Hindawi
Security and Communication Networks
Volume 2022, Article ID 7836461, 11 pages
https://doi.org/10.1155/2022/7836461

Research Article

A Provable Secure Cross-Verification Scheme for IoT Using Public Cloud Computing

Naveed Khan,[1] **Jianbiao Zhang**,[1] **Jehad Ali**,[2] **Muhammad Salman Pathan**,[3] **and Shehzad Ashraf Chaudhry**[4]

[1]*Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[2]*Department of Computer Science and Engineering, Sejong University, Seoul 05006, Republic of Korea*
[3]*Department of Computer Science, Maynooth University, Maynooth, Ireland*
[4]*Department of Computer Engineering, Faculty of Engineering and Architecture, Nisantasi University, Istanbul 34398, Turkey*

Correspondence should be addressed to Jehad Ali; jehadali@ajou.ac.kr

Received 6 June 2022; Revised 24 July 2022; Accepted 30 July 2022; Published 23 November 2022

Academic Editor: Mohammad Ayoub Khan

Public cloud computing has become increasingly popular due to the rapid advancements in communication and networking technology. As a result, it is widely used by businesses, corporations, and other organizations to boost the productivity. However, the result generated by millions of network-enabled IoT devices and kept on the public cloud server, as well as the latency in response and safe transmission, are important issues that IoT faces when using the public cloud computing. These concerns and obstacles can only be overcome by designing a robust mutual authentication and secure cross-verification mechanism. Therefore, we have attempted to design a cryptographic protocol based on a simple hash function, xor operations, and the exchange of random numbers. The security of the proposed protocol has formally been verified using the ROR model, ProVerif2.03, and informally using realistic discussion. In contrast, the performance metrics have been analyzed by looking into the security feature, communication, and computation costs. To sum it up, we have compared our proposed security mechanism with the state-of-the-art protocols, and we recommend it to be effectively implemented in the public cloud computing environment.

1. Introduction

Nowadays, cloud computing offers different services to the Internet enabling devices (IoT) for reducing cost and providing efficiency. These cloud servers are accessible via an Internet connection at any time and from any location. As the world moves toward globalization, IoT-enabled devices' importance and uses increase daily. IoT enables devices to be deployed and used in different applications and environments such as smart homes, smart cities, industries, Internet of Drones (IoD), space, underwater, and many more environments. IoT devices generate massive amounts of data that can be stored in the cloud servers. IoT is an emerging heterogeneous network industry, and 41 billion IoT devices will be connected to the Internet worldwide. These devices will generate 79 Zettabytes of data annually [1].

Different enterprises and individuals use three primary cloud deployment models: private, public, and hybrid. The public cloud is the most commonly used because it is cheaper than private and hybrid cloud deployment models. Cloud servers provide platform as a service, storage as a service, software as a service, and infrastructure as a service to different enterprises and users according to their needs.

The public cloud delivers services over a public network. Therefore, it raises security concerns when services are delivered over a public network channel. Thus, secure transmission plays a vital role in outsourcing data by corporations, businesses, government entities, and individuals. However, it also needs to notice the recent increase of cyberattacks on different networks and cloud servers and privacy leakage trying to stop those enterprises and individuals from using the cloud services. Therefore, to tackle

these issues and challenges for such massive use of the cloud, it is imperative to authenticate the communicating entities to protect the outsourced data from cybercriminals. However, authentication in IoT-enabled devices is not so easy because of limited resources and energy. Therefore, the authentication process should be efficient and reliable for network and energy constraints devices.

1.1. Motivation and Contribution. Recent developments in high-speed Internet, such as 5G and 6G architectures, increase the use of IoT-enabled devices. IoT enables devices to generate gigantic amounts of data annually. However, storing, analyzing, and processing vast amounts of data locally are complex. Therefore, cloud computing offers different services to consumers over the Internet to store and process data on servers with minimal cost. However, security is a big concern while transmitting data to the cloud servers over insecure channels because of cyberattacks. Thus, authenticating the communicating party is very important to transmit data securely. According to our analysis, the scheme [2] has vulnerabilities such as anonymity, untraceability, a man in the middle attacks, server impersonation attacks, and secret key disclosure attacks. Therefore, it motivates us to cryptanalysis the scheme [2] and proposes a secure and efficient scheme. Our contribution is to solve the security flaws in the scheme [2] and propose a more efficient and secure protocol. Further contributions are explained in detail below:

(i) The proposed scheme is efficient and based on symmetric-key cryptography to resist all known potential attacks.

(ii) The security analysis of the proposed scheme has been verified using. (A) ROR model. (B) ProVerif for key secrecy, confidentiality, and reachability.

(iii) The symmetric keys have been exchanged through the Diffie–Hellman method to confirm that no one can forge them.

(iv) The performance analysis of the proposed security mechanism has been made, bearing in mind. (A) computation overheads. (B) Communication overheads.

(v) Upon comparing the proposed scenario with the existing scheme, the proposed scheme is lightweight in terms of communication, computation costs, and efficiency.

1.2. System Model. Our system model consists of four entities, as shown in Figure 1. IoT devices, users, registration servers, and public cloud computing. The IoT devices generate data send to the public cloud servers over the Internet. The users and IoT devices first need to register with the registration server. Further details are given in the proposed scheme section.

1.3. Threat Model. We used the famous DY model [3] and CK model [4] as threat and adversary models in our article, where we consider the action and assume the power of $\mathcal{A}$ as follows:

(i) The $\mathcal{A}$ can intercept the exchanged messages transmitted among the participants and replay, listen, and forge messages.

(ii) The $\mathcal{A}$ can be insider or outsider dishonest participants.

(iii) The $\mathcal{A}$ can extract secret values from IoT devices and perform power analysis [5].

(iv) The $\mathcal{A}$ cannot extract secret keys from stored data in IoT devices, users, and servers.

(v) The $\mathcal{A}$ can intercept messages and try to modify, delete, insert, and intentionally temper them.

1.4. Paper Organization. The rest of this article is laid out as follows: The literature review is presented in detail in Section 2, and the proposed scenario is presented in Section 3. Then, in Sections 4 and 5, we examine the proposed framework's security, and in Section 6, we conduct a performance analysis. Finally, Section 7 brings the paper to a conclusion.

2. Literature Review

The integration of IoT enables devices in public cloud environments to make communication vulnerable to cybercriminals. Therefore, the biggest challenge is securely communicating over open network channels. Nevertheless, the researchers have proposed authentication schemes to communicate with IoT devices in the cloud server environment securely. However, these schemes have security vulnerabilities and high communication and computation costs. These high computations, communication costs, and vulnerable schemes are discussed below.

The author [6] proposed an authentication scheme for heterogeneous devices in wireless sensor networks. However, their scheme suffers from known session key and impersonation attacks and cannot provide perfect forward secrecy. Another scheme is proposed in [7] for wireless sensor networks. Nevertheless, their scheme is also vulnerable to known session keys and perfect forward secrecy. Finally, an ECC-based protocol is proposed in [8]. The protocol fulfils most security requirements except replay attacks and perfect forward secrecy. On the other hand, the protocol proposed in [9] has security vulnerabilities such as known session keys, insecure password change phase, and impersonation attacks.

