secrecy	✓	✓	✓	✓	✓	✓
Session key security	X	X	X	X	X	✓
User impersonation attack	X	X	✓	X	✓	✓
Server impersonation attack	✓	✓	✓	X	✓	✓
Ephemeral secret key leakage attack	X	X	X	X	✓	✓
User anonymity provision	✓	✓	✓	✓	X	✓
Efficient password change	X	✓	✓	NA	NA	✓
Login phase efficiency	X	✓	✓	X	X	✓
Revocation of smart card	NA	✓	✓	NA	NA	✓
Secure mutual authentication	✓	✓	X	X	✓	✓
Low computation overhead	X	X	X	X	✓	✓
Low communication overhead	X	✓	X	X	✓	✓
Formal security proof	X	X	✓	✓	✓	✓
Simulation using AVISPA/ProVerif	X	X	X	X	X	✓

protocol, if the random numbers are not properly deleted, an adversary \mathcal{A} might obtain them from a compromised device and also can launch ephemeral secret leakage attack. An authentication protocol must be able to resist this attack.

In the proposed scheme, the session key is computed as $SK_{MU_i,CS_j} = H(ID_i || ID_{S_j} || A_{ij} || RN_i || M_2 || TS_i || TS_j)$, where $A_{ij} = H(H(ID_i \oplus r_{ij}) || X_j)$, r_{ij} is an 1024-bit random secret and X_j is the master secret key of the cloud server CS_j . Hence, even if the values of random numbers RN_i and RN_j (that is, M_2) are known, the adversary \mathcal{A} cannot compute SK_{MU_i,CS_j} as it also depends on the long-term secret credentials, such as ID_i , ID_{S_j} and A_{ij} . As a result, SK_{MU_i,CS_j} can not derive other session keys established in other sessions between MU_i and CS_j using the ephemeral secret leakage attack.

10) USER IMPERSONATION ATTACK

Using user impersonation attack, an adversary \mathcal{A} can masquerade as a legitimate user and try to login to CS_j . However, the proposed scheme can resist this attack due to the following argument. \mathcal{A} needs to input correct inputs ID_i , PW_i and B'_i to prove its authenticity as a genuine user. As already discussed, an adversary has no computationally feasible way to guess these parameters.

\mathcal{A} can also try to generate a replay login message $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ and submit it to CS_j . But as explained in the replay attack protection, a duplicate value of the timestamp TS_i or random number RN_i will reveal that the message is a replayed one and it is not an original message. Note that $C_1 = A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j})$ and $A_{ij} = V'_{ij} \oplus RPBW_i$. As \mathcal{A} does not know ID_i , PW_i and B'_i , he/she is unable to guess correct value of A_{ij} and can not modify Msg_1 . Hence, the proposed scheme resists user impersonation attack.

TABLE 7. Actual execution time of different operations.

Symbol	Description	Execution Time (in ms)	
		HiPerSmart card (MU_i)	Pentium IV (CS_j)
T_P	Bilinear pairing operation	380	3.16
T_M	Elliptic curve point multiplication	130	1.17
T_{FE}	Fuzzy extractor operation	$\approx T_M$	$\approx T_M$
T_{sym}	Symmetric encryption/decryption	< 17.93	< 0.16
T_{GH}	map-to-point hash function	< 100	< 1
T_A	elliptic curve point addition	< 10	< 0.1
T_H	One way hash function	< 1	< 0.01
T_X	Bitwise XOR function	negligible	negligible

11) SERVER IMPERSONATION ATTACK

The proposed scheme protects server impersonation attack where an adversary \mathcal{A} can masquerade as a cloud server and try to respond with valid message to MU_i . When CS_j receives the user login message, it replies with an authorization message $Msg_2 = \{C_2, H_3, TS_j\}$. This message contains the hash value $H_3 = H(ID_i || M_1 || RN_j || TS_i || TS_j || SK_{CS_j, MU_i})$. Moreover, Msg_2 also contains $C_2 = B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i$. \mathcal{A} can not obtain $B_{ji} = A_{ij} = H(H(ID_i \oplus r_{ij}) || X_j)$ as it requires the server secret key X_j and random number r_{ij} . As a consequence, the proposed scheme also resists server impersonation attack.

