Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 2098530, 20 pages
https://doi.org/10.1155/2021/2098530

Research Article

Secure Three-Factor Anonymous User Authentication Scheme for Cloud Computing Environment

Hakjun Lee,[1] **Dongwoo Kang**,[2] **Youngsook Lee**,[1] **and Dongho Won**[2]

[1]*Department of IT Software and Security, Howon University, Gunsan, Republic of Korea*
[2]*Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Republic of Korea*

Correspondence should be addressed to Dongho Won; dhwon@security.re.kr

Received 21 May 2021; Revised 28 June 2021; Accepted 5 July 2021; Published 27 July 2021

Academic Editor: Chien-Ming Chen

Cloud computing provides virtualized information technology (IT) resources to ensure the workflow desired by user at any time and location; it allows users to borrow computing resources such as software, storage, and servers, as per their needs without the requirements of complicated network and server configurations. With the generalization of small embedded sensor devices and the commercialization of the Internet of Things (IoT), short- and long-range wireless network technologies are being developed rapidly, and the demand for deployment of cloud computing for IoT is increasing significantly. Cloud computing, together with IoT technology, can be used to collect and analyse large amounts of data generated from sensor devices, and easily manage heterogeneous IoT devices such as software updates, network flow control, and user management. In cloud computing, attacks on users and servers can be a serious threat to user privacy. Thus, various user authentication schemes have been proposed to prevent different types of attacks. In this paper, we discuss the security and functional weakness of the related user authentication schemes used in cloud computing and propose a new elliptic curve cryptography- (ECC-) based three-factor authentication scheme to overcome the security shortcomings of existing authentication schemes. To confirm the security of the proposed scheme, we conducted both formal and informal analyses. Finally, we compared the performance of the proposed scheme with those of related schemes to verify that the proposed scheme can be deployed in the real world.

1. Introduction

With the significant advances in information technology (IT), numerous types of devices can connect to the Internet and have a variety of features that can be used for different purposes. Using these devices with wireless network technologies, such as Wi-Fi, Bluetooth, 5G, 6lowPAN, and LoRa, has allowed the practical deployment of Internet of Things (IoT) [1]. IoT enables the networking of various types of embedded devices, such as home, mobile, and wearable devices, allowing them to communicate with people and objects at any time and location in our daily lives [2].

With combined with cloud computing, IoT technologies can collect and analyse large amounts of data from devices connected to an IoT network with hyperconnectivity and hyperintelligence surpassing the limits of time and space in various areas such as urban life, traffic, welfare, safety, healthcare, manufacturing, energy, finance, and logistics [3, 4]. Cloud computing provides IT resources on demand over the Internet. Cloud computing providers are building and maintaining physical data centers and servers, and users can enjoy the benefits of custom cloud services for greater computing power, storage, and database.

With such advantages, cloud computing is becoming a paradigm for the processing, storage, and utilization of large amounts of data generated by billions of smart devices because it can overcome the limitations of such devices, including low capacity and limited processing capability. The integration of IoT with cloud computing allows for better scalability, interoperability, reliability, efficiency, availability, and security through the utilization of various devices and technologies [5]. In addition, it provides benefits such as easy access, use, and deployment-cost reductions. A cloud computing environment can serve as a stable network environment for connection with IoT devices and provide storage for big data generated from IoT devices to securely keep and process

the data for analysis. With these advantages, both individuals and small companies can benefit from cloud services.

In general, there are three types of cloud services [6]:

(1) Infrastructure as a service (IaaS): providing the user with an infrastructure including storage and network use

(2) Platform as a service (PaaS): providing the user with a platform to develop various applications

(3) Software as a service (SaaS): providing the user with software applications

Because cloud computing is deployed in a practical manner, there have been growing concerns regarding its security. As mentioned earlier, clouds are used in various industries and services; thus, cloud servers can collect and process sensitive data, and it can seriously affect user privacy.

Security issues associated with cloud computing include various aspects such as embedded security, application security, trust and conviction, client management, cloud data storage, and operating systems. Among the different security requirements, the first security requirement for the protection of user privacy is user authentication that verifies a user's identity with a trusted party. There are three authentication factors used to verify the user identity: (i) what you know (e.g., secret information such as a password), (ii) what you have (e.g., things we own such as smart cards), and (iii) who you are (e.g., biometric such as fingerprint or iris data) [7].

In recent years, various user authentication schemes have been proposed [8–12], and user authentication studies using various cryptographic primitives have been proposed to protect a user's personal information in a cloud environment. However, the investigation into such studies has revealed that the level of security is still insufficient to authenticate and manage users in the current cloud computing environment. Therefore, in this paper, to strengthen the security of previously proposed schemes, we first report problems in the related authentication scheme used in a cloud computing environment and then propose a new authentication scheme to overcome these problems.

1.1. Motivations. Since Lamport [13] first proposed a password-based authentication scheme, many relevant studies on suitable two-factor authentication schemes in various network environments have been proposed to protect user privacy. After the introduction of cloud computing systems, authentication schemes using various encryption technologies, including the Advanced Encryption Standard (AES), hash function, Chebyshev polynomials, and Elliptic Curve Cryptography (ECC), began to be studied to provide secure user authentication and improve security and efficiency.

Amin et al. [14] proposed a user authentication scheme for a distributed IoT cloud environment. However, Wang et al. [15] found that Amin et al.'s scheme has some weaknesses—it is vulnerable to a stolen smart card attack, violates user anonymity and forward secrecy, has a time synchronization problem, and provides an insecure identity update phase. Wang et al. [15] also proposed a new authentication scheme to eliminate the security concern associated with Amin et al.'s scheme [14] by applying ECC to share the session key between the user and the cloud server. Nevertheless, their scheme does not provide a session key verification at the end of the authentication phase—an invalid session key may be generated between the user and the cloud server without detecting communication errors that may occur while sharing the parameters for the establishment of a session key.

In 2017, Kumari et al. [16] proposed a biometrics-based three-factor authentication scheme in a multicloud server environment using ECC and proved that the scheme is secure for cloud computing environments, but it does not provide an identity update phase.

In 2019, Zhou et al. [17] proposed a lightweight authentication scheme for an IoT-cloud architecture using only hash and exclusive-OR (XOR) operations; it is relatively lightweight in comparison with other schemes [14, 18] and satisfies some of the security properties required for cloud computing. However, Martínez-Peláez et al. [19] reported that Wang et al.'s scheme [15] has security vulnerabilities to insider attacks, man-in-the-middle attacks through a replay attack, and user impersonation attacks. Martínez-Peláez et al. [19] then proposed a new lightweight authentication scheme to provide secure access to user by improving the scheme developed by Zhou et al. [17]. However, Yu et al. [20] found that Martínez-Peláez et al.'s scheme [19] is vulnerable to impersonation attacks, session key-disclosure attacks, and replay attacks and that it does not ensure user anonymity. Yu et al. [20] then proposed a lightweight three-factor-based authentication scheme for IoT use in a cloud computing environment to enhance the level of security. In their scheme, the cloud server changes the identity of the user during each session. However, users cannot recover or update their own identity themselves.

In this paper, based on the same network model used in the abovementioned related schemes for cloud computing, we propose a new three-factor user authentication scheme to enhance the level of security and efficiently manage users by eliminating the security and functional flaws of the related schemes. In the proposed scheme, we selected the ECC from various cryptographic building blocks, which has various advantages. For example, the safety of the ECC system increases exponentially with the key length and has a shorter key length and faster operation speed than those of the RSA algorithm. This is particularly effective in applications where the processing capacity is limited; these include memory, smart cards, and wireless communication terminals [21]. ECC has been standardized for digital signature algorithms and key exchanges (e.g., ANSI X9.62 and X9.63) and is widely accepted in various network communication standards such as IPsec (RFC 2409) and TLS (RFC 4492).

1.2. Organization of the Paper. The remainder of this paper is organized as follows. Section 2 presents the preliminaries for security considerations and background of the network model. In Section 3, we detail a secure three-factor anonymous user authentication scheme for a cloud computing environment. We describe the informal and formal security analyses in Sections 4 and 5, respectively. In Section 6, we

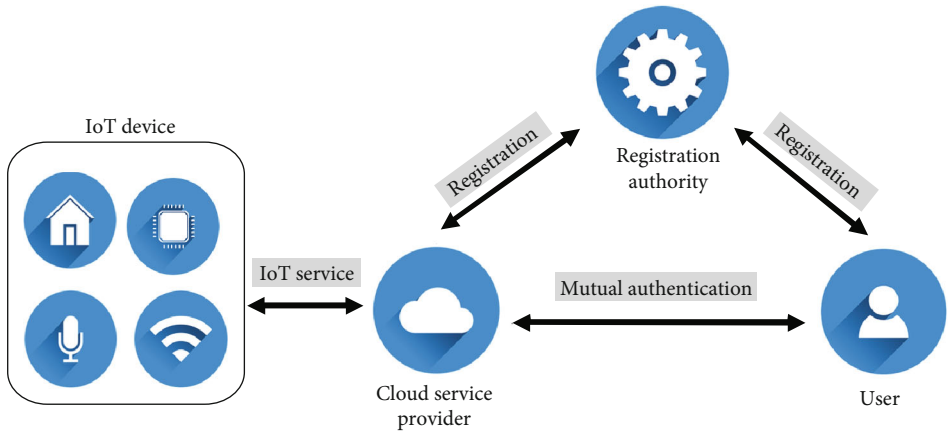

Figure 1: System architecture of cloud computing service with IoT.

evaluate the performance of the proposed scheme. Finally, we provide some concluding remarks in Section 7.

2. Preliminary

2.1. Network Model. The network model of the proposed protocol in the IoT environment is based on the cloud server environment adopted in the protocol described in [16–20], as shown in Figure 1. There are three participants in this model:

(1) Registration authority (RA): RA is a trusted authority that creates all system parameters and issues them to users and cloud servers through the registration process

(2) User (U_i): U_i wishes to access and enjoy the services provided by the cloud server using IoT device. For this purpose, the RA shares the session key with the cloud server

(3) Cloud server (CS_j): CS_j provides IoT cloud services to users

This network model is for a cloud server-centric service in which the cloud server collects and processes information from IoT devices and shares it with users. For example, a real-world scenario for this is as follows: Alex's grandfather has dementia, and his family is concerned about his grandfather's health and fear of getting lost when he goes out. The smartwatch worn by the grandfather can check the health condition through the built-in sensor and transmits the GPS information to the cloud server. Alex's family wants to use a service that can trace and check the location and health of their grandfather in real time. To this end, Alex's family and grandfather (U_i) first register their identifier with the registration authority (RA) to sign up for this IoT-based cloud service. RA issues security parameters to be used when establishing session key with CS_j (assume that CS_j is already registered). Information about the grandfather is sensitive. It should be shared only with the family. To this end, Alex creates a group through the interface provided by the cloud platform, adds family members as group members, and adds the devices of grandpa's smartwatch and family's smartphone to set permission to access information shared by registered devices. U_i and CS_j generate the session key through the authentication process; the family and grandfather can safely share information.

