Research Article

A Robust and Privacy-Preserving Anonymous User Authentication Scheme for Public Cloud Server

Naveed Khan, Jianbiao Zhang, and Saeed Ullah Jan

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
Department of Computer Science & IT, University of Malakand, Chakadara 18800, Pakistan

Correspondence should be addressed to Jianbiao Zhang; zjb@bjut.edu.cn

Received 18 December 2021; Revised 14 January 2022; Accepted 25 January 2022; Published 14 March 2022

Copyright © 2022 Naveed Khan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Everyone desires to avail online services provided by different service providers securely, efficiently, and effectively. In this regard, security is still a significant concern for them. However, no one guarantees secure communication by browsing different applications remotely. To ensure confidentiality, authorization, availability, nonrepudiation, and removing eavesdropping, without a robust authentication scheme, nothing will go right. Therefore, we attempted to design a robust and privacy-preserving authentication scheme for end-users to securely access public cloud servers’ services remotely without losing performance. Our proposed scheme security has been evaluated formally using the random oracle model (ROM) and ProVerif2.03 and informally using proposition and discussion. At the same time, the performance metric has been analyzed by considering the scheme’s computation and communication costs. Upon comparing the proposed scenario with state-of-the-art work, it has been demonstrated that the scheme is much better in terms of security and performance, as these are contradicting metrics, and the change in one conversely affects the other.

1. Introduction

With the advancement in high-speed Internet and the development of high-performance sensitive applications and smart devices, user privacy and authentication security have become more critical, such as in smartphone scenarios, a user interacts with a cloud server to send and receive data. The users access the cloud servers using smartphones or other portable devices over an insecure channel. From an intruder’s point of view, it is effortless and convenient to carry out malicious attacks and change the behaviour of smart devices.

Furthermore, these malicious attacks can be fatal and severely damage the users and cloud service providers. Moreover, with innovation in mobile technologies, portable and affordable lightweight smartphones, laptops, wearable devices access cloud servers for e-commerce, e-banking, chatting, and many more from anywhere and anytime. According to the author of [1], in 2021, the number of mobile users is estimated to be 7.1 billion, and in 2022, it will reach 7.26 billion, while in 2025, the estimation will be 7.49 billion. Therefore, it is challenging to guarantee user data security and privacy because of its openness in wireless communication and anonymity and authentication issues.

Furthermore, the communication among smartphones and cloud servers will suffer from malicious attacks such as man-in-the-middle, DoS, impersonation, and password-guessing attacks. Moreover, smartphone devices are resource-constrained and battery-powered. Thus, it is vulnerable to various network attacks.

Therefore, authentication is vital among smartphones and cloud servers to protect user data and communication from malicious attacks. But, unfortunately, many protocols prefer usability over security, such as [2–4]. As a result, their scheme cannot resist offline password-guessing attacks and impersonation attacks. Moreover, they did not follow the designing principle of [5] to achieve robust security. Thus, authentication and key agreement protocol that balance security and performance remain a challenge for researchers nowadays.
The first authentication and AKA scheme was proposed in 1981 by the author of [6], where the author used the password to authenticate. After that, several schemes were proposed for the radio frequency identification system [7], multiserver environment [8], and server-client [9]. In addition, the researcher tries to reduce the computational and communication cost in mobile scenarios using a hash function and symmetric key cryptography [7, 10, 11]. However, these schemes achieved computational and communication cost goals but failed to provide forward secrecy according to the author of [5].

Many asymmetric cryptography techniques such as RSA, bilinear pairing, ECC, and Chaotic maps are used to design AKA schemes. The first chaotic map, AKA scheme for e-commerce, was proposed in 2005 [12]. However, according to the author of [13], the scheme used in [12] cannot resist counterfeiting attacks. Later, the scheme used in [12] was improved by the author of [14] using chaotic maps. After that, many schemes were proposed for different applications using chaotic maps [15–19].

However, these proposed schemes used two-factor authentication such as password and smart card or password and biometric. Additionally, a two-factor authentication AKA scheme was proposed by the author of [15, 20] to achieve security and privacy. Moreover, compared to one-factor and two-factor AKA schemes, the three-factor AKA schemes are much more secure and have advantages over one-factor and two-factor AKA schemes. Nevertheless, the three-factor AKA schemes [2, 4, 20] cannot fulfill other security goals. Additionally, we attempt to use the three-factor AKA protocol in the mobile environment to achieve security and maintain performance properties.

1.1. Motivation and Contribution. The existing proposed schemes motivate us to design a three-factor authentication and key agreement scheme based on chaotic maps that provided maximum security over not compromising on performance for a mobile device in cloud computing. The scheme [2, 21] is not even indeed a three-factor authentication scheme, while the protocol proposed by the author of [16] cannot provide user un-traceability. Moreover, the [3, 21] schemes cannot resist offline/online password-guessing attacks and session key security. The scheme used in [20] is prone to counterfeiting attacks.

To overcome the existing scheme security issues, we proposed a new three-factor authentication and key agreement scheme to satisfy the following goals.

(i) The proposed scheme shall be actual three-factor authentication and key agreement scheme such as if the adversary gains two factors, it will not succeed in achieving the third factor.
(ii) The proposed scheme will resist DoS attacks, offline/online password-guessing attacks.
(iii) User anonymity and untraceability are provided.
(iv) The session key between a mobile device and cloud server from an attacker is protected.
(v) The proposed scheme will protect from server impersonation attacks, Mobile user impersonation attacks, and key compromise attacks.
(vi) The proposed scheme will be secured by analyzing the automated verification software toolkit ProVerif 2.03.
(vii) The proposed scheme will achieve maximum security, lower computation, and communication costs and reduce storage overheads compared to existing schemes.

1.2. Threat Model. Adversary A nowadays becomes more powerful; thus, all known possible attacks are possible. The adversary can modify, block, insert, delete, and intercept mobile user and public cloud server communication over an insecure channel based on the existing protocols. However, the adversary cannot obtain the secret key of the cloud server. Furthermore, the adversary can reveal all stored parameters in the smart card. Moreover, the possible threats are as follows:

(1) Spoofing threat: an attacker can impersonate a legal entity by spoofing the IDmu of a real mobile user or cloud server IDPCS.
(2) Routing threat: an adversary can launch wormhole, blockhole, and grey-hole attacks to change the route.
(3) Session key threat: the attacker can obtain the previous session key.
(4) Insider threat: an adversary gains mobile user credentials and tries to launch an insider attack.
(5) Unauthorized access threat: The attacker can gain access to any two of three factors.
(6) Untraceability threat: an adversary can track the physical location of a mobile user or cloud server, leading to fatal damage to both cloud server and mobile user.
(7) Perfect forward secrecy threat: an adversary compromises key impersonation attack, where the attacker can obtain the secret key of the cloud server.
(8) Data leakage threat: an adversary can steal any credentials in data leaks and copy them for later use.
(9) Masquerading threat: the attacker employs a mock identity through valid access identification, such as a network identity, to get unwanted access to the stored information in the public cloud server. A masquerade attack can make an authorization process exceedingly vulnerable if it is not adequately protected.
(10) Impersonation threat: an impersonation attack occurs when an attacker impersonates a trusted contact to deceive a real user into revealing their identity or disclosing critical information.
(11) Man-in-the-Middle threat: an unauthorized person intercepts a communication between two systems.
or individuals, posing a threat. The interceptor tries to eavesdrop on the conversation or impersonate one of the legitimate peers not to notice the intrusion.