Moreover, the authors [10] proposed a secure scheme in a multi-server environment based on the smartcard. However, their scheme has security flaws such as session key disclosure, spoofing, anonymity, traceability, and impersonation attacks. The author [11] proposed an authentication protocol and identified the security flaws in [12]. The protocol [13] proposed a scheme for smart home environments. However, their scheme grieves from offline password guessing attacks and insider attacks. Another scheme was also proposed for smart home environments in [14]. However, their scheme also has security vulnerabilities such as anonymity and cannot provide untraceability. Nevertheless, the protocols proposed in [15, 16] have

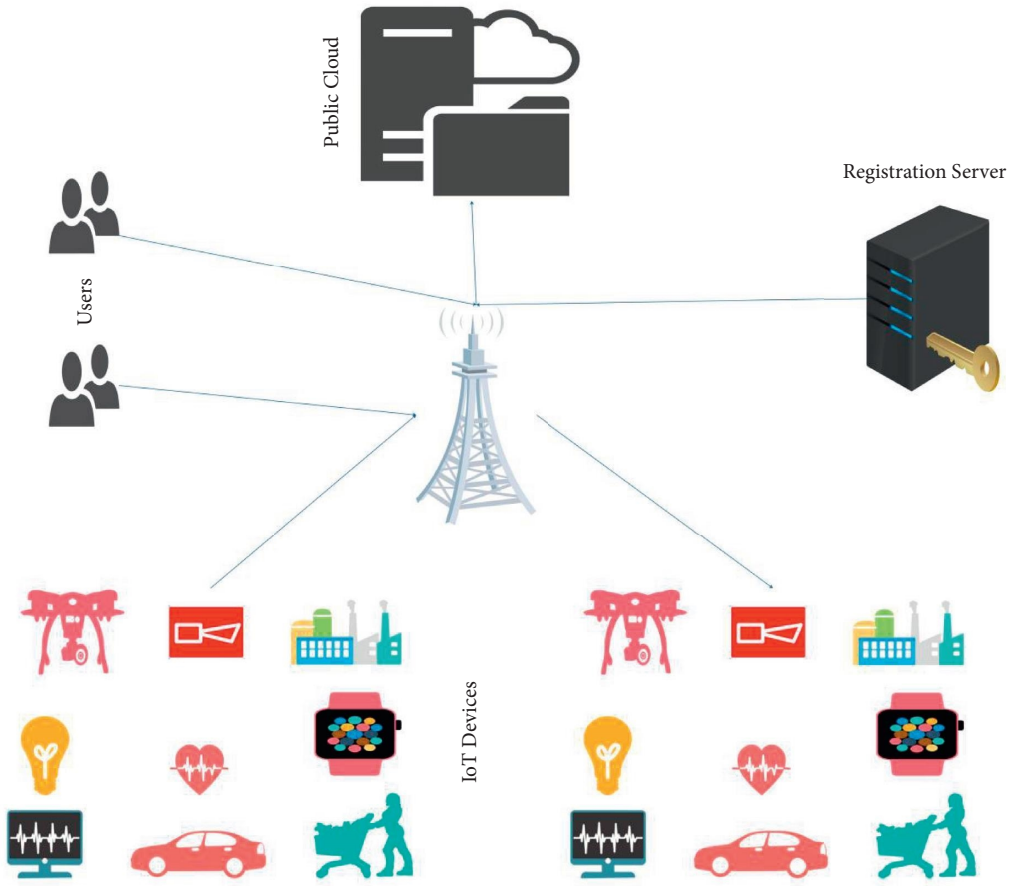

Figure 1: System model.

significant security flaws. These security flaws are mutual authentication, replay attacks, known session keys, anonymity, untraceability, and impersonation attacks.

The protocol proposed by [17] suffers from secret key guessing attacks. Therefore, the user and server can easily be compromised. The protocols [18, 19] suffer from session key attacks and secret key guessing attacks, and server and user can be compromised. At the same time, the scheme [14] is also vulnerable to impersonation attacks. The protocol [20] is based on the ECC, but the scheme has security vulnerabilities such as offline password guessing attacks, impersonation attacks, and anonymity issues. Finally, the scheme [21] has serious security vulnerabilities. The scheme [21] suffers from offline password guessing attacks, session key disclosure attacks, anonymity, perfect forward secrecy, impersonation attacks, desynchronization, and man-in-the-middle attacks. The author [22] proposed an ECC-based scheme for IoT devices in wireless sensor networks. According to the author [22], the protocol proposed in [23] is vulnerable to impersonation and password guessing attacks and unable to provide perfect forward secrecy.

Furthermore, the scheme [24] suffers from offline password guessing, impersonation, and perfect forward secrecy. The scheme proposed in [25] is based on ECC, but the scheme grieves from impersonation attacks, offline password guessing, man-in-the-middle, and session key disclosure attacks. Finally, the author proposed a scheme for a client-server environment in [26]. However, the scheme cannot resist impersonation, man-in-the-middle, password guessing, perfect forward secrecy, and insider attacks. Nevertheless, the scheme [27] suffers from offline password guessing attacks and anonymity, while the scheme [28] suffers from offline password guessing attacks. A multi-server cloud server authentication scheme based on biometrics has been proposed [29]. However, the scheme suffers from anonymity and man-in-the-middle attacks. The protocol in [30] is designed for a multi-server environment using biometrics. However, the scheme suffers from a known session temporary attack. Finally, a three-factor authentication scheme for a multi-server environment based on ECC is proposed in [31]. However, the scheme has significant security flaws such as impersonation, insider, and known session key temporary attacks and cannot provide perfect secrecy. The scheme [32] suffers from impersonation attacks and known session temporary attacks. The protocol in [33] suffers from DoS attacks and session key attacks, while the scheme [34] cannot resist offline password guessing attacks.

Moreover, the scheme [35] proposed for IoT enables devices, but the scheme suffers from insider attacks and cannot provide anonymity. Furthermore, the scheme [36] is vulnerable to impersonation and password guessing attacks.

An ECC-based authentication protocol was proposed in [37]. However, the scheme cannot resist impersonation and offline password guessing attacks. In contrast, the scheme [38] suffers from offline password guessing attacks and cannot provide anonymity. Furthermore, an ECC-based three-factor authentication for a multi-server environment is proposed by [31] and cannot resist impersonation attacks and is unable to provide perfect forward secrecy. Finally, authentication schemes [39–41] are proposed for VANETs. However, these schemes have security vulnerabilities. For example, the scheme [39] is vulnerable to replay attacks, while the schemes [40, 41] have traceability issues. Finally, the author [42] proposed an anonymous authentication scheme for mobile devices in a public cloud server. The scheme [42] achieved all the security vulnerabilities of the scheme [43], and the communication and computation costs are also less. In the end, we identify some flaws in the scheme [2], and these flaws are discussed in detail under:

(i) Anonymity and untraceability: In the protocol [2], the server identity is transmitted openly over an insecure network. Therefore, the $\mathscr{A}$ can easily intercept messages transmitted among users, the registration center, and the server. Thus, the proposed protocol cannot fulfil the property of anonymity and untraceability.

(ii) Man in the Middle Attack: As we know that the protocol did not provide anonymity and untraceability. Therefore, the $\mathscr{A}$ can pretend to be a fake server and start communicating with peers. Thus, the $\mathscr{A}$ easily launches a man in a middle attack.