2.2. Elliptic Curve Cryptography. In this study, we apply an elliptic curve cryptography to the proposed scheme, which provides a high level of security with a small key size [22]. ECC is based on the logarithm problems expressed in the point addition and multiplication of elliptic curves.

An elliptic curve is given by $E_p(a, b)$: $y^2 = x^3 + ax + b \bmod p$ over a finite field F_p, where p is the prime order and $a, b \in F_p$ such that $p > 3$ and $4a^3 + 27b^2 \neq 0 \bmod p$. The point multiplication over $E_p(a, b)$ is defined through a repetitive addition as $+P + \cdots + P(a \text{ times}) = aP$, where P is a point on $E_p(a, b)$ and $a \in F_p^*$ is a random integer. The security of ECC relies on the following assumption:

(1) Elliptic curve discrete logarithm problem (ECDLP): given $P, aP \in E_p(a, b)$, it is computationally infeasible to find a within polynomial time

(2) Elliptic curve computational Diffie-Hellman problem (ECCDHP): given $aP, bP \in E_p(a, b)$, it is computationally infeasible to find abP in polynomial time

2.3. Bio-Hash Function. In the proposed scheme, we use a bio-hash function. In 2004, Jin et al. [23] proposed a solution to the problem of false resection in which a genuine user is misidentified for various reasons, such as when experiencing dry or cracked skin. The bio-hash maps the biometric features to a binary string with a user-specific tokenized pseudo-random number. In three-factor authentication, many researchers use a bio-hash to identify the biometric features of the users [24–26]. It is a simple and efficient tool for resource-constrained devices such as IoT sensor devices.

2.4. Adversarial Model. For a security analysis in this paper, we consider the adversarial model as follows [27–29]:

(1) The attacker can control the public communication channel by interrupting, returning, amending, eliminating, or transmitting newly forged messages

(2) The attacker can extract the security parameters in the smart device using a side-channel attack

(3) The attacker can guess the user's identity and password by enumerating all possible items in polynomial time. The time of such an attack conducted to determine the correct identity and password is linear to the dictionary size

3. Proposed Scheme

In this section, we propose an improved three-factor authentication scheme in the cloud environment. Our scheme consists of (1) a registration phase, (2) a login and authentication phase, (3) a password change phase, and (4) an identity update phase. All notations used in this paper are listed in Table 1.

Table 1: Notations used.

Symbol	Description
U_i, CS_j, RA	User, cloud server, registration authority
ID_i, CID_j	Identity of U_i and CS_j
PW_i	Password of U_i
BIO_i	Biometrics of U_i
SK	Session key between U_i and CS_j
$h(\cdot)$	Hash function
$H(\cdot)$	Bio-hash function
$\|\|$	Concatenation
$\oplus$	XOR operation
x	Private key of RA
$xP = \text{Pub}$	Public key of RA

3.1. User Registration Phase. In this phase, U_i registers with RA and shares secret parameters for later login and authentication using IoT smart device. The registration phase of U_i shown in Figure 2 is as follows.

(1) U_i who wants to register in RA enters ID_i, PW_i, and BIO_i, and computes $PWB_i = h(PW_i \| BIO_i)$ and $UID_i = h(ID_i \| PWB_i)$

(2) U_i sends a registration request $<ID_i, PWB_i, UID_i, >$ through a secure channel

(3) After receiving the registration request message from U_i, RA first searches for UID_i in User_List to check whether the user's ID_i is already registered. If UID_i does not exist in User_List, RA selects a random number r and computes $D_i = h(UID_i \| x \| r)$, $C_i = h(ID_i \| PWB_i \| D_i)$, and $E_i = D_i \oplus PWB_i$

(4) Here, RA stores $<UID_i, r, \text{Honey\_List} = 0>$ into User_List and issues $<C_i, E_i, \text{Pub}, P>$. Then, U_i stores them to IoT smart device

3.2. Registration Phase for Cloud Server. In this phase, CS_j registers with RA and initials the system parameter. The registration phase of CS_j shown in Figure 3 is as follows.

(1) CS_j selects CID_j and sends it to RA

(2) RA computes $Ckey_j = h(CID_j \| x)$ and sends it to CS_j

3.3. Login and Authentication Phase. To access a cloud server CS_j, U_i begins a login and authentication protocol on the public channel through the support of RA. In this phase, RA confirms the legitimacy of U_i and CS_j; thus, they establish a session key for future communication. To this end, the following steps, shown in Figure 4, are executed.

(1) U_i enters ID_i, PW_i, and BIO_i, and computes $PWB_i = h(PW_i \| BIO_i)$, $D_i^* = PWB_i \oplus E_i$, $UID_i = h(ID_i \| PWB_i)$, and $C_i^* = h(ID_i \| PWB_i \| D_i^*)$. If C_i^* and C_i are not equal, U_i terminates the login phase; otherwise, it chooses a random number a, and computes $X_1 = a\text{Pub}$, $X_2 = aP$, $G_i = h(UID_i \| CID_j \| X_1 \| D_i^*)$, and $F_i = h(X_1 \| X_2) \oplus UID_i$

(2) U_i sends the login request message $M_1 = <CID_j, G_i, F_i, X_2>$ to CS_j

(3) CS_j chooses a random number b and computes $Y_1 = b\text{Pub}$, $Y_2 = bP$, and $K_j = h(Y_1 \| Ckey_j \| G_i \| F_i \| X_2)$

(4) CS_i sends the login request message $M_2 = <M_1, K_j, Y_2>$ to RA

(5) RA computes $X_1^* = xX_2$, $UID_i^* = F_i \oplus h(X_1^* \| X_2)$, $D_i^* = h(UID_i \| x \| r)$, $G_i^* = h(UID_i^* \| CID_j \| X_1^* \| D_i^*)$. If G_i^* and G_i are not equal, RA terminates the login phase; otherwise, it computes $Y_1^* = xY_2$, $Ckey_j^* = h(CID_j \| x)$, and $K_j^* = h(Y_1^* \| Ckey_j^* \| G_i^* \| F_i \| X_2)$. If K_i^* and K_i are not equal, RA terminates the login phase; otherwise, it computes $U_{cs} = h(Y_1^* \| Y_2 \| Ckey_j^* \| K_j^* \| X_2)$ and $V_{cs} = h(X_1^* \| X_2 \| UID_i \| D_i^* \| Y_2)$

(6) RA sends the response message $M_3 = <U_{CS}, V_{CS}>$ to CS_j

(7) CS_i computes $V_{cs}^* = h(Y_1 \| Y_2 \| Ckey_j \| K_j \| X_2)$ and checks whether V_{cs}^* and V_{cs} are equal. If they are equal, CS_i computes $Y_3 = bX_2$, $H_j = U_{cs} \oplus H(Y_1 \| Y_2 \| Ckey_j)$, $SK_{ji} = h(Y_2 \| X_2 \| Y_3 \| H_j)$, and $SV_j = h(SK_{ji} \| X_2)$

(8) CS_i sends the message $M_4 = <Y_2, SV_j>$ to U_i

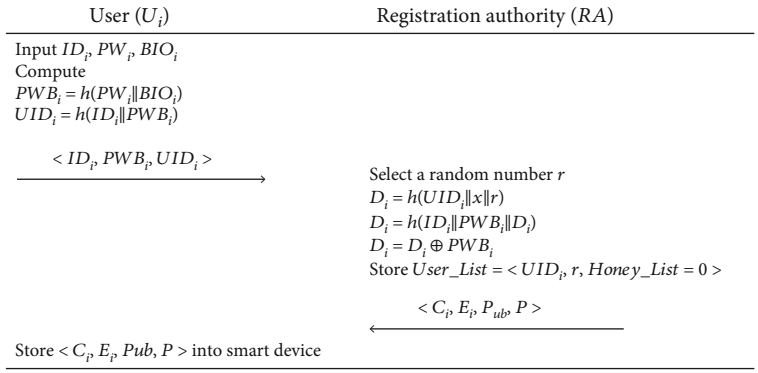

Figure 2: User registration phase of the proposed scheme.

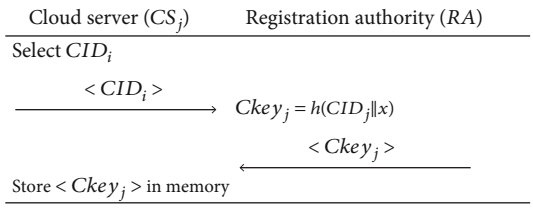

Figure 3: Registration phase for cloud server of the proposed scheme.

(9) U_i computes $X_3 = aY_2$, $\mathrm{SK}_{ij} = h(Y_2 \| X_2 \| X_3 \| \mathrm{UID}_i)$, and $\mathrm{SV}_i = h(\mathrm{SK}_{ij} \| X_2)$. If SV_i and SV_j are equal, U_i and CS_i have successfully shared the session key with each other; otherwise, the login authentication step has failed

3.4. Password Change Phase. In this phase, U_i can change to a new password offline. The password change phase shown in Figure 5 is as follows:

(1) U_i enters ID_i, PW_i, $\mathrm{PW}_i^{\mathrm{new}}$, and BIO_i, and computes $\mathrm{PWB}_i = h(\mathrm{PW}_i \| \mathrm{BIO}_i)$, $D_i = \mathrm{PWB}_i \oplus E_i$, and $C_i^* = h(\mathrm{ID}_i \| \mathrm{PWB}_i \| D_i)$

(2) If C_i^* and C_i are equal, U_i updates C_i and E_i with C_i^{new} and E_i^{new} by calculating $\mathrm{PWB}_i^{\mathrm{new}} = h(\mathrm{PW}_i^{\mathrm{new}} \| \mathrm{BIO}_i)$, $C_i^{\mathrm{new}} = h(\mathrm{ID}_i \| \mathrm{PWB}_i^{\mathrm{new}} \| D_i)$, and $E_i^{\mathrm{new}} = D_i \oplus \mathrm{PWB}_i \oplus \mathrm{PWB}_i^{\mathrm{new}}$, respectively

3.5. Identity Update Phase. When users want to change their identity or phone number, they need to update their identity. In the proposed scheme, U_i can perform an identity update process through RA by entering the old identity ID_i and new identity $\mathrm{ID}_i^{\mathrm{new}}$, shown in Figure 6, as follows:

