sensors

MDPI

Article

# Provably Secure Lightweight Mutual Authentication and Key Agreement Scheme for Cloud-Based IoT Environments

**Sieun Ju and Yohan Park ***

Department of Computer Engineering, College of Engineering, Keimyung University, Daegu 42601, Republic of Korea; juse0204@gmail.com
* Correspondence: yhpark@kmu.ac.kr; Tel.: +82-53-580-5229

**Abstract:** A paradigm that combines cloud computing and the Internet of Things (IoT) allows for more impressive services to be provided to users while addressing storage and computational resource issues in the IoT environments. This cloud-based IoT environment has been used in various industries, including public services, for quite some time, and has been researched in academia. However, various security issues can arise during the communication between IoT devices and cloud servers, because communication between devices occurs in open channels. Moreover, issues such as theft of a user's IoT device or extraction of key parameters from the user's device in a remote location can arise. Researchers interested in these issues have proposed lightweight mutual authentication key agreement protocols that are safe and suitable for IoT environments. Recently, a lightweight authentication scheme between IoT devices and cloud servers has been presented. However, we found out their scheme had various security vulnerabilities, vulnerable to insider, impersonation, verification table leakage, and privileged insider attacks, and did not provide users with untraceability. To address these flaws, we propose a provably secure lightweight authentication scheme. The proposed scheme uses the user's biometric information and the cloud server's secret key to prevent the exposure of key parameters. Additionally, it ensures low computational costs for providing users with real-time and fast services using only exclusive OR operations and hash functions in the IoT environments. To analyze the safety of the proposed scheme, we use informal security analysis, Burrows–Abadi–Needham (BAN) logic and a Real-or-Random (RoR) model. The analysis results confirm that our scheme is secure against insider attacks, impersonation attacks, stolen verifier attacks, and so on; furthermore, it provides additional security elements. Simultaneously, it has been verified to possess enhanced communication costs, and total bit size has been shortened to 3776 bits, which is improved by almost 6% compared to Wu et al.'s scheme. Therefore, we demonstrate that the proposed scheme is suitable for cloud-based IoT environments.

**Keywords:** cloud computing; internet of things; authentication; cryptanalysis; real-or-random model; Burrow–Abadi–Needham logic



**Citation:** Ju, S.; Park, Y. Provably Secure Lightweight Mutual Authentication and Key Agreement Scheme for Cloud-Based IoT Environments. *Sensors* **2023**, *23*, 9766. https://doi.org/10.3390/s23249766

Academic Editor: Jian Li

Received: 10 November 2023
Revised: 1 December 2023
Accepted: 8 December 2023
Published: 11 December 2023



## 1. Introduction

The Internet of Things (IoT) is a network in which Internet-enabled objects interact with each other through the internet [1,2]. IoT objects collect data from their surroundings, provide web services to users, and communicate with each other. Therefore, IoT objects such as smart devices need significant resources to store data collected from sensors and perform real-time computations using limited hardware. Hence, addressing the limitations of storage and computing capacities is crucial for the formation of a network of IoT objects [3–5]. However, cloud computing technology refers to the practice of moving computational power and storage space from individual devices to larger shared data centers [6]. Cloud computing allows access to a shared pool of computing resources such as networks, servers, storage, and applications. By using cloud computing, it becomes possible to overcome the limitations inherent in IoT devices [7–10]. The development and discussion of cloud-based IoT (CloudIoT) have been ongoing since before 2008 and continue to evolve. Figure 1 illustrates the structure of CloudIoT. This structure comprises three entities: user, cloud server, and control server. Users with IoT devices can access the resources provided by the cloud service provider's server anytime and anywhere through IoT objects. The cloud server collects user's requests and delivers the right service through IoT. The control server, acting as a trusted entity, generates the necessary parameters for communication between authenticated users and the cloud server through the registration process. Additionally, it monitors the key agreement phase to ensure that users and the cloud server establish the same session key for subsequent communications when needed.

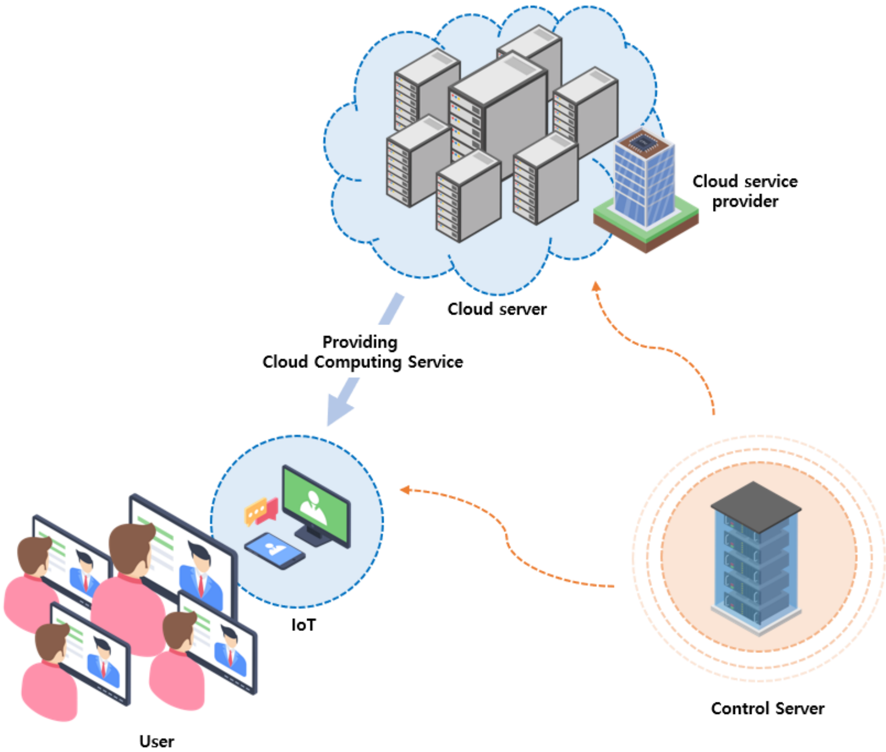


**Figure 1.** Cloud-based IoT enviroment.

In 2022, Wu et al. [11] proposed a lightweight authentication protocol for IoT-enabled cloud computing environments. The authors argued that their scheme could resist various attacks, such as man-in-the-middle, insider, DDoS, and masquerade attacks, and provides privacy, traceability, and integrity. However, we identified several vulnerabilities in Wu et al.'s scheme, including susceptibility to insider attacks, verification table attacks, user impersonation, and cloud server impersonation. Furthermore, the scheme lacks user untraceability, allowing an attacker to track the same user across different sessions through message eavesdropping alone. To address these vulnerabilities, we propose a provably secure lightweight mutual authentication and key agreement (MAKA) scheme. In our proposed scheme, we protect crucial parameters stored in the user's IoT smart card using the user's biometric information to prevent attacks like user impersonation and offline password-guessing. We also enhanced security by adding a secret key to the cloud server, preventing attackers from exploiting leaked database values. Additionally, we reduced the communication and computation overhead by employing only hash functions and exclusive-OR operations.

### 1.1. Research Contributions

We review and conduct a security analysis of Wu et al.'s authentication scheme. We demonstrate that Wu et al.'s scheme is vulnerable to insider attacks, verification table leakage attacks privileged insider attacks, user impersonation, and cloud server impersonation. Additionally, we propose an MAKA for cloud-based IoT environments that leverages biometric information. The proposed scheme is tailored to the IoT environments, using only exclusive OR operations and hash functions to align with a lightweight architecture. Additionally, we use a Real-or-Random (RoR) model and Burrow–Abadi–Needham (BAN) logic to demonstrate formally the security and robustness of the proposed. Moreover, we substantiate the security of our scheme against different attacks, including insider attacks, impersonation attacks, reply and man-in-the-middle (MITM) attacks, privileged insider attacks, ephemeral security leakage, stolen verifier attacks, DoS attacks, and session key disclosure attacks. In addition, we confirmed that our scheme can provide user anonymity, user untraceability, perfect forward secrecy, and mutual authentication. Last, we evaluate the security features, communication costs, and computation costs of the proposed scheme with related schemes, including Wu et al.'s.

### 1.2. Organization

In Section 2, we introduce studies related to cloud-based IoT, IoT, and cloud computing. We present the system model and adversary model used in our proposed scheme in Section 3. Following that, we discuss Wu et al.'s scheme in Section 4. We then delve into the vulnerabilities we identified in Wu et al.'s scheme in Section 5. In Section 6, we introduce our proposed scheme, and in Section 7, we provide security analyses using tools such as BAN logic, and RoR model. Performance analyses, including security features, communication, and computation costs, are presented in Section 8. Finally, in Section 9, we conclude our paper and outline future plans.

## 2. Related Works

When providing services to users over the internet, application security is crucial in gaining user trust. To access various services, including storage services provided by cloud service providers, the environments should be well prepared to handle various attacks and security threats that may exist. Furthermore, in IoT environments, lightweight protocol computations are essential to provide users with a seamless real-time service anytime, anywhere. In the following sections, we will review the authentication protocols in the existing cloud-based IoT environments.

In 2019, Schouqi et al. [12] introduced an authentication protocol for IoT built on Nikooghadam et al.'s [13] protocol. The protocol of Nikooghadam et al. was developed as a responses to issues with the authentication protocol proposed by Kumari et al. [14]. However, Nikooghadam et al.'s scheme has already been analyzed by researchers in the field, including Limbasiya et al., Chandrakar-Om, and Sharma-Kalra [15–17]. These researchers raised concerns about its security, highlighting vulnerabilities to various attacks such as password-guessing, insiders, and modification attacks. They also indicated that the protocol lacked forward secrecy and did not provide session key verification and a biometric update phase. The author of the new scheme reviewed the security issues known in Nikooghadam et al.'s protocol and proposed enhancements based on these findings.

Prosanta and Biplab (Prosanta-Biplab) [18] proposed lightweight two-factor authentication scheme for IoT devices in 2019. They argued that two-factor authentication schemes that use a passwords and smartcards, often vulnerable to physical attacks. To overcome these security issues they suggested physically uncloneable functions(PUF) as an authentication factor for IoT devices. However, in 2020, Siddiqui et al. [19] demonstrated that the scheme is vulnerable to man-in-the-middle, impersonations, session- hijacking and conventional and differential template attacks.

In 2019, Zhou et al. [20] presented a lightweight two-factor authentication scheme for IoT devices available in the cloud environments. In the same year, Rafael et al. [21] indicated that Zhou et al.'s scheme has several security issues. Rafael et al. demonstrated that Zhou et al.'s scheme failed to provide mutual authentication, was unsuccessful in protecting the secret key, and was vulnerable to various attacks, including insider attacks and man-in-the-middle attacks.

In 2020, Alzahrani et al. [22] presented an authentication protocol for IoT environments based on self-certified public keys and elliptic curve cryptography (ECC). Alzahrani conducted research on protocols proposed by Islam-Biswas [23] and Mandal et al. [24], highlighting their failure to ensure user anonymity and vulnerability to impersonation attacks. Therefore, the author developed a protocol that guarantees anonymity among connected devices and addresses security vulnerabilities. However, this scheme does not guarantee security against physical attacks.

Chen et al. [25] proposed a lightweight user authentication and key-agreement scheme for IoT. Chen et al. utilized XOR operations, hash functions, and elliptical multiplication. Lee et al. [26] indicated that Chen et al.'s scheme did not provide a steal-resistant smart-card offline password, offline identity guessing and reply attack. Subsequently, in 2020, Ye et al. [27] proposed an authentication and key agreement scheme for IoT-based cloud computing environments by advancing the protocol developed by He et al. [28]. Ye et al. addressed various security issues in the scheme proposed by He et al, such as failure to resist insider attacks, offline password-guessing, user impersonation, and potential DoS attacks.