12) PRIVILEGED-INSIDER ATTACK

In this attack, we assume that the registration information $ID_i, (RPBW_i \oplus k)$ from the mobile user registration request message is known to a privileged-insider user of the RC , who acts an adversary \mathcal{A} . Later, after completing the mobile user registration process, it is also assumed that \mathcal{A} also attains the stolen/lost mobile device, and then extract the information stored in the device using the power analysis attack [34]. As discussed in Section IV-C.3, it is computationally difficult task for \mathcal{A} to obtain PW_i and the biometric key θ_i

TABLE 8. Comparison of computational costs among related schemes.

Phase	Entity	He-Wang [32]	Yoon-Yoo[30]	Shen <i>et al.</i> [31]	Tsai-Lo [10]	Tseng <i>et al.</i> [29]	Our
LAP	MU_i	$7T_H + 3T_M$ ≈ 397 ms	$5T_H + 2T_M$ ≈ 265 ms	$5T_H + 3T_M$ ≈ 395 ms	$3T_H + 3T_M$ ≈ 393 ms	$3T_H + T_M$ ≈ 133 ms	$9T_H + 8T_X + T_{FE}$ ≈ 139 ms
	CS_j	$5T_H + 2T_M$ ≈ 2.39 ms	$5T_H + 2T_M$ ≈ 2.39 ms	$5T_H + 2T_M$ ≈ 2.39 ms	$2T_H + 4T_M$ $+2T_P$ ≈ 11.02 ms	$3T_H + 2T_M + T_P$ $2T_{G_H} + 2T_A$ ≈ 7.63 ms	$7T_H + 7T_X$ ≈ 0.07 ms
	RC	$9T_H + 2T_M$ ≈ 2.43 ms	$5T_H$ ≈ 0.05 ms	$7T_H + T_M$ ≈ 1.24 ms	–	–	–
Cryptographic primitive	ECC	ECC	ECC	Pairing	Pairing	Hash	

Note: LAP: Login and authentication phases

from V'_{ij} and A'_{ij} in polynomial time. Furthermore, without having the random secret k , \mathcal{A} can not compute $RPWB_i$ from $RPWB_i \oplus k$. Therefore, \mathcal{A} can not also obtain PW_i and θ_i from $RPWB_i$. Hence, the proposed scheme is free from the privileged-insider attack.

V. FORMAL SECURITY VERIFICATION USING PROVERIF

The formal security verification of the proposed scheme is presented in this section using the applied pi calculus based ProVerif simulation tool [60]. This tool can be practically used for testing whether an attacker is able to attack (or compromise) the session key in a security protocol.

In Fig. 7, we provide the code for declaration of channels, free variables, constants, functions, equations, queries and events required for the proposed scheme. The code for the process of the mobile user in the registration, login and authentication phases is modeled in Fig. 8. The process of the cloud sever CS_j is modeled as parallel composition of the process of registration (SReg) and process of authentication (SAuth). Fig. 9 shows the program code for the processes related to CS_j .

Finally, we execute the codes of the previous three tables in ProVerif latest version (i.e., ProVerif 1.93). The complete obtained results of session key secrecy (from both user and server side) and authentication are shown in Fig. 10. The result shows the following observations:

- RESULT inj-event(UserAuth(id)) ==> inj-event (UserStart(id)) is true.
- RESULT not attacker(SKus[]) is true.
- RESULT not attacker(SKus[]) is true.

Hence, the proposed scheme passes the security verification.