(12) Ephemeral secret leakage (ESL) threat: the opponent can divulge the user’s private keys, and the session key can be deduced from intercepted messages.

(13) Brute force threat: in this scenario, the attacker is attempting to crack passwords, encryption keys, and login credentials by using a hacking method on a trial basis. It is a simple but effective method for gaining unauthorized access to individual accounts as well as systems and networks of businesses.

(14) Denial-of-Service (DoS) threat: in denial of service threat, floods on the public cloud server or network have been launched by an attacker in preventing it from responding to queries.

(15) Phishing threat: the purpose is to steal sensitive data such as smart card and login information or infect the victim’s computer with malware.

1.3. Adversary Model. Modeling the role of attackers is an important topic in cyber defense because it aids in ensuring that security assessments are scientifically correct, particularly for conceptual contributions that are difficult to test or for which complete testing is impossible [22]. An adversary model is an operationalization of an attacker in a computer or networked system. The opponent could be an algorithm or a collection of statements/programs about abilities and intentions, depending on how extensive the formalization is? These strategies are used in a variety of computer security domains [23] by an attacker to reach the system and hack its credentials [23]. In light of this model, an adversary interacts with our mobile cloud architecture by posing as a malicious user with a cloud server and acting in the following manner:

(i) An adversary may extract stored data from the cloud server’s memory and use it to verify secret credentials

(ii) An adversary may alter, erase, upgrade, corrupt, or insert false information into a public network channel

(iii) Adversaries may replay, alter, or erase beneficial information exchanged between participants over a private channel

(iv) An adversary may acquire the internal sensitive credential from a stolen mobile device from a user or shape the memory of a stolen or misplaced mobile device using reverse engineering techniques or vital tags in offline mode, but not both simultaneously

1.4. Network Architecture. The proposed scheme consisted of a mobile device ($\mu_{m}$), a public cloud registration centre (PCRC), and a public cloud server (PCS). The mobile device ($\mu_{m}$) is battery-powered with limited resources, while PCRC does not require battery-powered and rich communication and computation costs. All mobile devices and PCRC have unique identities. The PCRC is trusted; thus, an attacker will not compromise it. The mobile devices used PCRC to communicate with PCS for data transmission. The performance of the PCRC is highly impacting the communication cost between mobile devices and PCS. Therefore, the calculated equations of PCRC capacity in noninterference and interference scenarios are worth mentioning. The equations are calculated by the author of [24] and as given as $C = c N a N M m, 2, 3, 3 (a N) 1 - N(m + m s) = N M m, 1 - N M 0, N M m, N M m, N m, N m, N m, N m, N m, N m, N m, N m, N m, N m, N m, N m, N m, N m,$ when the capacity is in noninterference mode, but it will be $f (\Phi) = c N a N m, N M r (N m + N m s) \varphi N m - 1$, when the capacity is in interference mode. The diagrammatic representation of the proposed network model is shown in Figure 1.

2. Related Work

Although there are many benefits of mobile cloud computing, despite the advantages, there are many risks, and the most notable one is outsourcing data storage. The data are distributed at more locations in cloud computing. Therefore, it triggers the risk of unauthorized physical access. However, encryption is the best possible solution to protect the data from unauthorized access. Data encryption and sending it to the cloud can stop unauthorized access from malicious users and cloud service providers. However, these encryption techniques need enhancement. For example, when an attacker compromises a secret key, the data must be protected from unauthorized access.

The authentication and key agreement schemes allow users to log in to the remote servers over an insecure network. The first authentication and key agreement scheme were proposed by the author of [6] in 1981. In this scheme, the server verifies the user through username and password. However, the scheme maintains a password table; therefore, the intruder can intercept the previous password, launch a replay attack, and successfully log in to the server. Moreover, to overcome replay attacks, the author [25] proposed a two-factor authentication scheme in 1990. The two-factor scheme use username, password, and a second factor, a smart card, has been used.

Two-factor authentication and key agreement scheme were proposed by the author of [26] using chaos theory in 2013. However, in [27], the author found out that the scheme used in [26] cannot provide session key security and anonymity. Additionally, the author of [15] cryptanalysis of the scheme [27] concludes that the scheme is vulnerable to DoS attacks and insider attacks and cannot ensure secure key agreement. Therefore, the author of [15] proposed two-factor authentication and a key agreement scheme to eliminate the issue in the scheme [27]. According to the author of [28], the scheme used in [15] cannot resist impersonation, key compromise, information leakage attacks, and the inability to provide local password updates and detect incorrect passwords.
Additionally, three-factor authentication and key agreement protocols recently got attention in dealing with smart card loss attacks. The three-factor typically used username with password, smart card, and biometric identification. However, the traditional authentication and key agreement scheme are used only for a single server environment, whereas commercial services are based on a multiserver environment [23]. Therefore, these conventional authentications and key agreement schemes do not provide users anonymity and untraceability. Furthermore, in [27], the author proposed a three-factor scheme where the third factor is biometric authentication. However, according to the author of [29], the scheme used in [3] fails to provide user anonymity and impersonating attacks. Nevertheless, another three-factor multiserver scheme was proposed in [30, 31], and according to the author of [32], the scheme used in [31] is defenseless against user impersonation attacks.

The security improvements of the scheme [33] are proposed in [34]. However, the [34] scheme also has security drawbacks, including privileged insider attacks and smart card losses. Therefore, the scheme [35, 36] provides the solution to the flaws of [34]. However, the security defects of [37, 38] are identified by the author of [39] and improve the scheme of [40] to achieve user anonymity. Additionally, the scheme in [41] stored the user’s public keys on the server-side. On the other hand, the scheme developed by the author of [42] suffers from insider attacks and cannot provide user untraceability. According to the author of [43], the protocol used in [44] is vulnerable to impersonation, reply, DoS attacks, and fails to provide strong user anonymity. However, the author also claims that their protocol has high computation and communication costs, storage costs, and no balance between performance and security. Meanwhile, the author [43] proposed a solution for [44] to achieve high security, tractability, robustness, and lightweight feature. Hence, an authentication scheme using a smart card is proposed the author of [45], which offers reliable information delivery and mutual authentication between server and client. Furthermore, BioHashing techniques have been used to prevent biometrics information from being stolen or misplaced smart card.

Consequently, in 2014, a three-factor AKA scheme was proposed in [46] and claimed that it provides security against smart card loss attacks and many more threats. Nevertheless, according to the author of [19], the scheme is vulnerable to offline password-guessing attacks, smart card loss attacks, and biometric sample leaks. Furthermore, in 2015, the author [16] cryptanalysis the scheme [47] and found out that the scheme cannot provide mutual authentication and also vulnerable to replay attacks, DoS attacks, and password guessing attacks. However, according to the author of [16], the scheme used in [16] has not indeed achieved three-factor authentication and key agreement and cannot resist offline password-guessing attacks.

Later in 2017, a scheme was proposed in [2] that cannot provide perfect forward secrecy and truly three-factor authentication and resist offline password-guessing attacks. In addition, the schemes [3, 4] are also vulnerable to password-guessing attacks, and the scheme in [4] is also prone to impersonation attacks.