Table 1, summarizes cryptographic technologies and limitations of various authentication schemes related to IoT, cloud-based IoT, and cloud computing environments. Related papers propose various protocols to provide users with secure and fast services in the CloudIoT environment. However, there are still vulnerabilities and challenges in fully supporting security features, as some attacks persist. Additionally, methods using symmetric keys like ECC may incur higher computation costs in IoT environments. Therefore, our goal is to design a lightweight protocol tailored for IoT environments using XOR and to achieve higher security in our scheme.

**Table 1.** Authentication scheme overview.

| **Schemes** | **Cryptographic Technologies** | **Limitaion** |
|---|---|---|
| Islam-Biswas [23] | - ECC<br>- Self-certified public keys | - Cannot provide anonymity<br>- Vulnerable to reply and clogging attacks |
| He et al. [28] | - Asymmetric cryptography | - Vulnerable to insider attacks, offline password-guessing,<br>user impersonation attacks, and DoS attacks |
| Chen at al. [25] | - XOR operation<br>- Hash function<br>- Elliptic multiplication | - Vulnerable to stolen smartcard, offline password-guessing,<br>offline identity guessing, and reply attacks |
| Prosanta-Biplab [18] | - PUF<br>- Fuzzy extractor | -Vulnerable to man-in-the-middle attacks, impersonation attacks,<br>session key hijaking, conventional and differential template attacks |
| Zhou et al. [20] | - XOR operation<br>- Hash function | - Cannot provide mutual authentication<br>- Vulnerable to insider attacks and man-in-the-middle attacks |
| Nikooghadam et al. [13] | - XOR operation<br>- Hash function | - Vulnerable to reply attacks, privileged insider attacks,<br>offline password-guessing, known key temporary information attacks,<br>server spoofing attacks and impersonation attcks |
| Tsai-Lo [29] | - Single Sign On scheme | - Cannot provide sesssion key security, mutual authentication, and<br>user anonymity<br>- Vulnerable to impersonation attacks |

**Table 1.** *Cont.*

| Schemes | Cryptographic Technologies | Limitaion |
|---|---|---|
| Kumari et al. [30] | - Hash function<br>- Diffie-Hellman | - Cannot provide user unlinkability and anonymity, data confidentiality<br>- Vulnerable to known session-specific temporary information attacks, impersonation attacks, and desynchronization attacks |
| Bhuarya et al. [31] | - ECC | - Cannot provide mutual authentication<br>- Vulnerable to impersonation attacks and man-in-the-middle attacks |

## 3. Preliminaries

### *3.1. System Model*

As shown in Figure 2, an IoT-enable cloud computing environment includes three entities: user, cloud server, and control server. Users can also use cloud computing provided by a cloud server, using IoT-enabled devices. Therefore, the user and cloud server should register and authenticate it through the control server. Finally, the user and cloud server share a session key for communication. The details are as follows

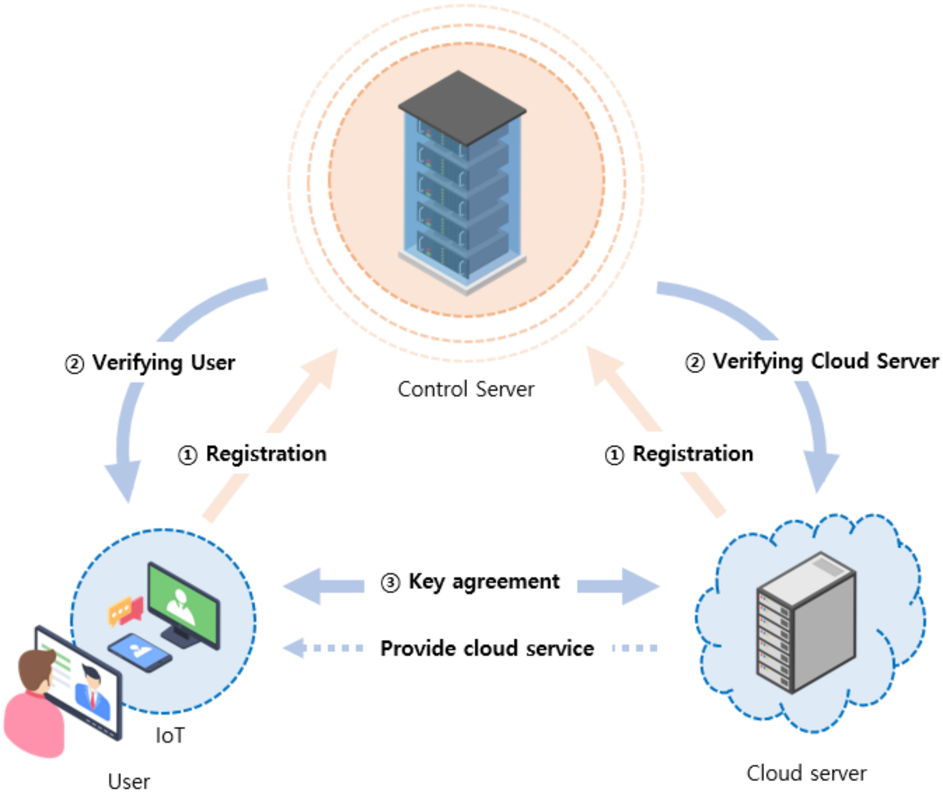


**Figure 2.** System model.

- User ($U_i$): User uses IoT devices with cloud services. Communicates with the cloud servers, then the user should register with the control server. The user can use smart cards and biometric technology to store sensitive information or the user's identity and password. We assumed that the user is an untrusted entity, implying that the user can execute unauthorized or malicious attacks.
- Cloud server ($S_j$): A cloud server provides cloud services to users using IoT devices. To achieve this, the cloud server should be registered with the control server. As a semi-trusted entity, the cloud server can misbehave; however, it cannot directly collude or participate.
- Control server ($CS$): This manages the registration of the user and cloud server, and helps generate the session key for authentication and subsequent communication.

As a semi-trusted entity, the control server can misbehave; however, it cannot directly collude or participate.

### *3.2. Adversary Model*

We employ the widely used "Dolev–Yao (DY) model" [32] to define the capabilities of the adversary. The details are as follows:

- Within the DY model, entities in the IoT environments are considered trustworthy, and the communication channel is also considered insecure. Consequently, the adversary can engage in various actions through the insecure channel, including resending, eavesdropping, blocking, and deleting any messages transmitted.
- The adversary can extract sensitive information through power analysis attacks from stolen user smart cards. Additionally, because the control and cloud servers are semi-trusted entities, the adversary can also extract information from their databases.

Furthermore, the "Canetti–Krawczyk (CK) model" [33] assumes that stronger adversaries can also be adapted to our protocol. The adversaries in the CK model can obtain and use ephemeral values or long-term values, and using those, ephemeral leakage attacks can be performed.

## **4. Revisit of Wu et al.'s Scheme**

### *4.1. Registration Phase*

Before generating a session key for communication, the user and cloud server must go through the registration process via a secure channel. The detailed process is as follows.

#### 4.1.1. User Registration Phase

**Step 1:** The $U_i$ enters $ID_i$, $PW_i$ and imprints $B_i$ on the device. Then, calculates $Gen(B_i) = \sigma_i, \tau_i$, $HPW_i = h(PW_i||\sigma_i)$ and sends $ID_i, HPW_i$ to $CS$ as a registration request message through a secure channel.

**Step 2:** $CS$ checks if $U_i$'s identity is new, and generates a random number $n_i$ to calculate $TID_i = h(ID_i)$, $A_1 = h(ID_{CS}||HPW_i) \oplus (n_i \oplus x)$. Then, stores $\{TID_i, HPW_i\}$ in its database, and stores $\{A_1, ID_{CS}\}$ to smart card $SC$. After that, sends $SC$ to $U_i$ through a secure channel.

**Step 3:** $U_i$ computes $A_2 = h(ID_i||HPW_i)$ and store $\{A_1, A_2, ID_{CS}, Gen(\cdot), Rep(\cdot), \tau_i\}$ in $SC$.

#### 4.1.2. Cloud Server Registration Phase

**Step 1:** $S_j$ selects $SID_j$ and random number $n_j$ and sends $\{SID_j, n_j\}$ as a request message to $CS$ through a secure channel.

**Step 2:** $CS$ checks if $S_j$'s identity is new, and chooses $S_j$'s pseudo identity $QID_j$, computes $A_3 = h(SID_j||x \oplus n_j)$, then stores $\{QID_j, n_j\}$ in its database. Next, $CS$ sends $QID_j, n_j$ to $S_j$ through secure channel.

**Step 3:** $S_j$ computes $A_3^* = A_3 \oplus SID_j$, and stores $\{A_3^*, QID_j\}$.

### *4.2. Login and Authentication Phase*

In this phase, the control server first verifies the identities of the user and the cloud server. If both are confirmed, a shared session key for subsequent communication is generated. The detailed process is as follows and illustrated in Figure 3.

| User $U_m$ | Cloud Server | Control Server $D_n$ |
|---|---|---|
| Input $ID_i, PW_i$, imprint $B_i$<br>Compute $Rep(B_i, \tau_i) = \sigma_i$<br>$HPW_i = h(PW_i\|\|\sigma_i)$<br>$A'_2 = h(ID_i\|\|HPW_i)$<br>Check $A'_2 \stackrel{?}{=} A_2$<br>Generate $r_i, TS_1$<br>Compute<br>$(n_i \oplus x) = A_1 \oplus h(ID_{CS}\|\|HPW_i)$<br>$B_1 = r_i \oplus h(ID_{CS}\|\|HPW_i \oplus SID_j)$<br>$B_2 = SID_j \oplus h(ID)CS\|\|HPW_i)$<br>$B_3 = h(TID_i\|\|ID_{CS}\|\|n_i \oplus x) \oplus HPW_i$ | | |
| $\xrightarrow{M_1 = \{TID_i, A_1, B_1, B_2, B_3, TS_1\}}$ | | |
| | Check $\|TS_1 - TS_c\| \leqq \Delta T$<br>Generate $r_j, TS_2$<br>Compute $A_3 = SID_j \oplus A_3^*$<br>$B_4 = r_j \oplus h(A_3\|\|SID_j)$<br>$B_5 = h(r_j\|\|A_3\|\|SID_j)$ | |
| | $\xrightarrow{M_2 = \{M_1, QID_j, B_4, B_5, TS_2\}}$ | |
| | | Check $\|TS_2 - TS_c\| \leqq \Delta T$<br>According $TID_i$ to find $HPW_i$<br>Compute $SID_j = B2 \oplus h(ID_{CS}\|\|HPW_i)$<br>$r_i = B_1 \oplus h(ID_{CS}\|\|HPW_i \oplus SID_j)$<br>$(n_i \oplus x) = A_1 \oplus h(ID_{CS}\|\|HPW_i)$<br>$B'_3 = h(TID_i\|\|ID_{CS}\|\|n_i \oplus x) \oplus HPW_i$<br>Check $B'_3 \stackrel{?}{=} B_3$<br>According $QID_j$ to find $n_j$<br>Compute $A_3 = h(SID_j\|\|x \oplus n_j)$<br>$r_j = B_4 \oplus h(A_3\|\|SID_j)$<br>$B'_5 = h(r_j\|\|A_3\|\|SID_j)$<br>Check $B'_5 \stackrel{?}{=} B_5$<br>Generate $r_k, TS_3$<br>Compute<br>$SK = h(r_i \oplus HPW_i\|\|r_j\|\|r_k\|\|SID_j)$<br>$B_6 = (r_i \oplus HPW_i) \oplus A_3$<br>$B_7 = h(A_3 \oplus r_j\|\|SID_j) \oplus r_k$<br>$B_8 = h(r_j\|\|r_k\|\|SK\|\|TS_3)$<br>$B_9 = h(n_i \oplus x\|\|SID_j) \oplus r_j$<br>$B_10 = h(HPW_i\|\|r_i) \oplus r_k$<br>$B_11 = h(SK\|\|n_i \oplus x\|\|r_k\|\|r_j)$ |
| | $\xleftarrow{M_3 = \{B_6, B_7, B_8, B_9, B_{10}, B_{11}, TS_3\}}$ | |
| | Check $\|TS_3 - TS_c\| \leqq \Delta T$<br>Compute $(r_i \oplus HPW_i) = B_6 \oplus A_3$<br>$r_k = h(A_3\|\|r_j\|\|SID_j) \oplus B_7$<br>$SK = h(r_i \oplus HPW_i\|\|r_j\|\|r_k\|\|SID_j)$<br>$B'\|8 = h(r_j\|\|r_k\|\|SK\|\|TS_3)$<br>Check $B'_8 \stackrel{?}{=} B_8$ | |
| $\xleftarrow{M_4 = \{B_9, B_{10}, TS_4\}}$ | | |
| Checks $\|TS_4 - TS_c\| \leqq \Delta T$<br>$r_j = h(n_i \oplus x\|\|SID_j) \oplus B_9$<br>$r_k = h(HPW_i\|\|r_i) \oplus B_{10}$<br>$SK = h(r_i \oplus HPW_i\|\|r_j\|\|r_k\|\|SID_j)$<br>$B'_{11} = h(SK\|\|n_i \oplus x\|\|r_k\|\|r_j)$<br>Checks $B'_{11} \stackrel{?}{=} B_{11}$<br>Compute $B_{12} = h(SK\|\|r_j)$ | | |
| $\xrightarrow{M_5 = \{B_{12}\}}$ | | |
| | Compute $B'_{12} = h(SK\|\|r_j)$<br>Check $B'_{12} \stackrel{?}{=} B_{12}$ | |