VI. PERFORMANCE COMPARISON

In this section, we compare the security and functionality of the proposed scheme with the recently developed multi-server authentication schemes designed for mobile cloud computing services [10], [29]–[32]. A detailed comparison on different security attacks is shown in Table 6. It is seen that a large number of the recent schemes suffer from denial of service attack and stolen mobile device attack. Further, most of the existing schemes fail to provide efficiency in login phase and password change phase, and they do not provide

TABLE 9. Comparison of communication costs.

Scheme	No. of rounds	No. of bits
He-Wang [32]	5	3520
Yoon-Yoo [30]	5	2496
Shen <i>et al.</i> [31]	5	1856
Tsai-Lo [10]	4	1696
Tseng <i>et al.</i> [29]	3	992
Our	2	864

revocation of lost mobile device phase. It is clear from Table 6 that the proposed scheme overcomes such security and functionality weaknesses of the existing schemes.

As implemented by Scott *et al.* [61] and Tseng *et al.* [29], we have considered Philips HiPersmart card device and Pentium IV computer for user side and cloud server side computation, respectively. Philips HiPersmart card has a clock speed of 36MHz with 32-bit RISC MIPS processor. It has flash memory of 256 KB with 16KB RAM. On the other side, Pentium IV has maximum clock speed of 3GHz operating under Windows XP OS with 512 MB RAM [16]. Bilinear pairing and other cryptographic operations are implemented in C language under specific IDE and specific C/C++ Library (MIRACL). Table 7 shows the notations for different cryptographic operations along with their execution time in Philips HiPersmart card device and Pentium IV computer, respectively.

In Table 8, we tabulate and compare the computation overhead of the proposed scheme with the relevant schemes [10], [29]–[32]. For all the given schemes, we separately tabulated computation for MU_i and cloud service provider CS_j under Philips HiPersmart card device and Pentium IV computer, respectively. Also, we mention the underlying cryptographic operations for each relevant scheme in comparison. We study that the total user side computation overhead of the proposed scheme in login and authentication phases is $9 * T_H + 8 * T_X + T_{FE}$. Considering the execution time needed for XOR operation is negligible, the total execution time of MU_i is then approximately $(9 * 1 + 130) = 139$ ms. On the other hand, the cloud service provider CS_j has a computation overhead of $7 * T_H + 7 * T_X$. Hence, total execution time in Pentium IV server is less than $7 * 0.01 = 0.07$ ms.

Comparison on communication cost of the proposed scheme with related mobile user authentication

schemes [10], [29]–[32] is also tabulated in Table 9. Since the user and server registration phases, password change phase and lost mobile device revocation phase are executed only once, we consider only login and authentication phases for calculation of communication cost for the proposed scheme and other schemes. The proposed scheme needs two messages $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ and $Msg_2 = \{C_2, H_3, TS_j\}$, which require $(160 + 160 + 32 + 160) = 512$ bits and $(160 + 32 + 160) = 352$ bits, respectively. So, the overall communication cost of the proposed scheme is $(512 + 352) = 864$ bits. Note that the proposed scheme does not involve *RC* during login and authentication phases, which causes significant reduction of overall communication cost. In addition, the proposed scheme requires only two rounds of message communication, whereas other related schemes require five, four or three rounds of message communication. It is observed from Table 8 that the user mobile device in He-Wang's scheme [32] and Yoon-Yoo's scheme [30] takes ≈ 397 ms and ≈ 265 ms, respectively. Shen *et al.*'s scheme [31] and Tsai-Lo's scheme [10] take ≈ 395 ms and ≈ 393 ms, respectively. Reason behind the high computation cost in the existing schemes is that they either use ECC based cryptosystem or bilinear pairing based cryptosystem. Quite clearly the user side computation cost of the proposed scheme is much less than that for these schemes. As a result, the proposed scheme comparatively more suited for the mobile users with low-power computing devices.