The most recent schemes were proposed from 2018–2021 to achieve security features and reduce storage overhead, communication, and computation costs. However, these schemes have the vulnerability to provide perfect security in current scenarios. In contrast, the scheme [48] did not offer traceability and mutual authentication. Therefore, [49, 50] proposed a three-factor authentication scheme based on ECC to achieve perfect forward secrecy. However, these schemes cannot provide perfect forward secrecy and user anonymity and resist replay attacks.

Furthermore, the protocol [51] provides security features over the cost of computation. The author [52] proposed a lightweight authentication scheme, although its key generation time is very high. Therefore, it contracted with the feature of a lightweight scenario.

Furthermore, a scheme [8] was proposed which used symmetric en/decryption, hash function, and chaotic maps to provide authentication and key agreements for multi-server environments. However, according to the author of [28], their scheme is prone to offline password-guessing attacks, biometric, and smart card leaks. Recently, Jiang et al. [53] proposed a scheme for cloud-assisted autonomous vehicles in which they used biometrics and fuzzy extractor for authentication consisting of user registration, user authentication, and biometric extraction phases. After extensive analysis, the scheme proposed by the author of [53] shows the following loopholes.

2.1. Lack of Strong User Anonymity and Unlinkability. If adversary A finds a misplaced mobile device or stolen from a legitimate user and restore C, tpi from its memory by reverse engineering, they can also get UID, from the public channel easily due to the availability of C, tpi, credentials. Moreover, the adversary also gets pk from message 1 and vk from message 2 easily due to its publicly transmitted network channel. Moreover, in [53], the key is unchanged, and the
adversary can easily figure out other credentials. Therefore, Jiang et al.'s [53] scheme is vulnerable to user anonymity and unlinkability.

2.2. Lack of Perfect Forward Security. If by power analysis method, the adversary recovers the public key from user smart card/mobile device, then they can be allowed to compute secret session key vuk = h(UID||VID||CID||PKmu||n_1||n_3), and vcsk = h(VID||CID||n_2||n_3||vk). Due to easily computing, pk = h(ps||n_3||CID) and vk = h(VID||k||r_1). Therefore, jiang et al. [53] does not provide perfect forward secrecy.

2.3. Side-Channel Attack. Jiang et al. [53] do not use a timestamp in each round trip, which leads to a side-channel attack.

3. Proposed Scheme

This section of the research paper will demonstrate the proposed mechanism for such a crucial infrastructure that everyone needs to browse information securely. Our proposed scheme has three participants. The first participant is the legitimate user who inputs their identity IDmu and password PWmu, and the second is the public cloud server PCS, and the last is the public cloud registration centre (PCRC). The public cloud registration centre selects secret key SKPCRC, public key PKPCRC, and random number r_1, which are only known to PCRC and publicly available public key. Furthermore, the notation is presented in Table 1. The proposed scenario consists of registration, login, and authentication, and password/biometric change phases, and each of these phases are described one by one under the following headings.

3.1. User’s Registration Phase. This phase of the proposed scenario competes in the following steps:

(i) Step 1: In this phase, the mobile user selects his/her identity IDmu and password PWmu, imprints biometric bio(Bmu), and chooses a random number r_2 and calculates S_1 = h(IDmu||PWmu||r_2||bio(Bmu)), S_2 = h(PWmu||r_2), and S_3 = S_1 ⊕ r_2. After calculation, the mobile user sends IDmu, S_2, S_3 towards the public cloud registration centre.

(ii) Step 2: After receiving IDmu, S_2, S_3, the public cloud registration centre calculates PKmu = r_1P, SKmu = r_1P ⊕ h(IDmu ⊕ PKmu), N = S_3 ⊕ S_1, M = h(IDmu ⊕ r_2) ⊕ h(IDmu||S_3||SKmu), and O = h(IDmu||S_3||PKmu). The public cloud registration server sends smart card with credentials [PKmu, N, M, O] towards the mobile user.

(iii) Step 3: After receiving smart card with credentials [PKmu, N, M, O] from the public cloud registration centre, the mobile user calculates further C = r_2 ⊕ bio(Bmu), N’ = N ⊕ r_2, and store [O, M, N’, PKmu] in smart card as shown in Table 2.

3.2. Public Cloud Server (PCS)’s Registration Phase. This phase is completed in the following steps:

(i) Step 1: The public cloud server selects identity IDPCS, chooses 160 bits integer q, and calculates PIDPCS = IDPCS||q. After calculation, the public cloud server sends PIDPCS towards the public cloud registration centre.

(ii) Step 2: After receiving PIDPCS from public cloud server, the public cloud registration centre chooses a random number r_1 and calculates PKPCS = r_1Q, SKPCS = (r_1P)⊕(PIDPCS||PKmu), and send PKPCS, SKPCS, r_1 back to public cloud server.

(iii) Step 3: The public cloud server stores [SKPCS, r_1] and publishes PKPCS, as shown in Table 3.

3.3. Login and Authentication Phase. This phase is a crucial stage of the protocol, which is accomplished in the following steps:

(i) Step 1: The legitimate user first inputs their identity, provides password, and imprints biometrics, the EEPROM inside the chip while computing e = c ⊕ bio(Bmu), O’ = h(IDmu||S_3||SKmu), and confirms O’ = 0, and if not matched, the process will terminate locally, else, proceed f = (r_2-P), and transmit MSG_1 over a public network channel.

(ii) Step 2: When receiving MSG_1, the public cloud server computes PKmu = (r_1P), Q = r_1 ⊕ SKPCS h(r_1P⊕r_2P) and transmits MSG_2 back towards the user again on the same public channel.

(iii) Step 4: The user, upon receiving MSG_2, verifies Q’ = r_1 ⊕ PKPCS h(r_1P⊕r_2P), computes S_1 = h(IDmu||PWmu||r_2||bio(Bmu)), D = (r_1 ⊕ PKmu) h(PIDPCS ⊕ (r_1P)) ⊕ (r_2P), M = h(IDmu || PWmu || r_2 || bio(Bmu)), and if not matched, considers for potential replay attack, else, computes the session secret shared key SKPCS = h(r_1P, r_2P), PKmu = (r_1P), SKmu = h(IDmu||PKmu||N||M||O), and computes Y = h(SKmu, r_1, r_1P, r_2P, IDmu), W = h((r_1P, r_2P)), and transmits MSG_3 towards public cloud server over open channel.

(iv) Step 3: The server, when receiving MSG_3, computes W’ = h((r_1P, r_2P)), D’ = (r_1 ⊕ PKmu) ⊕ (PIDPCS ⊕ (r_1P) ⊕ (r_2P)), confirms W’ = W/D = D, if not matched, considers for potential replay attack, else, computes the session secret shared key SKPCS = h(r_1P, r_2P), PKmu = (r_1P), SKmu = h(IDmu||PKmu||N||M||O), and keeps SKmu, SKPCS is the session secret key for future communication, as shown in Table 4.