(iii) Secret Key disclosure Attack: The server's identity is known to $\mathscr{A}$. Therefore, the $\mathscr{A}$ can easily impersonate the server and fool the registration server. Once the $\mathscr{A}$ can impersonate the server, it easily gets the registration server secret key. Therefore, the scheme is vulnerable to secret key disclosure attacks.

(iv) Server Impersonation Attack: As we know, the $\mathscr{A}$ can easily obtain the server's identity, which is transmitted openly on an insecure channel. Therefore, the $\mathscr{A}$ can easily impersonate the server.

3. Proposed Protocol

Our proposed scheme is based on a symmetric key authentication protocol for IoT devices in public cloud environments. Our protocol is described under:

3.1. Deployment Phase. The registration server generates secret key SK_{ps} and sends them to the public cloud server (PS). The public cloud server stores the SK_{ps}. Furthermore, the registration server assigns unique identities to IoT-enabled devices. $ID_i = \{1, 2, 3, 4, \ldots, n\}$. The registration server generates a secret key for IoT devices SK_i and stores it in each IoT device. Table 1 shows the notations and their descriptions of our proposed scheme.

Table 1: Notations and description.

Notations	Description
ID_u	User identity
ID_{ps}	Public cloud server identity
PS	Public cloud server
MSK_{ups}	Master shared key b/w user and PS
SK_{ps}	Secret key of public cloud server
SK_u	Secret key of user
$\|\|$, $h(.)$	Concatenation, hash function
Gen()	Generate
ID_i	IoT device identity
RS	Registration server
ϕ Φ	Fuzzy extractor
MSK_{ips}	Master shared key b/w IoT and PS
SK_i	The secret key to IoT device
S_K	Session key
$\oplus$, $\mathscr{A}$	XOR operator, adversary
Rep()	Reproduce

3.2. User Registration Phase. The user generates a random number r_u and selects identity and password ID_u, PW_u. The user calculates Gen (BIO) = (ϕ Φ). The user further computes $PID_u = h(ID_u||\Phi)$, $P_u = h$ ($PW_u||$ Φ). The user sends (PID_u, P_u, r_u) toward the registration server. The registration server calculates $MSK_{ups} = h(PID_u||SK_{ps}||r_u)$, and $U_1 = h(r_u||P_u) \oplus MSK_{ups}$. The registration server sends MSK_{ups} to the public cloud server while sending U_1 to the user. The PS computes: $N_u = h(ID_{ps}||SK_{ps}) \oplus MSK_{UPS}$ and $X = h(ID_u||r_u||MSK_{ups})$. The cloud server store (X, N_u) and send X to a user. After receiving (U_1, X), the user further calculates $M = h(ID_u||PW_u||\Phi)$, $U_2 = E_{MSKups}(U_1)$, $U_3 = h(PID_u||P_u) \oplus r_u$, $U_4 = h(PID_u||P_u||r_u)$. The user store (U_2, U_3, U_4, X).

3.3. IoT Device Registration Phase. The IoT device select random number r_i and calculate $PID_i = h(ID_i||r_i)$ and send (PID_i, r_i) towards registration server. After receiving the credentials from IoT device, the registration server further calculates $MSK_{ips} = h(PID_i||SK_{ps}||r_i)$. The registration server sends (PID_i, r_i) to a public cloud server. The public cloud server is stored (PID_i, r_i). The registration server sends MSK_{ips} to the IoT device. The IoT device calculate $D_1 = h(ID_i||SK_i) \oplus r_i$, $D_2 = MSK_{ips} \oplus h(SK_i||r_i)$. The IoT device stores D_1 and D_2.

3.4. Login and Authentication Phase. This phase of the protocol is shown in Table 2 and completed in the following steps:

(i) The user enters identity and password ID_u, PW_u and computes Φ = rep (BIO, ϕ), $PID_u = h(ID_u||\Phi)$ $P_u = h(PW_u||\Phi)$, $M = h(ID_u||PW_u||\phi)$, $U_1 = D_M(U_2)$, $r_u = U_3 \oplus h(PID_u||P_u)$, $MSK_{ups} = U_1 \oplus h(r_u||P_u)$ and check $U_4? = h(PID_u||P_u||r_u)$. The user selects Select T_{LA1} and r_2 and further calculate $S_1 = (ID_i||r_2) \oplus MSK_{ups} \oplus T_{LA1}$, $S_2 = PID_u \oplus h(MSK_{ups}||r_2||T_{LA1})$, $S_3 = h(PID_U||MSK_{ups}||r_2||T_{LA1})$ and forward Message$_1$\{S_1, S_2, S_3, X, T_{LA1}\} towards PS.

Table 2: Login and authentication phase.

User	Public cloud server (PS)	IOT device
ID_u, PW_u Computes: $\oiint = \text{rep}(BIO, \oint)$ $PID_u = h(ID_u \| \oiint)$ $P_u = h(PW_u \| \oiint)$ $M = h(ID_u \| PW_u \| \oiint)$ $U_1 = D_M(U_2)$ $r_u = U_3 \oplus h(PID_u \| P_u)$ $MSK_{ups} = U_1 \oplus h(r_u \| P_u)$ $U_4? = h(PID_u \| P_u \| r_u)$ Select T_{LA1} and r_2 $S_1 = (ID_u \| r_2) \oplus MSK_{ups} \oplus T_{LA1}$ $S_2 = PID_u \oplus h(MSK_{ups} \| r_2 \| T_{LA1})$ $S_3 = h(PID_U \| MSK_{ups} \| r_2 \| T_{LA1})$ $\text{Message}_1\{S_1, S_2, S_3, T_{LA1}\}$ $\longrightarrow$	Check T_{LA1}-$T \le \Delta T$ $MSK_{ups} = h(ID_{ps} \| SK_{ps}) \oplus N_u$ $(ID_i \| r_2) = S_1 \oplus MSK_{ups} \oplus T_{LA1}$ $PID_u = S_2 \oplus h(MSK_{ups} \| r_2 \| T_{LA1})$ $S_3? = h(PID_u \| MSK_{ups} \| r_2 \| T_{LA1})$ Select T_{LA2}, r_3 $MSK_{ips} = h(ID_i \| SK_{ps})$ $S_4 = (PID_u \| ID_{ps} \| r_2 \| r_3) \oplus h(ID_i \| MSK_{ips} \| T_{LA2})$ $S_5 = h(PID_u \| ID_{ps} \| MSK_{ips} \| r_2 \| r_3 \| T_{LA2})$ $\text{Message}_2\{S_4, S_5, T_{LA2}\}$ $\longrightarrow$	Check T_{LA2}-$T \le \Delta T$ $r_i = D_1 \oplus h(ID_i \| SK_i)$ $MSK_{ips} = D_2 \oplus h(SK_i \| r_i)$ $(PID_u \| ID_{ps} \| r_2 \| r_3) = S_4 \oplus h(ID_i \| MSK_{ips} \| T_{LA2})$ $S_5? = h(PID_u \| ID_{ps} \| MSK_{ips} \| r_2 \| r_3 \| T_{LA2})$ Select T_{LA3} and r_4 $S_6 = h(MSK_{ips} \| PID_i \| ID_{ps} \| T_{LA3}) \oplus r_4$ $S_K = h(r_2 \| r_3 \| r_4 \| PID_u \| ID_{ps} \| ID_i)$ $S_7 = h(ID_i \| r_4 \| MSK_{ips} \| S_K \| T_{LA3})$ $\text{Message}_3\{S_6, S_7, T_{LA3}\}$ $\text{Message}_3\{S_6, S_7, T_{LA3}\}$ $\longleftarrow$
	Check T_{LA3}-$T \le \Delta T$ $r_4 = S_6 \oplus h(MSK_{ips} \| PID_i \| ID_{ps} \| T_{LA3})$ $S_K = h(r_2 \| r_3 \| r_4 \| PID_u \| ID_{ps} \| ID_i)$ $S_7? = h(ID_i \| r_4 \| MSK_{ips} \| S_K \| T_{LA3})$ Select T_{LA4} $S_8 = (ID_{ps} \| r_3 \| r_4) \oplus h(PID_u \| MSK_{ups} \| r_2 \| T_{LA4})$ $S_9 = h(PID_u \| ID_{ps} \| r_2 \| r_3 \| S_K \| T_{LA4})$ $X^{new} = h(ID_u \| r_3 \| MSK_{ups})$ $\text{Message}_4\{S_8, S_9, T_{LA4}\}$ $\longleftarrow$	
Check T_{LA4}-$T \le \Delta T$ $(ID_{ps} \| r_3 \| r_4) = S_8 \oplus h(PID_u \| MSK_{ups} \| r_2 \| T_{LA4})$ $S_K = h(r_2 \| r_3 \| r_4 \| PID_u \| ID_{ps} \| ID_i)$ $S_9? = h(PID_u \| ID_{ps} \| r_2 \| r_3 \| S_K \| T_{LA4})$ $X = h(ID_u \| r_3 \| MSK_{ups})$		