VII. CONCLUSION

Before providing any access of cloud service to a mobile user, mutual authentication of a mobile user and cloud service provider is necessary. Authentication scheme should be lightweight with respect to resource constrained user mobile device. In this paper, we have proposed a mobile user authentication scheme on mobile cloud computing, which is based on cryptographic hash, bitwise XOR and fuzzy extractor functions only. We have provided the formal security proof through the ROR model and also the formal security verification through the ProVerif 1.93 simulation tool. Moreover, mutual authentication proof is provided by the BAN logic. Since the proposed scheme does not exploit any resource constrained cryptosystem, it has the low computation cost as compared to that for the existing related schemes. As the proposed scheme does not involve the *RC* in the authentication process, it has also low communication cost as compared to that for the existing related schemes. Overall, high security and low communication and computation costs make the proposed scheme very suitable for the practical applications in the mobile cloud computing domain.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable feedback on the paper which helped us to improve its quality and presentation.

REFERENCES

- [1] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 337–368, 1st Quart., 2014.
- [2] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: A survey, state of art and future directions," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 133–143, 2014.
- [3] *Heavy Reading Real World Research (2013) The Mobile Cloud Market Outlook to 2017*. Accessed: Jul. 2017. [Online]. Available: <http://www.snjtoday.com/story/36836393/global-mobile-cloud-market-2017-global-production-growth-share-demand-and-applications-market-research-report-to-2022>
- [4] ABI Research Report. *Mobile Cloud Applications*. Accessed: Jul. 2017. [Online]. Available: <http://www.abiresearch.com/research/1003385-Mobile+Cloud+Computing>
- [5] L. Wei *et al.*, "Security and privacy for storage and computation in cloud computing," *Inf. Sci.*, vol. 258, pp. 371–386, Feb. 2014.
- [6] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," *Inf. Sci.*, vol. 305, pp. 357–383, Jun. 2015.
- [7] J. Zhou, Z. Cao, X. Dong, N. Xiong, and A. V. Vasilakos, "4S: A secure and privacy-preserving key management scheme for cloud-assisted wireless body area network in m-healthcare social networks," *Inf. Sci.*, vol. 314, pp. 255–276, Sep. 2015.
- [8] J. Zhou, X. Dong, Z. Cao, and A. V. Vasilakos, "Secure and privacy preserving protocol for cloud-based vehicular DTNs," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1299–1314, Jun. 2015.
- [9] Z. Yan, X. Li, M. Wang, and A. Vasilakos, "Flexible data access control based on trust and reputation in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 5, no. 3, pp. 485–498, Jul./Sep. 2017, doi: 10.1109/TCC.2015.2469662.
- [10] J. L. Tsai and N.-W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Syst. J.*, vol. 9, no. 3, pp. 805–815, Sep. 2015.
- [11] OpenID Foundation. *The OpenID User Interface Extension Best Practices for Identity Providers 2009*. Accessed on Jul. 2017. [Online]. Available: <http://wiki.openid.net/w/page/12995153/Details-of-UXBest-Practices-for-OPs>
- [12] Google. (2008). *SAML Single Sign-On (SSO) Service for Google Apps*. Accessed: Jul. 2017. [Online]. Available: <https://support.google.com/a/answer/60224?hl=en>
- [13] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.
- [14] G. Yang, D. S. Wong, and X. Deng, "Anonymous and authenticated key exchange for roaming networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 9, pp. 3461–3472, Sep. 2007.
- [15] V. Odelu, A. K. Das, and A. Goswami, "A secure and efficient ECC-based user anonymity preserving single sign-on scheme for distributed computer networks," *Secur. Commun. Netw.*, vol. 8, no. 9, pp. 1732–1751, 2015.
- [16] H. J. Jo, J. H. Paik, and D. H. Lee, "Efficient privacy-preserving authentication in wireless mobile networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1469–1481, Jul. 2014.
- [17] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, G. Pellegrino, and A. Sorniotti, "An authentication flaw in browser-based single sign-on protocols: Impact and remediations," *Comput. Secur.*, vol. 33, pp. 41–58, Mar. 2013.
- [18] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions," *Comput. Netw.*, vol. 73, pp. 41–57, Nov. 2014.
- [19] D. Wang and P. Wang, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks," *Ad Hoc Netw.*, vol. 20, pp. 1–15, Sep. 2014.
- [20] D. Wang, D. He, P. Wang, and C. H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Depend. Sec. Comput.*, vol. 12, no. 4, pp. 428–442, Jul. 2015.
- [21] X. Huang, X. Chen, J. Li, Y. Xiang, and L. Xu, "Further observations on smart-card-based password-authenticated key agreement in distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1767–1775, Jul. 2014.
- [22] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Trans. Depend. Secure Comput.*, vol. 11, no. 6, pp. 568–581, Nov./Dec. 2014.

