Hindawi
Security and Communication Networks
Volume 2021, Article ID 6658041, 16 pages
https://doi.org/10.1155/2021/6658041

Research Article

Improved Authenticated Key Agreement Scheme for Fog-Driven IoT Healthcare System

Tsu-Yang Wu,[1,2,3] **Tao Wang**,[2,3] **Yu-Qi Lee**,[2,3] **Weimin Zheng**,[1,2] **Saru Kumari**,[4] **and Sachin Kumar**[5]

[1] *College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2] *School of Information Science and Engineering, Fujian University of Technology, Fuzhou 350118, China*
[3] *Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, China*
[4] *Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh 250004, India*
[5] *Department of Computer Science and Engineering, Ajay Kumar Garg Engineering College, Ghaziabad 201009, India*

Correspondence should be addressed to Weimin Zheng; zhengwm901@126.com

Received 30 October 2020; Revised 4 January 2021; Accepted 15 January 2021; Published 31 January 2021

Academic Editor: Shehzad Chaudhry

The Internet of things (IoT) has been widely used for various applications including medical and transportation systems, among others. Smart medical systems have become the most effective and practical solutions to provide users with low-cost, noninvasive, and long-term continuous health monitoring. Recently, Jia et al. proposed an authentication and key agreement scheme for smart medical systems based on fog computing and indicated that it is safe and can withstand a variety of known attacks. Nevertheless, we found that it consists of several flaws, including known session-specific temporary information attacks and lack of per-verification. The opponent can readily recover the session key and user identity. In this paper, we propose a secure authentication and key agreement scheme, which compensates for the imperfections of the previously proposed. For a security evaluation of the proposed authentication scheme, informal security analysis and the Burrows–Abadi–Needham (BAN) logic analysis are implemented. In addition, the ProVerif tool is used to normalize the security verification of the scheme. Finally, the performance comparisons with the former schemes show that the proposed scheme is more applicable and secure.

1. Introduction

A wireless sensor network (WSN) [1–5] (also called sensor network) is a multihop self-organizing network system formed by several inexpensive minisensor nodes distributed in the detection region by wireless communication. The aim of WSN is to gather and process the information of the sensing objects in the network coverage area and transmit it to the observer. The WSN is a significant foundation of the Internet of things and has been used in several fields, such as smart healthcare. Wireless medical sensor networks (WMSNs) [6] can be used to build universal medical systems, which can immediately verify patient emergency situations through the remote monitoring function and can increase the quality of patient medical treatment. In a WSN-based healthcare system, medical sensors are physically applied on patients, and then the acquired data are forwarded to authorized entities in a secure manner. However, the sensors deployed in the wireless medical sensor network have limited storage and computing capabilities; therefore, when excessive data are collected, the real-time nature of all the data processing may not be guaranteed.

To resolve the aforementioned critical problems, the concept of a fog-driven IoT healthcare system [7–9] (Figure 1) is proposed to move computing functions to users and devices at more remote locations. The fog-driven IoT healthcare system consists of the three following layers: healthcare device layer, medical fog layer, and medical cloud layer. In fog computing [10–16], fog nodes (including routers, gateways, switchers, and access points) are distributed at the margin of the network and approach terminal facilities in a geographic location. By expanding cloud services to the margin of the network, fog computing transforms cloud data centers into distributed platforms while preserving cloud services for users. Therefore, the waiting time for wireless medical sensor data processing is minimized [17–19], improving user experience and service quality.

Generally, sensor nodes are resource-constrained devices with computing, communication, and storage functions. In addition, sensor nodes are usually distributed in a sparsely populated environment. Because the nodes are vulnerable to threats from adversaries, the security of the deployed equipment cannot be guaranteed. Hence, the security of wireless sensor networks has become a significant challenge for researchers, particularly in WMSN because medical data, security, and privacy issues are more serious considering key patient private information. A few challenges need to be overcome to exploit the entire mechanism and run it efficiently. Maintaining the integrity of the medical data gathered from sensor nodes, providing only legitimate users with secure access to these data, and preventing misuse of data transmitted through public channels are the main challenges that need to be addressed and must be handled carefully. The integrity and confidentiality of data transmitted between the parties must be guaranteed [20].

To establish trust between communication parties and prevent counterfeiting, it is necessary to provide a unique identification [21] and authentication [22] to each user or fog node in the system. In addition, data transmitted through public channels and stored in fog nodes or cloud servers need to be encrypted to ensure data security and privacy [23–25]. However, owing to the mobility of deployed fog nodes and terminal devices, it is not practical to share session keys between them in advance. The authenticated key agreement (AKA) [26–29] is a sufficient scheme for user or node authentication and generating public session keys; however, it is rarely used for fog computing.

Recently, numerous AKA protocols [28–41] have been proposed in WSN, fog computing, and IoT environments. Turkanovic et al. [31] proposed an effective AKA scheme for heterogeneous WSNs, in which the user authenticates through the sensor node without communicating with the gateway node. However, Farash et al. [33] found that their protocol is vulnerable to theft attacks of smart cards and does not provide the untraceability and anonymity of sensor nodes to the user. Wang and Wang [32] indicated that the realization of anonymous authentication cannot be accomplished only through a symmetric cryptographic system. Therefore, it has always focused on designing AKA schemes based on asymmetry. Hayajneh et al. [34] proposed a lightweight authentication scheme based on the Rabin signature, which is used for the remote monitoring of patients by wireless sensor networks. In 2018, Amin et al. [35] proposed a lightweight AKA protocol that is applied to IoT devices in a distributed cloud computing environment. The mutual authentication between the user, service provider, and control server is implemented in their protocol, and a common session key is shared between the user and the server provider. In the scheme indicated above, only a symmetric cryptographic system is used to make the scheme highly efficient. Yeh et al. [30] proposed the first AKA elliptic curve cryptography (ECC) wireless sensor network solution, leading to other researchers proposing an increasing number of ECC-based AKA protocols [36, 41–46].

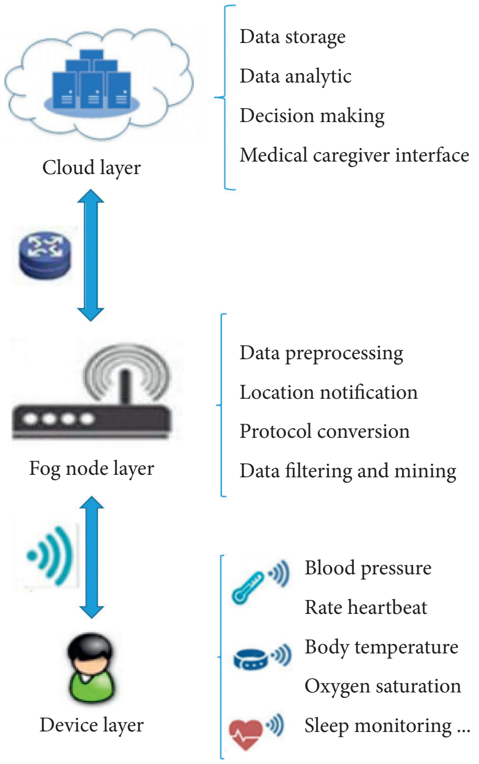


Figure 1: The concept of fog-driven IoT healthcare system.

Although several AKA schemes have been proposed for IoT environments, these protocols are rarely suitable for directly deployed fog computing environments. Hamid et al. [45] proposed a third-party single-round AKA protocol with bilinear pairing for this feature and indicated that it can ensure the privacy of medical data of the fog-based medical system. However, because the session key generated by this scheme is static, it cannot provide forward privacy. The key exchange mechanism of this scheme is based on Joux's three-party Diffie–Hellman key exchange algorithm [43]; thus, it is also vulnerable to man-in-the-middle attacks. Recently, Jia et al. [46] proposed an AKA scheme for a fog-driven IoT healthcare system using bilinear pairs, in which the cloud server authenticates the IoT device as well as the fog node and generates a shared common session key between them. Based on the Bellare–Rogaway–Pointcheval (BRP) security model [42], they claim that the proposed scheme can resist various known attacks. Informal security analysis also indicates that this scheme retains user anonymity and untractability. Some important related works are summarized in Table 1.

In this study, we first analyzed Jia et al.'s scheme and revealed that it is vulnerable to a random number impersonation attack and key compromise impersonation attack. Then, we proposed an enhancement based on their proposal and remedied the shortcomings of their scheme. In our

Table 1: The summary of authentication schemes.

Scheme	Cryptographic techniques	Limitations
Ref. [31]	Smart card One-way hash function	Vulnerable to smart card theft attacks Does not support anonymity Does not support untraceability
Ref. [35]	Symmetric encryption One-way hash function Elliptic curve cryptography	Does not support anonymity Vulnerable to impersonation attacks
Ref. [36]	Bilinear pairing One-way hash function Smart card	Vulnerable to replay attacks Does not support mutual authentication
Ref. [46]	Elliptic curve cryptography Bilinear pairing Identity-based cryptography	Insecure session key establishment Does not support anonymity
Ref. [41]	Bilinear pairing One-way hash function	Vulnerable to impersonation attacks

proposed scheme, the mutual authentication and key agreement between the three entities can be achieved only by one round of communication. After the cloud server verifies the identity of the IoT devices and fog nodes, it generates shared common session keys between them. For a security analysis, we adopted the BAN logic, ProVerif, and an informal security analysis. These approaches can provide evidence indicating that our improvement can resist several well-known security threats.

2. Cryptanalysis of Jia et al.'s AKA Scheme

2.1. Review of Jia et al.'s AKA Scheme. Here, we briefly review the scheme proposed by Jia et al. [46], which mainly consists of the following four phases: system setup, user registration, and fog node registration, as well as authentication and key agreement.

2.1.1. System Setup. The cloud service provider (CSP) selects a nonsingular elliptic curve on the finite field F_p, where p is a large prime number, and $l = \log_2 p$ is the security parameter. Let G be a cyclic group of order n generated by a base point P. Then, CSP selects a random $s \in Z_n^*$ and computes $P_{\text{pub}} = s \cdot P$. (G, P, P_{pub}) are published as the public system parameters, while s remains hidden. Six secure hash functions $\{h_0, h_1, h_2, h_3, h_4, h_5\}$, are selected by CSP, where h_0: $G_1 \longrightarrow \{0,1\}^*$, h_1: $\{0,1\}^* \times \{0,1\}^* \longrightarrow Z_p^*$, h_2: $\{0,1\}^* \times Z_p^* \times Z_n^* \longrightarrow Z_p^*$, h_3: $G_1 \times Z_p^* \times G_1 \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \longrightarrow Z_p^*$, h_4: $G_1 \times G_1 \times G_1 \times G_1 \times \{0,1\}^* \times \{0,1\}^* \longrightarrow Z_p^*$, and h_5: $G_2 \times G_1 \times G_1 \times G_1 \longrightarrow Z_p^*$. We assume that the CSP is fully trusted and also holds a database to record registered users and fog nodes.

2.1.2. User Registration. U_i inputs respective identity ID_i and password PW_i, and then computes $\text{RID}_i = h_1(\text{ID}_i \| \text{PW}_i) \oplus r_i$, where $r_i \in Z_p^*$ is a random number chosen by U_i. Then, U_i sends $(\text{ID}_i, \text{RID}_i)$ to CSP via a secure channel. After receiving the U_i request, CSP randomly chooses $x_i \in Z_p^*$ and computes $R_i = h_2(\text{ID}_i || s || x_i) \oplus \text{RID}_i$. The CSP then stores R_i in the smart card and the (ID_i, x_i) in its database and finally sends the smart card to the user over a secure channel. After the user receives the smart card, U_i calculates $R_i^* = R_i \oplus r_i$ and replaces R_i on the card with R_i^*.