(ii) The public cloud server check Check T_{LA1}-$T \leq \Delta T$, $MSK_{ups} = h(ID_{ps}||SK_{ps}) \oplus N_u$, $(ID_i||r_2) = S_1 \oplus MSK_{ups} \oplus T_{LA1}$, $PID_u = S_2 \oplus h(MSK_{ups}||r_2||T_{LA1})$, $S_3? = h(PID_u||MSK_{ups}||r_2||T_{LA1})$. The public cloud server selects timestamp T_{LA2}, and random number r_3. The PS further calculates $MSK_{ips} = h(ID_i||SK_{ps})$, $S_4 = (PID_u||ID_{ps}||r_2||r_3) \oplus h(ID_i||MSK_{ips}||T_{LA2})$, $S_5 = h(PID_u||ID_{ps}||MSK_{ips}||r_2||r_3||T_{LA2})$ and send $Message_2\{S_4, S_5, T_{LA2}\}$ towards IoT device through open network channel.

(iii) The IoT device Check T_{LA2}-$T \leq \Delta T$ and further calculate $r_i = D_1 \oplus h(ID_i||SK_i)$, $MSK_{ips} = D_2 \oplus h(SK_i||r_i)$, $(PID_u||ID_{ps}||r_2||r_3) = S_4 \oplus h(ID_i||MSK_{ips}||T_{LA2})$, and verify $S_5? = h(PID_u||ID_{ps}||MSK_{ips}||r_2||r_3||T_{LA2})$. The IoT selects timestamp T_{LA3} and random number r_4. Now, the IoT device further calculates, $S_6 = h(MSK_{ips}||PID_i||ID_{ps}||T_{LA3}) \oplus r_4, S_K = h(r_2||r_3||r_4||PID_u||ID_{ps}||ID_i)$, $S_7 = h(ID_i||r_4||MSK_{ips}||S_K||T_{LA3})$. The IoT device sends back $Message_3\{S_6, S_7, T_{LA3}\}$ towards PS.

(iv) The PS first check T_{LA3}-$T \leq \Delta T$ and computes $r_4 = S_6 \oplus h(MSK_{ips}||PID_i||ID_{ps}||T_{LA3})$, $S_K = h(r_2||r_3||r_4||PID_u||ID_{ps}||ID_i)$, and verify $S_7? = h(ID_i||r_4||MSK_{ips}||S_K||T_{LA3})$. Now, the PS select timestamp T_{LA4} and further calculate $S_8 = (ID_{ps}||r_3||r_4) \oplus h(PID_u||MSK_{ups}||r_2||T_{LA4})$, $S_9 = h(PID_u||ID_{ps}||r_2||r_3||S_K||T_{LA4})$, $X^{new} = h(ID_u||r_3||MSK_{ups})$ and send $Message_4 = \{S_8, S_9, T_{LA4}\}$ back to user.

(v) The User verify timestamp T_{LA4}-$T \leq \Delta T$ and computes $(ID_{ps}||r_3||r_4) = S_8 \oplus h$ $(PID_u||MSK_{ups}||r_2||T_{LA4})$, $S_K = h(r_2||r_3||r_4||PID_u||ID_{ps}||ID_i)$, and verify $S_9 ? = h(PID_u||ID_{ps}||r_2||r_3||S_K||T_{LA4})$. The user update $X = h(ID_u||r_3||MSKups)$.

3.5. Biometric and Password Change Phase

(i) Enter identity ID_u, and old password PW_u^P, and imprints old biometric BIO^P.

(ii) Computes $\iint\!\!\iint^* = rep\ (BIO^P, \oint^*)$, $PID_u{}^* = h(ID_u||\iint\!\!\iint^*)$, $P_u{}^* = h(PW_u{}^P||\iint\!\!\iint^*)$, $MSK_{ups}{}^* = h(PID_u{}^*||SK_{ps}||r_u)$, $U_1{}^* = D_{MSKups}{}^*(U_2)$, $U_3{}^* = h(PID_u{}^*||P_u{}^*) \oplus r_u$, and $U_4{}^* = h(PID_u{}^*||P_u{}^*||r_u)$ and check $U_4{}^*? = U_4$. If true, then allowed to input new password and imprint new BIO otherwise, terminate the connection.

(iii) The User inputs a new password $PW_U{}^N$ and imprints a new biometric BIO^N.

(iv) Computes $\iint\!\!\iint^N = rep\ (BIO^N, \oint^N)$,, $PID_u{}^N = h(ID_u||\iint\!\!\iint^N)$, $P_u{}^N = h(PW_u{}^N||\iint\!\!\iint^N)$, $MSK_{ups}{}^N = h(PID_u{}^N||SK_{ps}||r_u)$, $U_2{}^N = E_{MSKups}{}^N(U_1)$, $U_3{}^N = h(PID_u{}^N||P_u{}^N) \oplus r_u$, and $U_4{}^N = h(PID_u{}^N||P_u{}^N||r_u)$ and update $(U_2{}^N, U_3{}^N, U_4{}^N)$.

4. Formal Security Analysis

In the section of our research article, we will investigate, analyze, discuss, and explain our proposed scheme against all potential attacks using ProVerif, the ROR model, and informal security discussions.

4.1. ProVerif Code. ProVerif is a simulation toolkit that is used to simulate cryptographic algorithms. ProVerif checks the key secrecy, reachability, and confidentiality [44]. Figure 2 shows our proposed scheme simulation code result, and according to the ProVerif simulation result, our proposed scheme is secure.

4.2. ROR Model. In this section, we evaluate our proposed scheme S_K by using the ROR model [45]. Three participants are involved in our scheme such as user P_u^{T1}, public cloud server P_{ps}^{T2}, and IoT-enabled device P_i^{T3}. We demonstrate each query used in ROR model such as Execute, CorruptSC, Reveal, Send, and Test.