- [23] C.-G. Ma, D. Wang, and S.-D. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *Int. J. Commun. Syst.*, vol. 27, no. 10, pp. 2215–2227, 2014.
- [24] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: Preserving security and privacy in distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 8, pp. 1390–1397, Aug. 2011.
- [25] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Depend. Secure Comput.*, to be published, doi: [10.1109/TDSC.2016.2605087](https://doi.org/10.1109/TDSC.2016.2605087).
- [26] D. Wang, Q. Gu, H. Cheng, and P. Wang, "The request for better measurement: A comparative evaluation of two-factor authentication schemes," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur. (ASIA CCS)*, Xi'an, China, 2016, pp. 475–486.
- [27] V. Odelu, A. K. Das, S. Kumari, X. Huang, and M. Wazid, "Provably secure authenticated key agreement scheme for distributed mobile cloud computing services," *Future Generat. Comput. Syst.*, vol. 68, pp. 74–88, Mar. 2017.
- [28] P. Gope and A. K. Das, "Robust anonymous mutual authentication scheme for n-times ubiquitous mobile cloud computing services," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1764–1772, Oct. 2017, doi: [10.1109/JIOT.2017.2723915](https://doi.org/10.1109/JIOT.2017.2723915).
- [29] Y. M. Tseng, S. S. Huang, T. T. Tsai, and J. H. Ke, "List-free ID-based mutual authentication and key agreement protocol for multi-server architectures," *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 1, pp. 102–112, Jan. 2016.
- [30] E.-J. Yoon and K.-Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *J. Supercomput.*, vol. 63, no. 1, pp. 235–255, Jan. 2013.
- [31] H. Shen, C. Gao, D. He, and L. Wu, "New biometrics-based authentication scheme for multi-server environment in critical systems," *J. Ambient Intell. Humanized Comput.*, vol. 6, no. 6, pp. 825–834, 2015.
- [32] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Syst. J.*, vol. 9, no. 3, pp. 816–823, Sep. 2015.
- [33] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.
- [34] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.
- [35] W. Stallings, *Cryptography and Network Security: Principles and Practices*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003.
- [36] P. Sarkar, "A simple and generic construction of authenticated encryption with associated data," *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 4, 2010, Art. no. 33.
- [37] *Secure Hash Standard*, document FIPS PUB 180-1, U.S. Department of Commerce, National Institute of Standards and Technology (NIST), Apr. 1995. Accessed: Dec. 2016. [Online]. Available: <http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>
- [38] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology*. Interlaken, Switzerland: Springer, 2004, pp. 523–540.
- [39] K. Simoons, J. Bringer, H. Chabanne, and S. Seys, "A framework for analyzing template security and privacy in biometric authentication systems," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 833–841, Apr. 2012.
- [40] Q. Zhang, Y. Yin, D.-C. Zhan, and J. Peng, "A novel serial multimodal biometrics framework based on semisupervised learning techniques," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1681–1694, Oct. 2014.
- [41] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things," *IEEE Internet Things J.*, to be published, doi: [10.1109/JIOT.2017.2714179](https://doi.org/10.1109/JIOT.2017.2714179).