(1) U_i enters ID_i, $\mathrm{ID}_i^{\mathrm{new}}$, PW_i, and BIO_i, and computes $\mathrm{PWB}_i = h(\mathrm{PW}_i \| H(\mathrm{BIO}_i))$, $D_i^* = \mathrm{PWB}_i \oplus E_i$, and $C_i^* = h(ID_i \| \mathrm{PWB}_i \| D_i^*)$

(2) If C_i^* and C_i are equal, U_i chooses a random number a, and computes $X_1 = a\mathrm{Pub}$, $X_2 = aP$, $\mathrm{UID}_i^{\mathrm{new}} = h(\mathrm{ID}_i^{\mathrm{new}} \| \mathrm{PWB}_i)$, $G_i = h(\mathrm{UID}_i \| X_1 \| D_i \| \mathrm{UID}_i^{\mathrm{new}})$, $F_i = h(X_1 \| X_2) \oplus \mathrm{UID}_i$, and $\mathrm{NUID}_i = h(X_1 \| X_2) \oplus \mathrm{UID}_i^{\mathrm{new}}$

(3) U_i sends the request message $M_1 = <G_i, F_i, X_2, \mathrm{NUID}_i>$ to RA

(4) RA computes $X_1^* = xX_2$, $\mathrm{UID}_i^* = F_i \oplus h(X_1^* \| X_2)$, $D_i^* = h(\mathrm{UID}_i \| x \| r)$, $\mathrm{UID}_i^{\mathrm{new}} = h(X_1 \| X_2) \oplus \mathrm{NUID}_i$, and $G_i^* = h(\mathrm{UID}_i^* \| X_1^* \| D_i^* \| \mathrm{UID}_i^{\mathrm{new}})$. If G_i^* and G_i are not equal, RA rejects the request and sets Honey_List = Honey_List + 1. Once it exceeds the present value, U_i is suspended

(5) Otherwise, RA updates UID_i with $\mathrm{UID}_i^{\mathrm{new}}$ in User_List and computes $D_i^{\mathrm{new}} = h(\mathrm{UID}_i^{\mathrm{new}} \| x \| r)$, $M_{\mathrm{CS}} = h(\mathrm{UID}_i^{\mathrm{new}} \| \mathrm{UID}_i^* \| D_i^* \| D_i^{\mathrm{new}} \| D_i^* \| X_1^*)$, and $\mathrm{ND}_i = D_i^{\mathrm{new}} \oplus h(X_1^*)$

(6) RA sends $M_2 = <M_{\mathrm{CS}}, \mathrm{ND}_i>$ to U_i

(7) U_i computes $D_i^{\mathrm{new}} = \mathrm{ND}_i \oplus h(X_1)$ and $M_{\mathrm{CS}}^* = h(\mathrm{UID}_i^{\mathrm{new}} \| \mathrm{UID}_i \| D_i^{\mathrm{new}} \| D_i^* \| X_1)$ and checks whether M_{CS}^* and M_{CS} are equal. If so, U_i computes $C_i^{\mathrm{new}} = h(\mathrm{ID}_i^{\mathrm{new}} \| \mathrm{PWB}_i \| D_i^{\mathrm{new}})$, and $E_i^{\mathrm{new}} = D_i^{\mathrm{new}} \oplus \mathrm{PWB}_i$, and updates C_i and E_i with C_i^{new} and D_i^{new}, respectively

4. Security Comparison with the Related Scheme

This section provides a security comparison with other relevant schemes [15, 16, 19, 30]. For detailed procedures of the compared schemes, see Appendices A, B, C, and D.

(1) Wang et al. [15]: Wang et al. scheme does not provide the verification process of session key. In their scheme, after CS_j sends $<Y_2, Q_{\mathrm{cs}}>$ and U_i check the legitimacy of Q_{cs}, U_i generates a session key $\mathrm{SK}_i = h(Y_2 \| X_2 \| X_3)$. Here, after U_i considers that the session key has been shared with CS_j, the authentication process is terminated. However, U_i does not check whether the SK_i is same as SK_j generated by CS_j. If the session key is created incorrectly due to some error, the session key establishment process has to be executed again

(2) Kumari et al. [16] and Wang et al. [30]: their scheme consists of registration, login, authentication, and password change steps. However, it does not provide

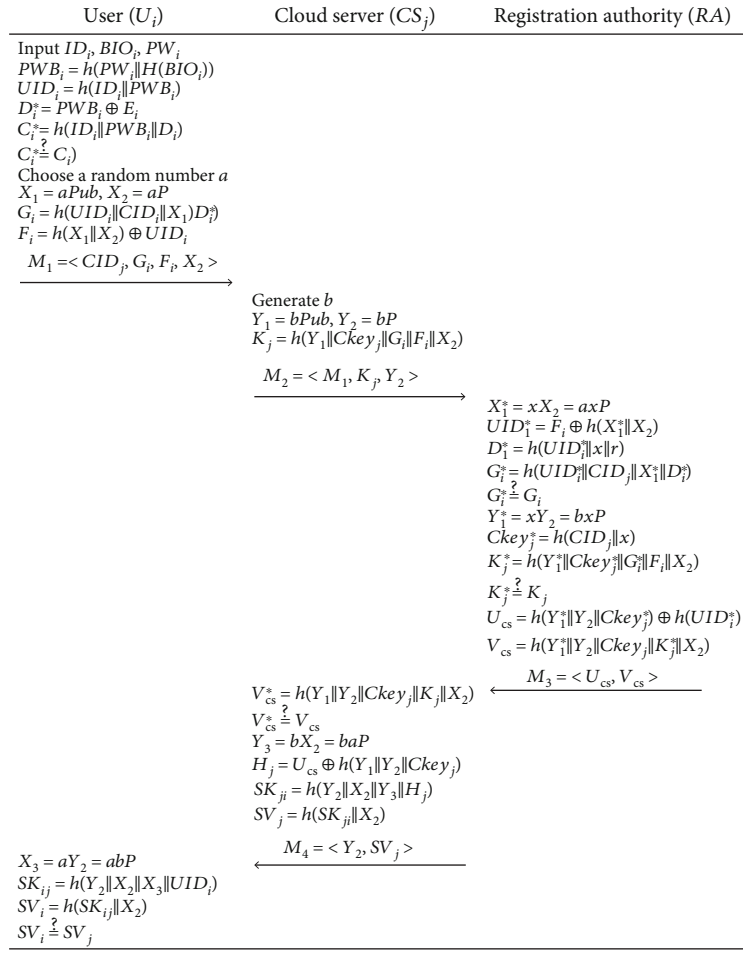

Figure 4: Login and authentication phase of the proposed scheme.

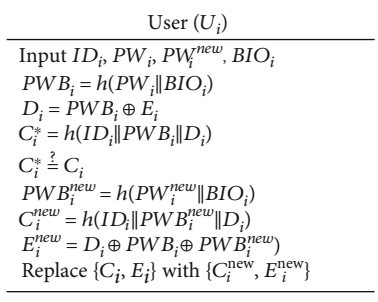

Figure 5: Password change phase of the proposed scheme.

an identity update process for the user. This process is necessary when the mobile phone number of U_i changes or the identity has to be changed for security reasons

(3) Martínez-Peláez et al. [19]: in their scheme, the adversary can extractor the secret parameters <PID$_i$, C_2, C_3, C_4, $h(n_U)$ > from the smart card of the user and intercept D_1 from message M_1. Then, the adversary can compute real identity ID$_i = C_2 \oplus D_1$. From now on, by creating T_u and n_U, an attacker can construct a malicious message M_1 that can easily impersonate a user. After that, the message generated by the attacker is delivered to the cloud server and the RA and processed. Eventually, the attacker will share the session key with the cloud server. In the aftermath of this attack, the attacker can acquire T_U^{new}, T_S^{new}, $T_{\text{CS}}^{\text{new}}$, n_U^{new}, n_S^{new}, and $n_{\text{CS}}^{\text{new}}$. Therefore, their scheme cannot resist user impersonation attacks and replay attacks, and violates mutual authentication and user anonymity. In addition, their scheme does not support the three-factor authentication and the identity update for user

In Table 2, we summarize the results of an informal analysis wherein the proposed scheme is compared with other relevant schemes. In the next section, we prove that our scheme satisfies all the security properties mentioned in Table 2.

5. Informal Analysis of the Proposed Scheme

In this section, we describe an informal analysis of the proposed scheme and show that it satisfies the desired security features and is secure against known attacks. In Table 2, we summarize the results of an informal analysis wherein the proposed scheme is compared with other relevant schemes.

Figure 6: Identity update phase of the proposed scheme.

Table 2: Security comparisons of the proposed scheme with related schemes in cloud computing environment.

Scheme	Wang et al. [15]	Kumari et al. [16]	Martínez-Peláez et al. [19]	Wang et al. [30]	Proposed
SP1	O	O	×	O	O
SP2	O	O	O	O	O
SP3	O	O	O	O	O
SP4	O	O	×	O	O
SP5	×	O	O	O	O
SP6	O	O	×	O	O
SP7	O	O	×	O	O
SP8	O	O	O	O	O
SP9	O	O	O	O	O
SP10	O	O	O	O	O
SP11	O	O	O	O	O
SP12	O	O	O	O	O
SP13	×	×	×	×	O
SP14	O	O	×	O	O

SP1: user anonymity; SP2: user untraceability; SP3: resistance to stolen-device attack; SP4: mutual authentication and session key agreement; SP5: verification of session key; SP6: resistance to user impersonation attack; SP7: resistance to replay attack; SP8: local user verification; SP9: resistance to privileged-insider attack; SP10: forward secrecy; SP11: resistance to stolen-verifier attack; SP12: user-friendly password change; SP13: providing identity update phase; SP14: providing three-factor user authentication.

5.1. User Anonymity. In the user authentication phase of our scheme, the user's ID_i is protected in UID_i and is preserved by the user's secret value X_1. The attacker must know two values, X_1 and $H(BIO_i)$, to know the user's ID_i from messages sent over a public channel. The value $X_1 = a\text{Pub} = axP$ is under the ECCDH problem, and the BIO_i is unique to each individual. Therefore, it is extremely difficult for an attacker to determine these two values, and thus, the proposed scheme guarantees user anonymity.

5.2. User Untraceability. In the login authentication phase, U_i sends a message M_1 to CS_j and receives a message M_4 from CS_j on a public channel that an attacker can eavesdrop on. In message M_1, G_i, F_i, and X_2 contain a random number a, and in message M_4, Y_2 and SV_j include random numbers a and b. Because both random numbers are changed during each session, and it is difficult to solve the ECCDH problem, the connection between the messages in each session cannot be determined, and the user activity cannot be tracked. Thus, the proposed scheme ensures user untraceability.