3.4. Biometric and Password Change Phase. First, a legitimate user enters his/her old identity and password and imprints biometrics (IDmu, PWmu, bio(Bmu)), takes a random number, and calculates S_1 = h(IDmu||PWmu) ⊕ h(r_2||bio(Bmu)), S_2 = h(PWmu||r_2), S_3 = S_1 ⊕ r_2. If this set
of calculations is performed successfully, they will locally be asked for the provision of the new identity and password and imprints biometrics \( \{ID_{mu}^{\text{new}}, PW_{mu}^{\text{new}}, \text{bio}(B_{mu})^{\text{new}} \} \). The EEPROM locally makes necessary computations by using replication and generation functions to make the credentials secure for any resisting any insider attack; to do so, \( S_1^{\text{new}} = h(ID_{mu}^{\text{new}}||PW_{mu}^{\text{new}}||r_2^{\text{new}}||\text{bio}(B_{mu})^{\text{new}}) \), \( S_2^{\text{new}} = S_{new}^{\text{new}} \oplus r_2^{\text{new}} \). Furthermore, \( S_1, S_2, \text{and} S_3 \) are replaced with \( S_1^{\text{new}}, S_2^{\text{new}}, \text{and} S_3^{\text{new}} \). Similar is the case for the password and other credentials changing.

## 4. Security Analysis

In this section of the research, we will investigate, scrutinize, and analyze the security of the proposed protocol by using two methods. First, to check whether the random number exchanges among the participants are securely communicated or not? Whether the hash code created will create a collision with other code or not? Similarly, we also will check the advantage of an adversary to break our protocol. To do so, we use the following methods.

| Table 1: Notation explanation. |
|--------------------------------|
| **Notation** | **Description** |
| \( u_{mu} \) | Mobile user |
| PCS | Public cloud server |
| PCRC | Public cloud registration centre |
| \( r_1, r_2 \) | Random numbers |
| \( SK_{mu}, PK_{mu} \) | Secret and public key of mobile users |
| \( SK_{PCRC}, PK_{PCRC} \) | Secret and public key of public cloud server |
| \( ID_{mu}, ID_{PCRC} \) | Identity of a mobile user and public cloud server |
| \( PW_{mu} \) | Mobile user password |
| \( \text{bio}(B_{mu}) \) | Mobile user biometric |
| \( h(.) \) | One-way hash function |
| \( \oplus \) | XOR |
| \( || \) | Concatenation |

| Table 2: User registration. |
|-----------------------------|
| **User** | **Public cloud registration centre (PCRC)** |
| User choose identity \( ID_{mu} \) | |
| Password \( PW_{mu} \) and imprints biometric \( \text{bio}(B_{mu}) \) | |
| Choose a random number \( r_2 \) and calculate | |
| \[ S_1 = h(h(ID_{mu}||PW_{mu}||r_2||\text{bio}(B_{mu}))) \] | |
| \[ S_2 = h(PW_{mu}||r_2) \] | |
| \[ S_3 = S_1 \oplus r_2 ID_{mu}, S_2, S_3 \] | |
| Calculate: | |
| \[ ID_{mu}, S_2, S_3 \] | |
| \[ PK_{mu} = r_1, P \] | |
| \[ SK_{mu} = r_1, P \oplus h(ID_{mu} \oplus PK_{mu}) \] | |
| \[ N = S_3 \oplus SK_{mu} \] | |
| \[ M = h(ID_{mu} \oplus S_3) \oplus S_2 || SK_{mu} \] | |
| \[ O = h(ID_{mu} || S_3 || PK_{mu}) \] | |
| Calculate: SC \( \{PK_{mu}, N, M, O\} \) | |
| \[ C = r_2 \oplus \text{bio}(B_{mu}) \] | |
| \[ N' = N \oplus r_2 \] | |
| Store \( \{O, M, N', PK_{mu}\} \) in smart card | |

| Table 3: Public cloud server registration phase. |
|-----------------------------------------------|
| **Public cloud server (PCS)** | **Public cloud registration centre (PCRC)** |
| Select \( ID_{PCRC} \) and 160-bits big integer \( q \) | |
| \[ PID_{PCRC} = 1ID_{PCRC}||q \] | |
| \[ PK_{PCRC} = r_1, q \] | |
| \[ SK_{PCRC} = (r_1, P) \oplus h(PID_{PCRC} || PK_{PCRC}) \] | |
| Store \( [SK_{PCRC}, r_1] \) and publish \( PK_{PCRC} \) | |
4.1. Formal Security Analysis. To determine the security of the proposed protocol using a formal approach, we, in this subsection of the research, will use a random oracle model (ROM) (advantage with the adversary to breach the proposed protocol).

4.1.1. Random Oracle Model (ROM) Analysis. Suppose $X$ means protocol; the external user who is currently using our protocol is denoted as $U$, the registration centre is denoted by $G$, and the public cloud server is $PCS$. When running $X$, each participant has many occurrences to be touched with $X$ via $r_1, r_2, p, q, PK, SK$. Furthermore, suppose we create a table of random numbers called an oracle. Also, let $I'$ be the $x$th occurrence of $U$, $J'$ is the $y$th occurrence of $G$, and $K'$ is the $z$th occurrence of $PCS$. While $I'$ is supposed to be the occurrence of all participants, then definitely, there are possibly three occurrences available $A, B, C$. $A$ is successfully usage of the protocol, and the user is securely authenticating with the destination, $B$ never authenticate, and $C$ = no result. Afore running $X, U$ has MSG1, $G$ has MSG2, and $PCS$ has MSG3, and suppose the shared secret key $SK$ is stored securely in the memory of $U, G$, and $PCS$.

Suppose an adversary desires to enter our protocol over the open channel and try to start their own session or terminate the $U$ session by arbitrating the participation. The adversary must be known $I' = (r_2, P)$ and $PK_{u} = (r_1, P)$, and then, he/she can execute $[P', P]$ and $[P', P']$ queries. In this regard, $A$’s advantage to breach the proposed protocol is

$$\text{Adv}_x (A) = |2\text{Prob}[C] + [C' - 1]|,$$

whereas $C$ means flipping a coin by the adversary, and when flipping the coin, the result is $C'$. For hash queries, the advantage with the adversary $A$ is

$$\text{Adv}_x (A) = |\text{Prob}(S_2) - \text{Prob}(S_2)| + q'(h^{2^a})S_1 + q'(h^{2^a+1})S_1 + q'(h^{2^a})S,$$  

whereas $q'(h^{2^a})S_1 + q'(h^{2^a+1})S_1 + q'(h^{2^a})S$ are at most chances of collision of hash code with each other in the oracle. By expanding (2), we get

$$|\text{Prob}(S_2) - \text{Prob}(S_2)| \leq \frac{2q_{U} + 2q_{PCS}S}{2q_{G}}$$

The advantage with the adversary to capture shared session key as

$$|\text{Prob}(S_2) - \text{Prob}(S_2)| \leq q_{PCS}.\text{Adv}_x (A).$$

If we keep another list of numbers/dictionary ($D$), then probability with the adversary $A$ is

Table 4: Login and authentication phase.