**Figure 3.** AKA phase of Wu et al.'s scheme.

**Step 1:** $U_i$ enters $ID_i$, $PW_i$ and imprints $B_i$, and calculates $Rep(B_i, \tau_i) = \sigma_i$, $HPW_i = h(PW_i||\sigma_i)$, $A'2 = h(ID_i||HPW_i)$. Then, by confirming $A'_2 \stackrel{?}{=} A2$, $U_i$ can be verified as a legitimate user. If this is valid, $U_i$ selects a random number $r_i$ and timestamp $TS_1$, then calculates $(n_i \oplus x) = A_1 \oplus h(ID_{CS}||HPW_i)$, $B_1 = r_i \oplus h(ID_{CS}||HPW_i \oplus SID_j)$, $B_2 = SID_j \oplus h(ID_{CS}||HPW_i)$, and $B_3 = h(TID_i||ID_{CS}||n_i \oplus x) \oplus HPW_i$. Finally, generates a message $M_1 = \{TID_i, A_1, B_1, B_2, B_3, TS_1\}$ and sends it to $S_j$ via open channel.

**Step 2:** Upon receiving $U_i$'s message, $CS$ confirms timestamp $|TS_1 - TS_c| \leqq \Delta T$. If the timestamp is valid, $S_j$ chooses a random value $r_j$ and timestamp $TS_2$. $S_j$ computes $A_3 = SID_j \oplus A_3^*$ , $B_4 = r_j \oplus h(A_3||SID_j)$, and $B_5 = h(r_j||A_3||SID_j)$. Finally, message $M_2 = \{M_1, QID_j, B_4, B_5, TS_2\}$ is sent through an open channel.

**Step 3:** After receiving the $M_2$, $S_j$ confirms timestamp $|TS_2 - TS_c| \leqq \Delta T$. If the timestamp is successfully verified, $CS$ uses $TID_i$ to find $HPW_i$ and performs the following computations: $SID_j = B_2 \oplus h(ID_{CS}||HPW_i)$, $r_i = B_1 \oplus h(ID_{CS}||HPW_i \oplus SID_j)$, and $B'_3 = h(TID_i||ID_{CS}||n_i \oplus x) \oplus HPW_i$. And by checking $B'_3 \stackrel{?}{=} B_3$, the $CS$ confirms whether $U_i$ is the legitimate user. Next, $CS$ utilizes the value of $QID_j$ to find $n_j$ and then performs the following computations: $A_3 = h(SID_j||x \oplus n_j)$, $r_j = B_4 \oplus h(A_3||SID_j)$, and $B'_5 = h(r_j||A_3||SID_j)$. After checking $B'_5 \stackrel{?}{=} B_5$ is valid, CS then selects $r_k$, $TS_3$, computes $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$, $B_6 = (r_i \oplus HPW_i) \oplus A_3$, $B_7 = h(A_3||r_j||SID_j) \oplus r_k$, $B_8 = h(r_j||r_k||SK||TS_3)$, $(n_i \oplus x) = A_1 \oplus h(ID_{CS}||HPW_i)$, $B_9 = h(n_i \oplus x||SID_j) \oplus r_j$, and $B_{10} = h(HPW_i||r_i) \oplus r_k$, $B_{11} = h(SK||n_i \oplus x||r_k||r_j)$. At last, $CS$ generates message $M_3 = \{B_6, B_7, B_8, B_9, B_{10}, B_{11}, TS_3\}$ and sends to $S_j$ through an open channel.

**Step 4:** Upon receiving $M_3$, $S_j$ checks timestamp $|TS_3 - TS_c| \leqq \Delta T$. If the timestamp is valid, $S_j$ calculates following computations: $(r_i \oplus HPW_i) = B_6 \oplus A_3$, $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$, and $B'_8 = h(r_j||r_k||SK||TS_3)$, and confirms $B'_8 \stackrel{?}{=} B_8$. If it confirms, $S_j$ generates message $M_4 = \{B_9, B_{10}, TS_4\}$ to $U_i$ via open channel.

**Step 5:** $U_i$ verifies timestamp $|TS_4 - TS_c| \leqq \Delta T$. If the timestamp is valid, $U_i$ calculates $r_j = h(n_i \oplus x||SID_j) \oplus B_9$, $r_k = h(HPW_i||r_i) \oplus B_{10}$, $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$, and $B'_{11} = h(SK||n_i \oplus x||r_k||r_j)$ and calculates $B'_{11} \stackrel{?}{=} B_{11}$. If it confirms, $U_i$ computes $B_{12} = h(SK||r_j)$ and generates $M_5 = \{B_{12}\}$ and sends to $S_j$.

**Step 6:** $S_j$ calculates the equation $B'_{12} = h(SK||r_j)$ and then checks $B'_{12} \stackrel{?}{=} B_{12}$. If they match, $S_j$ stores $SK$ for future communication.

## 5. Cryptanalysis of Wu et al.'s Scheme

Following the description of Section 3, adversary $\mathcal{A}$ can obtain important values from the user's smart card by using a power analysis attack. Furthermore, $\mathcal{A}$ can extract parameters from the cloud server and control server itself, because they are considered semi-trusted. With this information, various security attacks, including insider attack, verification table leakage attack, privileged insider attack, user impersonation, and cloud server impersonation, can be executed by $\mathcal{A}$. Details are described below.

### *5.1. Insider Attack*

An adversary $A$, who has undergone the registration process as a legitimate user, can obtain session keys from another user $U_i$'s sessions or impersonate $U_i$. The detailed process is as follows:

**Step 1:** After completing the registration process, $A$ obtains $B_6$ of $M_3$ during their AKA process. Subsequently, $A$ calculates $A_3$ of $S_j$ using their own $HPW_a$ and $r_a$.

**Step 2:** In another user $U_i$'s session, $A$ obtains message $M_2$ and uses $B_4$ and the previously acquired $A_3$ to deduce $r_j$.

**Step 3:** From $B_6$ of $M_3$, $A$ calculates user $U_i$'s $r_i$ and $HPW_i$, and from $B_7$, $A$ calculates $r_k$.

**Step 4:** Using the computed values, A can generate the session key $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$ for another user $U_i$ and potentially disclose or exploit it.

Therefore, Wu et al.'s scheme cannot resist insider attacks.

*5.2. Verification Table Leakage Attack*

If $\mathcal{A}$ extracts verification table of cloud server, $\mathcal{A}$ can disclose session key. The following procedures are below:

**Step 1:** $\mathcal{A}$ extracts the verification table to take $\{A_3^*, QID_j\}$ from $S_j$. And also intercept message $M_2 = \{M_1, QID, B_4, B_5, TS_2\}$ transmitted in public channel.

**Step 2:** $\mathcal{A}$ calculates $A_3 = SID_j \oplus A_3^*$, and $r_j = B_4 \oplus h(A_3||SID_j)$to extract $A_3$ and $r_j$.

**Step 3:** $\mathcal{A}$ takes message $M_3 = \{B_6, B_7, B_8, B_9, B_{10}, B_{11}, TS_3\}$.

**Step 4:** $\mathcal{A}$ computes $(r_i \oplus HPW_i) = B6 \oplus A_3$, and $r_k = h(A_3||r_j||SID_j) \oplus B_7$. In addition by calculating $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$, A can generate a session key to disclose or exploit it.

Therefore, Wu et al.'s scheme cannot resist verification table leakage attacks.

*5.3. Privileged Insider Attack*

A privileged insider can take important information like $\{ID_j, HPW_j\}$ from the registration message and values stored in the user's smart card such as $\{A_1, A_2, ID_{cs}, Gen(), Rep(), \}$. Through support from this privileged insider, a malicious $\mathcal{A}$ can generate a session key through the following:

**Step 1:** $\mathcal{A}$ computes $SID_j = B_2 \oplus h(ID_{CS}||HPW_i)$, and $r_i = B_1 \oplus h(ID_{CS}||HPW_i \oplus SID_j)$, $(n_i \oplus x) = A_1 \oplus h(ID_{CS}||HPW_i)$; therefore, $\mathcal{A}$ can extract parameters $SID_j$, $r_i$, and $(n_i \oplus x)$.

**Step 2:** $\mathcal{A}$ intercepts message $M_4 = \{B_9, B_{10}, TS_4\}$.

**Step 3:** $\mathcal{A}$ calculates $r_j = h(n_i \oplus x||SID_j) \oplus B_9$ , and $r_k = h(HPW_i||r_i) \oplus B_{10}$. Hence, $\mathcal{A}$ can compute session key $SK = h(r_i \oplus HPW_i||r_j||r_k||SID_j)$ and disclose it.

Thus, Wu et al.'s scheme is insecure against privileged insider attacks.

*5.4. Impersonation*

When $\mathcal{A}$ obtains the table information $\{k_n, SID_n\}$ of the control center, $\mathcal{A}$ can calculate $SK_{nm} = h(SID_m||SID_n||SID_c||A_8)$.

(1) User impersonation: If the privileged insider described in Section 5.3 generates random number $r_i$ and time stamp $TS_1$, $\mathcal{A}$ can forge message $M_1 = \{TID_i, A_1, B_1, B_2, B_3, TS_1\}$. In addition, by $\mathcal{A}$ to take message $M_4 = \{B_9, B_{10}, TS_4\}$ from an unsecured public channel, $\mathcal{A}$ can generate session key and $rj$. Thus, $\mathcal{A}$ can send message $M_5 = \{B12\}$ impersonates user.