5.3. Resistance to Stolen-Device Attack. According to the attacker model, the attacker may extract the secret parameters $<C_i, E_i, P, \text{Pub}>$ by applying a side-channel attack if the attacker acquires a user's IoT smart device. In this attack, an attacker attempts to guess ID_i and PW_i using the two values, C_i and E_i, to impersonate a user or obtain the user's personal information. The attacker needs to know the $H(BIO_i)$ value to obtain the user's ID_i and PW_i. However, the user's biometric information is unique to each individual. Therefore, the proposed scheme resists a stolen-device attack.

5.4. Mutual Authentication and Session Key Agreement. In the proposed scheme, U_i and CS_j establish a session key through mutual authentication based on the support of RA. First, RA authenticates U_i by validating G_i included in M_1. The value of D_i included in G_i contains the server's private keys x and r stored in the user list maintained by RA. In addition, U_i must calculate D_i correctly at the login stage to verify its legitimacy to RA. Moreover, CS_j verifies its identity to the RA by validating the value of K_j contained in message M_2. To do so, CS_j must include a valid $Ckey_j$ that contains the secret key x of RA in K_j.

If CS_j and U_i are authenticated, the RA calculates X_1 and Y_1 (both can be calculated by a valid RA) and includes these two values in U_{CS} and V_{CS}, respectively. After receiving message M_3, CS_j calculates V^*_{CS} and compares it with the received V_{CS} to check the validity of the RA. Then, CS_j multiplies X_2 by its random nonce b to generate a value of Y_3, calculates the session key SK_{ji} and SV_j for the verification of session key, and sends SV_j to U_i.

After receiving message M_4, U_i multiplies Y_2 by its random nonce a to compute X_3, and calculates session key SK_{ij} and verification value SV_i for the session key.

Because the validity of Y_2 received from CS_j is confirmed through the soundness of SV_j by adding UID^*_i into SK_{ji}, even if Y_3 is generated by multiplying any random nonce of the malicious attacker by X_2, the validation process for SV_j generated by a malicious CS_j is not passed. Therefore, the proposed scheme supports mutual authentication between U_i, CS_j, and RA and provides a secure session key establishment between U_i and CS_j.

5.5. Verification of Session Key. In the proposed scheme, CS_j calculates SV_j including the session key calculated by itself and transmits it to U_i. Then, U_i also calculates the session key SV_i calculated by itself and compares whether it is the same as SV_j. Therefore, the proposed scheme can prevent session key disagreement that may occur due to communication errors.

5.6. Resistance to User-Impersonation Attack. For an attacker to conduct a user impersonation attack, the attacker must either maliciously control the session key establishment process or extract secret parameters from the user's IoT smart device. However, as mentioned earlier, the proposed scheme guarantees mutual authentication and protection from stolen-device attacks. Therefore, the proposed scheme provides a safe protection technique using biometrics and under the ECCDH problem against user impersonation attacks.

5.7. Resistance to Replay Attack. To establish a malicious session by pretending to be a participant in the communication, an attacker must know the random number a or b to create a session key by eavesdropping on the messages sent and received over the public channel and then reuse them. However, with the proposed scheme, the random numbers included in the session key are protected from the ECCDH problem. Even if an attacker attempts to connect a malicious session by replaying the message, communication in the next step cannot be continued without knowing the secret parameter or random nonce. Therefore, the proposed scheme resists a replay attack.

5.8. Local User Verification. In the login and authentication phase of the proposed scheme, the user first enters ID_i, PW_i, and BIO_i to calculate C^*_i and then checks whether C^*_i is equal to C_i stored in the IoT smart device. Only U_i who have passed this local user verification procedure can perform the next mutual authentication phase. Because U_i cannot log in without inputs of the user's legitimate personal information, the proposed scheme can block unauthorized access in the local area.

5.9. Resistance to Privileged-Insider Attack. In the registration phase of the proposed scheme, the user sends ID_i, UID_i, and PWB_i to the RA. To perform the privileged-insider attack, here, the insider of the RA needs BIO_i to guess PW_i of U_i. However, the user's BIO_i is protected by PW_i, and thus, the malicious insider cannot impersonate U_i to communicate with CS_j. In addition, the proposed scheme can change PWB_i locally in U_i's IoT smart device without an intervention of RA. Therefore, the proposed scheme is safe from an insider attack.

5.10. Forward Secrecy. With the proposed scheme, the session key is not transmitted directly. Instead, the user and CS_j agree on the session key by calculating the secret parameters constituting the session key. Here, the session key shared between them includes a different random nonce for each session. Therefore, it is difficult for an attacker to attempt to guess the session key by collecting the session key verification value SV_j. Therefore, the proposed scheme guarantees forward secrecy.

```
(*.....chanels.....*)
free cha:channel [private].
free chb:channel.
free chc:channel.

(*.....constants.....*)
free IDi:bitstring [private].
free CIDj:bitstring.
free RA:bitstring.
free PWi:bitstring [private].
free BIOi:bitstring [private].

(*.....secret key.....*)
free X:bitstring [private].
free P:bitstring [private].

(*.....shared key.....*)
free SKij:bitstring [private].
free SKji:bitstring [private].

(*.....functions.....*)
fun concat(bitstring,bitstring) : bitstring.
fun syme(bitstring,bitstring):bitstring.
fun xor(bitstring,bitstring):bitstring.
fun h(bitstring):bitstring.
fun H(bitstring):bitstring.
fun ecpm(bitstring,bitstring):bitstring.
equation forall p:bitstring, q:bitstring); xor(xor(p,q), q)=p

(*.....events.....*)
event beingRAgent(bitstring).
event endRAgent(bitstring).
event beingCSnodet(bitstring).
event endCSnode(bitstring).
event beingUserNode(bitstring).
event beingUserNode(bitstring).
```

Figure 7: ProVerif code for predefined identifiers and definitions.

```
(*.....Ui's process.....*)
let pUser=
let PWBi = h(concat(PWi,H(BIOi))) in
let UIDi=h(concat(IDi,PWBi)) in
out(cha,(UIDi,PWBi));
in(cha,(XCi:bistring,XEi:bistring,XP:bistring,
XPub:bistring));
event beingUserNode(IDi);
let Di'=xor(PWBi, XEi) in
let Ci'=h(concat(IDi,concat(PWBi,Di))) in
if XCi=Ci' then
new a:bistring;
let X1=ecpm(a,XPub) in
let X2=ecpm(a,XPub) in
let Gi=h(concat(UIDi,concat(CIDj,concat(X1,Di')))) in
let Fi=xor(h(concat(X1,Di'))) in
let M1=concat(CIDj,concat(Gi,concat(X1,Di'))) in
out(chc,(M1));
in(chc,XM4:bitstring);
let(XXY2:bitstring, XSVj:bitstring = XM4 in
let X3=ecpm(a,XXY2) in
let SKij+h(XXY2,concat(X2,concat(X3,UIDi)))) in
let SVi=h(concat(SKij,X2)) in
if(SVi = XSVj) then event endUserNode(IDi).
```

Figure 8: ProVerif code for the entire user process.

5.11. Resistance to Stolen-Verifier Attack. In the proposed scheme, the personal login information of U_i, including PW_i and $H(BIO_i)$, is not directly transmitted to CS_j. RA retains User_List = <RID_i, r, Honey_List = 0 > in the database. Even if the attacker misappropriates User_List, the attacker cannot obtain the real identity of U_i because it is protected by the secret parameter x. Therefore, the proposed scheme is secure against stolen-verifier attacks.

5.12. User-Friendly Password Change. In the proposed scheme, the user can change to a new password by applying the user's information offline without interacting with the server. Therefore, the proposed scheme supports a change in the password in a user-friendly manner.

5.13. Providing Identity Update Phase. In the proposed scheme, users can request RA to change their identity. Thus, the user can set a new identity ID_i^{new} when he or she wants, such as changing his or her mobile phone number. When RA receives a request from the user, RA calculates new secret parameters for ID_i^{new} and stores them in the User_List, so that the user's legitimacy can be authenticated at a later login and authentication step.

6. Formal Analysis of the Proposed Scheme

6.1. ProVerif. To prove the security of the proposed scheme, we adapted ProVerif [31], which is an automated tool used to analyse cryptographic protocols. ProVerif supports an analysis of protocols based on various cryptographic primitives such as symmetric and asymmetric cryptography, digital signatures, and hash functions. ProVerif is widely used by many researchers [32–35] to validate a security analysis of the key agreement and authentication schemes for various network environments. In this section, we introduce the ProVerif code and present the analysis results to verify the proposed scheme's security.

We present the process of predefined identifiers and the definitions of the proposed scheme in Figure 7. Here, we define the public and secure channels used among U_i, CS_j,

```
(*.....CS's process.....*)
let pCS=
out(chb,(CIDj));
in(chb,(XCkeyj:bitstring,XXP:bitstring,XXPub:bitstring));
in(chc, (XM1i,bitstring));
let (XCIDj:bitstring,XGi:bitstring,XFi:bitstring,
XX2:bitstring) = XM1 in
event beginCSnode(CIDj);
new b:bistring;
let Y1=ecpm(b,XXPub) in
let Y2=ecpm(b,XXP) in
let Kj=h(concat(Y1,concat(XCkeyj,concat(XG1i,concat
(XFi,XX2))))) in
let M2=concat(Kj,concat(Y2,XM1)) in
out(chb,(M2));
in(chb,(XM3:bitstring));
let (XUcs:bitstring,XVcs:bitstring)=XM3 in
let Vcs'=h(concat(Y1,concat(Y2,concat(XCkeyj,concat
(Kj,XX2))))) in
if Vcs'=XVcs then
let Y3=ecpm(b,XX2) in
let Hj=xor(XUcs,h(concat(Y1,concat(Y2,XCkeyj)))) in
let SKji=h(concat(Y2,concat(XX2,concat(Y3,Hj)))) in
let SVj=h(concat(SKji,XX2)) in
let M4 =concat(Y2,SVj) in
out(chc,(M4));
event endCSnode(CIDj).
```