| User (U) | Public cloud server (PCS) |
|------------------|------------------|
| Input IDmu, PWmu, and bio(Bmu) | Compute: PKmu = (r1||P) |
| Calculate: e = C ⊕ bio(Bmu) | $Q = r_1 \oplus SKPCS||h(r_1||P) \oplus r_2||P$ |
| $O' = h(IDmu||S_2||SKmu)$ | Compute: $W' = h((r_1||P) \oplus (r_2||P))$ |
| Check $O' = O$ proceed further | $D' = (r_1 \oplus PKmu) \oplus (PIDPCS \oplus (r_1||P) \oplus (r_2||P))$ |
| $J = (r_2||P)$ | Check $W' = W$, and $D' = D$, proceed further |
| Compute: $Q' = r_1 \oplus PKPCS \oplus h(r_1||P) \oplus (r_2||P)$ | $S_{PCS} = h(r_1||P) \oplus (r_2||P)||h(IDmu||PKmu)||h(IDmu||PIDPCS)$ |
| $S_{PCS} = h(r_1||P) \oplus (r_2||P)||h(IDmu||PIDPCS)||h(IDmu||PIDPCS)$ | Compare: $Y' = h(SKmu||r_1) \oplus (r_1||P) \oplus (r_2||P)||IDma$ |
| $Y' = h(SKmu||r_1) \oplus (r_1||P) \oplus (r_2||P)||IDma$ | $Y' = h(SKmu||r_1) \oplus (r_1||P) \oplus (r_2||P)||IDma$ |

Session key $SK_{mu} = SK_{PCS}$
\[ |\text{prob} (S_3)| = \frac{1}{2} \max \left( \frac{q_U}{2}, \frac{q_{\text{PCS}}}{D} \right) \]  

(5)

By combining (1)–(5), we get

\[
\text{Adv}_x (A) = \text{Prob} (S_t) - 1
\]

\[
= 2|\text{Prob} (S_0 - \text{Prob} (S_t)| + \max \left( \frac{q_U}{2}, \frac{q_{\text{PCS}}}{D} \right)
\]

\[
\leq 2|\text{Prob} (S_0 - \text{Prob} (S_t)| + \max \left( \frac{q_U}{2}, \frac{q_{\text{PCS}}}{D} \right)
\]

\[
= 2\left[ \text{Prob} (S_1) - \text{Prob} (S_2) + \text{Prob} (S_3) - \text{Prob} (S_4) \right] + \max \left( \frac{q_U}{2}, \frac{q_{\text{PCS}}}{D} \right)
\]

\[
\leq \frac{q_U^2 + q_{\text{PCS}}^2 + q_G^2}{2^q} + \frac{(q_U + q_{\text{PCS}})^2}{2(q-1)} + 2q_{\text{PCS}} \cdot A \cdot \nu (A_{\text{PCS}}) + 2 \left( \frac{q_U}{2^q} \cdot \frac{q_{\text{PCS}}}{|D|} \right).
\]

4.1.2. **ProVerif2.03 Simulation.** To check whether the session secret is secured, confidentiality computed, and exchanged security among different peers or to check its reachability and test whether an attacker can forage it during a starting session or not, ProVerif2.03 verification software toolkit is used. This is a widely accepted software verification toolkit. The whole code and result are generated as follows:

\[
\text{query id:bitstring; inj-event(end_U(IDUi))} \Rightarrow \text{inj-event(start_U(IDmu))}.
\]

\[
\text{query id:bitstring; inj-event(end_S(IDS))} \Rightarrow \text{inj-event(start_S(PIDpcs))}.
\]

\[
\text{query attacker(SKmu).}
\]

\[
\text{(* = = = = = Events* = = = = = = = =)}
\]

\[
\text{event start_U(bitstring).}
\]

\[
\text{event end_U(bitstring).}
\]

\[
\text{event start_PCS(bitstring).}
\]

\[
\text{event end_PCS(bitstring).}
\]

\[
\text{(* = = = = = Constructors = = = = = = = =)}
\]

\[
\text{fun XOR(bitstring, bitstring): bitstring.}
\]

\[
\text{fun H(bitstring): bitstring.}
\]

\[
\text{fun CONCAT(bitstring, bitstring): bitstring.}
\]

\[
\text{(* = = = = = Equations = = = = = = = =)}
\]

\[
\text{equation forall m: bitstring, n: bitstring; XOR(XOR(m,n), m) = n.}
\]

\[
\text{(* = = = = = USER = = = = = = = = = =)}
\]

\[
\text{let U = event start_U(IDmu);}
\]

\[
\text{free e:bitstring.}
\]

\[
\text{let e = XOR(C, Bmu) in}
\]

\[
\text{let O = H(H(IDmu, S3), SKmu) in}
\]

\[
\text{if Odash = O then}
\]

\[
\text{let j = H(r2, P) in}
\]

\[
\text{out(PublCh, (j));}
\]

\[
\text{in(PublCh, (r1:bitstring, P:bitstring, Q:bitstring)) in}
\]

\[
\text{let Qdash = XOR(XOR(r1, PKpcs), H(H(r1, P), r2)) in}
\]

\[
\text{let S1 = H(H(H(IDmu, PWmu), r2), Bmu) in}
\]

\[
\text{let D = XOR(XOR(XOR(r1, PKmu), PIDpcs), r1), r2)) in}
\]
Upon running the code, the following result will generate, which shows no attacker at any stage cannot forge, crack, and hack the secret session key. Also, its reachability is confirmed among the participants.

Completing equations...
Completing equations...

-- Process 1-- Query not attacker(SK[]) in process 1
Translating the process into Horn clauses...
Completing...

200 rules inserted. Base: 136 rules (27 with conclusion selected). Queue: 24 rules.
Starting query not attacker(SK[])
RESULT not attacker(SK[]) is true.

-- Query inj-event(end_U(IDmu[])) => inj-event(start_U(IDmu[])) in process 1
Translating the process into Horn clauses...
Completing...

200 rules inserted. Base: 126 rules (27 with conclusion selected). Queue: 28 rules.
Starting query inj-event(end_PCS(PIDmu[])) => inj-event(start_PCS(PIDmu[]))
RESULT inj-event(end_PCS(PIDmu[])) => inj-event(start_PCS(PIDmu[])) is true.

- Query inj-event(end_U(IDmu[])) => inj-event(start_U(IDmu[])) in process 1
Translating the process into Horn clauses...
Completing...

200 rules inserted. Base: 136 rules (27 with conclusion selected). Queue: 24 rules.
Starting query inj-event(end_PCS(IDmu[])) => inj-event(start_U(IDmu[]))
RESULT inj-event(end_PCS(PIDmu[])) => inj-event(start_U(IDmu[])) is true.

Verification summary:
Query not attacker(SK[]) is true.
Query inj-event(end_U(IDmu[])) => inj-event(start_U(IDmu[])) is true.
Query inj-event(end_PCS(PIDmu[])) => inj-event(start_PCS(PIDmu[])) is true.

4.2. Informal Security Analysis. This section will intensely discuss the possible attacks and their prevention and discuss our proposed scheme features.

(1) Provide user anonymity: the proposed scheme provides user anonymity. When mobile user sends ID_{mu} by calculating, \( W = h(r_1, P) \). The \( (r_1, P) \) and \( (r_2, P) \) are only known to the public cloud server. Therefore, the proposed scheme provides user anonymity.

(2) Provide user untraceability: the A cannot extract information from \( W = h(r_1, P) \) because it is only known to a public cloud server. Thus, the proposed is a protected user untraceability feature.

(3) Resists to offline password-guessing attacks: the A can get \( W = h(r_1, P) \), \( Y = h(SK_{mu}, r_1, r_1, P) \), \( ID_{mu} \), \( D = (r_1 \oplus PK_{mu}) h(PID_{PCS} \oplus (r_1, P) \oplus (r_2, P)) \) on insecure medium. Although the A can also extract contents from smart card \( \{PK_{mu}, N, M, O\} \), the A cannot get mobile user password PW_{mu} from these credentials. Therefore, the proposed scheme resists offline password attacks.