(2) Cloud server impersonation: According to the previous verification table attack in Section 5.2, $\mathcal{A}$ generates random number $r_j$ and time stamp $TS_2$, and $\mathcal{A}$ can send $M_2 = \{M_1, QID_j, B_4, B_5, TS_2\}$. Second, $\mathcal{A}$ can generate $M_4 = \{B_9, B_{10}, TS_4\}$ after intercept message $M_3 = \{B_6, B_7, B_8, B_9, B_{10}, B_{11}, TS_3\}$. Hence $\mathcal{A}$ can impersonate cloud server.

Therefore, Wu et al.'s scheme cannot resist user and cloud impersonation attack.

*5.5. Lack of Untraceability*

If an attacker $A$ continues to eavesdrop on $M_1 = \{TID_i, A_1, B_1, B_2, B_3, TS_1\}$ and compares the value of $TID_i$ contained in $M_1$, $A$ can track the user $U_i$. The reason is that the pseudo identity of $U_i$, $TID_i$, is a fixed value, and an attacker can easily obtain it through eavesdropping on the message. Indeed, by verifying whether the value of $TID_i$ matches the values from previous or subsequent communications, $A$ can detect the user. In conclusion, Wu et al.'s scheme lacks anonymity and untraceability.

### *5.6. Impossibility of Offline Password Update*

In the user registration phase in Wu et al.'s scheme, the value of $HPW_i$ is created by concatenating the user's password $PW_i$ with their biometric information $\sigma_i$. Additionally, this $HPW_i$ is transmitted to the control server $CS$ and undergoes the operation $A_1 = h(ID_{CS}||HPW_i) \oplus (n_i \oplus x)$, and stored in the $CS$'s database as $A_1$. However, this design leads to a problem where users must communicate with the $CS$ to update the $A_1$ value stored in the $CS$ if they wish to change their password, because $CS$ cannot create the $HPW_i$ on its own. Consequently, Wu et al.'s scheme does not support offline password updates.

## 6. Proposed Protocol

### *6.1. Registration Phase*

Before generating a session key for communication, the user and the cloud server must go through the registration process with the control server via a secure channel. In this phase, users register the information, such as identity, password, and biometrics, with the control server. The detailed process is as follows and illustrated in Figure 4.

| **User $U_i$** | | **Control Server $CS$** |
|---|---|---|
| Select $ID_i, PW_i, B_i$<br>$Gen(B_i) = (\sigma_i, \tau_i)$ | | |
| | $\xrightarrow{\{ID_i\}}$ | |
| | | Check $U_i$'s identity, if it is new<br>Select random $n_i$<br>Compute $SID_i = h(ID_i \oplus x_{cs})$<br>$k_i = h(SID_i\|\|x_{cs}\|\|n_i)$<br>$SID_i^* = SID_i \oplus h(x_{cs}\|\|n_i)$<br>$PID_i = h(SID_i\|\|n_i)$<br>Saves $\{PID_i, SID_i^*, n_i\}$ in database |
| | $\xleftarrow{\{PID_i, ID_{CS}, k_i, SID_i\}}$ | |
| $RPW_i = h(ID_i\|\|PW_i\|\|\sigma_i)$<br>$A_1 = k_i \oplus h(ID_i\|\|HP_i)$<br>$A_2 = SID_i \oplus h(\sigma_i\|\|PW_i)$<br>Stores $\{PID_i, ID_{CS}, RPW_i, A_1, A_2, Gen(\cdot), Rep(\cdot), \tau_i\}$ | | |

**Figure 4.** User registration phase of proposed scheme.

#### 6.1.1. User Registration Phase

**Step 1:** The $U_i$ enters $ID_i$, $PW_i$ and imprints $B_i$ on the device. Then, calculates $Gen(B_i) = \sigma_i$ and sends $ID_i$ to $CS$ as a registration request message through a secure channel.

**Step 2:** $CS$ checks if $U_i$'s identity is new, and generates a random number $n_i$ to calculate $SID_i = h(ID_i \oplus x_{cs}, k_i = h(SID_i||x_{cs}||n_i)$. $SID_i^* = SID_i \oplus h(x_{cs}||n_i)$ and $PID_i = h(SID_i||n_i)$. Then, stores $\{PID_i, SID_i^*, n_i\}$ in its database, and sends $\{PID_i, ID_{cs}, k_i, SID_i\}$ to $U_i$ through secure channel.

**Step 3:** $U_i$ computes $RPW_i = h(ID_i||PW_i||\sigma_i)$, $A_1 = k_i \oplus h(ID_i||HP_i)$, and $A_2 = SID_i \oplus h(\sigma_i||PW_i)$. Then, store $\{PID_i, ID_{CS}, RPW_i, A_1, A_2, Gen(\cdot), Rep(\cdot), \tau_i\}$ in $SC$.

#### 6.1.2. Cloud Server Registration Phase

In this phase, cloud servers register the information with the control server. The detailed process is as follows and illustrated in Figure 5.

**Step 1:** $S_j$ selects $SID_j$ and sends $\{ID_j\}$ as a request message to $CS$ through a secure channel.

**Step 2:** $CS$ checks if $S_j$'s identity is new, and chooses random number $n_j$, computes $k_j = h(ID_j||n_j||x_{cs})$. Then, stores $\{ID_j, n_j\}$ in its database. Next, $CS$ sends $k_j$ to $S_j$ through secure channel.

**Step 3:** $S_j$ computes $A_3 = k_j \oplus x_j$, and stores $\{A_3\}$.

| **Cloud Server $S_j$** | | **Control Server $CS$** |
|---|---|---|
| Select $ID_j$ | $\xrightarrow{\{ID_j\}}$ | |
| | | Check $S_j$'s identity, if it is new<br>Select random $n_j$<br>Compute $k_j = h(ID_j \| \| n_j \| \| x_{cs})$<br>Saves $\{ID_j, n_j\}$ in database |
| | $\xleftarrow{\{k_j\}}$ | |
| $A3 = k_j \oplus x_j$<br>Stores $\{A_3\}$ | | |

**Figure 5.** Cloud server registration phase of proposed scheme.

### 6.2. Login and Authentication Phase

In this phase, the control server first verifies the identities of the user and the cloud server. If both are confirmed, a shared session key for subsequent communication is generated. The detailed process is as follows and illustrated in Figure 6.

**Step 1:** $U_i$ enters $ID_i$, $PW_i$, imprints $B_i$, and calculates $Rep(B_i, \tau_i) = \sigma_i$, $RPW_i' = h(ID - I||PW_i||\sigma_i)$, $k_i = A_1 \oplus (ID_i||PW_i)$ and $SID_i = A_q \oplus (\sigma_i||PW_i)$. Then, by confirming $RPW_i' \stackrel{?}{=} RPW_i$, $U_i$ can be verified as a legitimate user. If this is valid, $U_i$ selects a random number $r_i$ and timestamp $TS_1$ then calculates $B_1 = r_i \oplus h(SID_i||ID_{CS}||k_i)$, $B_2 = ID_j \oplus h(ID_{CS}||SID_i||r_i)$ and $V_1 = h(ID_{CS}||TS_1||SID_i||k_i||r_i)$. Finally, it generates a message $M_1 = \{PID_i, B_1, B_2, V_1, TS_1\}$ and sends it to $S_j$ via open channel.

**Step 2:** Upon receiving $U_i$'s message, $CS$ confirms timestamp $|TS_1 - TS_c| \leqq \Delta T$. If thetimestamp is valid, $S_j$ chooses a random value $r_j$ and timestamp $TS_2$. $S_j$ computes $k_j = A_3 \oplus x_j$ , $B_3 = r_j \oplus h(k_j||ID_j)$, and $V_2 = h(k_j||TS_2||ID_j||r_j)$. Finally, message $M_2 = \{M_1, B_3, V_2, TS_2\}$ is sent through an open channel.

**Step 3:** After receiving the $M_2$, $S_j$ confirms timestamp $|TS_2 - TS_c| \leqq \Delta T$. If the timestamp is successfully verified, $CS$ uses $PID_i$ to find $\{SID_i^*, n_i\}$ and performs the following computations: $SID_i = SID_i^* \oplus (x_{CS}||n_i)$, $k_i = h(SID_i||x_{CS}||n_i)$, $r_i = B_1 \oplus h(SID_i||ID_{CS}||k_i)$ and $V_1' = h(ID_{CS}||TS_1||SID_i||k_i||r_i)$. And by checking $V_1' \stackrel{?}{=} V_1$, the $CS$ confirms whether $U_i$ is the legitimate user.

**Step 4:** Next, $CS$ calculates $ID_j = B_2 \oplus h(ID_{CS}||SID_i||r_i)$ and utilizes the value of $ID_j$ to find $n_j$. Then, it performs the following computations: $k_j = h(ID_j||n_j||x_{CS})$, $r_j = B_3 \oplus h(k_j||ID_j)$, and $V_2' = h(k_j||TS_2||ID_j||r_j)$. Subsequently, it checks if $V_2' \stackrel{?}{=} V_2$ is valid.

**Step 5:** CS then selects $r_k$, $TS_3$, computes $SID_j = ID_j \oplus h(k_j||r_k)$, $C_1 = h(SID_i \oplus SID_j \oplus ID_{CS})$, $PID_i^* = PID_i \oplus h(PID_i||r_k||r_i)$. Next, it updates the old $PID_i$ to $PID_i^*$.

**Step 6:** $B_6 = (r_k||r_i) \oplus h(r_j||k_j)$, $B_7 = C_1 \oplus h(k_j||r_k)$, $B_8 = (r_j||r_k) \oplus h(SID_i||ID_{CS}||r_i)$, $B_9 = SID_j \oplus h(SID_i||r_k)$, $V_3 = h(r_j||r_k||C_1||ID_j||TS_3)$ and $V_4 = h(SID_i||r_j||r_k||C_1||TS_3)$. At last, $CS$ generates message $M_3 = \{B_6, B_7, B_8, B_9, V_3, V_4, TS_3\}$ and sends to $S_j$ through an open channel.

**Step 7:** Upon receiving $M_3$, $S_j$ checks timestamp $|TS_3 - TS_c| \leqq \Delta T$. If the timestamp is valid, $S_j$ calculates following computations: $(r_k||r_i) = B6oplush(r_j||k_j)$, $C_1 = B_7 \oplus h(k_j||r_k)$, $SK = h(C_1||r_i||r_j||r_k)$, and $V_3' = h(r_j||r_k||C_1||ID_j||TS_3)$. Subsequently, it confirms $V_3' \stackrel{?}{=} V_3$. If it confirms, $S_j$ generates message $M_4 = \{B_8, B_9, V_4, TS_4\}$ to $U_i$ via an open channel.