Figure 9: ProVerif code for the entire cloud server process.

```
(*.....RA's process.....*)
let pRA=
let Pub=ecpm(x,P) in
in(chb,(XUIDi:bitstring, XPWBi:bitstring));
new r:bitstring;
let Di=h(concat(XUIDi,concat(x,r))) in
let Ci=h(concat(XUIDi,concat(XPWBi,Di))) in
let Ei=xor(Di,XPWBi) in
out(cha,(Ci,Ei,P,Pub));
in(chb,(XCIDj:bitstring));
let Ckeyj=h(concat(XCID,x)) in
out(chb,CKeyj,P,Pub));
event beginRAgent(RA);
in(chb, XM2:bitstring);
let (XXM2:bitstring, XKj:bitstring, XY2:bitstring,
XXM1:bitstring) = XM2 in
let (XXCIDj:bitstring,XXGi:bitstring,XXFi:bitstring,
XXX2:bitstring) = XXM1 in
let XX1=ecpm(x,XXX2) in
let UIDi'=xor(XXFi,h(concat(XX1,XXX2))) in
let Di"=h(concat(UID',concat(x,r))) in
letGi'=h(concat(UIDi',concat(XXCIDj,concat(XX1,Di"
)))) in
if Gi'=XXGi then
let XY1=ecpm(x,XY2) in
let Ckeyj'=h(concat(XXCIDj,x)) in
let Kj'=h(concat(XY1,concat(Ckeyj',concat(Gi',concat
(XXFi,XXX2))))) in
if Kj'=XKj then
let Ucs=xor(UIDi',h(concat(XY1,concat(XY2,Ckeyj')))) in)
let Vcs=h(concat(XY1,concat(XY2,concat(Ckeyj,concat
(Kj',XXX2))))) in
let M3=concat(Ucs,Vcs) in
out(chb,(M3));
event endRAgent(RA).
```

Figure 10: ProVerif code for the entire registration agency process.

```
(*.....quries.....*)
query attacker(SKij).
query attacker(SKji).
query id:bitstring; inj-event(endlotNode(id)) ==> inj-event(beginlotNode(id)).
query id:bitstring; inj-event(endGateWay(id)) ==> inj-event(beginGateWay(id)).
query id:bitstring; inj-event(endMNode(id)) ==> inj-event(beginMNode(id)).
set traceDisplay=long
process
((!pMNNode)|(!pFAgent)|(!pHAgent))
```

Figure 11: ProVerif code for adversary capabilities and verifying equivalences.

```
RESULT inj - event (endUserNode(id)) ==> inj- event(beginUserNode(id)) is true.
RESULT inj - event (endRAgent(id_3057)) ==> inj- event(beginRAgent(id_3057)) is true.
RESULT inj - event (endCSnode(id_8030)) ==> inj- event(beginCSnode(id_8030)) is true.
RESULT not attacker (SKij[]) is true.
RESULT not attacker (SKji[]) is true.
```

Figure 12: ProVerif code for adversary capabilities and verifying equivalence verification.

Table 3: Comparisons in terms of the computational time and the communication costs.

Scheme	Wang et al. [15]	Kumari et al. [16]	Martínez-Peláez et al. [19]	Wang et al. [30]	Proposed
U_i	$9T_h + 3T_e$	$7T_h + 3T_e$	$7T_h + 3T_s$	$9T_h + 2T_C + 1T_S$	$12T_h + 3T_e$
CS_j	$3T_h + 3T_e$	$5T_h + 3T_e$	$5T_h + 3T_s$	$5T_h + 2T_C + 1T_S$	$5T_h + 3T_e$
RA	$7T_h + 2T_e$	$6T_h + 2T_e$	$21T_h + 2T_s$	$9T_h + 4T_s$	$10T_h + 2T_e$
Total	$19T_h + 8T_e$	$18T_h + 6T_e$	$33T_h + 8T_s$	$21T_h + 4T_c + 6T_s$	$27T_h + 8T_e$
Time	514.14 ms	513.64 ms	86.1 ms	147.78 ms	514.64 ms
Communication cost	2080 bits	2304 bits	3200 bits	1696 bits	1792 bits

and RA; the cryptographic parameters and operations; and the start and end of communication between nodes to be verified for the correspondence relationship of the messages.

We define the overall U_i process code for the proposed scheme, as shown in Figure 8. We model the registration phase in lines 37-41 and the login and authentication phase in lines 42-58.

We define the overall CS_j process code for the proposed scheme, as shown in Figure 9. We model the registration phase in lines 60-64 and the login and authentication phase in lines 65-82.

Figure 10 shows the overall RA process code for the proposed scheme. We model the registration phase in lines 84-94 and the login and authentication phase in lines 95-113.

The code shown in Figure 11 is intended to model the attacker's capabilities and verify the equivalencies of inter-process communication. The code in lines 115 and 116 checks whether the session keys SK_{ij} and SK_{ji} are secure against the attacker. The code in lines 117-119 verifies whether the internodal relationships of the proposed scheme are accurate during the procedure.

The execution of all codes described earlier verifies the effectiveness and availability of the simulated events and queries and generates the results of the simulation, as presented in Figure 12. This indicates that U_i, CS_j, and RA in the proposed scheme achieve a successful mutual authentication and securely establish the session key. Furthermore, it can be considered that the proposed scheme is secure against simulated attacks.

6.2. BAN Logic. Burrows-Abadi-Needham (BAN) logic [36] is used to prove the trust of each party in the authentication protocol on the formal logic. We utilize this logic to prove that U_i and CS_j share a valid and fresh session key through mutual authentication. We define the notations of BAN logic as follows:

(1) $U \triangleleft C$: U sees condition C

(2) $U \mid \equiv C$: condition C is believed by U

(3) $\#(C)$: it makes a fresh C

(4) $U \mid \sim C$: U expresses the condition C

(5) $U \overset{K}{\leftrightarrow} S$: U and S share a secret key K

(6) $U \Longrightarrow C$: condition C is handled by U

(7) $(C)_K$: C is encrypted under key K

We define five rules of BAN logic to prove the mutual authentication of the proposed scheme.

(1) Rule 1: message-meaning rule $U \mid \equiv U \overset{K}{\leftrightarrow} S, U \triangleleft (C)_K / U \mid \equiv S \mid \sim C$: if U trusts that key K is shared with S, U sees C combined with K and trusts S once said C

(2) Rule 2: nonce-verification rule $U \mid \equiv \#(C), U \mid \equiv S \mid \sim C / U \mid \equiv S \mid \equiv C$: if U trusts that freshness of C and trusts S once said C, then U trusts that S trusts C

(3) Rule 3: belief rule $U \mid \equiv C, U \mid \equiv M / A \mid \equiv (C, M)$: if U trusts C and M, (C, M) are also trusted by U

(4) Rule 4: freshness-conjuncatenation rule $U \mid \equiv \#(C) / A \mid \equiv \#(C, M)$: if the freshness of C is trusted by U, then U can trust the freshness of the full condition

(5) Rule 5: jurisdiction rule $U \mid \equiv S \mid \Longrightarrow C, U \mid \equiv S \mid \equiv C / A \mid \equiv C$: if U trusts that S has jurisdiction over C, and U trusts that S trusts a condition C, then U also trusts C

We must satisfy the following four goals:

(1) Goal 1: $U_i \mid \equiv (U_i \overset{SK_{ij}}{\leftrightarrow} CS_j)$

(2) Goal 2: $CS_j \mid \equiv (U_i \overset{SK_{ij}}{\leftrightarrow} CS_j)$

(3) Goal 3: $U_i \mid \equiv CS_j \mid \equiv (U_i \overset{SK_{ij}}{\leftrightarrow} CS_j)$

(4) Goal 4: $\mathrm{CS}_j| \equiv U_i| \equiv (U_i \overset{\mathrm{SK}_{ij}}{\leftrightarrow} \mathrm{CS}_j)$

The four messages transmitted in the proposed scheme can be converted into the idealized form as follows:

(1) Using $M_1 = <\mathrm{CID}_j, G_i, F_i, X_2>$, $U_i \longrightarrow \mathrm{CS}_j : G_i = h(\mathrm{UID}_i \| \mathrm{CID}_j \| X_1 \| D_1)$, $F_i = h(X_1 \| X_2) \oplus \mathrm{UID}_i$. This is reduced to MSG_1:$(\mathrm{UID}_i, \mathrm{CID}_j, D_1, X_1, X_2)$

(2) Using $M_2 = <M_1, K_j, Y_2>$, $\mathrm{CS}_j \longrightarrow \mathrm{RA} : K_j = h(Y_1 \| \mathrm{Ckey}_j \| G_i \| F_i \| X_2)$. This is reduced to $\mathrm{MSG}_2 : (M_1, Y_1, \mathrm{Ckey}_j, X_2)$

(3) Using $M_3 = <U_{\mathrm{cs}}, V_{\mathrm{cs}}>$, $\mathrm{RA} \longrightarrow \mathrm{CS}_j : U_{\mathrm{cs}} = h(Y_1^* \| Y_2 \| \mathrm{Ckey}_j^* \| K_j^* \| X_2) \oplus h(\mathrm{UID}_i^*)$, $V_{\mathrm{cs}} = h(X_1^* \| X_2 \| \mathrm{UID}_i \| D_i^* \| Y_2)$. This is reduced to MSG_3:$(Y_1, Y_2, \mathrm{Ckey}_j, X_2, \mathrm{UID}_i)$

(4) Using $M_4 = <Y_2, \mathrm{SV}_j>$, $\mathrm{CS}_j \longrightarrow U_i : \mathrm{SV}_j = h(\mathrm{SK}_{ji} \| X_2)$. This is reduced to $\mathrm{MSG}_4 : (Y_2, X_2)$

To derive the goals of the proposed scheme, we define the following assumptions.

(1) $U_i| \equiv \#(\mathrm{UID}_i)$

(2) $\mathrm{CS}_j| \equiv \#(\mathrm{Ckey}_j)$

(3) $\mathrm{RA}| \equiv \#(x)$

(4) $\mathrm{CS}_j| \equiv \#(b)$

(5) $\mathrm{CS}_j| \equiv (U_i \overset{a}{\leftrightarrow} \mathrm{CS}_j)$

(6) $\mathrm{CS}_j| \equiv (\mathrm{CS}_j \overset{b}{\leftrightarrow} \mathrm{RA})$

(7) $\mathrm{RA}| \equiv (\mathrm{CS}_j \overset{b}{\leftrightarrow} \mathrm{RA})$

(8) $U_i| \equiv (U_i \overset{a}{\leftrightarrow} \mathrm{CS}_j)$

(9) $U_i| \equiv \mathrm{CS}_j| \equiv (U_i \overset{\mathrm{SK}_{ji}}{\leftrightarrow} \mathrm{CS}_j)$

(10) $\mathrm{CS}_j| \equiv U_i| \equiv (U_i \overset{\mathrm{SK}_{ij}}{\leftrightarrow} \mathrm{CS}_j)$

We describe the main proof of the proposed scheme using the BAN logic rules, messages, and assumptions as follows:

(1) From M_1, we obtain $V_1 : \mathrm{CS}_j \triangleleft (\mathrm{CID}_j, \mathrm{UID}_i, P)_a$

(2) From A_5 and rule 1, we obtain $V_2 : \mathrm{CS}_j| \equiv U_i| \sim (\mathrm{CID}_j, \mathrm{UID}_i, P)_a$

(3) From A_1 and rule 4, we obtain $V_3 : \mathrm{CS}_j| \equiv \#(\mathrm{CID}_j, \mathrm{UID}_i, P)_a$