Theorem 1. *The Adv_A has the advantage of violating S_K of our scheme, the inequality $ADV_A \leq (q_h^2/|HASH|) + 2\{C.(q^s/2^{lf})\}$. q^s denoted the hash queries, and C, l, and f are Zipf values [46].*

Proof. Four Games in a sequence $Game_g$: $\{g = 0, 1, 2, 3\}$ are played by $\mathscr{A}$. The Adv_A has the probability of winning all the Games. These Games are discussed below:

$Game_{g0}$: In this $Game_{g0}$, the $\mathscr{A}$ executes a real attack and tries to guess a bit in order to win the $Game_{g0}$.

$$ADV_A = \left|2.ADV_{A,Gameg0} - 1\right|. \quad (1)$$

$Game_{g1}$: The $\mathscr{A}$ trying to eavesdrop attack on a proposed scheme where all messages transmitted are intercepted by using Execute. The $\mathscr{A}$ perform Test and Reveal to check that the message has S_K or random numbers. The $\mathscr{A}$ need secret values such as SK_u, SK_{ps}, SK_i, PID_u, PID_i, and random numbers to construct $S_K = h(r_2||r_3||r_4||PID_u||ID_{ps}||ID_i)$. Therefore, based on this, we obtained

$$ADV_{A,Gameg1} = ADV_{A,Gameg0}. \quad (2)$$

$Game_{g2}$: In this game, $Game_{g2}$, the $\mathscr{A}$ trying actively/passively attack our scheme. The $\mathscr{A}$ using the Send query and Hash query. The $\mathscr{A}$ intercepted all exchanged messages such as $Message_1\{S_1, S_2, S_3, T_{LA1}\}$, $Message_2\{S_4, S_5, T_{LA2}\}$, $Message_3\{S_6, S_7, T_{LA3}\}$, and $Message_4 = \{S_8, S_9, T_{LA4}\}$. Furthermore, these messages are protected using secret keys, random numbers, and hashing $h(.)$. Therefore, we obtain

$$\left|ADV_{A,Gameg2} - ADV_{A,Gameg1}\right| \leq \left\{C.q_{send}\frac{q^s}{2^{lf}}\right\}. \quad (3)$$

$Game_{g3}$: The $\mathscr{A}$ trying to get $\{U_2, U_2, U_3\}$ from IoT device memory using CorruptSC through power analysis attack. The $\mathscr{A}$ trying to get password PWu using offline password guessing attack. However, in our scheme, the $\mathscr{A}$ cannot get a password using Send query. Therefore, we get

$$\left|ADV_{A,Gameg2} - ADV_{A,Gameg1}\right| \leq \frac{q_h^2}{|HASH|}. \quad (4)$$

```
(*==========RESULT==========*)
Completing equations...
Completing equations...
-- Process 1-- Query not attacker (SK[]) in process 1
Translating the process into Horn clauses...
Completing...
Starting query not attacker(SK[])
RESULT not attacker(SK[]) is true.
-- Query inj-event(end_U(IDu)) ==> inj-event(start_U(IDu)) in process 1
Translating the process into Horn clauses...
Completing...
RESULT inj-event(end_U(IDu)) ==> inj-event(start_U(IDu)) is true.
-- Query inj-event(end_PS(IDps)) ==> inj-event(start_PS(IDps)) in process 1
Translating the process into Horn clauses...
Completing...
RESULT inj-event(end_PS(IDps)) ==> inj-event(start_PS(IDps)) is true.
-- Query inj-event(end_D(IDi)) ==> inj-event(start_D(IDi)) in process 1
Translating the process into Horn clauses...
Completing...
RESULT inj-event(end_D(IDi)) ==> inj-event(start_D(IDi)) is true.

------------------------------------------------------------
Verification summary:

Query not attacker(SK[]) is true.

Query inj-event(end_U(IDu)) ==> inj-event(start_U(IDu)) is true.

Query inj-event(end_PS(IDps)) ==> inj-event(start_PS(IDps)) is true.

Query inj-event(end_D(IDi)) ==> inj-event(start_D(IDi)) is true.

------------------------------------------------------------
```

Figure 2: ProVerif simulation result.

After playing Game_{g0}, Game_{g1}, Game_{g2}, and Game_{g3}. The $\mathcal{A}$ tries to guess the bit to win the game using the Test query. Hence, we get

$$\text{ADV}_{A,\text{Game}g3} = \frac{1}{2}. \tag{5}$$

By applying (1), (2) and (5), we obtained

$$\begin{aligned}\frac{1}{2}\text{ADV}_A &= \left|\text{ADV}_{A,\text{Game}g0} - \frac{1}{2}\right| \\ &= \left|\text{ADV}_{A,\text{Game}g1} - \frac{1}{2}\right| \\ &= \left|\text{ADV}_{A,\text{Game}g2} - \frac{1}{2}\right| \\ &= \left|\text{ADV}_{A,\text{Game}g1} - \text{ADV}_{A,\text{Game}g3}\right|.\end{aligned} \tag{6}$$

Now by using (4), (5), and (6), we get

$$\begin{aligned}\frac{1}{2}\text{ADV}_A &= \left|\text{ADV}_{A,\text{Game}g1} - \text{ADV}_{A,\text{Game}g3}\right| \\ &\le \left|\text{ADV}_{A,\text{Game}g1} - \text{ADV}_{A,\text{Game}g2}\right| \\ &\quad + \left|\text{ADV}_{A,\text{Game}g2} - \text{ADV}_{A,\text{Game}g3}\right| \\ &\le \frac{q_h^2}{|\text{HASH}|} + \left\{C.q_{\text{send}}^2 \frac{q^s}{2^{lf}}\right\}.\end{aligned} \tag{7}$$

Equation (7) is multiplied by 2 on both sides, and we get

$$\text{ADV}_A \le \frac{q_h^2}{|\text{HASH}|} + 2\left\{C.q_{\text{send}}^2, \frac{q^s}{2^{lf}}\right\}. \tag{8}$$

Hence, the theorem is proved. □

4.3. Shared Session Key Correctness. In this section, we will prove that the shared session key for communicating participants is the same. During in login and authentication phase the shared session key is calculated by IoT device is $S_K = h(r_2||r_3||r_4||\text{PID}_u||\text{ID}_{ps}||\text{ID}_i)$ and the receiver end the user calculated the shared session key $S_K = h(r_2||r_3||r_4||\text{PID}_u||\text{ID}_{ps}||\text{ID}_i)$. In the initiator IoT device received $S_4 = (\text{PID}_u||\text{ID}_{ps}||r_2||r_3) \oplus h(\text{ID}_i||\text{MSK}_{ips}||T_{\text{LA2}})$ and $S_5 = h(\text{PID}_u||\text{ID}_{ps}||\text{MSK}_{ips}||r_2||r_3||T_{\text{LA2}})$. It successfully computed $(\text{PID}_u||\text{ID}_{ps}||r_2||r_3) = \text{S}_4 \oplus h\ (\text{ID}_i||\text{MSK}_{ips}||T_{\text{LA2}})$ and verify $S_5 = h(\text{PID}_u||\text{ID}_{ps}||\text{MSK}_{ips}||r_2||r_3||T_{\text{LA2}})$. Furthermore, the IoT device computed $S_6 = h(\text{MSK}_{ips}||\text{PID}_i||\text{ID}_{ps}||T_{\text{LA3}}) \oplus r_4$ and $S_7 = h(\text{ID}_i||r_4||\text{MSK}_{ips}||S_K||T_{\text{LA3}})$ and forward it to the public cloud server. Similarly, likewise IoT device, the public cloud server successfully generated $r_4 = S_6 \oplus h(\text{MSK}_{ips}||\text{PID}_i||\text{ID}_{ps}||T_{\text{LA3}})$ and verify $S_7 = h(\text{ID}_i||r_4||\text{MSK}_{ips}||S_K||T_{\text{LA3}})$ and further calculated $S_8 = (\text{ID}_{ps}||r_3||r_4) \oplus h(\text{PID}_u||\text{MSK}_{ups}||r_2||T_{\text{LA4}})$ and $S_9 = h(\text{PID}_u||\text{ID}_{ps}||r_2||r_3||S_K||T_{\text{LA4}})$. The public cloud server forward S_8 and S_9. The user successfully computes $(\text{ID}_{ps}||r_3||r_4) = S_8 \oplus h(\text{PID}_\text{u}||\text{MSK}_{ups}||r_2||T_{\text{LA4}})$ and verify S_9. Therefore, the communicating participants successfully get the required credentials to construct the shared session key.