2.1.3. Fog Node Registration. Each fog node F_N must be registered with the CSP before deployment. F_N transmits its identity ID_j to CSP. Then, CSP randomly selects $y_j \in Z_p^*$ and computes $R_j = h_2(\text{ID}_j || s || y_j)$; CSP sends R_j to the fog node over a secure channel and stores (ID_j, y_j) into its database.

2.1.4. Authentication and Key Agreement. In this phase, CSP can help U_i and F_N to authenticate each other and establish a session key SK after executing the following steps:

(a) U_i randomly chooses $a \in Z_n^*$ and computes $A = a \cdot P$, $\overline{A} = a \cdot P_{\text{pub}}$, $\text{PID}_i = \text{ID}_i \oplus h_0(\overline{A})$, $M_i = h_1(\text{ID}_i \| \text{PW}_i) \oplus R^*$, $|N_i = h_3(\overline{A} \| M_i \| A \| \text{ID}_i \| \text{ID}_j \| T_u)|$, where T_u is the current timestamp. U_i sends $\text{Msg}_1 = \{A, \text{PID}_i, N_i, T_u\}$ to F_N.

(b) Upon receiving Msg_1, F_N first checks that the freshness of the timestamp T_u meets the requirements. Then, F_N randomly selects $b \in Z_n^*$ and calculates $B = b \cdot P$, $\overline{B} = b \cdot P_{\text{pub}}$, $\text{PID}_j = \text{ID}_j \oplus h_0(\overline{B})$, $| \ L_j = h_3(\overline{B} \| R_j \| A \| \text{PID}_j \| \text{ID}_j \| T_f)$, where T_f is the current timestamp. Finally, F_N sends $\text{Msg}_2 = \{A, B, \text{PID}_i, \text{PID}_j, N_i, L_j, T_u, T_f\}$ to the CSP.

(c) After receiving Msg_2, CSP first checks the validity of two timestamps T_u, T_f and then executes the following steps:

  (i) CSP computes $\overline{A}' = sA$, $\overline{B}' = sB$,$\text{ID}_i' = \text{PID}_i \oplus h_0(\overline{A}')$, and $\text{ID}_j' = \text{PID}_j \oplus h_0(\overline{B}')$.

  (ii) CSP searches its database to find entries that match (ID_i', x_i) and (ID_j', y_j). If there are no matching entries, CSP denies the request and immediately terminates the session. Otherwise, CSP computes $M_i' = h_2(\text{ID}_i' || s || x_i)$, $R_j' = h_2(\text{ID}_j' || s || y_i)$, $N_i' = h_3(\overline{A}' || M_i' || A || \text{ID}_i' || \text{ID}_j' \| T_u)$, and $L_j' = h_3(\overline{B}' || R_i' || A || \text{ID}_i' || \text{ID}_j' \| T_f)$.

(iii) CSP checks whether $N_i = N_i'$ and $L_j = L_j'$. If one of these equations is not true, the CSP rejects the request and terminates. Otherwise, it randomly chooses $c \in Z_n^*$ and computes $C = c \cdot P$ $\mathrm{Auth}_i = h_4$ $(A\|B\|C\|\overline{A}\,'||\mathrm{ID}_i'||T_c)|\mathrm{Auth}_j = h_4$ $(A\|B\|C\|\overline{B}'||\mathrm{ID}_j'||T_c)$, $K_c = e(A,B)^C$, and $\mathrm{SK}_c = h_5(K_c \|A\|B\|C)$; note, the current timestamp is T_c. Finally, CSP forwards $\mathrm{Msg}_3 = \{C, \mathrm{Auth}_i, \mathrm{Auth}_j, T_c\}$ to FN_j.

(d) Upon receiving Msg_3, F_N checks the freshness of T_c and verifies whether $\mathrm{Auth}_j = h_4(A\|B\|C\|\overline{B}||\mathrm{ID}_j||T_c)$. If the equation is not true, F_N terminates the session. Otherwise, F_N calculates $\mathrm{SK}_f = h_5(K_f\|A\|B\|C)$, where $K_f = e(A,C)^b$. Then, F_N sends $\mathrm{Msg}_4 = \{B, C, \mathrm{Auth}_i, T_c\}$ to U_i.

(e) Upon receiving Msg_4, U_i checks the freshness of T_c and verifies whether $\mathrm{Auth}_i = h_4(A\|B\|C\|\overline{A}||\mathrm{ID}_i||T_c)$. If not, U_i aborts the session. Otherwise, U_i computes $\mathrm{SK}_u = h_5(K_u\|A\|B\|C)$, where $K_u = e(B,C)^a$.

2.2. Security Weakness of Jia et al.'s Scheme

2.2.1. Known Session-Specific Temporary Information Attack. Here, we demonstrate that Jia et al.'s scheme suffered from a known session-specific temporary information attack. This attack is indicated in Canetti and Krawczyk's (CK) adversary model [47]. We allow an attacker E to fully control the communications over the user, fog node, and CSP for "authentication and key agreement phase." Thus, E can intercept the messages and obtain the hidden information of a current session from either side over a public channel, which enabled the recovery of key information from the session, such as the session key and the entity's identity.

(a) *Session key recovery.* Based on the CK adversarial model, we may assume that an attacker E can obtain a random number a of users U_i. Note, E can also be intercepted $\{A, B, \mathrm{PID}_i, \mathrm{PID}_j, N_i, L_j, T_u, T_f, C, \mathrm{Auth}_i, \mathrm{Auth}_j\}$ in the open channel. Then, E can compute $\mathrm{SK}_u = h_5(A\|B\|C\|Ku)$, where $K_u = e(B,C)^a$. Note, we may assume that E can obtain b or c from F_N and CSP. The session key SK can also be computed by $e(A,C)^b$ and $e(A,B)^c$ because $\mathrm{SK} = e(B,C)^a = e(A,C)^b = e(A,B)^c$ in Jia et al.'s scheme; note, a, b, and c are random numbers chosen by U_i, F_N, and CSP, respectively.

(b) *Identity recovery (anonymity violation).* By the same assumption in (a), E can recover the U_i identity $\mathrm{ID}_i = \mathrm{PID}_i \oplus h_0(\overline{A^*})$, where $\overline{A^*} = a \cdot P_{\mathrm{pub}}$. Similarly, E can recover $\mathrm{ID}_j = \mathrm{PID}_j \oplus h_0(\overline{B^*})$, where $\overline{B^*} = b \cdot P_{\mathrm{pub}}$, while E obtains the F_N random value b.

2.2.2. Lack of Per-Verification. Step (a) of the authentication and key agreement phase lacks verifying the user input ID_i and PW_i. This will increase the redundant computational cost, while the user inputs an incorrect ID_i or PW_i. The incorrect input will be identified by CSP in step (c) of the authentication and key agreement phase.

3. Our Improved Scheme

In this section, we propose an improvement based on Jia et al.'s scheme to overcome the previously indicated security weaknesses in Section 2. In our improvement, the system setup is the same as in Jia et al.'s scheme.

3.1. Modified User Registration. This phase is depicted in Figure 2.

(a) U_i randomly chooses $r_i \in Z_p^*$, inputs the password PW_i and the identity ID_i to compute $\mathrm{RID}_i = h_1(\mathrm{ID}_i\|\mathrm{PW}_i) \oplus r_i$. Then, U_i sends $(\mathrm{ID}_i, \mathrm{RID}_i)$ to CSP via a secure channel.

(b) After receiving $(\mathrm{ID}_i, \mathrm{RID}_i)$, CSP randomly chooses $x_i \in Z_p^*$ and computes $q_i = h_2(\mathrm{ID}_i||s||x_i)$, $R_i = q_i \oplus \mathrm{RID}_i$, $D_i = h_2(q_i\|\mathrm{ID}_i) \oplus \mathrm{RID}_i$. The CSP then stores (R_i, D_i) in the smart card and the (ID_i, x_i) in its own database and finally sends the smart card to the user over a secure channel.

(c) After the user receives the smart card, U_i calculates $R_i^* = R_i \oplus r_i$, $V_i = D_i \oplus r_i$, and replaces R_i, D_i with R_i^* and V_i.

3.2. Modified Fog Node Registration. F_N transmits its identity ID_j to the CSP. It randomly selects $y_j \in Z_p^*$ and computes $g_j = h_2(\mathrm{ID}_j||s||y_j)$. Then, CSP sends g_j to the fog node via a secure channel and stores (ID_j, y_j) in its own database. This phase is shown in Figure 3.

3.3. Modified Authentication and Key Agreement. This phase is depicted in Figure 4.

(a) U_i inputs ID_i and PW_i and computes $q_i = R_i^* \oplus h_1(\mathrm{ID}_i\|\mathrm{PW}_i)$, $V_i^* = h_2(q_i\|\mathrm{ID}_i) \oplus h_1(\mathrm{ID}_i\|\mathrm{PW}_i)$. Then, whether $V_i^* = V_i$ is checked. If the equation is true, U_i randomly chooses $a \in Z_n^*$ and computes $v_u = a \cdot q_i$, $A = v_u \cdot P$, $\overline{A} = v_u \cdot P_{\mathrm{pub}}$, $\mathrm{PID}_i = \mathrm{ID}_i \oplus h_0(\overline{A})$, $M_i = q_i = h_1(\mathrm{ID}_i\|\mathrm{PW}_i) \oplus R^*$, $N_i = h_3(\overline{A}||M_i||A||\mathrm{ID}_i||\mathrm{ID}_j\|T_u)$, where T_u is the current timestamp. Finally, U_i sends $\mathrm{Msg}_1 = \{A, \mathrm{PID}_i, N_i, T_u\}$ to F_N.

(b) Upon receiving Msg_1, F_N first checks that the freshness of the timestamp T_u meets the requirements. Then, it randomly selects $b \in Z_n^*$ and calculates $v_f = b \cdot g_j$, $B = v_f \cdot P$, $\overline{B} = v_f \cdot P_{\mathrm{pub}}$, $\mathrm{PID}_j = \mathrm{ID}_j \oplus h_0(\overline{B})$, $L_j = h_3(\overline{B}||R_j||A||\mathrm{PID}_j||\mathrm{ID}_j\|T_f)$, where T_f is the current timestamp. Finally, F_N forwards $\mathrm{Msg}_2 = \{A, B, \mathrm{PID}_i, \mathrm{PID}_j, N_i, L_j, T_u, T_f\}$ to the CSP.

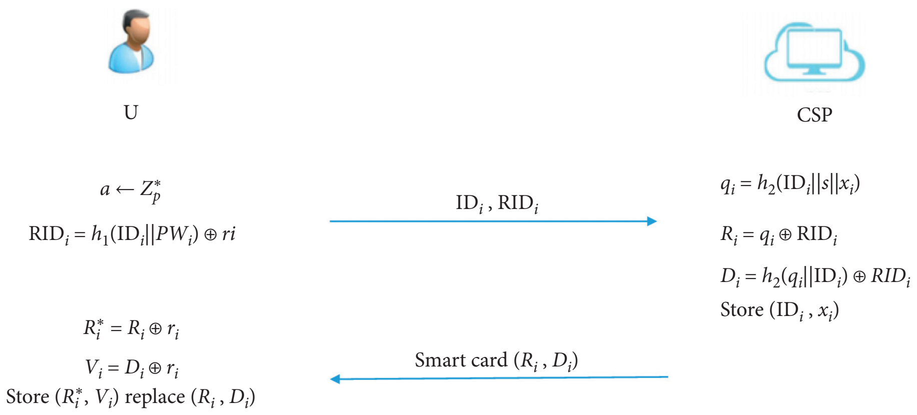

U
CSP
$a \leftarrow Z_p^*$
$\mathrm{RID}_i = h_1(\mathrm{ID}_i||PW_i) \oplus ri$
ID_i , RID_i
$q_i = h_2(\mathrm{ID}_i||s||x_i)$
$R_i = q_i \oplus \mathrm{RID}_i$
$D_i = h_2(q_i||\mathrm{ID}_i) \oplus RID_i$
Store (ID_i , x_i)
$R_i^* = R_i \oplus r_i$
$V_i = D_i \oplus r_i$
Store (R_i^*, V_i) replace (R_i , D_i)
Smart card (R_i , D_i)

Figure 2: Modified user registration phase.