(4) From V_1, V_2, and rule 2, we obtain $V_4 : \mathrm{CS}_j| \equiv U_i| \equiv (\mathrm{CID}_j, \mathrm{UID}_i, P)_a$

(5) From M_2, we obtain $V_5 : \mathrm{RA} \triangleleft (\mathrm{Ckey}_j, \mathrm{CID}_j, \mathrm{UID}_i, P)_b$

(6) From A_7 and rule 1, we obtain $V_6 : \mathrm{RA}| \equiv \mathrm{CS}_j| \sim (\mathrm{Ckey}_j, \mathrm{CID}_j, \mathrm{UID}_i, P)_b$

(7) From A_2 and rule 4, we obtain $V_7 : \mathrm{CS}_j| \equiv \#(\mathrm{Ckey}_j, \mathrm{CID}_j, \mathrm{UID}_i, P)_b$

(8) From V_5, V_6, and rule 2, we obtain $V_8 : \mathrm{RA}| \equiv \mathrm{CS}_j| \equiv (\mathrm{Ckey}_j, \mathrm{CID}_j, \mathrm{UID}_i, P)_b$

(9) From M_3, we obtain $V_9 : \mathrm{CS}_j \triangleleft (a, x, P, \mathrm{Ckey}_j, \mathrm{UID}_i)_b$

(10) From A_6 and rule 1, we obtain $V_{10} : \mathrm{CS}_j| \equiv \mathrm{RA}| \sim (a, x, P, \mathrm{Ckey}_j, \mathrm{UID}_i)_b$

(11) From A_3 and rule 4, we obtain $V_{11} : \mathrm{CS}_j| \equiv \#(a, x, P, \mathrm{Ckey}_j, \mathrm{UID}_i)_b$

(12) From V_9, V_{10}, and rule 2, we obtain $V_{12} : \mathrm{CS}_j| \equiv \mathrm{RA}| \equiv (a, x, P, \mathrm{Ckey}_j, \mathrm{UID}_i)_b$

(13) From M_4, we obtain $V_{13} : U_i \triangleleft (b, x, P)_a$

(14) From A_8 and rule 1, we obtain $V_{14} : U_i| \equiv \mathrm{CS}_j| \sim (b, x, P)_a$

(15) From A_4 and rule 4, we obtain $V_{15} : U_i| \equiv \#(b, x, P)_a$

(16) From V_{13}, V_{14}, and rule 2, we obtain $V_{16} : U_i| \equiv \mathrm{CS}_j| \equiv (b, x, P)_a$

(17) From V_{12}, V_{16}, and SK, we obtain $V_{17} : U_i| \equiv (U_i \overset{\mathrm{SK}_{ij}}{\leftrightarrow} \mathrm{CS}_j)$ (goal 1)

(18) From V_4, V_8, and SK, we obtain $V_{18} : CS_j| \equiv (U_i \overset{\mathrm{SK}_{ij}}{\leftrightarrow} \mathrm{CS}_j)$ (goal 2)

(19) From A_9, V_{17}, and rule 5, we obtain $V_{19} : U_i| \equiv \mathrm{CS}_j| \equiv (U_i \overset{\mathrm{SK}_{ij}}{\leftrightarrow} \mathrm{CS}_j)$ (goal 3)

(20) From A_{10}, V_{18}, and rule 5, we obtain $V_{17} : \mathrm{CS}_j| \equiv U_i| \equiv (U_i \overset{\mathrm{SK}_{ij}}{\leftrightarrow} \mathrm{CS}_j)$ (goal 4)

From goals 1, 2, 3, and 4 that we achieved earlier, we see that U_i and C_j establish a session key through a secure mutual authentication.

7. Performance Analysis

In this section, we compare the computational and communication costs for the proposed scheme with those of other related schemes for cloud computing environments. We considered the computational cost and number of

communications occurring during the login and authentication process. As described by Kocarev and Lian [37], we consider the execution time of cryptographic operations as follows:

(1) 160-bit elliptic multiplication operation: $T_e \approx 63.08$ ms

(2) 128-bit Advanced Encryption Standard (AES) algorithm: $T_s \approx 8.7$ ms

(3) 128-bit hash function: $T_h \approx 0.5$ ms

(4) 128-bit Chebyshev polynomial computation: $T_h \approx 21.02$ ms

We summarize the results of the comparison in terms of the computational time and communication costs in Table 3. The results reveal that Martínez-Peláez et al.'s scheme [19] is significantly faster in terms of computational time than the other schemes. However, as described in Section 1.2, Yu et al. [20] revealed that Martínez-Peláez et al.'s scheme [19] is vulnerable to various attacks. Wang et al.'s scheme [30] applies a Chebyshev chaotic map as cryptography primitive to strengthen the security of the session key. However, their scheme does not provide the identity update phase. The securities of schemes proposed by Kumari et al. [16] and Wang et al. [15] are based on the ECC for which the communication participants agree on the session key. However, Wang et al.'s scheme [15] does not provide the session key verification procedure to check its validation, and Kumari et al. [16] do not design the identity update phase in their scheme. Meanwhile, our scheme has slightly higher computational costs than those of Kumari et al.'s [16] and Wang et al.'s scheme [30], although the proposed scheme satisfies all security requirements, as mentioned in Section 5.

According to the results of previous analysis [28, 38], we assume that the lengths of the identity, random number, and timestamp are 128, 64, and 32 bits, respectively, for a comparison of the communication costs. The hash function produces 160 bits; the block size of the symmetric encryption is 128 bits; the size of the Chebyshev polynomial is 128 bits; the size of the point multiplication on the elliptic curve is 160 bits.

Table 3 also provides data from the comparisons of the communication costs. The total communication cost of the proposed scheme is 1792 bits, whereas those of Amin et al.'s [14], Kumari et al.'s [16], Martínez-Peláez et al.'s [19], and Wang et al.'s schemes [30] are 2080, 2304, 3200, and 1696 bits, respectively. Table 3 shows that the scheme proposed by Wang et al. [30] requires the lowest communication cost, whereas the proposed scheme has the second-lowest communication cost. However, as shown in Table 2, Wang et al.'s scheme [30] does not support the identity update phase. Therefore, the proposed scheme is a more practical option in a cloud computing environment.

8. Conclusion

In this study, we conducted an informal analysis to demonstrate the security of the proposed scheme against various known attacks. In addition, using ProVerif and BAN logic, we applied a formal analysis to prove that the user and cloud server establish a session key through secure mutual authentication. Moreover, we conducted an analysis of the proposed scheme in terms of the security features and performance; we compared it with those of existing related schemes and proved that our proposed scheme ensures better safety and efficiency in user management and that it is suitable for use in a practical cloud computing environment.

Appendix

A. Wang et al.'s Authentication Scheme [15]

Wang et al.'s authentication scheme is shown in Figures 13–16.

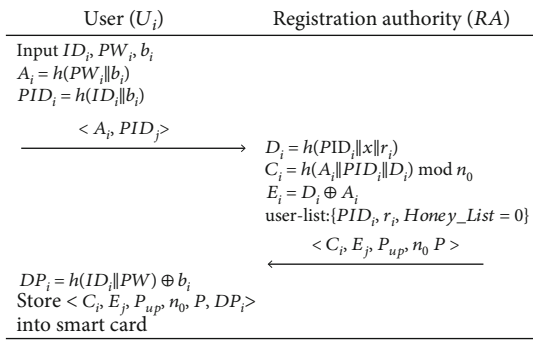

Figure 13: Registration phase of user in Wang et al.'s scheme [15].

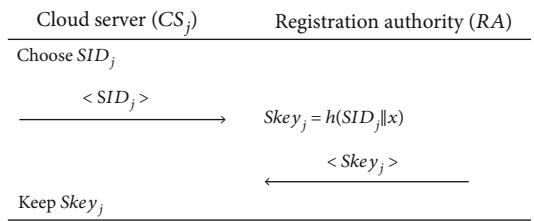

Figure 14: Registration phase of cloud server in Wang et al.'s scheme [15].

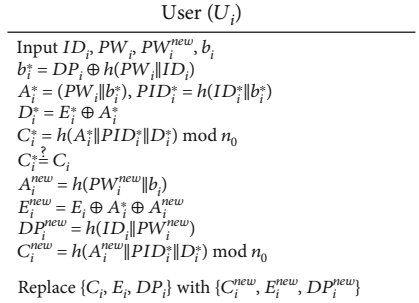

Figure 15: Password change phase in Wang et al.'s scheme [15].

Figure 16: Login and authentication phase in Wang et al.'s scheme [15].

B. Kumari et al.'s Authentication Scheme [16]

Kumari et al.'s authentication scheme is shown in Figures 17–20.

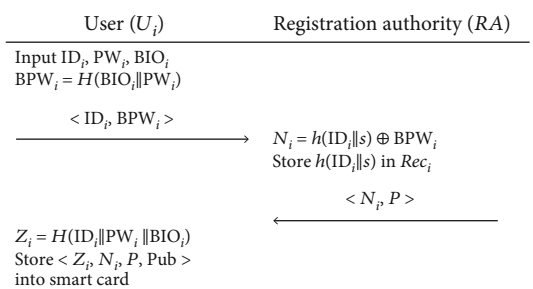

Figure 17: Registration phase of user in Kumari et al.'s scheme [16].

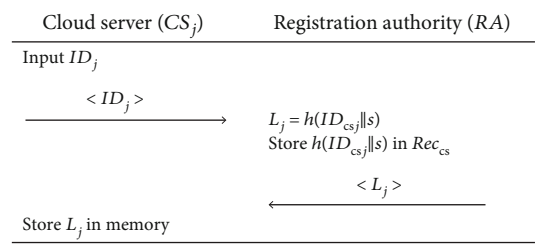

Figure 18: Registration phase of cloud server in Kumari et al.'s scheme [16].

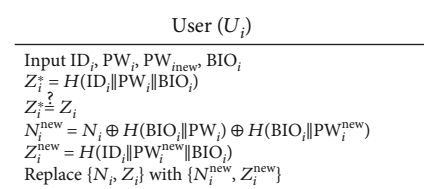

Figure 19: Password change phase in Kumari et al.'s scheme [16].