(4) Resist to masquerading attacks: in the proposed scheme, the A cannot launch an impersonation attack because of unable to get PW_{mu} or the secret keys of the mobile user. The A cannot modify the \( Y = h(SK_{mu}, r_1, r_1, P, r_2, P) \) because of \( (r_1, P), (r_2, P) \). Therefore, the proposed scheme resists masquerading attacks.
(5) Resist to smart card stolen attacks: The A guesses mobile user password PWmu and launches a user impersonation attack. However, the A cannot launch forgery, modification, and replay attacks in the proposed scheme because the A will need a mobile user secret key and the random number to compute $Y = h(SK_{\text{mu}} r_1, r_1 P, r_2 P, ID_{\text{mu}})$. Therefore, the proposed scheme resists smart card stolen attacks.

(6) Provide perfect forward secrecy: the proposed scheme provides perfect forward secrecy even if the secret key of the public cloud registration server is lost. The proposed scheme generates session key $S_{\text{Kmu}} = h(r_1 P, r_2 P) h(ID_{\text{mu}}) [h(SK_{\text{mu}}) r_1 || h(ID_{\text{mu}}) || PID_{\text{PCS}}]$ by using random numbers with multiplication with $P$. It is very difficult for $A$ to compute those numbers because of ECDLP in a short period of time.

(7) Provide known key security: the A cannot get mobile user and public cloud server secret keys and random numbers. Although if the A gets the session key, it cannot extract secret keys and random numbers. Therefore, the proposed scheme protects against the known key attacks.

(8) Resist to session key attacks: if the A gets a random number, it still requires a mobile user secret key to generate a session key. Therefore, the proposed scheme resists session key attacks.

(9) Resist to credentials leakage attacks: The A may get some mobile user leaked credentials and try to launch a masquerading server attack. Our proposed protocol resists such attacks even if mobile user credentials are leaked. The A cannot generate $Q = r_1 \oplus SK_{\text{PCS}} h(r_1 P \oplus r_2 P)$ because of the public cloud server secret key. Therefore, the proposed scheme resists credentials leak attacks.

(10) Resist to replay attacks: the A may get the previous session key and extract meaningful information to launch the replay attack. In the proposed scheme, the public cloud server checks $W' = W$ and $D' = D$ to stop replay attacks. Thus, the proposed protocol resists replay attacks.

(11) Resist to denial-of-service attacks: each session starts with a fresh and unique session key. The proposed scheme random number $r_1, r_2$ is very hard to compute while the proposed scheme checks $Q' = Q$, thus if A sends incorrect information to any participants, such request is rejected. Therefore, the proposed scheme resists denial of service attacks.

(12) Resist man-in-the-middle attacks: in our proposed scheme, the mobile user and public cloud server share session key after authenticating each other. Therefore, A cannot construct a connection with the mobile user and public cloud server because A needs random numbers, $r_1, r_2$, bio$(B_{\text{mu}})$, ID_o, PWmu and public cloud server secret key SK_{PCS}. Thus, the proposed scheme resists a man-in-the-middle attack.

(13) Provide mutual authentication: in our proposed scheme’s login and authentication phase, the public cloud server computes $Q = r_1 \oplus SK_{\text{PCS}} [h(r_1 P \oplus r_2 P)]$ while the mobile user verifies $Q' = Q$. On the other side, the mobile user calculates $D = (r_1 \oplus PK_{\text{mu}}) \oplus (PID_{\text{PCS}} \oplus (r_1 P \oplus (r_2 P)))$ while the public cloud server verifies $D' = D$. Therefore, mobile users and public cloud servers authenticate each other.

5. Performance Analysis

This section will calculate communication and computation cost and compare our proposed scheme with other recent related protocols regarding communication and computation cost and possible attacks and scheme features.

5.1. Communication Cost. In this section, we will discuss the communication cost in detail. The computation of communication cost is calculated based on the messages transmitted between the mobile user and public cloud server in the login and authentication phase. MSG1 {J}, MSG2 {r_1, P, Q}, MSG3 {W, Y, D} is equal to 160 bits + 160 bits + 160 bits + 160 bits + 160 bits + 160 bits + 160 bits = 1120.

5.2. Computation Cost. We will consider the work done by the author of [54–62] for the time consumed during session key establishment among different participants, as shown in Table 5.

Therefore, keeping in view the aforementioned values, the computation time for the proposed authentication protocol is given as

\[
\text{USER SIDE COMPUTATION:} \quad 12T_o + 1T_h + 1TR + 11T_{\text{MLT}}.
\]

By putting values from Table 5, we get
\[
\begin{align*}
&= 12(0.0288) + 11(0.0023) + 1(0.539) + 11(0.0171) \\
&= 0.3456 + 0.0253 + 0.539 + 0.1881 \\
&= 1.098 \text{ ms.} \quad (7)
\end{align*}
\]

\[
\text{SERVER SIDE COMPUTATION:} \quad 7T_o + 7T_h + 2TR + 11T_{\text{MLT}}.
\]

By putting values from Table 5, we get
\[
\begin{align*}
&= 7(0.0288) + 7((0.0023) + 2(0.539) + 11(0.0171)) \\
&= 0.2016 + 0.0161 + 1.078 + 0.1881 \\
&= 1.4838 \text{ ms.} \quad (8)
\end{align*}
\]

By adding equations (7) and (8), we get
\[
\begin{align*}
&= 1.098 + 1.183 \\
&= 2.281 \text{ ms.} \quad (9)
\end{align*}
\]

Therefore, the time required to compute session shared key among user and server is just 2.281 milliseconds.

5.3. Comparison Analysis. In this section, we will compare our scheme with recently published protocols such as
Table 5: Computation time for different cryptographic functions.

| Function descriptions                      | Notation | Computation cost in ms |
|--------------------------------------------|----------|------------------------|
| Time for a collision-free hash function    | $T_h$    | 0.0023 ms              |
| Time for bitwise XOR operation             | $T_{\oplus}$ | 0.0288 ms            |
| Extraction of random number                | $T_{RN}$ | 0.539 ms               |
| Multiplication time complexity             | $T_{MLT}$| 0.0171 ms              |

Table 6: Communication cost comparison.

| Schemes | No of messages | Total cost |
|---------|----------------|------------|
| [53]    | 4              | 5280       |
| [54]    | 3              | 1728       |
| [55]    | 3              | 1536       |
| [56]    | 3              | 1696       |
| [57]    | 3              | 2528       |
| [58]    | 4              | 1312       |
| [59]    | 3              | 2560       |
| [60]    | 3              | 2016       |
| [61]    | 2              | 1184       |
| [62]    | 3              | 1184       |
| Our     | 3              | 1120       |

Table 7: Security features.

| Features | [53] | [54] | [55] | [56] | [57] | [58] | [59] | [60] | [61] | [62] | Our |
|----------|------|------|------|------|------|------|------|------|------|------|-----|
| A        | X    | X    | ✓    | ✓    | ✓    | ✓    | X    | X    | ✓    | ✓    | ✓   |
| B        | ✓    | ✓    | X    | ✓    | ✓    | ✓    | ✓    | ✓    | X    | ✓    | ✓   |
| C        | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| D        | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| E        | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| F        | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| G        | ✓    | ✓    | X    | X    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| H        | ✓    | ✓    | ✓    | ✓    | ✓    | X    | ✓    | ✓    | ✓    | ✓    | ✓   |
| I        | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| J        | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| K        | ✓    | ✓    | X    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| L        | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |

A: Provide user anonymity. B: Provide mutual authentication. C: Resist privileged insider attack. D: Resist offline password-guessing attack. E: Resist stolen smart card attack. F: Resist replay attack. G: Session key. H: Provide key management. I: Resist impersonation attack. J: Resist to DoS attack. K: Provide untraceability. L: Resist brute-force attack.