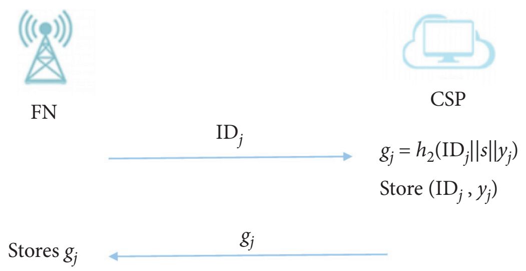

FN
CSP
ID_j
$g_j = h_2(\mathrm{ID}_j||s||y_j)$
Store (ID_j , y_j)
Stores g_j
g_j

Figure 3: Modified fog node registration phase.

(c) After receiving Msg_2, CSP first checks the validity of two timestamps T_u, T_f and then executes the following steps:

(i) To compute $\overline{A}' = s \cdot A$, $\overline{B}' = s \cdot B$, $\mathrm{ID}_i' = \mathrm{PID}_i \oplus h_0(\overline{A}')$, $\mathrm{ID}_j' = \mathrm{PID}_j \oplus h_0(\overline{B}')$ and then searches for (ID_i', x_i) and (ID_j', y_j) in its database. If there are no matching entries, CSP denies the request and immediately terminates the session.

(ii) To compute $M_i' = h_2(\mathrm{ID}_i'||s||x_i)$, $R_j' = h_2(\mathrm{ID}_j'||s||y_i)$, $N_i' = h_3(\overline{A}'||M_i'||A||\mathrm{ID}_i'||\mathrm{ID}_j'\|T_u)$, and $L_j' = h_3(\overline{B}'||R_j'||A||\mathrm{PID}_j'||\mathrm{ID}_j'\|T_f)$. Then, it checks whether $N_i = N_i'$ and $L_j = L_j'$. If one of these equations is not true, the CSP rejects the request and terminates.

(iii) CSP randomly chooses $c \in Z_n^*$ and computes $z_c = h_2(y_i||s||x_i)$, $v_c = c \cdot z_c$, $C = v_c \cdot P$, $\mathrm{Auth}_i = h_4(A||B||C||\overline{A}'||\mathrm{ID}_i'\|T_c)$, $\mathrm{Auth}_j = h_4(A||B||C||\overline{B}'||\mathrm{ID}_j'\|T_c)$, $K_c = e(A, B)^{v_c}$, $\mathrm{SK}_c = h_5(K_c||A||B\|C)$, where T_c is the current timestamp. Finally, CSP sends $\mathrm{Msg}_3 = \{C, \mathrm{Auth}_i, \mathrm{Auth}_j, T_c\}$ to F_N.

(d) Upon receiving $\mathrm{Msg}_3 = \{C, \mathrm{Auth}_i, \mathrm{Auth}_j, T_c\}$, F_N checks the freshness of T_c and verifies whether $\mathrm{Auth}_j = h_4(A\|B\|C\|\overline{B}||\mathrm{ID}_j||T_c)$. If the equation is not true, then F_N immediately terminates the session. Otherwise, F_N calculates $K_f = e(A, C)^{v_f}$, $\mathrm{SK}_f = h_5(K_f||A||B\|C)$, and forwards $\mathrm{Msg}_4 = \{B, C, \mathrm{Auth}_i, T_c\}$ to U_i.

(e) Upon receiving Msg_4, U_i checks the freshness of T_c and verifies if $\mathrm{Auth}_i = h_4(A\|B\|C\|\overline{A}||\mathrm{ID}_i||T_c)$. If the equation is not true, U_i immediately terminates the session. Otherwise, U_i calculates $K_u = e(A, C)^{v_u}$, $\mathrm{SK}_u = h_5(K_u||A||B\|C)$.

4. Security Analysis of Our Improved Scheme

In this section, the security of our scheme is illustrated by the BAN logic, ProVerif, and an informal security analysis.

4.1. Formal Security Analysis Using BAN Logic. In this subsection, the sharing session *SK* calculated by CSP between U_i, F_N, and CSP is presented, which can be used to send request information to the server when the user wants to obtain data from the server. Note, the following notations and rules for the BAN logic can be found in previous studies [33, 35, 39, 48].

4.1.1. Related Rules

Messages meaning rule $(A| \equiv A \overset{K}{\leftrightarrow} B, A \triangleleft \langle X \rangle K/A| \equiv B \sim X)$: if principal A believes that hidden K value is shared between principals A and B, and A receives the message X enciphered with K and then A believes that B is the sender of X.

Nonce verification rule $(A| \equiv \#(X), A| \equiv B \sim X/A| \equiv B| \equiv X)$: if A believes that message X is fresh and that B has sent X, then A believes that B also believes in message X.

Jurisdiction rule $(A| \equiv B| \Rightarrow X, A| \equiv B| \equiv X/A| \equiv X)$: if A believes that B has jurisdiction over X and that B believes on statement X, then A believes on X.

Session key introduction rule $A| \equiv \#(X), A| \equiv B| \equiv X/A| \equiv A \overset{K}{\leftrightarrow} B$: if A believes that message X is fresh and that B also believes on X, then A believes they share the session key.

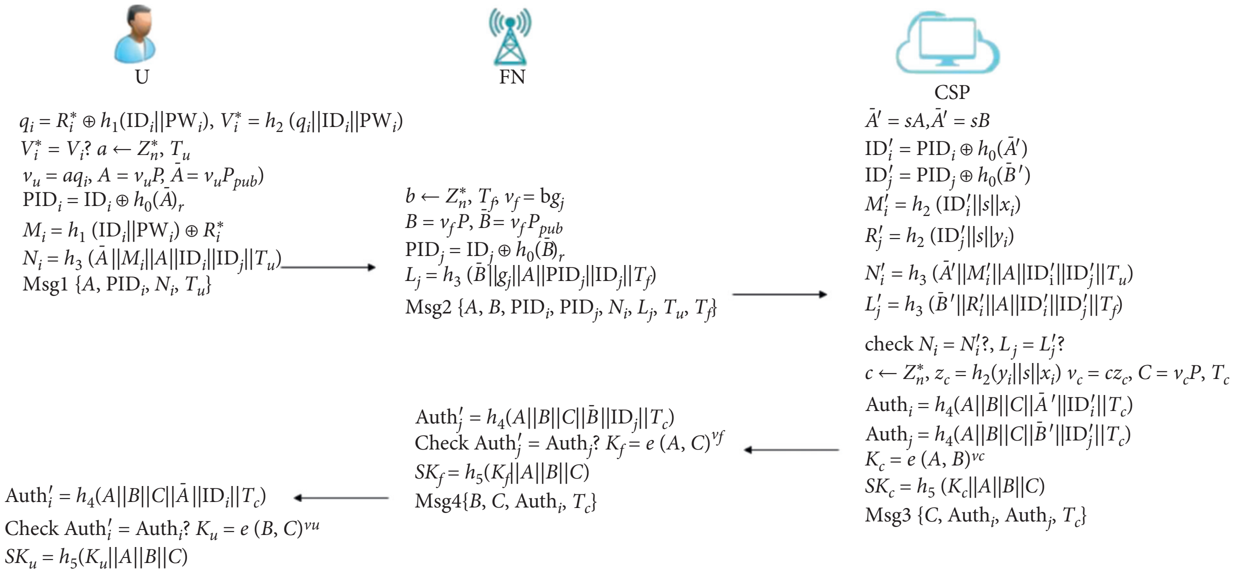

U
FN
CSP
$q_i = R_i^* \oplus h_1(\mathrm{ID}_i||\mathrm{PW}_i)$, $V_i^* = h_2\ (q_i||\mathrm{ID}_i||\mathrm{PW}_i)$
$V_i^* = V_i$? $a \leftarrow Z_n^*$, T_u
$v_u = aq_i$, $A = v_uP$, $\bar{A} = v_uP_{pub})$
$\mathrm{PID}_i = \mathrm{ID}_i \oplus h_0(\bar{A})_r$
$M_i = h_1\ (\mathrm{ID}_i||\mathrm{PW}_i) \oplus R_i^*$
$N_i = h_3\ (\bar{A}\,||M_i||A||\mathrm{ID}_i||\mathrm{ID}_j||T_u)$
Msg1 $\{A, \mathrm{PID}_i, N_i, T_u\}$
$b \leftarrow Z_n^*$, T_f, $v_f = \mathrm{b}g_j$
$B = v_fP$, $\bar{B} = v_fP_{pub}$
$\mathrm{PID}_j = \mathrm{ID}_j \oplus h_0(\bar{B})_r$
$L_j = h_3\ (\bar{B}||g_j||A||\mathrm{PID}_j||\mathrm{ID}_j||T_f)$
Msg2 $\{A, B, \mathrm{PID}_i, \mathrm{PID}_j, N_i, L_j, T_u, T_f\}$
$\bar{A}' = sA, \bar{A}' = sB$
$\mathrm{ID}_i' = \mathrm{PID}_i \oplus h_0(\bar{A}')$
$\mathrm{ID}_j' = \mathrm{PID}_j \oplus h_0(\bar{B}')$
$M_i' = h_2\ (\mathrm{ID}_i'||s||x_i)$
$R_j' = h_2\ (\mathrm{ID}_j'||s||y_i)$
$N_i' = h_3\ (\bar{A}'||M_i'||A||\mathrm{ID}_\mathrm{i}'||\mathrm{ID}_j'||T_u)$
$L_j' = h_3\ (\bar{B}'||R_i'||A||\mathrm{ID}_i'||\mathrm{ID}_j'||T_f)$
check $N_i = N_i'$?, $L_j = L_j'$?
$c \leftarrow Z_n^*$, $z_c = h_2(y_i||s||x_i)$ $v_c = cz_c$, $C = v_cP$, T_c
$\mathrm{Auth}_i = h_4(A||B||C||\bar{A}'||\mathrm{ID}_i'||T_c)$
$\mathrm{Auth}_j = h_4(A||B||C||\bar{B}'||\mathrm{ID}_j'||T_c)$
$K_c = e\ (A, B)^{vc}$
$SK_c = h_5\ (K_c||A||B||C)$
Msg3 $\{C, \mathrm{Auth}_i, \mathrm{Auth}_j, T_c\}$
$\mathrm{Auth}_j' = h_4(A||B||C||\bar{B}||\mathrm{ID}_j||T_c)$
Check $\mathrm{Auth}_j' = \mathrm{Auth}_j$? $K_f = e\ (A, C)^{vf}$
$SK_f = h_5(K_f||A||B||C)$
Msg4$\{B, C, \mathrm{Auth}_i, T_c\}$
$\mathrm{Auth}_i' = h_4(A||B||C||\bar{A}\,||\mathrm{ID}_i||T_c)$
Check $\mathrm{Auth}_i' = \mathrm{Auth}_i$? $K_u = e\ (B, C)^{vu}$
$SK_u = h_5(K_u||A||B||C)$

Figure 4: Modified authentication and key agreement phase.

Belief rule $(A| \equiv B| \equiv (XY)/A| \equiv B| \equiv X)$: if A believes that B believes formula (X, Y), then A believes that B also believes the X or Y part of the formula.