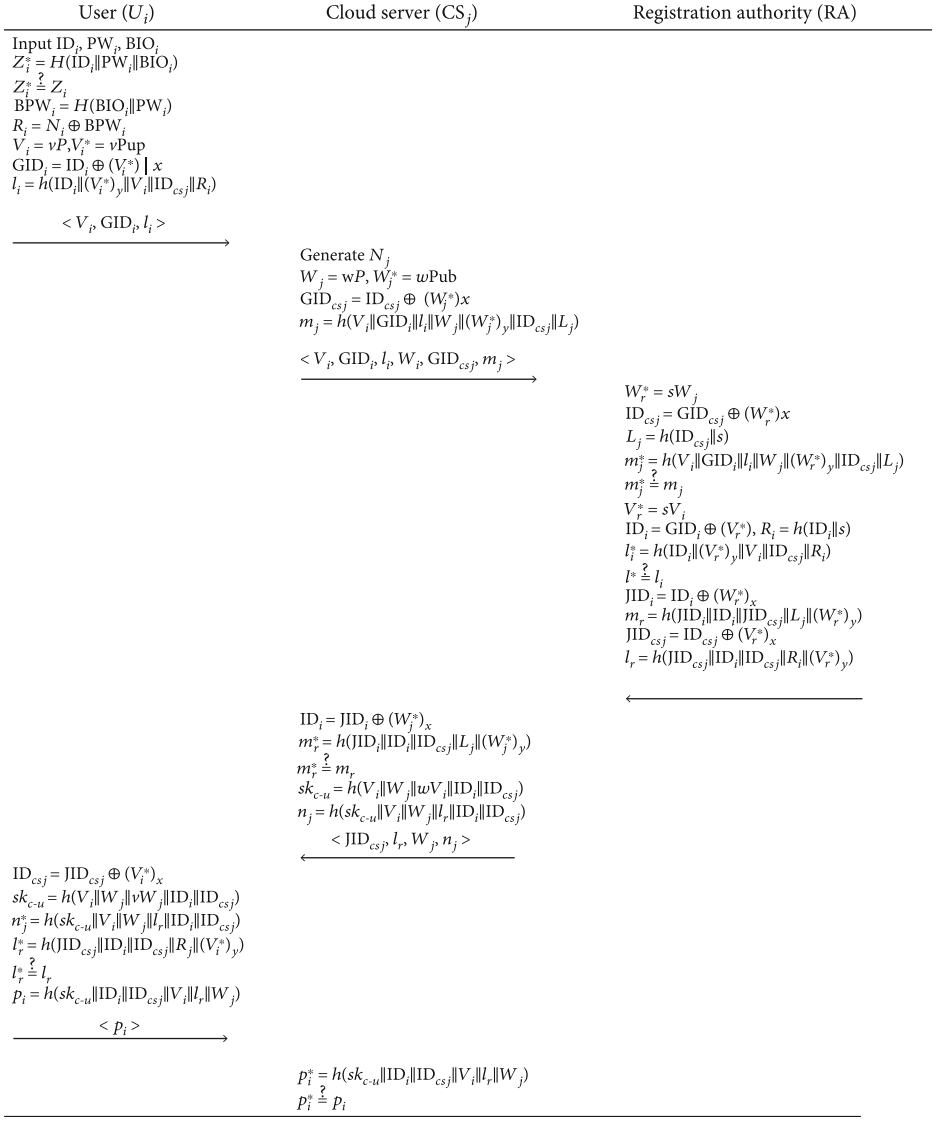

Figure 20: Login and authentication phase in Kumari et al.'s scheme [16].

C. Martínez-Peláez et al.'s Authentication Scheme [19]

Martínez-Peláez et al.'s authentication scheme is shown in Figures 21–24.

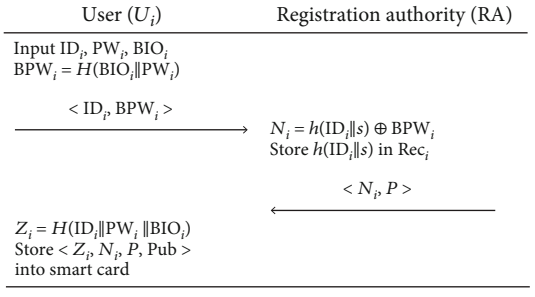

Figure 21: Registration phase of user in Martínez-Peláez et al.'s scheme [19].

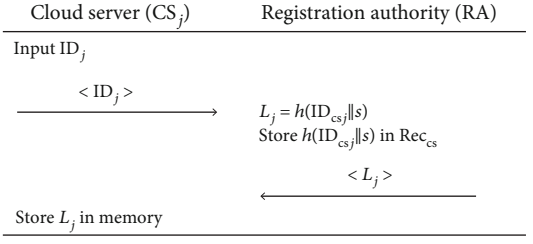

Figure 22: Registration phase of cloud server in Martínez-Peláez et al.'s scheme [19].

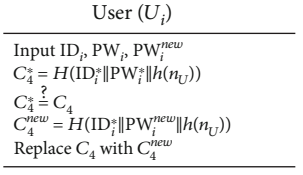

Figure 23: Password change phase in Martínez-Peláez et al.'s scheme [19].

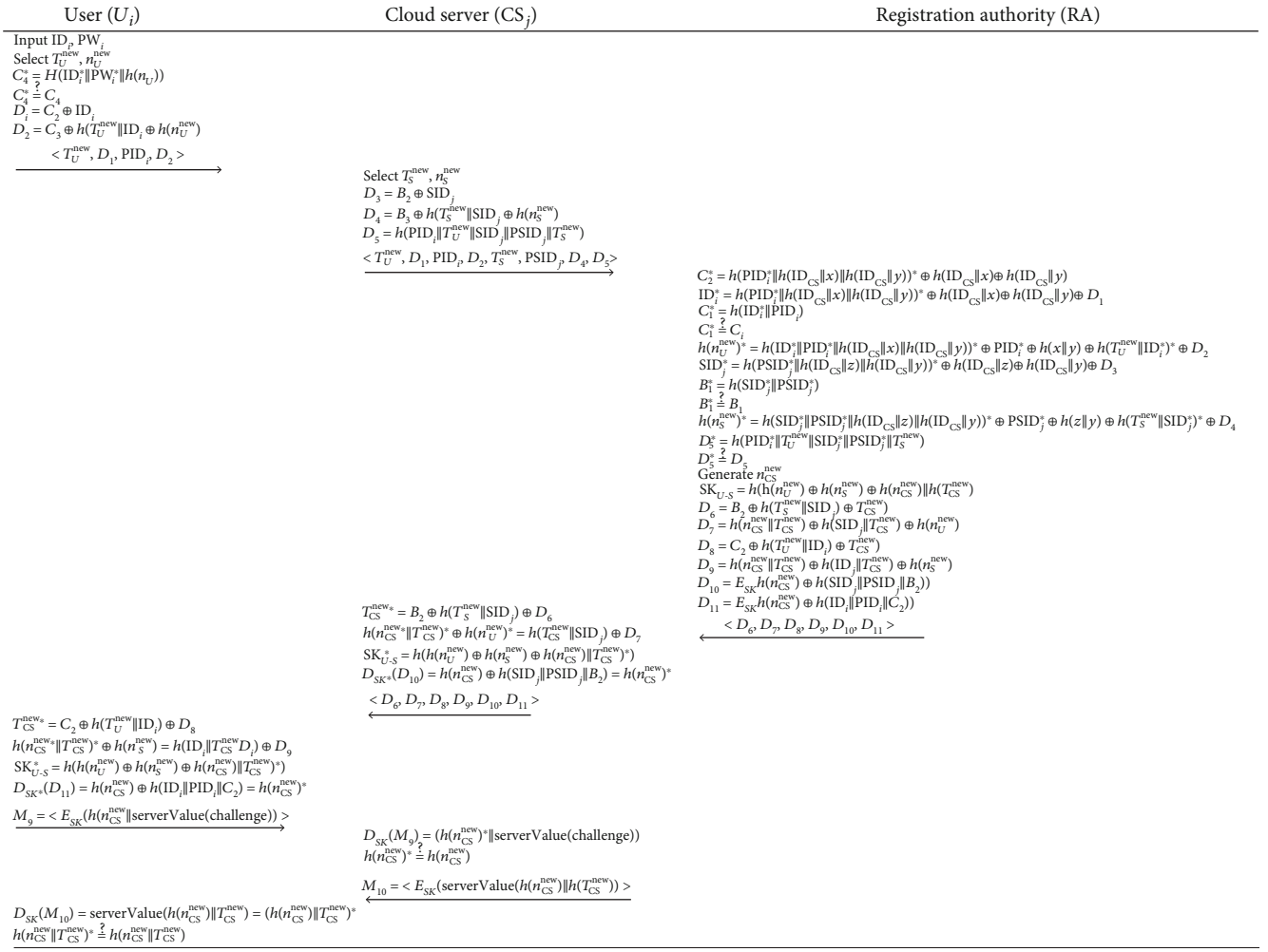

Figure 24: Login and authentication phase in Martínez-Peláez et al.'s scheme [19].

D. Wang et al.'s Authentication Scheme [30]

Wang et al.'s authentication scheme is shown in Figures 25–28.

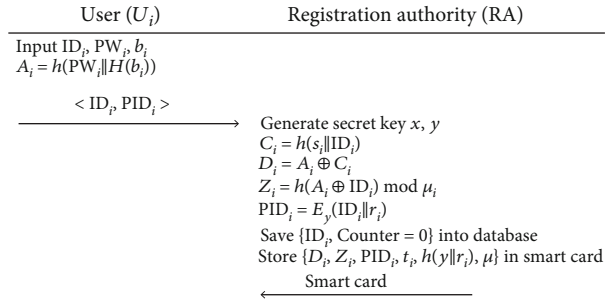

Figure 25: Registration phase of user in Wang et al.'s scheme [30].

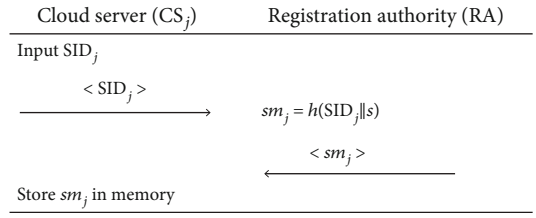

Figure 26: Registration phase of cloud server in Wang et al.'s scheme [30].

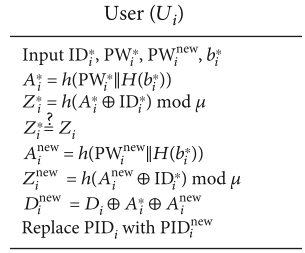

Figure 27: Password change phase in Wang et al.'s scheme [30].