**Step 8:** $U_i$ verifies timestamp $|TS_4 - TS_c| \leqq \Delta T$. If the timestamp is valid, $U_i$ calculates $(r_j||r_k) = B_8 \oplus h(SID_i||ID_{CS}||r_i)$, $SID_j = B_9 \oplus h(SID_i||r_k)$, $C_1 = h(SID_i \oplus SID_j \oplus ID_{CS})$, $SK = h(C_1||r_i||r_j||r_k)$, and calculates $V_4' = h(SID_i||r_j||r_k||C_1||TS_3)$ to check $B_{11}' \stackrel{?}{=} B_{11}$. If it confirms, $U_i$ computes $PID_i^* = PID_i \oplus h(PID_i||r_k||r_i)$ and update old $PID_i$ to $PID_i^*$.

| **User $U_i$** | **Cloud Server $S_j$** | **Control Server $CS$** |
|---|---|---|
| Inputs $ID_i, PW_i$, imprint $B_i$<br>Compute $Rep(B_i, \tau_i) = \sigma_i$<br>$RPW_i' = h(ID_i\|\|PW_i\|\|\sigma_i)$<br>$k_i = A_1 \oplus (ID_i\|\|PW_i)$<br>$SID_i = A_2 \oplus (\sigma_i\|\|PW_i)$<br>Checks $RPW_i' \stackrel{?}{=} RPW_i$<br>Generate random $r_i$, timestamp $TS_1$<br>$B_1 = r_i \oplus h(SID_i\|\|ID_{CS}\|\|k_i)$<br>$B_2 = ID_j \oplus h(ID_{CS}\|\|SID_i\|\|r_i)$<br>$V_1 = h(ID_{CS}\|\|TS_1\|\|SID_i\|\|k_i\|\|r_i)$ | | |
| $\xrightarrow{M_1 = \{PID_i, B_1, B_2, V_1, TS_1\}}$ | | |
| | Check $\|TS_1 - TS_c\| \leqq \Delta T$<br>Generate random $r_j$, timestamp $TS_2$<br>$k_j = A_3 \oplus x_j$<br>$B_3 = r_j \oplus h(k_j\|\|ID_j)$<br>$V_2 = h(k_j\|\|TS_2\|\|ID_j\|\|r_j)$ | |
| | $\xrightarrow{M_2 = \{M_1, B_3, V_2, TS_2\}}$ | |
| | | Check $\|TS_2 - TS_c\| \leqq \Delta T$<br>According $PID_i$ to find $\{SID_i^*, n_i\}$<br>$SID_i = SID_i^* \oplus (x_{CS}\|\|n_i)$<br>$k_i = h(SID_i\|\|x_{CS}\|\|n_i)$<br>$r_i = B_1 \oplus h(SID_i\|\|ID_{CS}\|\|k_i)$<br>$V_1' = h(ID_{CS}\|\|TS_1\|\|SID_i\|\|k_i\|\|r_i)$<br>Check $V_1' \stackrel{?}{=} V_1$<br><br>$ID_j = B_2 \oplus h(ID_{CS}\|\|SID_i\|\|r_i)$<br>According $ID_j$ to find $\{n_j\}$<br>$k_j = h(ID_j\|\|n_j\|\|x_{CS})$<br>$r_j = B_3 \oplus h(k_j\|\|ID_j)$<br>$V_2' = h(k_j\|\|TS_2\|\|ID_j\|\|r_j)$<br>Check $V_2' \stackrel{?}{=} V_2$<br><br>Generate random $r_k$, timestamp $TS_3$<br>$SID_j = ID_j \oplus h(k_j\|\|r_k)$<br>$C_1 = h(SID_i \oplus SID_j \oplus ID_{CS})$<br>$PID_i^* = PID_i \oplus h(PID_i\|\|r_k\|\|r_i)$<br>Update $PID_i$ to $PID_i^*$<br><br>$B_6 = (r_k\|\|r_i) \oplus h(r_j\|\|k_j)$<br>$B_7 = C_1 \oplus h(k_j\|\|r_k)$<br>$B_8 = (r_j\|\|r_k) \oplus h(SID_i\|\|ID_{CS}\|\|r_i)$<br>$B_9 = SID_j \oplus h(SID_i\|\|r_k)$<br>$V_3 = h(r_j\|\|r_k\|\|C_1\|\|ID_j\|\|TS_3)$<br>$V_4 = h(SID_i\|\|r_j\|\|r_k\|\|C_1\|\|TS_3)$ |
| | $\xleftarrow{M_3 = \{B_6, B_7, B_8, B_9, V_3, V_4, TS_3\}}$ | |
| | Check $\|TS_3 - TS_c\| \leqq \Delta T$<br>$(r_k\|\|r_i) = B_6 \oplus h(r_j\|\|k_j)$<br>$C_1 = B_7 \oplus h(k_j\|\|r_k)$<br>$SK = h(C_1\|\|r_i\|\|r_j\|\|r_k)$<br>$V_3' = h(r_j\|\|r_k\|\|C_1\|\|ID_j\|\|TS_3)$<br>Check $V_3' \stackrel{?}{=} V_3$ | |
| $\xleftarrow{M_4 = \{B_8, B_9, V_4, TS_4\}}$ | | |
| Checks $\|TS_4 - TS_c\| \leqq \Delta T$<br>$(r_j\|\|r_k) = B_8 \oplus h(SID_i\|\|ID_{CS}\|\|r_i)$<br>$SID_j = B_9 \oplus h(SID_i\|\|r_k)$<br>$C_1 = h(SID_i \oplus SID_j \oplus ID_CS)$<br>$SK = h(C_1\|\|r_i\|\|r_j\|\|r_k)$<br>$V_4' = h(SID_i\|\|r_j\|\|r_k\|\|C_1\|\|TS_3)$<br>Checks $V_4' \stackrel{?}{=} V_4$<br>$PID_i^* = PID_i \oplus h(PID_i\|\|r_k\|\|r_i)$<br>Update $PID_i$ to $PID_i^*$ | | |

**Figure 6.** AKA phase of proposed scheme.

### 6.3. Offline Password and Biometric Template Update

In this phase, an authenticated user $U$ can locally change their password and biometrics without a connection to CS. $U$ must perform the login process on the IoT device before updating data offline. A logged-in user can update their password or biometric template. The detailed process is as follows and illustrated in Figure 7.

**Step 1:** $U_i$ enters $ID_i$, $PW_i$ and imprint $B_i$ on the device. Compute $Rep(Bio_i, \tau_i) = \sigma_i$ and check $RPW_i' = h(ID_i||PW_i||\sigma_i)$ for login phase and confirm user.

**Step 2:** Then, ask $U_i$ to change password and biometric data. $U_i$ select new password $PW_i^{new}$, and compute $RPW_i^{new} = h(ID_i||PW_i^{new}||\sigma_i)$, $A_1^{new} = k_i \oplus h(ID_i||PW_i^{new})$ and $A_2^{new} = SID_i \oplus h(\oplus_i||PW_i^{new})$. Subsequently, update $RPW_i$, $A_1$, and $A_2$ with new data to change the password.

**Step 3:** Compute $Rep(Bi_i, \tau_i) = \sigma_i^{new}$, $RPW_i^{new} = h(ID||PW_i||\sigma_i^{new})$ and $A_2^{new} = SID_i \oplus h(\sigma_i^{new}||PW_i)$. Subsequently, update $RPW_i^{new}$ and $A_2^{new}$ with new data to change the biometric template.

| **User $U_i$** | | **Smart Card $SC$** |
|---|---|---|
| Inputs $ID_i, PW_i$ and<br>Imprint $B_i$ to $SC$ | | |
| | $\xrightarrow{\{ID_i, PW_i, B_i\}}$ | |
| | | Compute $Rep(B_i, \tau_i) = \sigma_i$<br>$RPW_i' = h(ID_i\|\|PW_i\|\|\sigma_i)$<br>If $RPW_i' \stackrel{?}{=} RPW_i$, login is successful<br>Ask $U_i$ to change password<br>and biometric data |
| Select $PW_i^{new}$ | | |
| | $\xrightarrow{\{PW_i^{new}\}}$ | |
| | | $RPW_i^{new} = h(ID\|\|PW_i^{new}\|\|\sigma_i)$<br>$A_1^{new} = k_i \oplus h(ID_i\|\|PW_i^{new})$<br>$A_2^{new} = SID_i \oplus h(\sigma_i\|\|PW_i^{new})$<br>Update $RPW_i \rightarrow RPW_i^{new}$<br>$A_1 \rightarrow A_1^{new}$<br>$A_2 \rightarrow A_2^{new}$ |
| | | $Rep(Bi_i, \tau_i) = \sigma_i^{new}$<br>$RPW_i^{new} = h(ID\|\|PW_i\|\|\sigma_i^{new})$<br>$A_2^{new} = SID_i \oplus h(\sigma_i^{new}\|\|PW_i)$<br>Update $RPW_i \rightarrow RPW_i^{new}$<br>$A_2 \rightarrow A_2^{new}$ |

**Figure 7.** Offline password and biometric template update of proposed scheme.

## 7. Security Analysis

### 7.1. ROR Model

In this section, we conduct an analysis of session key security using the ROR model [34]. To apply the proposed protocol to the ROR model, we first define participants, especially $U_{US}^{i_1}$, $U_{SJ}^{i_2}$, and $U_{CS}^{i_3}$ as user, cloud server, and control server, respectively. Note that $i_k$ ($k = 1, 2, 3$) is an instance for each participant. In ROR model, the adversary can eavesdrop, delete, intercept, and send messages through the public channel. Moreover, the adversary can extract secret parameters from the user $U_{US}^{i_1}$. These actions of the adversary can be defined as queries in the ROR model.

- $EX(U_{US}^{i_1}, U_{SJ}^{i_2}, U_{CS}^{i_3})$: This query is an eavesdropping attack that the adversary can obtain messages transmitted via a public channel. Thus, this query can be defined as a passive attack.

- $CoUD(U_{US}^{i_1})$: In this query, the adversary extracts secret parameters using the smart device of $U_{US}^{i_1}$. Therefore, we can define the query $CoUD$ is an active attack.
- $Sn(U_p^i)$: The adversary sends messages to legal participants through open channels. This query is an active attack.
- $Ts(U_p^i)$: In this query, the adversary flips an unbiased coin. When the result of the flipped coin is 0, the session key is not fresh. When the result of the flipped coin is 1, we can demonstrate that the session key is fresh. Otherwise, the result outputs $NULL$ ($\perp$).

**Theorem 1.** *We take a definition of $P_{AD}$, $HA$, $q_{HA}$, and $q_{Sn}$ as the possibility of breaking session key, range space of hash function, number of hash functions, and number of send queries, respectively. Moreover, we define that s and C are the Zipf's parameters [35]. From that, the adversary tries to reveal the session key of the proposed protocol in polynomial time. Following [36–38], the ROR model analysis of the proposed protocol is composed of four games ($GAME_m$, $m = 0, 1, 2, 3$) and the winning possibility of the adversary is $PW_{GAME_m}$ for each game $GAME_m$.*

$$P_{AD} \leq \frac{q_{HA}^2}{|HA|} + 2\{Cq_S^{Sn}\} \tag{1}$$

- *$GAME_0$: In this game, the adversary has no knowledge about the session key. Thus, the adversary picks a random bit B.*
$$P_{AD} = |2PW_{GAME_0} - 1| \tag{2}$$
- *$GAME_1$: The adversary conducts EX query to collect the messages transmitted via public channels. Thus, the adversary obtains $\{PID_i, B_1, B_2, V_1, TS_1\}$, $\{M_1, B_3, V_2\}$, $\{B_6, B_7, B_8, V_3, V_4, TS_3\}$, and $\{B_8, V_5, TS_4\}$. After that, the adversary flips an unbiased coin to execute the Ts query. However, the adversary has no knowledge of the session key $SK = h(C_1 \parallel r_i \parallel r_j \parallel r_k)$ because it is composed of random numbers $r_i$, $r_j$ and $r_k$ and masked in the hash functions. For these reasons, the adversary can obtain the following:*
$$PW_{GAME_1} = PW_{GAME_1} \tag{3}$$
- *$GAME_2$: The adversary conducts $HA$ and $Sn$ queries to reveal session key in this game. However, the session key is composed of fresh random numbers and a cryptographic hash function. Therefore, the adversary cannot make hash collisions to calculate the session key. Applying the birthday paradox [39], we obtain the following:*
$$|PW_{GAME_2} - PW_{GAME_1}| \leq \frac{q_{HA}^2}{|HA|} \tag{4}$$
- *$GAME_3$: In the last game, the adversary conducts $CoUD$ query to obtain the secret parameters $\{PID_i, ID_{CS}, RPW_i, A_1, A_2, Gen(.), Rep(.), \tau_i\}$. However, the adversary cannot decrypt the secret parameters because these parameters are encrypted using the identity $ID_i$, password $PW_i$, and biometrics $B_i$. Since simultaneously guessing $ID_i$, $PW_i$, and $B_i$ is a computationally infeasible task, the adversary has no advantage in this game. We obtain the following using Zipf's law [35].*
$$|PW_{GAME_3} - PW_{GAME_2}| \leq Cq_S^{Sn} \tag{5}$$
*When all the games end, the adversary becomes a random bit B.*
$$PW_{GAME_3} = \frac{1}{2} \tag{6}$$
*We can calculate (7) utilizing (2) and (3).*
$$\frac{1}{2}P_{AD} = |PW_{GAME_0} - \frac{1}{2}| = |PW_{GAME_3} - \frac{1}{2}| \tag{7}$$