4.1.2. Goals

GOAL 1: $U_i| \equiv (U_i \overset{\mathrm{SK}}{\leftrightarrow} F_N)$
GOAL 2: $U_i| \equiv F_N| \equiv (U_i \overset{\mathrm{SK}}{\leftrightarrow} F_N)$
GOAL 3: $F_N| \equiv (U_i \overset{\mathrm{SK}}{\leftrightarrow} F_N)$
GOAL 4: $F_N| \equiv U_i| \equiv (U_i \overset{\mathrm{SK}}{\leftrightarrow} F_N)$
GOAL 5: $\mathrm{CSP}| \equiv (U_i \overset{\mathrm{SK}}{\leftrightarrow} F_N)$
GOAL 6: $\mathrm{CSP}| \equiv U_i| \equiv (U_i \overset{\mathrm{SK}}{\leftrightarrow} F_N)$
GOAL 7: $\mathrm{CSP}| \equiv F_N| \equiv (U_i \overset{\mathrm{SK}}{\leftrightarrow} F_N)$

4.1.3. Idealize the Communication Messages

Msg1 $U_i \longrightarrow F_N$: $\{A, \mathrm{PID}_i, N_i, T_u\}$
Msg2 $F_N \longrightarrow \mathrm{CSP}$: $\left\{B, \mathrm{PID}_j, L_j, T_f, A, \mathrm{PID}_i, N_i, T_u\right\}$
Msg3 $U_i \longrightarrow \mathrm{CSP}$: $\{A, \mathrm{PID}_i, N_i, T_u\}$
Msg4 $\mathrm{CSP} \longrightarrow F_N$: $\left\{\mathrm{Auth}_j, C, T_c\right\}$
Msg5 $\mathrm{CSP} \longrightarrow U_i$: $\{\mathrm{Auth}_i, C, T_c\}$
Msg6 $F_N \longrightarrow U_i$: $\{B, \mathrm{Auth}_i, C, T_c\}$

4.1.4. Initial State Assumptions

A1: $U_i| \equiv \#(a)$
A2: $U_i| \equiv \#(A)$
A3: $U_i| \equiv (A)$
A4:$U_i| \equiv \#(B)$
A5: $U_i| \equiv \#(C)$
A6: $F_N| \equiv \#(b)$
A7: $F_N| \equiv \#(B)$
A8: $F_N| \equiv (B)$
A9:$F_N| \equiv \#(A)$
A10:$F_N| \equiv \#(C)$
A11: $\mathrm{CSP}| \equiv \#(c)$
A12: $\mathrm{CSP}| \equiv \#(C)$
A13: $\mathrm{CSP}| \equiv (C)$
A14: $\mathrm{CSP}| \equiv \#(A)$
A15:$\mathrm{CSP}| \equiv \#(B)$
A16: $U_i| \equiv U_i \overset{\mathrm{ID}_j}{\leftrightarrow} F_N$
A17: $U_i| \equiv U_i \overset{(\mathrm{ID}_i,\mathrm{PID}_i,R_i)}{\leftrightarrow} \mathrm{CSP}$
A 18: $U_i| \equiv F_N = > B$
A 19: $U_i| \equiv \mathrm{CSP} = > C$
A 20: $F_N| \equiv U_i \overset{\mathrm{ID}_j}{\leftrightarrow} F_N$
A 21: $F_N| \equiv F_N \overset{(\mathrm{ID}_j,g_j)}{\leftrightarrow} \mathrm{CSP}$
A 22: $F_N| \equiv U_i = > A$
A 23:$F_N| \equiv \mathrm{CSP} = > C$
A 24: $\mathrm{CSP}| \equiv U_i \overset{(\mathrm{ID}_i,\mathrm{PID}_i,R_i)}{\leftrightarrow} \mathrm{CSP}\ \alpha$
A 25: $\mathrm{CSP}| \equiv F_N \overset{(\mathrm{ID}_j,g_j)}{\leftrightarrow} \mathrm{CSP}$
A 26:$\mathrm{CSP}| \equiv F_N = > B$
A 27:$\mathrm{CSP}| \equiv U_i = > A$

If a is a random number chosen by U_i, we can obtain A1and A2; when Msg1 sends form U_i to F_N, A22 is obtained. From A22, we obtain A9; when Msg3 sends form U_i to CSP, we obtain A27. From A27, we obtain A14. Similarly, because b is a random number chosen by F_N, we obtain A6 and A7; when Msg6 sends from F_N to U_i, we obtain A18. From A18, we obtain A4; when Msg2 sends from F_N to CSP, we obtain A26. From A26, we obtain A15. c is a random number chosen by CSP; we obtain A26 and A27; when Msg5 sends from CSP to U_i, we obtain A19. From A19, we obtain A5;

when Msg4 sends from CSP to F_N, we obtain A23. From A23, we obtain A10.

4.1.5. Main Proofs Using BAN Rules and Assumptions

(1) For GOAL 1 and GOAL 2. From message Msg6 and using the seeing rule, we obtain S1: $U_i \lhd \{B, \text{Auth}_i, C, T_c\}$. Using the seeing rule, we obtain S2: $U_i \lhd \{B\}$. Using A16, S2, and the message meaning rule, we obtain S3: $U_i \mid\equiv F_N \mid\sim \{B\}$. Using A4, S3, and the nonce verification rule, we obtain S4: $U_i \mid\equiv F_N \mid\equiv B$. Using A18, S4, and the jurisdiction rule, we obtain S5: $U_i \mid\equiv B$. Based on message Msg5 and the seeing rule, we obtain S6: $U_i \lhd \{\text{Auth}_i, C, T_c\}$. Using the seeing rule, we obtain S7: $U_i \lhd \{C\}$. According to A17, S7, and the message meaning rule, we have S8: $U_i \mid\equiv \text{CSP} \mid\sim \{C\}$. Using A5, S8, and the nonce verification rule, we obtain S9: $U_i \mid\equiv \text{CSP} \mid\equiv C$. Using A19, S9, and the jurisdiction rule, we obtain S10: $U_i \mid\equiv C$. Based on A2, A4, A5, A3, S5, S10, and the belief rule, we obtain S11: $U_i \mid\equiv \#(A, B, C)$ and S12: $U_i \mid\equiv (A, B, C)$. Because $A = v_u P, B = v_f P, C = v_c P$, we can obtain S13: $U_i \mid\equiv (v_u, v_f, v_c)$. Because $K_u = K_f = K_c = e(B, C)^{v_u} = e(A, C)^{v_f} = e(A, B)^{v_c}$, $\text{SK}_u = \text{SK}_f = \text{SK}_c = h_5(K_u \| A \| B \| C) = h_5(K_f \| A \| B \| C) = h_5(K_c \| A \| B \| C)$. Using A2, A16, S12, S13, and the belief rule, we obtain S14: $U_i \mid\equiv (U_i \overset{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 1).

Using A2, S14, and the session key introduction rule, we obtain S15: $U_i \mid\equiv F_N \mid\equiv (U_i \overset{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 2).

(2) For GOAL 3 and GOAL 4. From message Msg1 and using the seeing rule, we obtain S16: $F_N \lhd \{A, \text{PID}_i, N_i, T_u\}$. Using the seeing rule, we obtain S17: $F_N \lhd \{A\}$. According to A20, S17, and the message meaning rule, we have S18: $F_N \mid\equiv U_i \mid\sim \{A\}$. Employing A9, S18, and the nonce verification rule, we obtain S19: $F_N \mid\equiv U_i \mid\equiv A$. Using A22, S19, and the jurisdiction rule, we have S20: $F_N \mid\equiv A$. From message Msg4 and using the seeing rule, we have S21: $F_N \lhd \{\text{Auth}_j, C, T_c\}$. We obtain S22: $F_N \lhd \{C\}$ via the seeing rule. According to A21, S22, and the message meaning rule, we obtain S23: $F_N \mid\equiv \text{CSP} \mid\sim \{C\}$. Using A10, S23, and the nonce verification rule, we obtain S24: $F_N \mid\equiv \text{CSP} \mid\equiv C$. According to A23, S24, and the jurisdiction rule, we have S25: $F_N \mid\equiv C$. According to A7, A10, A9, A8, S20, S25, and the belief rule, we obtain S26: $F_N \mid\equiv \#(A, B, C)$ and S27: $F_N \mid\equiv (A, B, C)$. Because $A = v_u P, B = v_f P, C = v_c P$, we can obtain S28: $F_N \mid\equiv (v_u, v_f, v_c)$. Using A7, A20, S27, S28, and the belief rule, we obtain S29: $F_N \mid\equiv (U_i \overset{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 3).

By using A7, S29, and the session key introduction rule, we obtain S30: $F_N \mid\equiv U_i \mid\equiv (U_i \overset{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 4).

(3) For GOAL 5, GOAL 6, and GOAL 7. According to Msg2 and using the seeing rule, we obtain S31: $\text{CSP} \lhd \{B, \text{PID}_j, L_j, T_f, A, \text{PID}_i, N_i, T_u\}$. Using the seeing rule, we obtain S32: $\text{CSP} \lhd \{B\}$. Using A25, S32, and the message meaning rule, we obtain S33: $\text{CSP} \mid\equiv F_N \mid\sim \{B\}$. Using A15, S33, and the nonce verification rule, we obtain S34: $\text{CSP} \mid\equiv F_N \mid\equiv B$. Using A26, S34, and the jurisdiction rule, we obtain S35: $\text{CSP} \mid\equiv B$. Based on Msg3 and the seeing rule, we obtain S36: $\text{CSP} \lhd \{A, \text{PID}_i, N_i, T_u\}$. We have S37: $\text{CSP} \lhd \{A\}$ via the seeing rule. According to A24, S37, and the message meaning rule, we obtain S38: $\text{CSP} \mid\equiv U_i \mid\sim \{A\}$. Using A14, S38, and the nonce verification rule, we obtain S39: $\text{CSP} \mid\equiv U_i \mid\equiv A$. According to A27, S39, and the jurisdiction rule, we obtain S40: $\text{CSP} \mid\equiv A$. According to A14, A12, A15, A13, S35, S40, and the belief rule, we obtain S41: $\text{CSP} \mid\equiv \#(A, B, C)$ and S42: $\text{CSP} \mid\equiv (A, B, C)$. Because $A = v_u P, B = v_f P, A = v_c P$, we can obtain S43: $U_i \mid\equiv (v_u, v_f, v_c)$. Using A12, S42, S43, and the belief rule, we obtain S44: $\text{CSP} \mid\equiv (U_i \overset{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 5).

Using A14, S44, and the session key introduction rule, we obtain S45: $\text{CSP} \mid\equiv U_i \mid\equiv (U_i \overset{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 6).

Using A15, S44, and the session key introduction rule, we obtain S46: $\text{CSP} \mid\equiv F_N \mid\equiv (U_i \overset{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 7).

4.2. Informal Security Analysis. In this section, we demonstrate that our improved scheme can achieve the following well-known security requirements.

4.2.1. Known Session-Specific Temporary Information Attacks. The session key $\text{SK}_u = \text{SK}_f = \text{SK}_c = h_5(K_u \| A \| B \| C) = h_5(K_f \| A \| B \| C) = h_5(K_c \| A \| B \| C)$ is generated utilizing the hidden values of $K_u = K_f = K_c = e(B, C)^{v_u} = e(A, C)^{v_f} = e(A, B)^{v_c}$, and $v_u = aq_i$, $v_f = bg_j$, $v_c = cz_c$; (A, B, C) can be intercepted on an open channel, but adversaries do not know (q_i, g_j, z_c) because they are the hidden values of U_i, F_N, and CSP, respectively, and, thus, cannot calculate (v_u, v_f, v_c). Therefore, despite adversaries determining (a, b, c), they cannot calculate (K_u, K_f, K_c) without (q_i, g_j, z_c). Therefore, an opponent cannot recover SK using temporarily leaked session-specific information {a, b, c}.