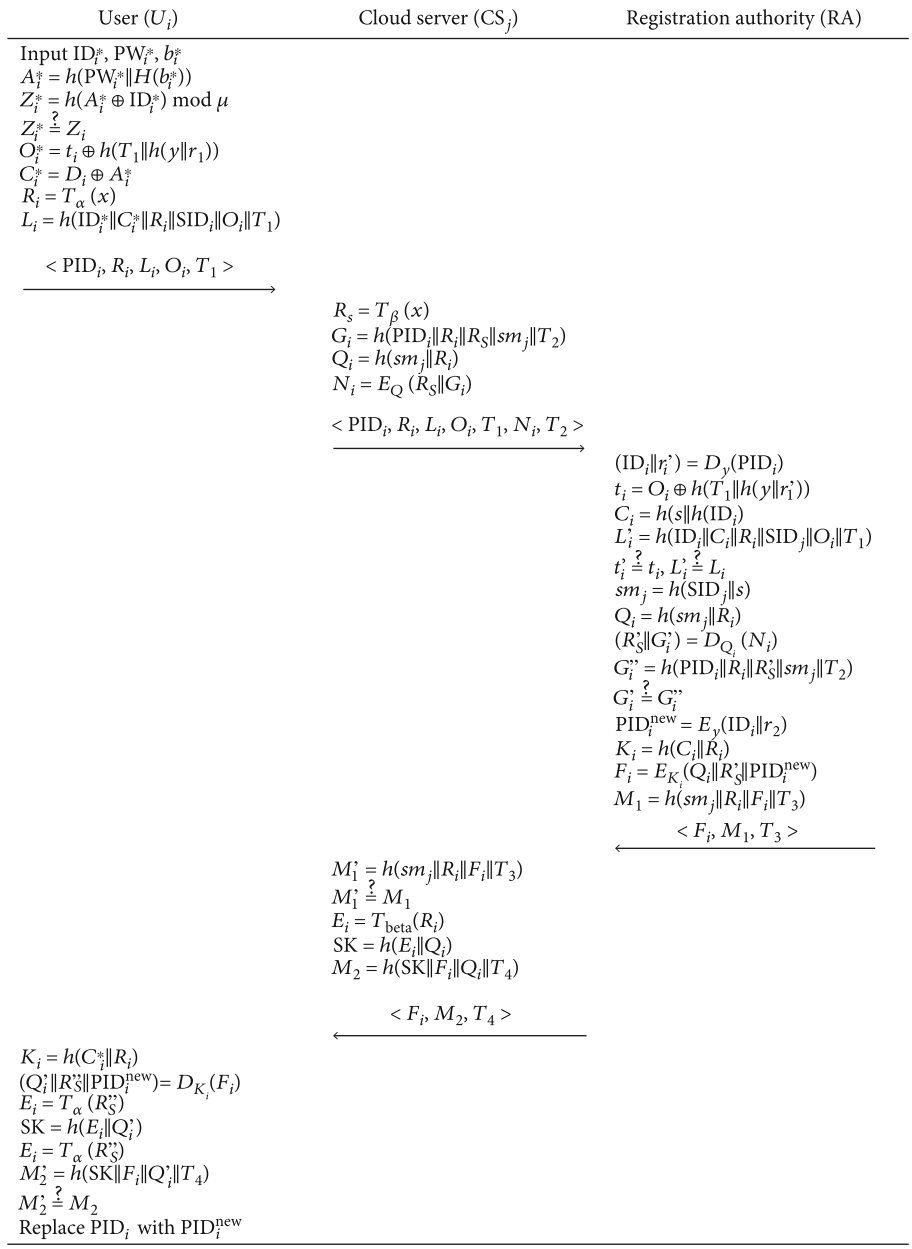

Figure 28: Login and authentication phase in Wang et al.'s scheme [30].

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by an Institute of Information & Communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00258, Development of On-chain-based Electronic Contract Application Platform Using Zero-Knowledge Proof).

References

[1] S. K. Goudos, P. I. Dallas, S. Chatziefthymiou, and S. Kyriazakos, "A survey of IoT key enabling and future technologies: 5G, mobile IoT, sematic web and applications," *Wireless Personal Communications*, vol. 97, no. 2, pp. 1645–1675, 2017.

[2] S. Moganedi and J. Mtsweni, "Beyond the convenience of the Internet of Things: security and privacy concerns," in *2017 IST-Africa Week Conference (IST-Africa)*, pp. 1–10, Windhoek, Namibia, May 2017.

[3] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: promises and challenges of IoT in medicine and healthcare," *Future Generation Computer Systems*, vol. 78, pp. 659–676, 2018.

[4] A. J. C. Trappey, C. V. Trappey, C.-Y. Fan, A. P. T. Hsu, X. K. Li, and I. J. Y. Lee, "IoT patent roadmap for smart logistic service provision in the context of Industry 4.0," *Journal of the Chinese Institute of Engineers*, vol. 40, no. 7, pp. 593–602, 2017.

[5] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: a survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[6] A. Rashid and A. Chaturvedi, "Cloud computing characteristics and services a brief review," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 2, pp. 421–426, 2019.

[7] A. C. Weaver, "Biometric authentication," *Computer*, vol. 39, no. 2, pp. 96-97, 2006.

[8] M. Kim, J. Moon, D. Won, and N. Park, "Revisit of password-authenticated key exchange protocol for healthcare support wireless communication," *Electronics*, vol. 9, no. 5, p. 733, 2020.

[9] Y. Park, K. S. Park, and Y. H. Park, "Secure user authentication scheme with novel server mutual verification for multiserver environments," *International Journal of Communication Systems*, vol. 32, no. 7, article e3929, 2019.

[10] S. Banerjee, V. Odelu, A. K. Das et al., "A provably secure and lightweight anonymous user authenticated session key exchange scheme for Internet of Things deployment," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8739–8752, 2019.

[11] A. Irshad, S. A. Chaudhry, M. Sher et al., "An anonymous and efficient multiserver authenticated key agreement with off-line registration centre," *IEEE Systems Journal*, vol. 13, no. 1, pp. 436–446, 2018.

[12] D. Kang, J. Jung, D. Lee, H. Kim, and D. Won, "Security analysis and enhanced user authentication in proxy mobile IPv6 networks," *PLoS One*, vol. 12, no. 7, article e0181031, 2017.

[13] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[14] R. Amin, N. Kumar, G. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment," *Future Generation Computer Systems*, vol. 78, pp. 1005–1019, 2018.

[15] C. Wang, K. Ding, B. Li et al., "An enhanced user authentication protocol based on elliptic curve cryptosystem in cloud computing environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3048697, 13 pages, 2018.

[16] S. Kumari, X. Li, F. Wu, A. K. Das, K. K. R. Choo, and J. Shen, "Design of a provably secure biometrics-based multi-cloud-server authentication scheme," *Future Generation Computer Systems*, vol. 68, pp. 320–330, 2017.

[17] L. Zhou, X. Li, K. H. Yeh, C. Su, and W. Chiu, "Lightweight IoT-based authentication scheme in cloud computing circumstance," *Future Generation Computer Systems*, vol. 91, pp. 244–251, 2019.

[18] T. Maitra, S. K. H. Islam, R. Amin, D. Giri, M. K. Khan, and N. Kumar, "An enhanced multi-server authentication protocol using password and smart-card: cryptanalysis and design," *Security and Communication Networks*, vol. 9, no. 17, pp. 4615–4638, 2016.

[19] R. Martínez-Peláez, H. Toral-Cruz, J. R. Parra-Michel et al., "An enhanced lightweight IoT-based authentication scheme in cloud computing circumstances," *Sensors*, vol. 19, no. 9, p. 2098, 2019.

[20] S. Yu, K. S. Park, and Y. H. Park, "A secure lightweight three-factor authentication scheme for IoT in cloud computing environment," *Sensors*, vol. 19, no. 16, p. 3598, 2019.

[21] M. Alam, I. Jahan, L. J. Rosario, and I. Jerin, "A comparative study of RSA and ECC and implementation of ECC on embedded systems," *Algorithms*, vol. 1, p. 2, 2016.

[22] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[23] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition*, vol. 37, no. 11, pp. 2245–2255, 2004.

[24] I. Khan, S. A. Chaudhry, M. Sher, J. I. Khan, and M. K. Khan, "An anonymous and provably secure biometric-based authentication scheme using chaotic maps for accessing medical drop box data," *The Journal of Supercomputing*, vol. 74, no. 8, pp. 3685–3703, 2018.

[25] D. Mishra, P. Vijayakumar, V. Sureshkumar, R. Amin, S. K. H. Islam, and P. Gope, "Efficient authentication protocol for secure multimedia communications in IoT-enabled wireless sensor networks," *Multimedia Tools and Applications*, vol. 77, no. 14, pp. 18295–18325, 2018.

[26] R. Amin and G. P. Biswas, "A novel user authentication and key agreement protocol for accessing multi-medical server usable in tmis," *Journal of Medical Systems*, vol. 39, no. 3, p. 33, 2015.

[27] J. Moon, Y. Lee, J. Kim, and D. Won, "Improving an anonymous and provably secure authentication protocol for a

mobile user," *Security and Communication Networks*, vol. 2017, article 1378128, 13 pages, 2017.

[28] M. Karuppiah, A. K. Das, X. Li et al., "Secure remote user mutual authentication scheme with key agreement for cloud environment," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 1046–1062, 2019.

[29] D. Wang, Q. Gu, H. Cheng, and P. Wang, "The request for better measurement: a comparative evaluation of two-factor authentication schemes," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pp. 475–486, Xi'an, China, May 2016.

[30] F. Wang, G. Xu, G. Xu, Y. Wang, and J. Peng, "A robust IoT-based three-factor authentication scheme for cloud computing resistant to session key exposure," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 3805058, 15 pages, 2020.

[31] B. Blanchet, "Modeling and verifying security protocols with the applied Pi calculus and ProVerif," *Foundations and Trends® in Privacy and Security*, vol. 1, no. 1-2, pp. 1–135, 2016.

[32] S. A. Chaudhry, M. T. Khan, M. K. Khan, and T. Shon, "A multiserver biometric authentication scheme for tmis using elliptic curve cryptography," *Journal of Medical Systems*, vol. 40, no. 11, p. 230, 2016.

[33] Q. Xie, B. Hu, X. Tan, M. Bao, and X. Yu, "Robust anonymous two-factor authentication scheme for roaming service in global mobility network," *Wireless Personal Communications*, vol. 74, no. 2, pp. 601–614, 2014.

[34] Q. Jiang, Z. Chen, B. Li, J. Shen, L. Yang, and J. Ma, "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1061–1073, 2018.

[35] J. Ryu, H. Lee, H. Kim, and D. Won, "Secure and efficient three-factor protocol for wireless sensor networks," *Sensors*, vol. 18, no. 12, p. 4481, 2018.

[36] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.

[37] L. Kocarev and S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*, Springer Science & Business Media, 2011.

[38] Y. Zhao, S. Li, L. Jiang, and T. Liu, "Security-enhanced three-factor remote user authentication scheme based on Chebyshev chaotic maps," *International Journal of Distributed Sensor Networks*, vol. 15, no. 4, 2019.