*Then, we use (6) and (7) to obtain (8).*

$$\frac{1}{2}P_{AD} = |PW_{GAME_1} - PW_{GAME_3}| \tag{8}$$

*We calculate (9) utilizing the triangular inequality.*

$$\begin{aligned}\frac{1}{2}P_{AD} &= |PW_{GAME_1} - PW_{GAME_3}| \\ &\leq |PW_{GAME_1} - PW_{GAME_2}| \\ &+|PW_{GAME_2} - PW_{GAME_3}| \\ &\leq \frac{q_{HA}^2}{2|HA|} + Cq_S^{Sn}\end{aligned} \tag{9}$$

*We calculate (10) multiplying (9) by 2.*

$$P_{AD} \leq \frac{q_{HA}^2}{|HA|} + 2\{Cq_S^{Sn}\} \tag{10}$$

*We obtain the inEquation (10) which is the same as (1). It means that the adversary cannot distinguish random nonce and the session key using various security attacks, such as EX, CoUD, and Sn. Thus, we can prove the session key security of the proposed protocol.*

### 7.2. BAN Logic

We analyze the mutual authentication of the proposed protocol using BAN logic [40]. Following [41–43], we define basic notations and descriptions of BAN logic in Table 2.

**Table 2.** Basic notations and decriptions.

| Notation | Description |
|---|---|
| $A_i, A_j$ | Principals |
| $SK$ | Session key |
| $T_1, T_2$ | Statements |
| $A_i\| \equiv T_1$ | $A_i$ **believes** $T_1$ |
| $A_1\| \sim T_1$ | $A_i$ once **said** $T_1$ |
| $A_i \mapsto T_1$ | $A_i$ **controls** $T_1$ |
| $A_i \lhd T_1$ | $A_i$ **receives** $T_1$ |
| $\#T_1$ | $T_1$ is **fresh** |
| $\{T_1\}_S$ | $T_1$ is **encrypted** with $S$ |
| $A_i \overset{SH}{\longleftrightarrow} A_j$ | $A_i$ and $A_j$ have a shared key $SH$ |

#### 7.2.1. Rules

In BAN logic, there are five rules, such as "Message meaning rule (MMR)", "Nonce verification rule (NVR)", "Jurisdiction rule (JR)", "Belief rule (BR)", and "Freshness rule (FR)".

1. Message meaning rule (MMR):

$$\frac{A_i \Big| \equiv A_i \overset{SH}{\leftrightarrow} A_j, \quad A_i \lhd \{T_1\}_{SH}}{A_i| \equiv A_j| \sim T_1}$$

2. Nonce verification rule (NVR):

$$\frac{A_i| \equiv \#(T_1), \quad A_i| \equiv A_j \Big| \sim T_1}{A_i| \equiv A_j| \equiv T_1}$$

3. Jurisdiction rule (JR):

$$\frac{A_i| \equiv A_j \Rightarrow T_1, \quad A_i| \equiv A_j| \equiv T_1}{A_i \Big| \equiv T_1}$$

4. Belief rule (BR):

$$\frac{A_i \Big| \equiv (T_1, T_2)}{A_i \Big| \equiv T_1}$$

5. Freshness rule (FR):

$$\frac{A_i \Big| \equiv \#(T_1)}{A_i \Big| \equiv \#(T_1, T_2)}$$

#### 7.2.2. Goals

In our protocol, each participant authenticate the communication partner by establishing session key $SK$. Thus, goals of the proposed protocol can be shown as follows:

**Goal 1:** $UI| \equiv UI \stackrel{SK}{\longleftrightarrow} CS$

**Goal 2:** $UI| \equiv CS| \equiv UI \stackrel{SK}{\longleftrightarrow} CS$

**Goal 3:** $CS| \equiv UI \stackrel{SK}{\longleftrightarrow} CS$

**Goal 4:** $CS| \equiv UI| \equiv CS \stackrel{SK}{\longleftrightarrow} UI$

**Goal 5:** $CS| \equiv CS \stackrel{SK}{\longleftrightarrow} SJ$

**Goal 6:** $CS| \equiv SJ| \equiv CS \stackrel{SK}{\longleftrightarrow} SJ$

**Goal 7:** $SJ| \equiv CS \stackrel{SK}{\longleftrightarrow} SJ$

**Goal 8:** $SJ| \equiv CS| \equiv SJ \stackrel{SK}{\longleftrightarrow} CS$

#### 7.2.3. Idealized Forms

In the proposed authentication phase, four messages are transmitted via open channels ($\{PID_i, B_1, B_2, V_1, TS_1\}$, $\{M_1, B_3, V_2\}$, $\{B_6, B_7, B_8, V_3, V_4, TS_3\}$, $\{B_8, V_5, TS_4\}$). To analyze these messages, we convert them into idealized forms.

$MSG_1: \ UI \rightarrow SJ : \{r_i, ID_j, TS_1\}_{k_i}$

$MSG_2: \ SJ \rightarrow CS : \{\{r_i, ID_j\}_{k_i}, \{r_j, TS_2\}_{k_j}\}$

$MSG_3: \ CS \rightarrow SJ : \{\{r_k, r_i, C_1, TS_3\}_{k_j}, \{r_j, r_k, TS_3\}_{r_i}\}$

$MSG_4: \ SJ \rightarrow UI : \{r_j, r_k, TS_4\}_{r_i}$

#### 7.2.4. Assumptions

In the proposed protocol, participants agree on the freshness of the random number and secret parameters. Therefore, we show the assumptions to analyze the proposed authentication phase.

$S_1$**:** $CS| \equiv \#(TS_2)$

$S_2$**:** $SJ| \equiv \#(TS_3)$

$S_3$**:** $UI| \equiv \#(TS_4)$

$S_4$**:** $CS| \equiv \#(r_i)$

$S_5$**:** $CS| \equiv UI \stackrel{k_i}{\longleftrightarrow} CS$

$S_6$**:** $CS| \equiv SJ \stackrel{k_j}{\longleftrightarrow} CS$

$S_7$: $SJ| \equiv CS \overset{k_j}{\longleftrightarrow} SJ$

$S_8$: $UI| \equiv CS \overset{r_j}{\longleftrightarrow} UI$

#### 7.2.5. BAN Logic Proof

**Step 1:** We obtain $P_1$ using $MSG_2$.

$$P_1\text{: } CS \lhd \{\{r_i, ID_j\}_{k_i}, \{r_j, TS_2\}_{k_j}\}$$

**Step 2:** We use $S_5$, $S_6$, and MMR to obtain $P_2$ and $P_3$ from $P_1$.

$$P_2\text{: } CS| \equiv UI| \sim (r_i, ID_j)$$
$$P_3\text{: } CS| \equiv SJ| \sim (r_j, TS_2)$$

**Step 3:** From $P_2$ and $P_3$, we use $S_1$, $S_4$, and FR to obtain $P_4$ and $P_5$.

$$P_4\text{: } CS| \equiv \#(r_i, ID_j)$$
$$P_5\text{: } CS| \equiv \#(r_j, TS_2)$$

**Step 4:** From $P_2$, $P_3$, $P_4$ and $P_5$, we use NVR to obtain $P_6$ and $P_7$.

$$P_6\text{: } CS| \equiv UI| \equiv (r_i, ID_j)$$
$$P_7\text{: } CS| \equiv SJ| \equiv (r_j, TS_2)$$

**Step 5:** We obtain $P_8$ using $MSG_3$.

$$P_8\text{: } SJ \lhd \{\{r_k, r_i, C_1, TS_3\}_{k_j}, \{r_j, r_k, TS_3\}_{r_i}\}$$

**Step 6:** We use $S_7$ and MMR to obtain $P_9$ from $P_8$.

$$P_9\text{: } SJ| \equiv CS| \sim (r_k, r_i, C_1, TS_3)$$

**Step 7:** From $P_9$, we use $S_2$ and FR to obtain $P_{10}$.

$$P_{10}\text{: } SJ| \equiv \#(r_k, r_i, C_1, TS_3)$$

**Step 8:** From $P_9$ and $P_{10}$, we use NVR to obtain $P_{11}$.

$$P_{11}\text{: } SJ| \equiv CS| \equiv (r_k, r_i, C_1, TS_3)$$

**Step 9:** Using $P_7$ and $P_{11}$, $CS$ and $SJ$ computes the session key $SK = h(C_1 \parallel r_i \parallel r_j \parallel r_k)$. Thus, we obtain the following:

$$P_{12}\text{: } SJ| \equiv CS| \equiv SJ \overset{SK}{\longleftrightarrow} CS \quad \textbf{(Goal 8)}$$
$$P_{13}\text{: } CS| \equiv SJ| \equiv CS \overset{SK}{\longleftrightarrow} SJ \quad \textbf{(Goal 6)}$$

**Step 10:** Using JR into $P_{12}$ and $P_{13}$, We obtain the following goals:

$$P_{14}\text{: } SJ| \equiv SJ \overset{SK}{\longleftrightarrow} CS \quad \textbf{(Goal 7)}$$
$$P_{15}\text{: } CS| \equiv CS \overset{SK}{\longleftrightarrow} SJ \quad \textbf{(Goal 5)}$$

**Step 11:** We obtain $P_{16}$ using $MSG_4$.

$$P_{16}\text{: } UI \lhd \{r_j, r_k, TS_4\}_{r_i}$$

**Step 12:** We use $S_8$ and MMR to obtain $P_{17}$ from $P_{16}$.

$$P_{17}\text{: } UI| \equiv CS| \sim (r_j, r_k, TS_4)$$

**Step 13:** From $P_{17}$, we use $S_3$ and FR to obtain $P_{18}$.

$$P_{18}\text{: } UI\#(r_j, r_k, TS_4)$$

**Step 14:** From $P_{17}$ and $P_{18}$, we use NVR to obtain $P_{19}$.

$$P_{19}\text{: } UI| \equiv CS| \equiv (r_j, r_k, TS_4)$$

**Step 15:** Using $P_6$ and $P_{19}$, $UI$ and $SJ$ agrees the session key $SK = h(C_1 \parallel r_i \parallel r_j \parallel r_k)$. Thus, we obtain the following:

$$P_{20}\text{: } UI| \equiv CS| \equiv UI \overset{SK}{\longleftrightarrow} CS \quad \textbf{(Goal 2)}$$
$$P_{21}\text{: } CS| \equiv UI| \equiv CS \overset{SK}{\longleftrightarrow} UI \quad \textbf{(Goal 4)}$$

**Step 16:** Using JR into $P_{20}$ and $P_{21}$, We obtain the following goals:

$$P_{22}\text{: } UI| \equiv CS \overset{SK}{\longleftrightarrow} UI \quad \textbf{(Goal 1)}$$
$$P_{23}\text{: } CS| \equiv UI \overset{SK}{\longleftrightarrow} CS \quad \textbf{(Goal 3)}$$

### *7.3. Informal Security Analysis*

#### 7.3.1. Insider Attack

Malicious actor A, who has gone through the registration phase as a legitimate user, can attempt an insider attack using the acquired information. However, the attacker is unable to know the random values$(ri, rj, rk)$ and $k_j$. As a result, the attacker cannot calculate $C_1$, rendering the attack impossible.