(q_i, g_j) are the hidden values of U_i, and F_N, respectively; if only (a, b) is found, but not (q_i, g_j), the adversaries cannot calculate $v_u = aq_i$, $v_f = bg_j$. $\overline{A} = v_u P_{\text{pub}}$, $\overline{B} = v_f P_{\text{pub}}$, $\text{PID}_i = \text{ID}_i \oplus h_0(\overline{A})$, $\text{PID}_j = \text{ID}_j \oplus h_0(\overline{B})$; $(\text{PID}_i, \text{PID}_j)$ can be intercepted on an open channel, but adversaries cannot retrieve $\text{ID}_i = \text{PID}_i \oplus h_0(\overline{A})$ and $\text{ID}_j = \text{PID}_j \oplus h_0(\overline{B})$ without (v_u, v_f). If adversaries intercept (A, B) on an open channel, they do not know the key s of the CSP and, thus, cannot calculate $\overline{A}' = sA$ and $\overline{B}' = sB$, or retrieve $\text{ID}_i = \text{PID}_i \oplus h_0(\overline{A}')$, $\text{ID}_j = \text{PID}_j \oplus h_0(\overline{B}')$ without s.

4.2.2. Mutual Authentication. CSP authenticates U_i by verifying whether ID_i' equals to the ID_i saved in the CSP database and whether N_i' equals to N_i, N_i sent from U_i. U_i authenticates CSP by verifying whether Auth_i' equals to $\text{Auth}_i = h_4(A \| B \| C \| \overline{A}' \| \text{ID}_i' \| T_c)$, which includes C calculated by CSP.

Similarly, CSP authenticates F_N by verifying whether ID_j' equals to the ID_j saved in the CSP database and whether L_j' equals L_j, L_j sent from F_N. F_N authenticates CSP by verifying whether Auth_j' equals to $\text{Auth}_j = h_4(A \| B \| C \| \overline{B}' \| \text{ID}_j' \| T_c)$, which includes C calculated by CSP.

F_N authenticates U_i by verifying whether Auth_j' equals to Auth_j which includes A calculated by U_i, and U_i authenticates

F_N by verifying whether Auth_i' equals to $\text{Auth}_i = h_4(A\|B\|C\|\overline{A}'\|\text{ID}_i'\|T_c)$, which includes B calculated by F_N.

4.2.3. Impersonation Attack. To impersonate a legitimate user, the adversary has to obtain the identity ID_i, password PW_i, and $q_i = h_2(\text{ID}_i\|s\|x_i)$ of U_i or construct $A = v_uP$, $\text{PID}_i = \text{ID}_i \oplus h_0(\overline{A})$, and $N_i = h_3(\overline{A}\|M_i\|A\|\text{ID}_i\|\text{ID}_j\|T_u\|)$. First, the opponent is unable to guess the correct identity and password of U_i through "password-guessing attack." Second, to construct $\{A, \text{PID}_i, N_i\}$, the adversary has to obtain the key s and parameter x_i. However, it cannot compute q_i without ID_i, s, and x_i, which are crucial for computing $\{A,\text{PID}_i, N_i\}$. Thus, the adversary cannot impersonate a legitimate user.

Similarly, to mimic a legitimate fog node, the opponent must obtain the identity ID_j and $q_j = h_2(\text{ID}_j\|s\|x_j)$ of F_N or construct $B = v_fP$,$\text{PID}_j = \text{ID}_j \oplus h_0(\overline{B})$, and $L_j = h_3(\overline{B}\|g_j\|A\|\text{ID}_j\|\text{ID}_j\|T_f\|)$; the adversary can obtain the identity ID_j, but it is impossible for the adversary to determine $g_j = h_2(\text{ID}_j\|s\|y_j)$, which is computed and assigned by CSP in F_N registration. g_j cannot be computed without s and y_j, which are crucial for computing $\{B, \text{PID}_j, L_j\}$. Thus, the adversary cannot impersonate a legitimate F_N.

The adversary is also unable to impersonate CSP. To compute $C = v_cP$, $\text{Auth}_i = h_4(A\|B\|C\|\overline{A}'\|\text{ID}_i'\|T_c)$, and $\text{Auth}_j = h_4(A\|B\|C\|\overline{B}'\|\text{ID}_j'\|T_c)$, s, x_i, and y_j are required to compute $C = v_cP = h_2(y_i\|s\|x_i)cP$. However, the adversary cannot obtain C unless it obtains all three factors at the same time. This is beyond the capacity of an adversary. Thus, the adversary cannot impersonate CSP.

4.2.4. Man-in-the-Middle Attacks. If the adversary obtains Msg1 or Msg2 from the public channel and modifies Msg1 or Msg2 to launch a man-in-the-middle attack, the identity authentication of CSP cannot be passed; the premise of the authentication of CSP is to determine the identity of U_i and F_N. From "(2),"we know that CSP will compute ID_i' and ID_j' and compare the values with ID_i and ID_j saved in the CSP database; if it is not equal, the session will immediately be terminated. From "(1)," we know that the adversary cannot obtain ID_i and ID_j. Meanwhile, from "(3)," we also know that the adversary cannot obtain the values of s, x_i, and y_j. Thus, the modified messages cannot pass the verification of $N_i' = N_i$ and $L_j' = L_j$ from CSP.

If the adversary obtains Msg3 or Msg4 from the open channel and modifies Msg3 or Msg4 to launch the man-in-the-middle attack, the authentication from U_i and F_N will still not be passed. As indicated by "(2)," we can see that if the messages are modified by the adversary, they cannot pass the verification of $\text{Auth}_i' = \text{Auth}_i$ and $\text{Auth}_j' = \text{Auth}_j$ from U_i and F_N.

4.2.5. Known Session Key Attacks. A scheme is considered vulnerable to known session key attacks if an adversary wants to use the old compromised session key to obtain sensitive parameters and keys for subsequent communication sessions. In our scheme, $\text{SK}_u = \text{SK}_f = \text{SK}_c = h_5(K_u\|A\|B\|C) = h_5(K_f\|A\|B\|C) = h_5(K_c\|A\|B\|C)$, $K_u = K_f = K_c = e(B,C)^{v_u} = e(A,C)^{v_f} = e(A,B)^{v_c}$, $A = v_uP, B = v_fP, C = v_cP$, $v_u = aq_i$, $v_f = bg_j$, $v_c = cz_c$, is refreshed using random numbers $\{a, b, c\}$ and the attacker does not know $\{q_i, g_j, z_c\}$. Thus, owing to the computational difficulty of the elliptic curve Diffie–Hellman problem, it is impossible for the attacker to obtain the new SK information from the old SK and extract $\{a, b, c\}$ from $\{A, B, C\}$; thus, the scheme we proposed can withstand the known session key attack.

4.2.6. Compromise Impersonation Attacks. If the CSP long-term key s is compromised, the adversary may use s to impersonate a legitimate user to determine F_N and CSP. However, all attack sessions are terminated immediately, as follows. In a worst case scenario, the adversary may have access to the data $R_i^* = h_2(\text{ID}_i\|s\|x_i) \oplus h_1(\text{ID}_i\|\text{PW}_i)$,$V_i = h_2(h_2(\text{ID}_i\|s\|x_i)\|\text{ID}_i) \oplus h_1(\text{ID}_i\|\text{PW}_i)$, in the stolen smart card SC. Despite knowing s, the adversary does not know the hidden values of $\{\text{ID}_i, x_i,\text{PW}_i\}$ to compute $h_2(\text{ID}_i\|s\|x_i) = R_i^* \oplus h_1(\text{ID}_i\|\text{PW}_i)$ or $q_i = h_2(\text{ID}_i\|s\|x_i)$ directly. Thus, the adversary cannot generate $\text{Msg1} = \langle A, \text{PID}_i, N_i, T_u\rangle$ to masquerade U_i to launch a new session.

The adversary may intercept messages sent by U_i during authentication and key negotiation and attempt to impersonate the initiator of the session. However, the session will terminate immediately because the attacker cannot calculate $K_u = e(B,C)^{v_u}$ correctly without knowing the hidden values of $\{a, q_i\}$, despite knowing s.

4.2.7. Parallel Session Attacks. When the entity is in session, the adversary may try to replay the old messages to launch a new session attack; however, this is impossible. When an attacker replays {M1, M2} to CSP, it can pass the verification of $N_i' = h_3(\overline{A}'\|M_i'\|A\|\text{ID}_i'\|\text{ID}_j'\|T_u)$, $L_j' = h_3(\overline{B}'\|R_j'\|A\|\text{ID}_i'\|\text{ID}_j'\|T_f)$. However, because the attacker does not know $\{a, b\}$ and $\{q_i, g_j\}$, it cannot compute one of $K_u = h_5(K_u\|A\|B\|C)$, $K_f = h_5(K_f\|A\|B\|C)$, $K_u = K_f = e(B,C)^{v_u} = e(A,C)^{v_f}$, and $v_u = aq_i$, $v_f = bg_j$, and the attacker session is immediately aborted.

4.2.8. Stolen Smart Card Attacks. If an attacker steals the smart card and extracts $R_i^* = h_2(\text{ID}_i\|s\|x_i) \oplus h_1(\text{ID}_i\|\text{PW}_i)$, $V_i = h_2(h_2(\text{ID}_i\|s\|x_i)\|\text{ID}_i) \oplus h_1(\text{ID}_i\|\text{PW}_i)$, he/she may impersonate U_i to F_N and CSP. However, the attacker does not know the sensitive parameter $\{\text{ID}_i, \text{PW}_i, x_i, \text{s}\}$ to generate the initiator message $\text{PID}_i = \text{ID}_i \oplus h_0(\overline{A})$, $N_i = h_3(\overline{A}\|M_i\|A\|\text{ID}_i\|\text{ID}_j\|T_u)$, thus cannot impersonate U_i to F_N and CSP. Hence, the proposed scheme can withstand stolen smart card attacks.

4.2.9. Password-Guessing Attacks. If an adversary obtains information regarding $\{A, B, \text{PID}_i, \text{PID}_j, N_i, L_j, C, \text{Auth}_i, \text{Auth}_j, T_u, T_f, T_c\}$ from the open channel, online password-guessing attacks may be launched. However, the adversary will fail because $A = v_u P$, $\mathrm{PID}_i = \mathrm{ID}_i \oplus h_0(\overline{A})$, $N_i = h_3(\overline{A} \| M_i \| A \| \mathrm{ID}_i \| \mathrm{ID}_j \| T_u)$, $B = v_f P$, $\mathrm{PID}_j = \mathrm{ID}_j \oplus h_0(\overline{B})$, $L_j = h_3(\overline{B} \| g_j \| A \| \mathrm{PID}_j \| \mathrm{ID}_j \| T_f \|)$, $C = v_c P$, $\mathrm{Auth}_i = h_4(A \| B \| C \| \overline{A}' \| \mathrm{ID}_i' \| T_c)$, $\mathrm{Auth}_j = h_4(A \| B \| C \| \overline{B}' \| \mathrm{ID}_j' \| T_c)$, and PW_i are not included in these values. Therefore, PW_i remains secure.

If the smart card is compromised by an opponent, the parameter $\{R_i^*, V_i\}$ in the SC can be obtained through the power analysis attack method, and then off-line dictionary attacks can be made based on the relevant parameter $R_i^* = h_2(\mathrm{ID}_i \| s \| x_i) \oplus h_1(\mathrm{ID}_i \| \mathrm{PW}_i)$, $V_i = h_2(h_2(\mathrm{ID}_i \| s \| x_i) \| \mathrm{ID}_i) \oplus h_1(\mathrm{ID}_i \| \mathrm{PW}_i)$, to guess the user password. However, because the values $\{x_i, s\}$ are only known by the CSP, the opponent cannot verify the accuracy of the guess value; therefore, all sensitive parameters are safe.