5. Informal Security Analysis

Informal security discussion and explanation of our proposed architecture are under:

(1) Impersonation Attack: The $\mathscr{A}$ trying to impersonate user, public cloud server, and IoT device. It will need to calculate the authentication request messages such as message_1 and message_4. However, it is challenging for $\mathscr{A}$ to generate secret key SK_{ps}, random numbers, and PID_u. Therefore, our proposed scheme resists impersonation attacks because the $\mathscr{A}$ is unable to compute the values mentioned above.

(2) IoT Device Capture Attack: Let us suppose the IoT device is physically captured by $\mathscr{A}$ and $\mathscr{A}$ trying to extract secret values such as $\{D_1, D_2\}$. However, the $\mathscr{A}$ cannot compute MSK_{ips} without knowing the secret key of public cloud SK_{ps}, random number r_1, and pseudo-identity PID_i. Therefore, the proposed scheme resists IoT device capture attacks.

(3) Man-in-the-Middle Attack: Suppose the $\mathscr{A}$ eavesdrop on all transmitted messages among IoT devices, users, and public cloud servers, then it is possible to launch a MITM attack. However, the $\mathscr{A}$ cannot construct the transmitted messages because these messages are protected with secret keys $\{\text{SK}_i, \text{SK}_u, \text{SK}_{ps}\}$, identities $\{\text{ID}_i, \text{ID}_u, \text{ID}_{ps}\}$, and random numbers $\{r_1, r_2, r_3, r_4\}$. Thus, our proposed scheme is secure against MITM attacks.

(4) Session Key Disclosure Attack: Let suppose the $\mathscr{A}$ obtain $\{U2, U3, U4\}$ that are stored on the user side. However, the $\mathscr{A}$ should get the random numbers $\{r_1, r_2, r_3, r_4\}$ to construct session key S_k. Moreover, the $\mathscr{A}$ also needs to know the pseudo-identity of user PID_u, cloud server identity ID_{ps}, and IoT identity ID_i. Hence, our scheme resists session key disclosure attacks.

(5) Offline Password Guessing Attack: Suppose the $\mathscr{A}$ access to $\{U_2, U_3, U_4\}$ is stored on the user side. These values are constructed in a way that the A cannot get a password from it, such as $U_2 = \text{E}_{\text{MK}}(U_1)$, $U_3 = h(\text{PID}_u||P_u) \oplus r_u$, and $U_4 = h(\text{PID}_u||P_u||r_u)$. The $\mathscr{A}$ needs random number r_1 and $\oplus$ to construct those values. Therefore, our scheme is secure against offline password guessing attacks.

(6) Anonymity and untraceability: Suppose the $\mathscr{A}$ access to all transmitted messages during the login and authentication phase. However, the $\mathscr{A}$ cannot get the identities $\{\text{ID}_u, \text{ID}_{ps}, \text{ID}_i\}$, pseudo identities $\{\text{PID}_i, \text{PID}_u\}$ without knowing the secret keys. Furthermore, the random numbers and timestamps are different in each session. Therefore, the $\mathscr{A}$ cannot trace any peers. Hence, the proposed scheme provides anonymity and untraceability.

(7) Mutual Authentication: In our proposed architecture, all parties mutual authenticate each other; after receiving, $\text{Message}_1\{S_1, S_2, S_3, T_{\text{LA1}}\}$ from a user, the public cloud server authenticates the user using $S_3? = h(\text{PID}_i||\text{MSK}_{ups}||r_2||T_{\text{LA1}})$ while the IoT device authenticate PS using $S_5? = h(\text{PID}_u||\text{ID}_{ps}||\text{MSK}_{ips}||r_2||r_3||T_{\text{LA2}})$. Furthermore, the PS authenticate IoT devices using $S_7? = h(\text{ID}_i||r_4||\text{MSK}_{ips}||S_k||T_{\text{LA3}})$ and the user authenticate PS using $S_9? = h(\text{PID}_u||\text{ID}_{ps}||r_2||r_3||S_K||T_{\text{LA4}})$. Hence, our proposed architecture provides mutual authentication.

(8) Replay Attack: If the $\mathscr{A}$ intercept previous session transmitted messages such as $\text{Message}_1\{S_1, S_2, S_3, T_{\text{LA1}}\}$, $\text{Message}_2\{S_4, S_5, T_{\text{LA2}}\}$, $\text{Message}_3\{S_6, S_7, T_{\text{LA3}}\}$, and $\text{Message}_4 = \{S_8, S_9, T_{\text{LA4}}\}$. After the interception, the $\mathscr{A}$ trying to resend those messages again, then our proposed scheme checks the validation of timestamps. Furthermore, all transmitted messages are protected using secret keys and random numbers. Hence, our scheme is resilient to replay attacks.

(9) Perfect Forward Secrecy: In our proposed scheme, the $\mathscr{A}$ cannot construct the session key if it is compromised previous session key S_K. Because the $\mathscr{A}$ will need MSK_{ups}, r_i, PID_i, PID_u, and MSK_{ips} to construct the session key. Therefore, the proposed scheme provides perfect forward secrecy.

6. Performance Analysis

We evaluate the proposed scenario regarding security features, communication, and computation costs. We consider the existing protocols and compare them with our scheme. Our scheme provides foolproof security and lower computation and communication costs.

6.1. Security Features. This section evaluates our proposed scheme in terms of security features. We compared our proposed protocol with other recent related existing schemes. Table 3 compares our scheme with the existing schemes and shows that our scheme performs better than other schemes in terms of security features.

6.2. Communication Cost. We calculate our proposed scheme communication cost in this section. We choose SH-1, where identities are equal to 160 bits, random numbers are 160, and timestamp 32 bits. For encryption and decryption, we select AES-128, which takes 128 bits as an input and output. The hash function is 160 bits. Our scheme authentication is completed in four rounds. The message transmitted from the user to the public cloud server is $\text{message}_1 = \{512\}$. From public cloud server to IoT device $\text{message}_2 = \{352\}$ while from IoT device to public cloud server is $\text{message}_3 = \{352\}$ and from public cloud server to user is $\text{message}_4 = \{352\}$. The total communication cost is 1568 bits, as shown in Figure 3.