#### 7.3.2. Impersonation Attack

(1) User impersonation: Adversary $A$ needs to create a valid message $M_1 = \{PID_i, B_1, B_2, V_1, TS_1\}$ to impersonate the legitimate user $U_i$. While $A$ might obtain $PID_i$ from the user's device, it is impossible for $A$ to access the necessary $k_i$ and $SID_i$ to calculate $B_1, B_2, V_1, TS_1$ needed to create the message. Therefore, $A$ cannot generate the $M_1$ message on behalf of the user $U_i$ and transmit it to the cloud server and control server. Thus, the proposed scheme is secure against user impersonation attacks.

(2) Cloud server impersonation: To execute this attack, $A$ needs to send the message $M_2 = \{M_1, B_3, V_2, TS_2\}$ to the control server on behalf of the cloud server $S_j$. Even if $A$ intercepts the transmission of $M_1$ over an open channel, they cannot generate the necessary $B_3, V_2$ for the message until they know $k_j$. Therefore, the proposed scheme is secure against cloud server impersonation attacks.

#### 7.3.3. Reply and MITM Attacks

All users, the cloud server, and the control server attempt to validate the received messages through $V_1', V_2', V_3', V_4'$. Also, the sent messages are masked with different random values for each session, ensuring freshness. Therefore, the proposed scheme is secure against reply attacks and MITM attacks.

#### 7.3.4. Privileged Insider Attack

In this attack scenario, external entity A is considered a privileged insider, implying that A possesses the user's registration request message $ID_i$ and confidential values such as $\{A_1, A_2, ID_{CS}, Gen(\cdot), Rep(\cdot), \tau_i\}$ However, without the precise biometric information, ID, or PW values of the user, calculating $k_i = A_1 \oplus (ID_i||PW_i)$ or $SID_i = A_2 \oplus (\sigma_i||PW_i)$ is not possible. As a result, the proposed scheme is secure against privileged insider attacks.

#### 7.3.5. Ephemeral Security Leakage Attack

To prevent adversary $A$ from carrying out valid attacks, such as obtaining the session key through this attack scenario, it is essential to ensure that the session key is preserved even if the random values used in the session are exposed. Therefore, assuming $A$ knows the values of $r_i, r_j, r_k$, it is postulated here that even with this knowledge, $A$ cannot calculate $SK$ without knowing $SID_i, SID_j$. Additionally, valid attacks like impersonating the user or cloud server using random values are not possible. Therefore, the proposed scheme is secure against ESL attacks.

#### 7.3.6. Stolen Verifier Attack

We can assume that a malicious $A$, upon obtaining $\{A_3\}$ from the cloud server's database, attempts to calculate the session key $SK = h(C_1||r_i||r_j||r_k)$ or impersonate the cloud server. However, without the cloud server's secret key $x_j$, $A$ cannot deduce the value of $k_j$ from the stored $A_3$, nor can $A$ determine the randomly generated values $r_i, r_j$, or $r_k$. Therefore, $A$ is unable to compute the session key or impersonate the cloud server. Consequently, the proposed scheme is secure against verification table leakage attacks.

#### 7.3.7. DoS Attack

The adversary $A$ may intentionally attempt to send the message $M_1 = \{PID_i, B_1, B_2, V_1, TS_1\}$ repeatedly. However, to generate message $M_1$, $A$ must go through the login process and pass the verification $RPW_i' \stackrel{?}{=} RPW_i$. However, to create a valid $RPW_i' = h(ID_i||PW_i||\sigma_i)$, $A$ cannot have the required $ID_i, PW_i, \sigma_i$. Therefore, $A$ cannot create and repeatedly send the message $M_1$, making the proposed scheme secure against DoS attacks.

#### 7.3.8. User Anonymity and Untraceability

Due to the use of $PID_i$ as a pseudo identity, the user's identity $ID_i$ cannot be deduced by an adversary $A$. Additionally, the $PID_i$ is updated as a new value with random elements for each session, making it impossible for $A$ to compare $PID_i$ values between previous and current sessions to compromise the user's untraceability. Therefore, the proposed scheme provides user anonymity and untraceability.

#### 7.3.9. Session Key Disclosure Attack

To calculate the session key $SK = h(C_1||r_i||r_j||r_k)$, adversary $A$ needs to have access to the values of $SID_i$, $SID_j$, $r_i$, $r_j$, and $r_k$. However, for $A$ to discover $SID_i$, $SID_j$, they would need access to the secret key $x_{cs}$ and the random values $n_i$ and $r_k$. Additionally, random values like $r_i$, $r_j$, $r_k$ are used temporarily and exist only within a single session. Therefore, the proposed scheme is secure against session key disclosure attacks.

#### 7.3.10. Perfect Forward Secrecy

If the control server's secret key $x_{cs}$ is compromised, adversary $A$ may attempt to calculate the session key $SK$ for a previous session. However, since $SK = h(C_1||r_i||r_j||r_k)$ does not contain $x_{cs}$ and the values of $r_i, r_j, r_k$ are random and cannot be deduced, A cannot perform the calculation. Furthermore, without $n_i$ through $x_{cs}$, A cannot compute $SID_i$. Therefore, the proposed scheme ensures perfect forward secrecy.

#### 7.3.11. Mutual Authentication

In the login and authentication phases, the messages $\{PID_i, B_1, B_2, V_1, TS_1\}$ and $\{M_1, B_3, V_2, TS_2\}$ included can be used by the control server to verify the legitimacy of the user and the cloud server through the transmitted $V_1$ and $V_2$. Additionally, messages $\{B_6, B_7, B_8, B_9, V_3, V_4, TS_3\}$ and $\{B_8, B_9, V_4, TS_4\}$ allow both the user and the cloud server to validate each other's identity using $V_3$ and $V_4$. Due to the unavailability of $SID_i$, $kj$ values, and random values to adversaries through the open channel, the transparency of authentication is ensured. Therefore, the provided scheme offers mutual authentication.

## 8. Performance Analysis

### *8.1. Security Features Comparison*

We visually compare the safety elements of the proposed scheme and related schemes [11,21,44–48] and record them in Table 3, which includes various types of safety elements such as "insider attack", "impersonation attack", "stolen verification attack", "ESL attack", "privileged attack", "perfect forward secrecy", "reply attack", "offline password-guessing attack", "session key disclosure", "mutual authentication", "DoS attack", "user anonymity", and "untraceability". Ultimately, the proposed scheme offers more security features compared to Wu et al.'s scheme, and it exhibits fewer features that are either unidentified or not provided, even when compared to the schemes of other related works.

**Table 3.** Security and functionality features(SFF) comparison.

| SFF | [21] | [44] | [45] | [46] | [47] | [48] | [11] | Proposed |
|---|---|---|---|---|---|---|---|---|
| *SP1* | ✓ | ✓ | ✓ | ✓ | △ | △ | × | ✓ |
| *SP2* | × | ✓ | ✓ | × | ✓ | ✓ | × | ✓ |
| *SP3* | ✓ | ✓ | △ | ✓ | △ | △ | × | ✓ |
| *SP4* | ✓ | ✓ | △ | × | ✓ | ✓ | ✓ | ✓ |
| *SP5* | ✓ | ✓ | △ | × | ✓ | ✓ | × | ✓ |
| *SP6* | ✓ | ✓ | △ | ✓ | ✓ | △ | ✓ | ✓ |
| *SP7* | × | ✓ | ✓ | ✓ | △ | ✓ | ✓ | ✓ |
| *SP8* | ✓ | ✓ | × | ✓ | ✓ | △ | ✓ | ✓ |
| *SP9* | × | ✓ | ✓ | × | △ | △ | × | ✓ |
| *SP10* | × | ✓ | ✓ | ✓ | △ | ✓ | ✓ | ✓ |
| *SP11* | ✓ | ✓ | △ | ✓ | △ | △ | ✓ | ✓ |
| *SP12* | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| *SP13* | ✓ | ✓ | △ | ✓ | ✓ | ✓ | × | ✓ |

Note: SP1: insider attack; SP2: impersonation attack; SP3: stolen verification attack; SP4: ESL attack; SP5: privileged insider attack; SP6: perfect forward secrecy; SP7: reply attack; SP8 offline password-guessing attack SP9: session key disclosure; SP10: mutual authentication; SP11: DoS attack; SP12: user anonymity; SP13: untraceability; ✓: provides safety/functional features; ×: does not provides safety/functional features; △: not verified.

### *8.2. Communication Costs Comparison*

We conducted a comparative analysis of communication costs between the related schemes [11,21,44–48] and the proposed scheme. Based on [11], we assume the bit lengths of hash function, timestamp, string, identity, random number, fuzzy extractor, and encryption operation to be 256, 32, 160, 160, 160, 8, and 256 bits, respectively. Therefore, during the MAKA phase of our proposed scheme, the exchanged message $M_1 = \{PID_i, B_1, B_2, V_1, TS_1\}$ requires (160 + 160 + 160 + 256 + 32 = 768 bits), message $M_2 = \{M_1, B_3, V_2, TS_2\}$ requires (768 + 160 + 256 + 32 = 1216 bits), message $M_3 = \{B_6, B_7, B_8, B_9, V_3, V_4, TS_3\}$ requires (160 + 160 + 160 + 160 + 256 + 256 + 32 = 1184 bits), and message $M_4 = \{B_8, V_5, TS_4\}$ requires (160 + 160 + 256 + 32 = 608 bits). Table 4 and Figure 8 present a summary of the communication costs for the associated schemes [11,21,44–48] and the proposed scheme.

**Table 4.** Comparison analysis of communication costs.

| Scheme | Total Cost (bits) | Number of Messages |
|---|---|---|
| Martinez-Pelaez et al. [21] | 4608 bits | 6 |
| Wu et al. (2020) [44] | 4416 bits | 5 |
| Kang et al. [45] | 4320 bits | 4 |
| Huang et al. [46] | 3232 bits | 4 |
| Alam-Kumar [47] | 2912 bits | 4 |
| Wu et al. (2023) [48] | 3360 bits | 4 |
| Wu el al. [11] | 4001 bits | 5 |
| Proposed | 3776 bits | 4 |