Figure 2: Communication and computation cost comparison.
References

[54–61] in terms of communication and computation cost and security features. Table 6 shows the comparison of our proposed scheme with other state-of-the-art schemes in terms of communication cost, while the security features are shown in Table 7. Moreover, Figure 2 shows communication and computation cost comparison with other schemes in graphical representation. The result shows that our scheme is lightweight and has more security features than other schemes.

6. Conclusion

This research article proposes a robust privacy-preserving authentication and key agreement scheme, which guarantees secure communication among participants. The possible threats to the system and power with an adversary have been highlighted and designed a scheme for three participants (user, PCRC, and PCS). The security analysis section shows that our protocol is secure and effective informally and formally. In the end, the result of performance metrics of the scheme offers a delicate balance with security which is most probabilistically missing in prior protocols. Upon checking the different security functionalities, our scheme is much accurate, robust, and lightweight and preserves the privacy of an end-user. In the future, we have planned to simulate the scheme using the AVISPA tool and NS3.

Data Availability: If anyone desires to need data like figures, tables, code, and modules of this article that are utilized in support of the study’s strength/findings, they can correspond with the principal author of the paper. They will be facilitated by sending the relevant materials upon request.

Data Availability

If anyone needs data like figures, tables, code, and modules of this article that are utilized in support of the study’s strength/findings, they can correspond with the principal author of the paper. They will be facilitated by sending the relevant materials upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Beijing Key Laboratory of Trusted Computing in the Beijing University of Technology, Beijing.

References

[1] S. O’Dea, “Forecast number of mobile users worldwide 2020-2025,” 2021, https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/.
[2] M. Wazid, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, “Secure three-factor user authentication scheme for renewable-energy-based smart grid environment,” IEEE Transactions on Industrial Informatics, vol. 13, no. 6, pp. 3144–3153, 2017.
[3] 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 of Things Journal, vol. 5, no. 4, pp. 2884–2895, 2017.
[4] S. H. Islam, P. Vijayakumar, M. Z. A. Bhuiyan, R. Amin, and B. Balasamy, “A provably secure three-factor session initiation protocol for multimedia big data communications,” IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3408–3418, 2018.
[5] C.-G. Ma, D. Wang, and S.-D. Zhao, “Security flaws in two improved remote user authentication schemes using smart cards,” International Journal of Communication Systems, vol. 27, no. 10, pp. 2215–2227, 2014.
[6] L. Lamport, “Password authentication with insecure communication,” Communications of the ACM, vol. 24, no. 11, pp. 770–772, 1981.
[7] P. Gope, J. Lee, and T. Q. S. Quek, “Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions,” IEEE Transactions on Information Forensics and Security, vol. 13, no. 11, pp. 2831–2843, 2018.
[8] 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 Transactions on Dependable and Secure Computing, vol. 15, no. 5, pp. 824–839, 2016.
[9] J. Arkko, V. Torvinen, G. Camarillo, A. Niemi, and T. Haukka, “Security Mechanism Agreement for SIP Sessions,” IETF Internet Draft (Draft-ietf-sipsecagree-04.txt), 2002.
[10] M. L. Das, A. Saxena, and V. P. Gulati, “A dynamic ID-based remote user authentication scheme,” IEEE Transactions on Consumer Electronics, vol. 50, no. 2, pp. 629–631, 2004.
[11] Z. Yang, J. He, Y. Tian, and J. Zhou, “Faster authenticated key agreement with perfect forward secrecy for industrial Internet-of-Things,” IEEE Transactions on Industrial Informatics, vol. 16, no. 10, pp. 6584–6596, 2019.
[12] D. Xiao, X. Liao, and K. Wong, “An efficient entire chaos-based scheme for deniable authentication,” Chaos, Solitons & Fractals, vol. 23, no. 4, pp. 1327–1331, 2005.
[13] G. Alvarez, “Security problems with a chaos-based deniable authentication scheme,” Chaos, Solitons & Fractals, vol. 26, no. 1, pp. 7–11, 2005.
[14] D. Xiao, X. Liao, and S. Deng, “A novel key agreement protocol based on chaotic maps,” Information Sciences, vol. 177, no. 4, pp. 1136–1142, 2007.
[15] T.-F. Lee, C.-H. Hsiao, S.-H. Hwang, and T.-H. Lin, “Enhanced smartcard-based password-authenticated key agreement using extended chaotic maps,” Plos one, vol. 12, no. 7, Article ID e0181744, 2017.
[16] Y. Liu and K. Xue, “An improved secure and efficient password and chaos-based two-party key agreement protocol,” Nonlinear Dynamics, vol. 84, no. 2, pp. 549–557, 2016.
[17] L. Han, Q. Xie, W. Liu, and S. Wang, “A new efficient chaotic maps based three factor user authentication and key agreement scheme,” Wireless Personal Communications, vol. 95, no. 3, pp. 3391–3406, 2017.
[18] X. Li, F. Wu, M. K. Khan, L. Xu, J. Shen, and M. Jo, “A secure chaotic map-based remote authentication scheme for telecare medicine information systems,” Future Generation Computer Systems, vol. 84, pp. 149–159, 2018.
[19] Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li, and A. Alelaiwi, “Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy,” Nonlinear Dynamics, vol. 83, no. 4, pp. 2085–2101, 2016.
[20] H. Zhu and X. Hao, “A provable authenticated key agreement protocol with privacy protection using smart card based on chaotic maps,” *Nonlinear Dynamics*, vol. 81, no. 1, pp. 311–321, 2013.

[21] J.-L. Tsai, N.-W. Lo, and T.-C. Wu, “Novel anonymous authentication scheme using smart cards,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2004–2013, 2012.

[22] S. A. Chaudhry, “Combating identity de-synchronization: an improved lightweight symmetric key based authentication scheme for IoV,” *Journal of Network Intelligence*, vol. 6, p. 12, 2021.

[23] S. U. Jan, I. A. Abbasi, and F. Algarni, “A key agreement scheme for IoD deployment civilian drone,” *IEEE Access*, vol. 9, Article ID 149311, 2021.

[24] M. K. Alshawaqfeh, O. S. Badarneh, and F. S. Almehmadi, “Performance Analysis of Digital Transmission in Interference-Limited Networks,” *Telecommunication Systems*, vol. 79, pp. 1–15, 2021.

[25] T. Hwang, Y. Chen, and C. J. Laih, “Non-active Password Authentications without Password Tables,” *Information & Security: An International Journal*, vol. 2, pp. 429–431, 1990.

[26] C. Guo and C.-C. Chang, “Chaotic maps-based password-authenticated key agreement using smart cards,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 6, pp. 1433–1440, 2013.