4.2.10. Privileged-Insider Attacks. When the attacker obtains U_i' registration information (ID_i, RID_i, x_i) and the key s of CSP, the intent is to compute the session key $\mathrm{SK}_u = \mathrm{SK}_f = \mathrm{SK}_c = h_5(K_u \| A \| B \| C) = h_5(K_f \| A \| B \| C) = h_5(K_c \| A \| B \| C)$, which is randomized using $\{a, b, c\}$ and $\{q_i, g_j, z_c\}$. By $K_u = K_f = K_c = e(B, C)^{v_u} = e(A, C)^{v_f} = e(A, B)^{v_c}$ and $v_u = a q_i$, $v_f = b g_j$, $v_c = c z_c$, the attacker can compute $q_i = h_2(\mathrm{ID}_i \| s \| x_i)$ and obtain (A, B, C) from the public channel. However, (a, b, c) are random numbers independently selected by U_i, F_N, and CSP, respectively, and are not available to the attacker; therefore, v_u and SK_u cannot be computed.

Similarly, when the attacker obtains the F_N registration information (ID_j, y_j) and the key s of CSP, the intent is to compute the session key SK_f; the attacker can compute $g_j = h_2(\mathrm{ID}_j \| s \| y_j)$ and obtain (A, B, C) from the public channel. However, (a, b, c) are random numbers independently selected by U_i, F_N, and CSP, respectively, and are not available to the attacker; therefore, v_f and SK_f cannot be computed.

The attacker also cannot compute SK_c; $z_c = h_2(y_i \| s \| x_i)$ can be computed, but $v_c = c z_c$ cannot be computed without the c selected by CSP. Thus, the modified scheme can withstand privileged-insider attacks.

4.2.11. Replay Attacks. The adversary may attempt to replay old messages {Msg1, Msg2, Msg3, and Msg4}. However, all communicated messages are refreshed and rely on the timestamp $\{T_u, T_f, T_c\}$ as well as random numbers $\{a, b, c\}$. Upon receiving the authentication request from the sender, the receiver first checks the freshness of the timestamp. If the timestamp is not fresh, the session is terminated immediately.

4.2.12. Perfect Forward Secrecy. Perfect forward secrecy indicates that if a long-term key is revealed to an attacker, the SK between U_i, F_N, and CSP, cannot be computed and remains secure. If an attacker attempts to calculate the session key, $\mathrm{SK}_u = \mathrm{SK}_f = \mathrm{SK}_c = h_5(K_u \| A \| B \| C \|) = h_5(K_f \| A \| B \| C \|) = h_5(K_c \| A \| B \| C \|)$, which is randomized using numbers $\{a, b, c\}$ and $\{q_i, g_j, z_c\}$; $K_u = K_f = K_c = e(B, C)^{v_u} = e(A, C)^{v_f} = e(A, B)^{v_c}$, $v_u = a q_i$, $v_f = b g_j$, $v_c = c z_c$. The attacker obtains (A, B, C) from the public channel; however, the attacker needs to compute one of the parameters v_u, v_f, v_c, which cannot be obtained, thus SK cannot be calculated. Therefore, the improved scheme can provide perfect forward secrecy.

4.2.13. No Key Control. Each entity cannot control the key agreement process to calculate *SK* individually, where $\mathrm{SK}_u = \mathrm{SK}_f = \mathrm{SK}_c = h_5(K_u \| A \| B \| C \|) = h_5(K_f \| A \| B \| C \|) = h_5(K_c \| A \| B \| C \|)$, $K_u = K_f = K_c = e(B, C)^{v_u} = e(A, C)^{v_f} = e(A, B)^{v_c}$, and $v_u = a q_i$, $v_f = b g_j$, $v_c = c z_c$, $A = v_u P$, $B = v_f P$, $C = v_c P$. The details are as follows:

(a, b, c) are random numbers independently selected by U_i, F_N, and CSP, respectively, and (A, B, C) are computed independently by each entity. If U_i does not know the values of B and C, which are contributed by F_N and CSP, SK_u cannot be computed. Similarly, F_N and CSP cannot compute SK_f and SK_c without the values of (A, C) and (A, B).

4.2.14. Unknown Key-Share. From “‘(2),” we know that all three entities are mutually identifiable. If U_i and entity-1 establish the session key and send the request message of entity-1 by mistake to entity-2, it is impossible to pass the validation $\mathrm{ID}_i' = \mathrm{ID}_i$, $\mathrm{ID}_j' = \mathrm{ID}_j$, $N_i = h_3(\overline{A} \| M_i \| A \| \mathrm{ID}_i \| \mathrm{ID}_j \| T_u) = N_i' = h_3(\overline{A}' \| M_i' \| A \| \mathrm{ID}_i' \| \mathrm{ID}_j' \| T_u)$, and $L_j = h_3(\overline{B} \| g_j \| A \| \mathrm{PID}_j \| \mathrm{ID}_j \| T_f \|) = L_j' = h_3(\overline{B}' \| R_j' \| A \| \mathrm{ID}_i' \| \mathrm{ID}_j' \| T_f)$, thus the session terminates immediately. Therefore, the proposed scheme can resist unknown key-share attacks.

4.3. Evaluation by ProVerif. In this section, we choose the widely accepted software tool ProVerif [49–53] to perform security simulation and testing of the scheme, which can fully guarantee the characteristics of confidentiality and authenticity.

The complete scheme shown in Figure 4 is implemented and validated in ProVerif. During the simulation, we assumed the two channels shown in Figure 5(a). The ch is a common channel used for the transmission of messages between entities in the authentication phase. The sch is a secure channel for user and fog node registration. Variables and constants are also defined in Figure 5(a). ID_i and ID_j are the identities of users and fog nodes, respectively, SK_u, SK_f, and SK_c are the keys negotiated between the three entities, respectively.

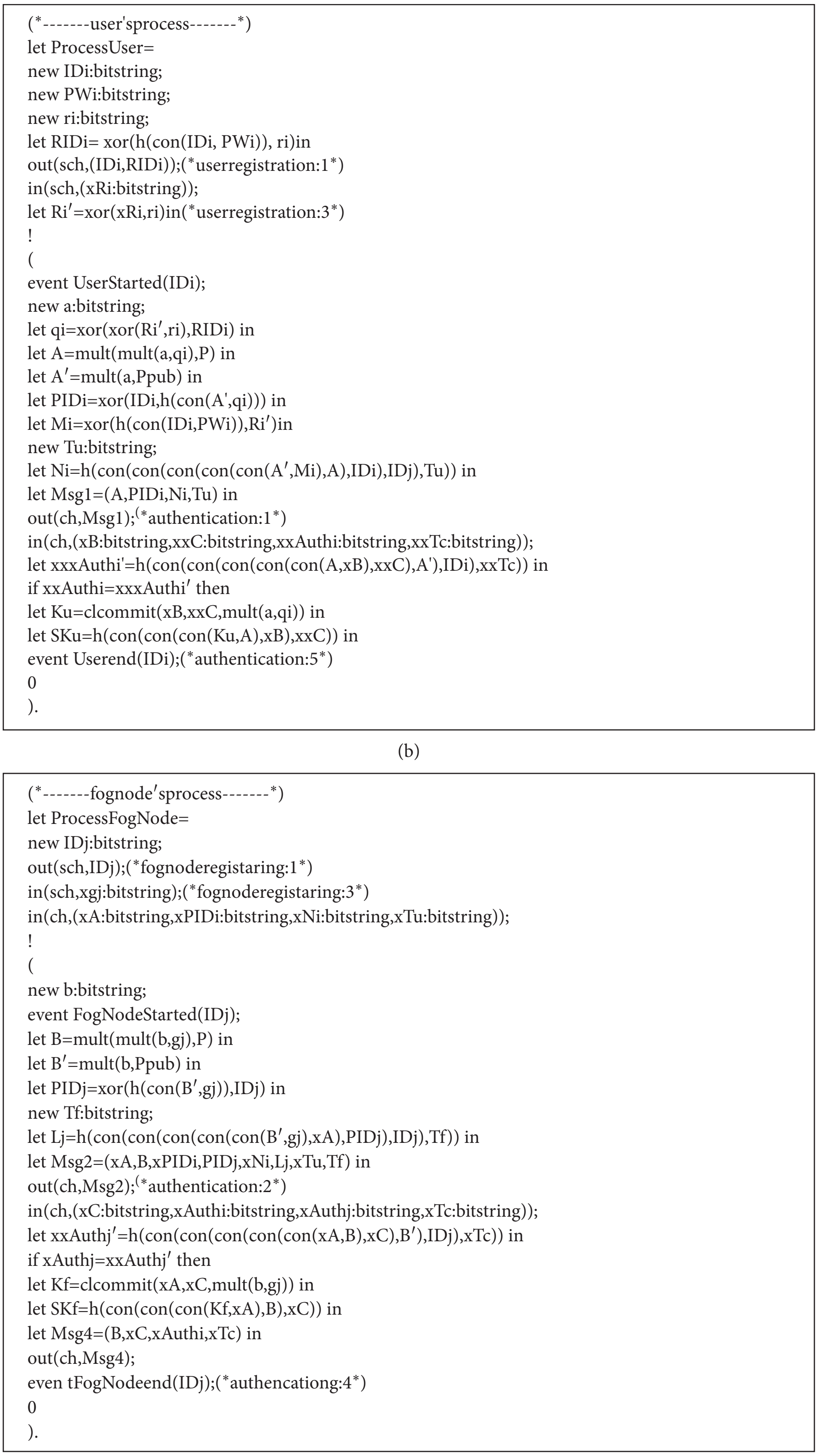
User and fog node are described by starting and ending events, and scheme authenticity is achieved by exposing the respective relationships between the start and end intervals of related events initiated by a particular participant. If no end event is reached, it means the scheme failed to terminate and the scheme is incorrect. Figures 5(b)–5(d) represent the user, fog node, and CSP implementation simulation processes, respectively, which are described in detail in Section 3 and executed in parallel.

The necessary queries are defined in Figure 5(a) to verify the security and correctness of the scheme. The query attacker simulates an actual attack to obtain the session key and secret random numbers, while the other three query in-events

```
(*-------channel-------*)
free ch:channel.(*public channel*)
free sch:channel[private].(*secure channel,used for registering*)
(*-------shared key-------*)
free SKu:bitstring [private].
free SKf:bitstring [private].
free SKc:bitstring [private].
(*-------constants and variables-------*)
free s:bitstring [private].(*the CSP's secret kay*)
(*free SKu:bitstring [private].
free SKf:bitstring [private].
free SKc:bitstring [private].*)
free ri:bitstring [private].
free a:bitstring [private].
free b:bitstring [private].
free c:bitstring [private].
const IDi:bitstring.(*user′sID*)
const IDj:bitstring.(*fognode′sID*)
const Ri:bitstring.
const gj:bitstring.
const Ppub:bitstring.
const P:bitstring.
(*-------functions & reductions & equations-------*)
fun h(bitstring):bitstring.(*hashfunction*)
fun mult(bitstring, bitstring):bitstring.(*scalar multiplication operation*)
fun con(bitstring, bitstring):bitstring.(*conncatention operation*)
reduc forall m:bitstring, n:bitstring;getmess(con(m, n))= m.
fun x or(bitstring, bitstring):bitstring.(*XOR operation*)
equation forall m:bitstring, n:bitstring;xor(xor(m, n), n)= m.
fun clcommit(bitstring, bitstring, bitstring):bitstring.(*pairing operation*)
(*-------queries-------*)
query attacker(SKu).
query attacker(SKf).
query attacker(SKc).
query attacker(ri).
query attacker(a).
query attacker(b).
query attacker(c).
query var:bitstring;inj-event(Userend(var))==> inj-event(UserStarted(var)).
query var:bitstring;inj-event(FogNodeend(var))==> inj-event(FogNodeStarted(var)).
(*query var inj-event(endCSP)==> inj-event(startCSP).*)
(*-------events-------*)
event UserStarted(bitstring).
event Userend(bitstring).
event FogNodeStarted(bitstring).
event FogNodeend (bitstring).
```