- [42] M. Wazid, A. K. Das, N. Kumar, and J. P. C. Rodrigues, "Secure three-factor user authentication scheme for renewable energy based smart grid environment," *IEEE Trans. Ind. Informat.*, to be published, doi: [10.1109/TII.2017.2732999](https://doi.org/10.1109/TII.2017.2732999).
- [43] M. Wazid, A. K. Das, N. Kumar, M. Conti, and A. V. Vasilakos, "A novel authentication and key agreement scheme for implantable medical devices deployment," *IEEE J. Biomed. Health Informat.*, to be published, doi: [10.1109/JBHI.2017.2721545](https://doi.org/10.1109/JBHI.2017.2721545).
- [44] A. K. Das, A. K. Sutrala, S. Kumari, V. Odelu, M. Wazid, and X. Li, "An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 2070–2092, 2016.
- [45] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment," *IEEE Trans. Depend. Secure Comput.*, to be published, doi: [10.1109/TDSC.2016.2616876](https://doi.org/10.1109/TDSC.2016.2616876).
- [46] S. Challa, A. K. Das, S. Kumari, V. Odelu, F. Wu, and X. Li, "Provably secure three-factor authentication and key agreement scheme for session initiation protocol," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5412–5431, 2016.
- [47] A. K. Das, "Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards," *IET Inf. Secur.*, vol. 5, no. 3, pp. 145–151, Sep. 2011.
- [48] X. Li, J.-W. Niu, J. Ma, W.-D. Wang, and C.-L. Liu, "Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 73–79, 2011.
- [49] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. 8th Int. Workshop Theory Pract. Public Key Cryptograph. (PKC)*, 2005, pp. 65–84.
- [50] M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu, "Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS," *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 1983–2001, 2016.
- [51] H. Yang, Y. Zhang, Y. Zhou, X. Fu, H. Liu, and A. V. Vasilakos, "Provably secure three-party authenticated key agreement protocol using smart cards," *Comput. Netw.*, vol. 58, pp. 29–38, Jan. 2014.
- [52] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. 1st ACM Conf. Comput. Commun. Secur. (CCS)*, Fairfax, VA, USA, 1993, pp. 62–73.
- [53] V. Shoup. (2004). "Sequences of games: A tool for taming complexity in security proofs," *Cryptol. ePrint Arch.*, Tech. Rep. 2004/332. [Online]. Available: <http://eprint.iacr.org/2004/332>
- [54] D. Pointcheval and S. Zimmer, "Multi-factor authenticated key exchange," in *Proc. 6th Int. Conf. Appl. Cryptograph. Netw. Secur. (ACNS)*, 2008, pp. 277–295.
- [55] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.
- [56] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2012, pp. 538–552.
- [57] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, "Targeted online password guessing: An underestimated threat," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Vienna, Austria, 2016, pp. 1242–1254.
- [58] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.
- [59] P. F. Syverson and I. Cervesato, "The logic of authentication protocols," in *Foundations of Security Analysis and Design (Lecture Notes in Computer Science)*, vol. 2171. Bertinoro, Italy: Springer, 2001, pp. 63–137.
- [60] M. Abadi, B. Blanchet, and H. Comon-Lundh, "Models and proofs of protocol security: A progress report," in *Proc. 21st Int. Conf. Comput. Aided Verification (CAV)*, 2009, pp. 35–49.
- [61] M. Scott, N. Costigan, and W. Abdulwahab, "Implementing cryptographic pairings on smartcards," in *Proc. 8th Int. Workshop Cryptograph. Hardw. Embedded Syst. (CHES)*, 2006, pp. 134–147.