[27] H.-Y. Lin, “Improved chaotic maps-based password-authenticated key agreement using smart cards,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 20, no. 2, pp. 482–488, 2015.

[28] S. Qiu, D. Wang, G. Xu, and S. Kumari, “Practical and Provably Secure Three-Factor Authentication Protocol Based on Extended Chaotic-Maps for mobile Lightweight Devices,” *IEEE Transactions on Dependable and Secure Computing*, 2020.

[29] H. Lin, F. Wen, and C. Du, “An improved anonymous multi-server authenticated key agreement scheme using smart cards and biometrics,” *Wireless Personal Communications*, vol. 84, no. 4, pp. 2351–2362, 2015.

[30] P. Jiang, Q. Wen, W. Li, Z. Jin, and H. Zhang, “An anonymous and efficient remote biometrics user authentication scheme in a multi server environment,” *Frontiers of Computer Science*, vol. 9, no. 1, pp. 142–156, 2015.

[31] D. He and D. Wang, “Robust biometrics-based authentication scheme for multi-server environment,” *IEEE Systems Journal*, vol. 9, no. 3, pp. 816–823, 2014.

[32] V. Odelu, A. K. Das, and A. Goswami, “A secure biometrics-based multi-server authentication protocol using smart cards,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1951–1966, 2015.

[33] W.-B. Hsieh and J.-S. Leu, “An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures,” *The Journal of Supercomputing*, vol. 70, no. 1, pp. 133–148, 2014.

[34] R. Amin and G. P. Biswas, “Design and analysis of bilinear pairing based mutual authentication and key agreement protocol useable in multi-server environment,” *Wireless Personal Communications*, vol. 84, no. 1, pp. 439–462, 2015.

[35] P. Chandrakar and H. Om, “A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC,” *Computer Communications*, vol. 110, pp. 26–34, 2017.

[36] P. Chandrakar and H. Om, “Cryptography and improvement of a biometric-based remote user authentication protocol useable in a multiserver environment,” *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 12, Article ID e3200, 2017.

[37] Y. Choi, Y. Lee, and D. Won, “Security improvement on biometric based authentication scheme for wireless sensor networks using fuzzy extraction,” *International Journal of Distributed Sensor Networks*, vol. 12, no. 1, Article ID 8572410, 2016.

[38] I.-P. Chang, T.-F. Lee, T.-H. Lin, and C.-M. Liu, “Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks,” *Sensors*, vol. 15, no. 12, Article ID 29841, 2015.

[39] Y. Park and Y. Park, “Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks,” *Sensors*, vol. 16, no. 12, p. 2123, 2016.

[40] E.-J. Yoon and C. Kim, “Advanced biometric-based user authentication scheme for wireless sensor networks,” *Sensor Letters*, vol. 11, no. 9, pp. 1836–1843, 2013.

[41] A. Irshad, M. Sher, S. A. Chaudhary, H. Naqvi, and M. S. Farash, “An efficient and anonymous multi-server authenticated key agreement based on chaotic map without engaging Registration Centre,” *The Journal of Supercomputing*, vol. 72, no. 4, pp. 1623–1644, 2016.

[42] A. G. Reddy, E.-J. Yoon, A. K. Das, V. Odelu, and K.-Y. Yoo, “Design of mutually authenticated key agreement protocol resistant to impersonation attacks for multi-server environment,” *IEEE access*, vol. 5, pp. 3622–3639, 2017.

[43] S. U. Jan and F. Qayum, “A robust authentication scheme for client-server architecture with provable security analysis,” *Network and Communication Technologies*, vol. 3, no. 1, 2018.

[44] A. Hassan, N. Eltayeb, R. Elhabob, and F. Li, “An efficient certificateless user authentication and key exchange protocol for client-server environment,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1713–1727, 2018.

[45] S. U. Jan and F. Qayum, “An improved lightweight privacy preserving authentication scheme for SIP-Based-VoIP using smart card,” MPhil–Dissertation, SSRN, Rochester, NY, USA, SSRN, 2016.

[46] S. H. Islam, “Provably secure dynamic identity-based three-factor password authentication scheme using extended chaotic maps,” *Nonlinear Dynamics*, vol. 78, no. 3, pp. 2261–2276, 2014.

[47] X. Guo and J. Zhang, “Secure group key agreement protocol based on chaotic hash,” *Information Sciences*, vol. 180, no. 20, pp. 4069–4074, 2010.

[48] S. A. Chaudhry, I. L. Kim, S. Rho, M. S. Farash, and T. Shon, “An improved anonymous authentication scheme for distributed mobile cloud computing services,” *Cluster Computing*, vol. 22, no. 1, pp. 1595–1609, 2019.

[49] X. Li, J. Peng, M. S. Obaidat, F. Wu, M. K. Khan, and C. Chen, “A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems,” *IEEE Systems Journal*, vol. 14, no. 1, pp. 39–50, 2019.

[50] J. J. Hathaliya, S. Tanwar, and R. Evans, “Securing electronic healthcare records: a mobile-based biometric authentication approach,” *Journal of Information Security and Applications*, vol. 53, Article ID 102528, 2020.

[51] M. Nikoooghadam and H. Amintoosi, “Secure communication in CloudIoT through design of a lightweight authentication and session key agreement scheme,” *International Journal of Communication Systems*, p. e4332, 2020.

[52] H. A. El Zouka and M. M. Hosni, “Secure IoT communications for smart healthcare monitoring system,” *Internet of Things*, vol. 13, Article ID 100036, 2021.
Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K. R. Choo, “Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles,” IEEE Transactions on Vehicular Technology, vol. 69, no. 9, pp. 9390–9401, 2020.

B. Bera, D. Chattaraj, and A. K. Das, “Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment,” Computer Communications, vol. 153, pp. 229–249, 2020.

J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, “TCALAS: temporal credential-based anonymous lightweight authentication scheme for Internet of drones environment,” IEEE Transactions on Vehicular Technology, vol. 68, no. 7, pp. 6903–6916, 2019.

M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. Rodrigues, “Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of drones deployment,” IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3572–3584, 2018.

S. Challa, M. Wazid, A. K. Das et al., “Secure signature-based authenticated key establishment scheme for future IoT applications,” IEEE Access, vol. 5, pp. 3028–3043, 2017.

D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, “Efficient privacy-aware authentication scheme for mobile cloud computing services,” IEEE Systems Journal, vol. 12, no. 2, pp. 1621–1631, 2016.

M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu, “Provably secure biometric-based user authentication and key agreement scheme in cloud computing,” Security and Communication Networks, vol. 9, no. 17, pp. 4103–4119, 2016.

M. Alizadeh, S. Abolfazli, M. Zamani, S. Baharun, and K. Sakurai, “Authentication in mobile cloud computing: a survey,” Journal of Network and Computer Applications, vol. 61, pp. 59–80, 2016.

A. Karati, R. Amin, S. H. Islam, and K.-K. R. Choo, “Provably secure and lightweight identity-based authenticated data sharing protocol for cyber-physical cloud environment,” IEEE Transactions on Cloud Computing, vol. 9, no. 1, pp. 318–330, 2018.

R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, “A lightweight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment,” Future Generation Computer Systems, vol. 78, pp. 1005–1019, 2018.