(a)

Figure 5: Continued.



```
(*-------user'sprocess-------*)
let ProcessUser=
new IDi:bitstring;
new PWi:bitstring;
new ri:bitstring;
let RIDi= xor(h(con(IDi, PWi)), ri)in
out(sch,(IDi,RIDi));(*userregistration:1*)
in(sch,(xRi:bitstring));
let Ri′=xor(xRi,ri)in(*userregistration:3*)
!
(
event UserStarted(IDi);
new a:bitstring;
let qi=xor(xor(Ri′,ri),RIDi) in
let A=mult(mult(a,qi),P) in
let A′=mult(a,Ppub) in
let PIDi=xor(IDi,h(con(A',qi))) in
let Mi=xor(h(con(IDi,PWi)),Ri′)in
new Tu:bitstring;
let Ni=h(con(con(con(con(con(A′,Mi),A),IDi),IDj),Tu)) in
let Msg1=(A,PIDi,Ni,Tu) in
out(ch,Msg1);(*authentication:1*)
in(ch,(xB:bitstring,xxC:bitstring,xxAuthi:bitstring,xxTc:bitstring));
let xxxAuthi'=h(con(con(con(con(con(A,xB),xxC),A'),IDi),xxTc)) in
if xxAuthi=xxxAuthi′ then
let Ku=clcommit(xB,xxC,mult(a,qi)) in
let SKu=h(con(con(con(Ku,A),xB),xxC)) in
event Userend(IDi);(*authentication:5*)
0
).
```

(b)

```
(*-------fognode′sprocess-------*)
let ProcessFogNode=
new IDj:bitstring;
out(sch,IDj);(*fognoderegistaring:1*)
in(sch,xgj:bitstring);(*fognoderegistaring:3*)
in(ch,(xA:bitstring,xPIDi:bitstring,xNi:bitstring,xTu:bitstring));
!
(
new b:bitstring;
event FogNodeStarted(IDj);
let B=mult(mult(b,gj),P) in
let B′=mult(b,Ppub) in
let PIDj=xor(h(con(B′,gj)),IDj) in
new Tf:bitstring;
let Lj=h(con(con(con(con(con(B′,gj),xA),PIDj),IDj),Tf)) in
let Msg2=(xA,B,xPIDi,PIDj,xNi,Lj,xTu,Tf) in
out(ch,Msg2);(*authentication:2*)
in(ch,(xC:bitstring,xAuthi:bitstring,xAuthj:bitstring,xTc:bitstring));
let xxAuthj′=h(con(con(con(con(con(xA,B),xC),B′),IDj),xTc)) in
if xAuthj=xxAuthj′ then
let Kf=clcommit(xA,xC,mult(b,gj)) in
let SKf=h(con(con(con(Kf,xA),B),xC)) in
let Msg4=(B,xC,xAuthi,xTc) in
out(ch,Msg4);
even tFogNodeend(IDj);(*authencationg:4*)
0
).
```

(c)

Figure 5: Continued.

```
(*-------CSP′sprocess-------*)
let UserReg=
in(sch,(rIDi:bitstring,rRIDi:bitstring));
new xi:bitstring;
new yj:bitstring;
let qi=h(con(con(rIDi,s),xi)) in
let Ri=xor(qi,rRIDi) in
out(sch,Ri).(*user registaring:2*)
let FogNodeReg=z
in(sch,(rIDj:bitstring));
new xi:bitstring;
new yj:bitstring;
let gj=h(con(con(rIDj,s),yj)) in
out(sch,gj).(*fognode registaring:2*)
let CSPAuth=
in(ch,(xxA:bitstring,xB:bitstring,xxPIDi:bitstring,xPIDj:bitstring,xxNi:bitstring,
xLj:bitstring,xxTu:bitstring,xTf:bitstring));
new xi:bitstring;
new yj:bitstring;
let A″=mult(s,xxA) in
let B″=mult(s,xB) in
let IDi′=xor(xxPIDi,h(A″)) in
let IDj′=xor(xPIDj,h(B″)) in
let Mi′=h(con(con(IDi′,s),xi)) in
let gj′=h(con(con(IDj′,s),yj)) in
let xxxNi′=h(con(con(con(con(con(A″,Mi′),xxA),IDi′),IDj'),xxTu)) in
let xxLj′=h(con(con(con(con(con(B″,gj′),xxA),IDi′),IDj′),xTf)) in
if xxNi=xxxNi′ then
if xLj=xxLj′ then
new c:bitstring;
let zc=h(con(con(xi,s),yj)) in
let C=mult(mult(c,zc),P) in
new Tc:bitstring;
let Authi=h(con(con(con(con(con(xxA,xB),C),A″),IDi′),Tc)) in
let Authj=h(con(con(con(con(con(xxA,xB),C),B″),IDj′),Tc)) in
let Kc=clcommit(xxA,xB,mult(c,zc)) in
let SKc=h(con(con(con(Kc,xxA),xB),C)) in
let Msg3=(C,Authi,Authj,Tc) in
out(ch,Msg3).
(*-------authentication:3--------*)
let ProcessCSP=UserReg|FogNodeReg|CSPAuth.
(*--------main-------*)
process
let Ppub=mult(s,P) in
(!ProcessUser|!ProcessFogNode|!ProcessCSP)
```

(d)

Figure 5: ProVerif simulation. (a) Declarations. (b) User's process. (c) Fog node's process. (d) CSP's process and main.

```
-- Query not attacker(SKu[])
nounif mess(sch[],rIDj_2969)/-5000
Completing...
Starting query not attacker(SKu[])
RESULT not attacker(SKu[]) is true.
-- Query not attacker(SKf[])
nounif mess(sch[],rIDj_7537)/-5000
Completing...
Starting query not attacker(SKf[])
RESULT not attacker(SKf[]) is true.
-- Query not attacker(SKc[])
nounif mess(sch[],rIDj_12015)/-5000
Completing...
Starting query not attacker(SKc[])
RESULT not attacker(SKc[]) is true.
```

(a)

```
-- Query not attacker(ri[])
nounif mess(sch[],rIDj_16493)/-5000
Completing...
Starting query not attacker(ri[])
RESULT not attacker(ri[]) is true.
--Query not attacker(a[])
nounif mess(sch[],rIDj_20971)/-5000
Completing...
Starting query not attacker(a[])
RESULT not attacker(a[])istrue.
--Query not attacker(b[])
nounif mess(sch[],rIDj_25449)/-5000
Completing...
Starting query not attacker(b[])
RESULT not attacker(b[]) is true.
--Query not attacker(c[])
nounif mess(sch[],rIDj_29927)/-5000
Completing...
Starting query not attacker(c[])
RESULT not attacker(c[]) is true.
```

(b)

```
-- Query inj-event(Userend(var))
==> inj-event(UserStarted(var))
nounif mess(sch[],rIDj_34441)/-5000
Completing...
Starting query inj-event(Userend(var))
==> inj-event(UserStarted(var))
RESULT inj-event(Userend(var))
==> inj-event(UserStarted(var))istrue.
-- Query inj-event(FogNodeend(var_52))
==> inj-event(FogNodeStarted(var_52))
nounif mess(sch[],rIDj_39848)/-5000
Completing...
Starting query inj-event(FogNodeend(var_52))
==> inj-event(FogNodeStarted(var_52))
RESULT inj-event(FogNodeend(var_52))
==> inj-event(FogNodeStarted(var_52)) is true.
```

(c)

Figure 6: Verification result. (a) Query results for SK. (b) Query results for secrecy. (c) Query results for events.

Table 2: Comparison of security.

Security properties	Ref. [36]	Ref. [46]	Ref. [41]	Our scheme
Known session-specific temporary information attack	Yes	No	Yes	Yes
User anonymity and untraceability	Yes	No	Yes	Yes
Mutual authentication	No [55]	—	Yes	Yes
Impersonation attacks	No [55]	—	No [56]	Yes
Man-in-the-middle attacks	Yes	Yes	Yes	Yes
Known session key attacks	Yes	Yes	Yes	Yes
Compromise impersonation attacks	—	—	Yes	Yes
Parallel session attacks	—	—	—	Yes
Stolen smart card attacks	Yes	Yes	Yes	Yes
Password-guessing attacks	Yes	Yes	Yes	Yes
Privileged-insider attacks	Yes	—	—	Yes
Replay attacks	No [55]	Yes	Yes	Yes
Perfect forward privacy	Yes	Yes	Yes	Yes
No key control	—	Yes	—	Yes
Unknown key-share	—	—	—	Yes

Table 3: Computation time of basic operations.

Operation	Description	Times (ms)
$\mathrm{TG}_{\widehat{e}}$	Bilinear pairing	17.4
TG_m	Scalar multiplication	13.5
TG_a	Point addition	0.48
T_h	Hash function	0.42
T_{fe}	Fuzzy extractor function [36]	17.1

Table 4: Performance comparisons (computation costs).

	Ref. [36]	Ref. [46]	Ref. [41]	Our scheme
Authentication and key agreement	$3\mathrm{TG}_m + 19T_h + 1T_{fe}$	$3\mathrm{TG}_{\widehat{e}} + 7\mathrm{TG}_m + 18T_h$	$4\mathrm{TG}_{\widehat{e}} + 10\mathrm{TG}_m + 25T_h$	$3\mathrm{TG}_{\widehat{e}} + 10\mathrm{TG}_m + 21T_h$
Total	65.58 ms	154.26 ms	215.1 ms	196.02 ms

correspond to the start and end events of the three processes. If any of these queries result in false, it means that the scheme is incorrect. The results of the discussion query are shown in Figure 6.

It can be seen from the results in Figures 6(a) and 6(b) that the session key negotiated between entities and the secret random number selected by each entity are secure when dealing with security threats, which proves that the authenticity and confidentiality of our scheme are guaranteed during the execution process. The results in Figure 6(c) show that each process started and ended successfully, which proves the correctness of our scheme.

5. Performance Evaluation

In this section, the security features and defense against various attacks are compared between our scheme and the previous schemes [36, 41, 46] in Table 2. We can conclude that our scheme is more secure than the compared schemes. Note that "Yes" represents that the scheme can resist the indicated attack, whereas "No" represents that the scheme cannot, and "–" represents that the attack method indicated is not in the scope of the scheme.

Subsequently, we evaluate the performance of the proposed scheme from the perspective of computational and communication costs. The improved scheme was implemented in JAVA with JDK version 1.3, and the simulation of the scheme was based on the JAVA paired cryptography library (JPBC) [54], version JPBC-2.0.0. A Windows 10 computer system was used as the experimental platform, which was configured with a quad-core 2.3 GHz Intel(R) Core i5-8300H processor and 16 GB memory. The software developed is the community version of IntelliJ IDEA 2020.2.1 and uses the widely accepted type A pairing, which is based on the curve $y^2 = x^3 + x$ structure in the field F_q of a specific $q = 3 \bmod 4$. We have listed the symbols ($\mathrm{TG}_{\widehat{e}}$, TG_m, T_h, TG_a) and time used in the performance comparison in Table 3. Table 4 presents the calculation costs for the different phases of the scheme.

As shown by the analysis in Table 4, the computing cost for our scheme is slightly higher than that of schemes [36, 46]; however, our scheme provides auxiliary security features, and the mandatory security objectives achieved by this scheme are greater than those achieved by other schemes [36, 41, 46]. Our solution provides security features that other solutions do not have, such as being able to resist replay attacks and impersonation attacks and providing user anonymity, mutual authentication, etc.

To calculate the communication and storage costs, we present that the length of the random nonce, password, and

Table 5: Performance comparisons (communication costs).