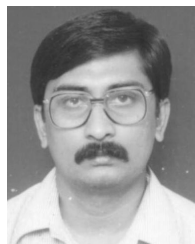
SANDIP ROY received the M.Tech. degree in computer science and technology from the West Bengal University of Technology, India. He is currently pursuing the Ph.D. degree in computer science and engineering from Jadavpur University, Kolkata, India. He is also an Assistant Professor with the Department of Computer Science and Engineering, Asansol Engineering College, India. He has authored five international journal and conference papers in his area of research. His current

research interests include cryptography, wireless sensor network security, and access control.



over 20 papers in international journals and conferences.

SANTANU CHATTERJEE received the Ph.D. degree in computer science and engineering and the master's degree in computer science and engineering from Jadavpur University, India. He is currently a Scientist with the Research Center Inmarat in Directorate of ICT, Defence Research and Development Organization, Hyderabad, India. His current research interests include cryptography, wireless sensor network security, data mining, and enterprise resource planning. He has authored



industry houses. He has authored over 110 papers in international journals and conferences.

SAMIRAN CHATTOPADHYAY received the bachelor's and master's degrees in computer science and engineering from IIT Kharagpur, India, and the Ph.D. degree from Jadavpur University, Kolkata, India. He is currently a Professor with the Department of Information Technology, Jadavpur University, Kolkata, India. He is having over 25 years of teaching experience with Jadavpur University, four years of industry experience, and 12 years of technical consultancy in the reputed



architectural access control, security in vehicular ad hoc networks, smart grid, Internet of Things (IoT), cyber-physical systems and cloud computing, and remote user authentication. He has authored over 145 papers in international journals and conferences in the above areas. Some of his research findings are published in top cited journals, such as the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON SMART GRID, the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, the IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS (Formerly, IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE), the *IEEE Consumer Electronics Magazine*, the IEEE ACCESS, the *IEEE Communications Magazine*, the *Future Generation Computer Systems*, and the *Journal of Network and Computer Applications*. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He is on the Editorial Board of the *KSII Transactions on Internet and Information Systems* and the *International Journal of Internet Technology and Secured Transactions* (Inderscience), and a Guest Editor of the *Computers and Electrical Engineering* (Elsevier) for the special issue on Big data and IoT in e-healthcare, and has served as a Program Committee Member in many international conferences.

ASHOK KUMAR DAS (M'17) received the M.Sc. degree in mathematics, the M.Tech. degree in computer science and data processing, and the Ph.D. degree in computer science and engineering from IIT Kharagpur, India. He is currently an Assistant Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. His current research interests include cryptography, wireless sensor network security, hier-



the IEEE, the Elsevier, the Springer, and the John Wiley. Some of his research findings are published in top cited journals, such as the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, THE IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, THE IEEE NETWORK, THE IEEE COMMUNICATIONS, THE IEEE WIRELESS COMMUNICATIONS, THE IEEE INTERNET OF THINGS JOURNAL, and the IEEE SYSTEMS JOURNAL. He has guided many research scholars leading to Ph.D. and M.E./M.Tech. He is on the editorial board of the *Journal of Network and Computer Applications* (Elsevier) and the *International Journal of Communication Systems* (Wiley).

NEERAJ KUMAR (M'16–SM'17) received the Ph.D. degree in computer science and engineering from Shri Mata Vaishno Devi University, Katra, India, in 2009. He was a Post-Doctoral Research Fellow with Coventry University, Coventry, U.K. He is currently an Associate Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, India. He has authored over 160 technical research papers authored in leading journals and conferences from



INFORMATION TECHNOLOGY IN BIOMEDICINE, the *ACM Transactions on Autonomous and Adaptive Systems*, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. He is also the General Chair of the European Alliances for Innovation.

ATHANASIOS V. VASILAKOS is currently a Chair Professor with Innopolis University. He served or is serving as an Editor for many technical journals, such as the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON NANOBIOSCIENCE, the IEEE TRANSACTIONS ON

• • •