6.3. Computation Cost. We compute the computation cost of our proposed scheme in this section. We adopted the work done by [54]. T_m represents multiplication time, T_h is a one-way hash function, and T_E and T_D are encryption and decryption. The operation execution time in ms is

Table 3: Security features.

Features↓ ⟶Schemes	[2]	[47]	[48]	[49]	[50]	[51]	[52]	[53]	[54]	[35]	[55]	Our
Impersonation attack	∞	∝	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Offline password guessing attack	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
Man-in-the-middle attack	∞	∞	✓	✓	✓	✓	✓	✓	✓	✓	-	✓
Session key disclosure attack	∞	∞	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Anonymity and untraceability	∞	∞	✓	✓	✓	✓	✓	✓	✓	∝	✓	✓
Mutual authentication	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Replay attack	✓	✓	✓	✓	✓	✓	✓	✓		✓	-	✓
Perfect forward secrecy	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
Stolen device attack	—	—	—	—	—	∝	∝	∝	—	—	—	✓

✓: secure, ∞: insecure, —: not considered.

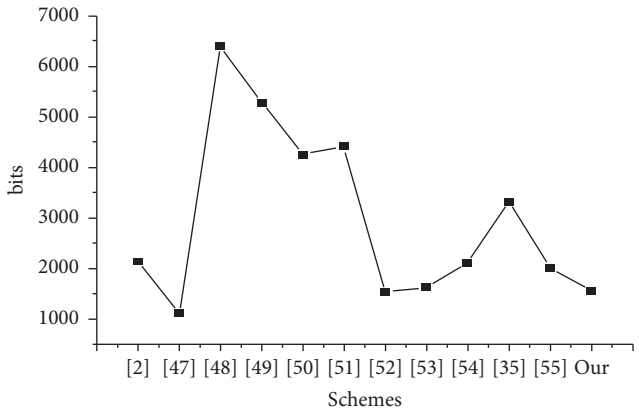

Figure 3: Communication cost.

Table 4: Operation and execution time.

Operation	Execution time in ms
T_h	0.00097
T_A	0.0028
T_E	0.109
T_D	0.0036
T_M	0.0035

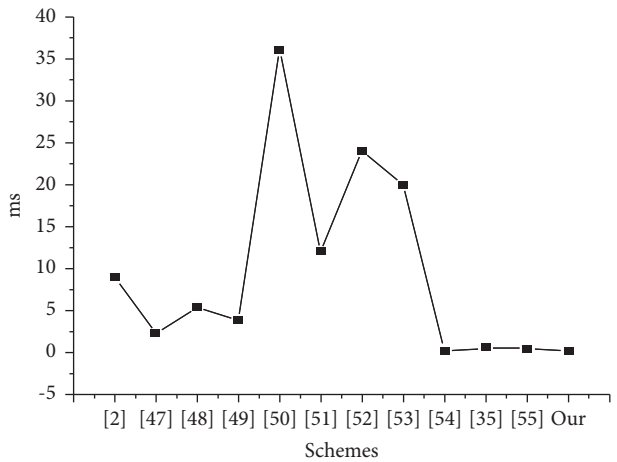

Figure 4: Computation cost.

mentioned in Table 4. Furthermore, our scheme computation cost equals = 0.266 ms, as shown in Figure 4.

7. Conclusions

As we know that the misconfiguration, unauthorized accessing of applications, and the response of cloud servers to the results generated by IoT of end-user in the cloud computing paradigm is yet to be addressed by the researchers. In this regard, we have attempted to design a security mechanism for mitigating the aforesaid issues to a maximum extent. The security analysis section of the proposed framework has been made using worldwide used techniques ROR model, ProVerif2.03, and realistic discussion. Furthermore, the performance analysis has been evaluated by considering three metrics, i.e., security features, communication, and computation costs. The comparison results show that the proposed scenario is suitable for practical implantation in the IoT using a public cloud server. In the future, we have planned to design a transitional authentication for end-users when using IoT. At the same time, its security will be conducted using AVISPA.

Data Availability

The data used to support the findings of this study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported partially by the BK21 FOUR program of the National Research Foundation of Korea, funded by the Ministry of Education (NRF5199991514504) and by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2022-2018-0-01431) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation).

References

[1] M. Saqib, B. Jasra, and A. H. Moon, "A lightweight three factor authentication framework for IoT based critical applications," *Journal of King Saud University-Computer and Information Sciences*, 2021.

[2] Z. Ali, S. Hussain, R. H. U. Rehman et al., "ITSSAKA-MS: An improved three-factor symmetric-key based secure AKA scheme for multi-server environments," *IEEE Access*, vol. 8, pp. 107993–108003, 2020.

[3] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[4] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Lecture Notes in Computer Science, Vol 2045*, pp. 453–474, Springer, Berlin, Heidelberg, 2001.

[5] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology-CRYPTO' 99*, pp. 388–397, Springer, Berlin, Heidelberg, 1999.

[6] R. Amin, S. H. Islam, N. Kumar, and K.-K. R. Choo, "An untraceable and anonymous password authentication protocol for heterogeneous wireless sensor networks," *Journal of Network and Computer Applications*, vol. 104, pp. 133–144, 2018.

[7] M. Alotaibi, "An enhanced symmetric cryptosystem and biometric-based anonymous user authentication and session key establishment scheme for WSN," *IEEE Access*, vol. 6, pp. 70072–70087, 2018.

[8] P. Chandrakar and H. Om, "An extended ECC-based anonymity-preserving 3-factor remote authentication scheme useable in TMIS," *International Journal of Communication Systems*, vol. 31, no. 8, p. e3540, 2018.

[9] A. H. Moon, U. Iqbal, and G. M. Bhat, "Mutual entity authentication protocol based on ECDSA for WSN," *Procedia Computer Science*, vol. 89, pp. 187–192, 2016.

[10] W.-i. Bae and J. Kwak, "Smart card-based secure authentication protocol in multi-server IoT environment," *Multimedia Tools and Applications*, vol. 79, no. 23, pp. 15793–15811, 2020.

[11] S. Shin and T. Kwon, "A lightweight three-factor authentication and key agreement scheme in wireless sensor networks for smart homes," *Sensors*, vol. 19, no. 9, p. 2012, 2019.

[12] J. Jung, J. Moon, D. Lee, and D. Won, "Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks," *Sensors*, vol. 17, no. 3, p. 644, 2017.

[13] M. Fakroon, M. Alshahrani, F. Gebali, and I. Traore, "Secure remote anonymous user authentication scheme for smart home environment," *Internet of Things*, vol. 9, p. 100158, 2020.

[14] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, and Y. Park, "An efficient, anonymous and robust authentication scheme for smart home environments," *Sensors*, vol. 20, no. 4, p. 1215, 2020.

[15] 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.

[16] 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.

[17] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Systems Journal*, vol. 14, no. 1, pp. 560–571, 2019.

[18] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Information Systems*, vol. 15, no. 9, pp. 1200–1215, 2021.

[19] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Networks*, vol. 25, no. 8, pp. 4737–4750, 2019.

[20] B. Ying and A. Nayak, "Lightweight remote user authentication protocol for multi-server 5G networks using self-certified public key cryptography," *Journal of Network and Computer Applications*, vol. 131, pp. 66–74, 2019.