	Ref. [36]	Ref. [46]	Ref. [41]	Our scheme
Authentication and key agreement	$5\|G_1\| + 4\|q\|$	$6\|G_1\| + 9\|q\| + 5\|T\|$	$3\|G_1\| + 4\|q\| + 4\|T\|$	$6\|G_1\| + 9\|q\| + 5\|T\|$
Total	5760 bit	7744 bit	3840 bit	7744 bit

Table 6: Performance comparisons (storage cost).

Scheme	Storage cost (bits)
Ref. [36]	320
Ref. [46]	640
Ref. [41]	736
Ours	640

identity is 160 bits, and the length of a point in the G_1 group is 1024 bits, denoted as $|G_1|$. The output length of the hash functions h_0, h_1, h_2, h_3, and h_4 in Z_p^* is 160 bits, denoted as $|q|$. The output length of h_5 and the key length are both 256 bits. The length of the timestamp is 32 bits, denoted as $|T|$. The communication and storage costs of our scheme and related schemes are listed in Tables 5 and 6.

As shown in Tables 5 and 6, the communication and the storage overhead of our scheme are slightly higher. The slightly higher cost of our scheme is mainly due to the increase in computing overhead while providing stronger security. However, because the primary purpose of a scheme is to ensure the security and privacy of data, it is acceptable to have a slightly higher communication cost but stronger security. After analyzing Tables 4 and 5, our scheme is concluded to be better than the other schemes [36, 41, 46], which can provide stronger security and withstand various known attacks.

6. Conclusion

The usage of fog-driven IoT healthcare systems has brought significant convenience to people. The authentication of the healthcare system is also the most important. Recently, a growing number of scholars have taken a closer look at healthcare systems and developed stronger authentication protocols for their certification environments. In this study, we proposed a secure authenticated and key agreement scheme in fog-driven IoT healthcare systems; the defects of the original scheme were analyzed and security improvements were proposed. An analysis of the performance evaluation and informal security in comparison to other related schemes is also presented in this study, which indicates that our scheme provides more security features. Our solution uses pairing technology, and the time cost is slightly higher than other solutions. Future studies can improve on this limitation, but our solution provides security features that other solutions do not have, which is more suitable for the practical application of medical system based on the IoT.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The work was supported in part by National Key Research and Development Project, China, under Grant no. 2018YFC1201102 and the Natural Science Foundation of Fujian Province, China, under Grant nos. 2018J01636 and 2018J01638.

References

[1] F. C. Chang and H. C. Huang, "A survey on intelligent sensor network and its applications," *Journal of Network Intelligence*, vol. 1, no. 1, pp. 1–15, 2016.

[2] J. S. Pan, L. Kong, T. W. Sung, P. W. Tsai, and V. Snasel, "Alpha-fraction first strategy for hierarchical wireless sensor networks," *Journal of Internet Technology*, vol. 19, no. 6, pp. 1717–1726, 2018.

[3] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, and S.-J. Lim, "An affinity propagation-based self-adaptive clustering method for wireless sensor networks," *Sensors*, vol. 19, no. 11, p. 2579, 2019.

[4] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.

[5] J. Wang, Y. Gao, C. Zhou, R. Simon Sherratt, and L. Wang, "Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs," *Computers, Materials & Continua*, vol. 62, no. 2, pp. 695–711, 2020.

[6] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, 2010.

[7] T. N. Gia, M. Jiang, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare internet of things: a case study on ECG feature extraction," in *Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, pp. 356–363, IEEE, Liverpool, UK, October 2015.

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

[9] A. M. Rahmani, T. N. Gia, B. Negash et al., "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of*

the First Edition of the MCC Workshop on Mobile Cloud Computing ACM, pp. 13–16, Helsinki, Finland, August 2012.

[11] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: a survey," in *Wireless Algorithms, Systems, and Applications. WASA 2015. Lecture Notes in Computer Science*, K. Xu and H. Zhu, Eds., vol. 9204, pp. 685–695, Springer, Cham, Switzerland, 2015.

[12] C. Huang, R. Lu, and K.-K. R. Choo, "Vehicular fog computing: architecture, use case, and security and forensic challenges," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 105–111, 2017.

[13] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K. R. Choo, and M. Dlodlo, "From cloud to fog computing: a review and a conceptual live vm migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.

[14] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.

[15] H. Xiong, Y. Wu, C. Jin, and S. Kumari, "Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT," *IEEE Internet of Things Journal*, .

[16] H. Xiong, Y. Zhao, Y. Hou et al., "Heterogeneous signcryption with equality test for IIoT environment," *IEEE Internet of Things Journal*, .

[17] Z. Meng, J.-S. Pan, and K.-K. Tseng, "PaDE: an enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80–99, 2019.

[18] A. Q. Tian, S. C. Chu, J. S. Pan, H. Cui, and W. M. Zheng, "A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station," *Sustainability*, vol. 12, no. 3, p. 767, 2020.

[19] S. C. Chu, X. Xue, J. S. Pan, and X. Wu, "Optimizing ontology alignment in vector space," *Journal of Internet Technology*, vol. 21, no. 1, pp. 15–22, 2020.

[20] Y. Huang, M. Hsieh, H. Chao, S. Hung, and J. Park, "Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 400–411, 2009.

[21] J. S. Pan, X. X. Sun, S. C. Chu, A. Abraham, and B. Yan, "Digital watermarking with improved SMS applied for QR code," *Engineering Applications of Artificial Intelligence*, vol. 97, Article ID 104049, 2021.

[22] R. Tso, "Two-in-one oblivious signatures," *Future Generation Computer Systems*, vol. 101, pp. 467–475, 2019.

[23] T.-Y. Wu, C.-M. Chen, K.-H. Wang, C. Meng, and E. K. Wang, "A provably secure certificateless public key encryption with keyword search," *Journal of the Chinese Institute of Engineers*, vol. 42, no. 1, pp. 20–28, 2019.

[24] J. Zhang, H. Liu, and L. Ni, "A secure energy-saving communication and encrypted storage model based on RC4 for EHR," *IEEE Access*, vol. 8, pp. 38995–39012, 2020.

[25] J. M.-T. Wu, G. Srivastava, A. Jolfaei, P. Fournier-Viger, and J. C.-W. Lin, "Hiding sensitive information in eHealth datasets," *Future Generation Computer Systems*, vol. 117, pp. 169–180, 2021.

[26] C. M. Chen, L. Xu, T. Y. Wu, and C. R. Li, "On the security of a chaotic maps-based three-party authenticated key agreement protocol," *Journal of Network Intelligence*, vol. 1, no. 2, pp. 61–66, 2016.

[27] C. M. Chen, Y. Huang, E. K. Wang, and T. Y. Wu, "Improvement of a mutual authentication protocol with anonymity for roaming service in wireless communications," *Data Science and Pattern Recognition*, vol. 2, no. 1, pp. 15–24, 2018.

[28] S. Kumari, P. Chaudhary, C.-M. Chen, and M. K. Khan, "Questioning key compromise attack on Ostad-Sharif et al.'s authentication and session key generation scheme for healthcare applications," *IEEE Access*, vol. 7, pp. 39717–39720, 2019.

[29] P. Wang, C.-M. Chen, S. Kumari et al., "HDMA: hybrid D2D message authentication scheme for 5G-enabled VANETs," *IEEE Transactions on Intelligent Transportation Systems*, p. 1, 2020.

[30] H.-L. Yeh, T.-H. Chen, P.-C. Liu, T.-H. Kim, and H.-W. Wei, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.

[31] M. Turkanovic, B. Brumen, and M. Holbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion," *Ad Hoc Networks*, vol. 20, pp. 96–112, 2014.

[32] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions," *Computer Networks*, vol. 73, pp. 41–57, 2014.

[33] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Networks*, vol. 36, pp. 152–176, 2016.

[34] T. Hayajneh, B. J. Mohd, M. Imran, G. Almashaqbeh, and A. V. Vasilakos, "Secure authentication for remote patient monitoring with wireless medical sensor network," *Sensors*, vol. 16, no. 4, p. 424, 2016.

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

[36] S. Challa, A. K. Das, V. Odelu et al., "An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks," *Computers & Electrical Engineering*, vol. 69, pp. 534–554, 2018.

[37] C.-M. Chen, K.-H. Wang, K.-H. Yeh, B. Xiang, and T.-Y. Wu, "Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3133–3142, 2019.

[38] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[39] T.-Y. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, and C.-M. Chen, "An authenticated key exchange protocol for multi-server architecture in 5G networks," *IEEE Access*, vol. 8, pp. 28096–28108, 2020.

[40] C.-M. Chen, Y. Huang, K.-H. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Information Systems*, p. 1, 2020.

[41] M. Nikravan and A. Reza, "A multi-factor user authentication and key agreement protocol based on bilinear pairing for the internet of things," *Wireless Personal Communications*, vol. 111, no. 1, pp. 463–494, 2020.

[42] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology—EUROCRYPT 2000. EUROCRYPT 2000.*

Lecture Notes in Computer Science, B. Preneel, Ed., Vol. 1807, Springer, Berlin, Germany, 2000.

[43] A. Joux, "A one round protocol for tripartite Diffie-Hellman," *Journal of Cryptology*, vol. 17, no. 4, pp. 263–276, 2004.

[44] C. T. Li, T. Y. Wu, C. L. Chen, C. C. Lee, and C. M. Chen, "An efficient user authentication and user anonymity scheme with provably security for IoT-based medical care system," *Sensors*, vol. 17, no. 7, p. 1482, 2017.

[45] H. A. Hamid, S. M. Rahman, M. S. Hossain, A. Almogren, and A. Alamri, "A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography," *IEEE Access*, vol. 5, pp. 22313–22328, 2017.

[46] 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.

[47] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels,," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques–Advances in Cryptology (EUROCRYPT'02)*, pp. 337–351, Amsterdam, Netherlands, April 2002.

[48] M. Burrows, R. A. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 24, no. 20, pp. 18–36, 1975.

[49] B. Blanchet, M. Sylvestre, M. X. Allamigeon, V. Cheval, B. Smyth, and C. Stentzel, "ProVerif: cryptographic protocol verifier in the formal model," 2019, http://prosecco.gforge.inria.fr/personal/bblanche/proverif/.

[50] K. Mansoor, A. Ghani, S. Chaudhry, S. Shamshirband, S. Ghayyur, and A. Mosavi, "Securing IoT-based RFID systems: a robust authentication protocol using symmetric cryptography," *Sensors*, vol. 19, no. 21, p. 4752, 2019.

[51] B. A. Alzahrani, S. A. Chaudhry, A. Barnawi, A. Al-Barakati, and M. H. Alsharif, "A privacy preserving authentication scheme for roaming in IoT-based wireless mobile networks," *Symmetry*, vol. 12, no. 2, p. 287, 2020.

[52] S. A. Chaudhry, "Correcting "PALK: password-based anonymous lightweight key agreement framework for smart grid," *International Journal of Electrical Power & Energy Systems*, vol. 125, p. 106529, 2021.

[53] T. Y. Wu, Y. Q. Lee, C. M. Chen, Y. Tian, and N. A. Al-Nabhan, "An enhanced pairing-based authentication scheme for smart grid communications," *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[54] A. D. Caro and V. Iovino, "JPBC: java pairing based cryptography," in *Proceedings of the 2011 IEEE Symposium on Computers and Communications (ISCC)*, pp. 850–855, IEEE, Corfu, Greece, June 2011.

[55] Z. Ali, A. Ghani, I. Khan, S. A. Chaudhry, S. H. Islam, and D. Giri, "A robust authentication and access control protocol for securing wireless healthcare sensor networks," *Journal of Information Security and Applications*, vol. 52, Article ID 102502, 2020.

[56] S. Shamshad, K. Mahmood, and S. Kumari, "Comments on "a multi-factor user authentication and key agreement protocol based on bilinear pairing for the internet of things," *Wireless Personal Communications*, vol. 112, no. 1, pp. 463–466, 2020.