[21] M. Nikooghadam, R. Jahantigh, and H. Arshad, "A lightweight authentication and key agreement protocol preserving user anonymity," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 13401–13423, 2017.

[22] B. Hu, W. Tang, and Q. Xie, "A two-factor security Authentication scheme for wireless sensor networks in IoT environments," *Neurocomputing*, 2022.

[23] C.-T. Chen, C.-C. Lee, and I.-C. Lin, "Efficient and secure three-party mutual authentication key agreement protocol for WSNs in IoT environments," *PLoS One*, vol. 15, no. 4, p. e0232277, 2020.

[24] R. Amin, T. Maitra, D. Giri, and P. Srivastava, "Cryptanalysis and improvement of an RSA based remote user authentication scheme using smart card," *Wireless Personal Communications*, vol. 96, no. 3, pp. 4629–4659, 2017.

[25] M. Luo, Y. Zhang, M. K. Khan, and D. He, "A secure and efficient identity-based mutual authentication scheme with smart card using elliptic curve cryptography," *International Journal of Communication Systems*, vol. 30, no. 16, p. e3333, 2017.

[26] T. Maitra, M. S. Obaidat, R. Amin, S. H. Islam, S. A. Chaudhry, and D. Giri, "A robust ElGamal-based password-authentication protocol using smart card for client-server communication," *International Journal of Communication Systems*, vol. 30, no. 11, p. e3242, 2017.

[27] S. H. Islam, "Design and analysis of an improved smartcard-based remote user password authentication scheme," *International Journal of Communication Systems*, vol. 29, no. 11, pp. 1708–1719, 2016.

[28] T. Maitra, M. S. Obaidat, S. H. Islam, D. Giri, and R. Amin, "Security analysis and design of an efficient ECC-based two-factor password authentication scheme," *Security and Communication Networks*, vol. 9, no. 17, pp. 4166–4181, 2016.

[29] 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.

[30] Q. Feng, D. He, S. Zeadally, and H. Wang, "Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment," *Future Generation Computer Systems*, vol. 84, pp. 239–251, 2018.

[31] R. Ali and A. K. Pal, "An efficient three factor-based authentication scheme in multiserver environment using ECC," *International Journal of Communication Systems*, vol. 31, no. 4, p. e3484, 2018.

[32] F. Wang, G. Xu, C. Wang, and J. Peng, "A provably secure biometrics-based authentication scheme for multi-server environment," *Security and Communication Networks*, vol. 2019, 2019.

[33] J. Wang, H. Liu, H. Shao, and H.-y. Xia, "Novel two-way security authentication wireless scheme based on hash

function," *Computer Science*, vol. 43, no. 11, pp. 205–209, 2016.

[34] S. D. Kaul and A. K. Awasthi, "Security enhancement of an improved remote user authentication scheme with key agreement," *Wireless Personal Communications*, vol. 89, no. 2, pp. 621–637, 2016.

[35] S. S. Sahoo, S. Mohanty, and B. Majhi, "A secure three factor based authentication scheme for health care systems using IoT enabled devices," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1419–1434, 2021.

[36] M. Qi and J. Chen, "New robust biometrics-based mutual authentication scheme with key agreement using elliptic curve cryptography," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 23335–23351, 2018.

[37] A. Ostad-Sharif, D. Abbasinezhad-Mood, and M. Nikooghadam, "A robust and efficient ECC-based mutual authentication and session key generation scheme for healthcare applications," *Journal of Medical Systems*, vol. 43, no. 1, pp. 1–22, 2019.

[38] A. K. Sutrala, A. K. Das, V. Odelu, M. Wazid, and S. Kumari, "Secure anonymity-preserving password-based user authentication and session key agreement scheme for telecare medicine information systems," *Computer Methods and Programs in Biomedicine*, vol. 135, pp. 167–185, 2016.

[39] I. Z. Ahmed, T. M. Mohamed, and R. A. Sadek, "A low computation message delivery and authentication VANET protocol," in *Proceedings of the 2017 12th International Conference on Computer Engineering and Systems (ICCES)*, pp. 204–211, IEEE, Cairo, Egypt, December 2017.

[40] H. Tan, Z. Gui, and I. Chung, "A secure and efficient certificateless authentication scheme with unsupervised anomaly detection in VANETs," *IEEE Access*, vol. 6, pp. 74260–74276, 2018.

[41] R. Ma, J. Cao, D. Feng et al., *A secure Authentication scheme for Remote Diagnosis and Maintenance in Internet of Vehicles*, pp. 1–7.

[42] N. Khan, J. Zhang, and S. U. Jan, "A robust and privacy-preserving Anonymous user Authentication scheme for public cloud server," *Security and Communication Networks*, vol. 2022, 2022.

[43] 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.

[44] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, *ProVerif 2.00: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*, pp. 05–16, 2018.

[45] S. A. Chaudhry, K. Yahya, F. Al-Turjman, and M.-H. Yang, "A secure and reliable device access control scheme for IoT based sensor cloud systems," *IEEE Access*, vol. 8, pp. 139244–139254, 2020.

[46] Z. Hou and D. Wang, "New Observations on Zipf's Law in passwords," *IEEE Transactions on Information Forensics and Security*, 2022.

[47] S. Barman, H. P. Shum, S. Chattopadhyay, and D. Samanta, "A secure authentication protocol for multi-server-based e-healthcare using a fuzzy commitment scheme," *IEEE Access*, vol. 7, pp. 12557–12574, 2019.

[48] X. Li, T. Liu, M. S. Obaidat, F. Wu, P. Vijayakumar, and N. Kumar, "A lightweight privacy-preserving authentication protocol for VANETs," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3547–3557, 2020.

[49] R. I. Abdelfatah, N. M. Abdal-Ghafour, and M. E. Nasr, "Secure VANET Authentication protocol (SVAP) using Chebyshev Chaotic Maps for Emergency Conditions," *IEEE Access*, vol. 10, pp. 1096–1115, 2021.

[50] K. Mahmood, S. Shamshad, M. Rana et al., "PUF enable lightweight key-exchange and mutual authentication protocol for multi-server based D2D communication," *Journal of Information Security and Applications*, vol. 61, p. 102900, 2021.

[51] M. Kaveh, D. Martín, and M. R. Mosavi, "A lightweight authentication scheme for V2G communications: A PUF-based approach ensuring cyber/physical security and identity/location privacy," *Electronics*, vol. 9, no. 9, p. 1479, 2020.

[52] P. Gope and B. Sikdar, "An efficient privacy-preserving authenticated key agreement scheme for edge-assisted internet of drones," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13621–13630, 2020.

[53] P. Gope, Y. Gheraibia, S. Kabir, and B. Sikdar, "A secure IoT-based modern healthcare system with fault-tolerant decision making process," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 3, pp. 862–873, 2020.

[54] S. Shamshad, M. F. Ayub, K. Mahmood, S. Kumari, S. A. Chaudhry, and C.-M. Chen, "An enhanced scheme for mutual authentication for healthcare services," *Digital Communications and Networks*, vol. 8, no. 2, pp. 150–161, 2022.

[55] S. S. Sahoo, S. Mohanty, and B. Majhi, "Improved biometric-based mutual authentication and key agreement scheme using ECC," *Wireless Personal Communications*, vol. 111, no. 2, pp. 991–1017, 2020.