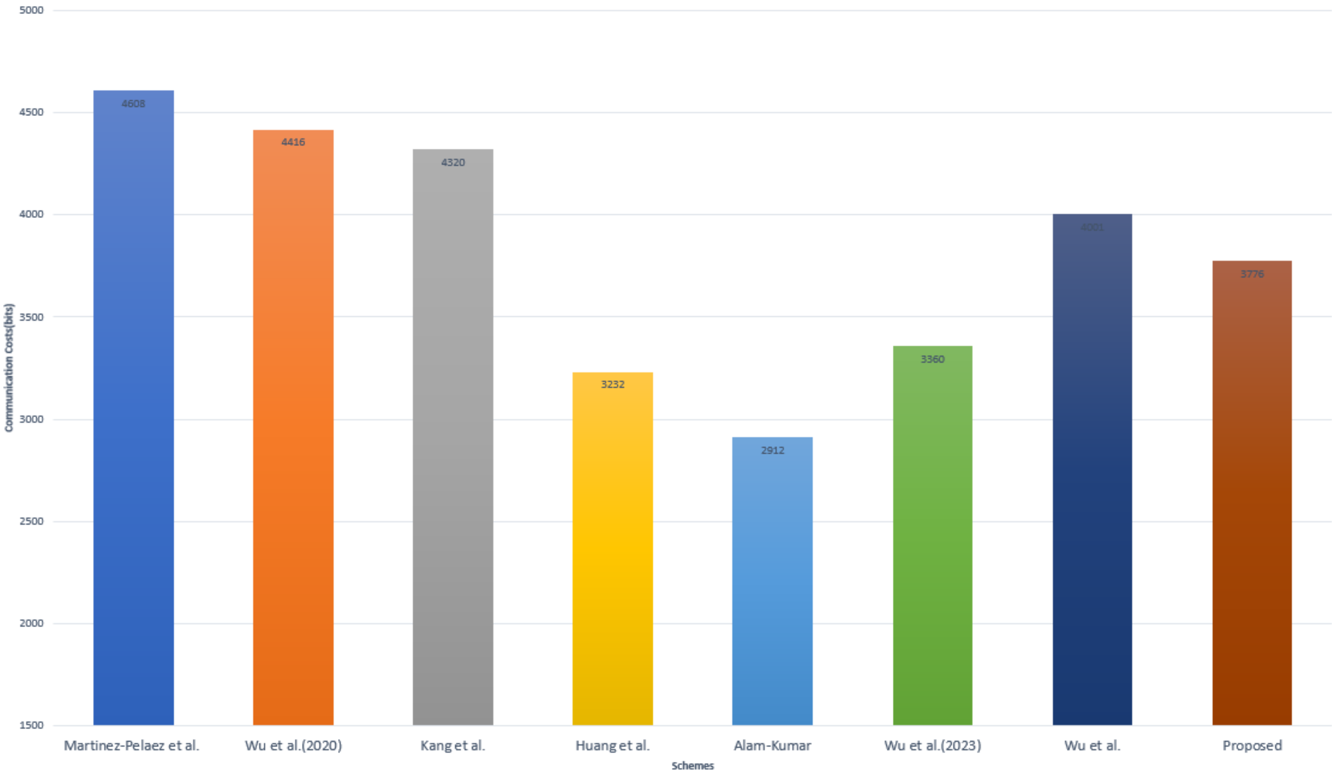


**Figure 8.** Communication costs comparison [7,11,17,21,40–48].

### 8.3. Computation Costs Comparison

We conducted a comparative analysis of computation costs for the AKA phase of the proposed scheme and related schemes [11,21,44–48]. Based on [49], we designed the environment for computing costs. The experimental environment and the performance of operation costs, including the minimum, maximum, and average values, are summarized in Table 5. We represent hash function as $T_h$ and encryption/decryption operations of AES-256 as $T_e$. Using these values, we conducted a comparison of computation costs as shown in Table 6 and Figure 9.

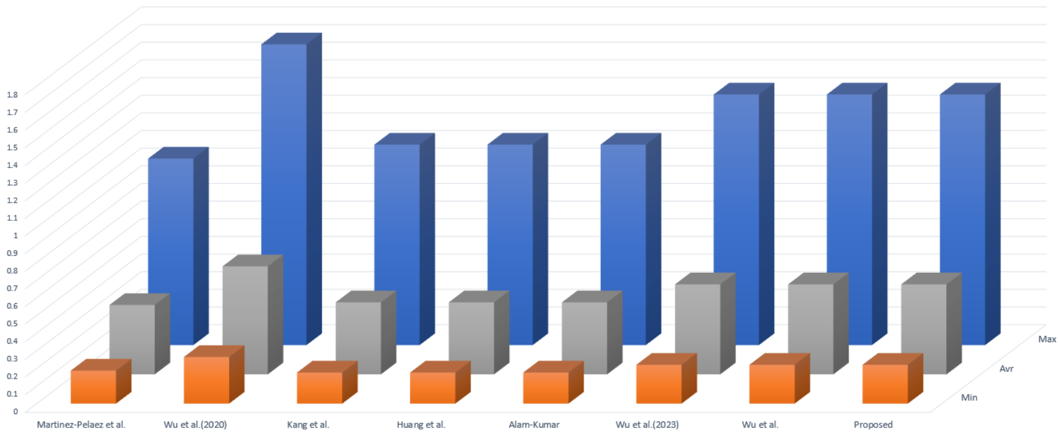


**Figure 9.** Computation costs comparison on user side devices [7,11,17,21,40–48].

**Table 5.** Hardware software enviroment and operation costs.

| Hardware/Software | Operation | Max | Min | Average |
|---|---|---|---|---|
| Raspberry PI 4B with Linux Ubuntu 18.04.4 LTS with 64-bits, 8 GB, and MIRACL library | Hash function $T_h$ | 0.142 ms | 0.022 ms | 0.051 ms |
| | AES-256 $T_e$ | 0.021 ms | 0.011 ms | 0.012 ms |

**Table 6.** Comparison analysis of user side computation costs.

| Protocol | User Side | Max | Min | Average |
|---|---|---|---|---|
| [21] | $7T_h + 3T_e$ | ≈1.057 ms | ≈0.187 ms | ≈0.393 ms |
| [44] | $12T_h$ | ≈1.704 ms | ≈0.264 ms | ≈0.612 ms |
| [45] | $8T_h$ | ≈1.136 ms | ≈0.176 ms | ≈0.408 ms |
| [46] | $8T_h$ | ≈1.136 ms | ≈0.176 ms | ≈0.408 ms |
| [47] | $8T_h$ | ≈1.136 ms | ≈0.176 ms | ≈0.408 ms |
| [48] | $10T_h$ | ≈1.42 ms | ≈0.22 ms | ≈0.51 ms |
| [11] | $10T_h$ | ≈1.42 ms | ≈0.22 ms | ≈0.51 ms |
| Proposed | $10T_h$ | ≈1.42 ms | ≈0.22 ms | ≈0.51 ms |

We can observe that the computational costs for users using the proposed scheme and users using Wu et al.'s scheme are the same. Next, we calculated the computational costs of the cloud server and control server for the proposed scheme and related schemes based on the environments provided in [49] as well. Table 7 represents the calculated computational costs for the proposed scheme and related schemes.

**Table 7.** Comparison analysis of cloud server side control server side computation costs.

| Scheme | Cloud Server | Control Server | Total Average (ms) |
|---|---|---|---|
| [21] | $5T_h + 3T_e$ | $21T_h + 2T_e$ | ≈1.386 ms |
| [44] | $8T_h$ | $19T_h$ | ≈1.377 ms |
| [45] | $4T_h$ | $11T_h$ | ≈0.765 ms |
| [46] | $4T_h$ | $10T_h$ | ≈0.714 ms |
| [47] | $3T_h$ | $6T_h$ | ≈0.459 ms |
| [48] | $5T_h$ | $12T_h$ | ≈0.867 ms |
| [11] | $5T_h$ | $13T_h$ | ≈0.918 ms |
| Proposed | $6T_h$ | $16T_h$ | ≈1.122 ms |

When comprehensively examining the results of the comparison with related schemes, we can elaborate as follows. Our proposed scheme offers more security elements compared to other schemes and is secure against various attacks such as insider attacks, impersonation attacks, stolen verification attacks, ESL attacks, privileged insider attacks, reply attacks, and offline password-guessing attacks. Simultaneously, it maintains reasonable user-side computation cost and communication cost suitable for the CloudIoT environment. However, it is noteworthy that to provide such robust security, additional computation operations on the server side have been introduced.

## 9. Conclusions

This study analyzed the key agreement protocol between cloud-enabled IoT devices and cloud servers as proposed by Wu et al. The scheme proposed by Wu et al. was found to be vulnerable to insider, privileged insiders, impersonation, and verification table leakage attacks and lacks user untraceability. In addition, it is inconvenient for users to update their passwords offline. To overcome these vulnerabilities and inconveniences, this study proposed a provably secure lightweight MAKA protocol for the cloud-based IoT environments.

The proposed protocol ensures safety against various attacks by preventing the exposure of critical parameters using user biometric information, and the cloud server's secret key. Furthermore, user untraceability was ensured by updating the user's pseudonym in every session and convenience was enhanced by adding an offline user password change and biometric template update phase. The safety of mutual authentication and the resulting session key was verified using the RoR model and BAN logic. Moreover, informal analysis was conducted to verify safety against attacks such as insider attacks, impersonation attacks, privileged attacks, ESL attacks, stolen verifier attacks and DoS attacks, while confirming security features such as user anonymity, untraceability, and perfect forward secrecy. The security features, communication costs, and computation costs of the proposed scheme were compared. This comparison demonstrated that the proposed scheme is rational in terms of communication and computation amounts in the cloud-based IoT environments, while being verified for safety.

In conclusion, the proposed scheme demonstrated robust safety and the ability to provide users with real-time services securely. Future research will focus on integrating the proposed scheme into real-world environments and various industrial settings where cloud-based IoT is applied.

**Author Contributions:** Conceptualization, S.J. and Y.P.; Methodology, S.J. and Y.P.; Formal analysis, Y.P.; Writing—original draft, S.J.; Writing—review editing, Y.P.; Supervision, Y.P.; Project administration, Y.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Bisa Research Grant of Keimyung University in 2022.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
2. Zhao, J.C.; Zhang, J.F.; Feng, Y.; Guo, J.X. The study and application of the IOT technology in agriculture. In Proceedings of the 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China, 9–11 July 2010; IEEE: Piscataway, NJ, USA, 2010; Volume 2, pp. 462–465.
3. Park, Y.; Park, Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 2123. [CrossRef] [PubMed]
4. Lee, S.; Kim, S.; Yu, S.; Jho, N.; Park, Y. Provably Secure PUF-Based Lightweight Mutual Authentication Scheme for Wireless Body Area Networks. *Electronics* **2022**, *11*, 3868. [CrossRef]
5. Park, Y.; Ryu, D.; Kwon, D.; Park, Y. Provably secure mutual authentication and key agreement scheme using PUF in internet of drones deployments. *Sensors* **2023**, *23*, 2034. [CrossRef] [PubMed]
6. Jadeja, Y.; Modi, K. Cloud computing-concepts, architecture and challenges. In Proceedings of the 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), Nagercoil, India, 21–22 March 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 877–880.
7. Dinh, T.; Kim, Y.; Lee, H. A location-based interactive model of internet of things and cloud (IoT-Cloud) for mobile cloud computing applications. *Sensors* **2017**, *17*, 489. [CrossRef] [PubMed]
8. Babu, S.M.; Lakshmi, A.J.; Rao, B.T. A study on cloud based Internet of Things: CloudIoT. In Proceedings of the 2015 Global Conference on Communication Technologies (GCCT), Thuckalay, India, 23–24 April 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 60–65.
9. Zargar, S.; Shahidinejad, A.; Ghobaei-Arani, M. A lightweight authentication protocol for IoT-based cloud environment. *Int. J. Commun. Syst.* **2021**, *34*, e4849. [CrossRef]
10. Kim, M.; Yu, S.; Lee, J.; Park, Y.; Park, Y. Design of secure protocol for cloud-assisted electronic health record system using blockchain. *Sensors* **2020**, *20*, 2913. [CrossRef]
11. Wu, T.Y.; Meng, Q.; Kumari, S.; Zhang, P. Rotating behind security: A lightweight authentication protocol based on iot-enabled cloud computing environments. *Sensors* **2022**, *22*, 3858. [CrossRef]
12. Shouqi, C.; Wanrong, L.; Liling, C.; Xin, H.; Zhiyong, J. An improved authentication protocol using smart cards for the Internet of Things. *IEEE Access* **2019**, *7*, 157284–157292. [CrossRef]

13. Nikooghadam, M.; Jahantigh, R.; Arshad, H. A lightweight authentication and key agreement protocol preserving user anonymity. *Multimed. Tools Appl.* **2017**, *76*, 13401–13423. [CrossRef]
14. Kumari, S.; Chaudhry, S.A.; Wu, F.; Li, X.; Farash, M.S.; Khan, M.K. An improved smart card based authentication scheme for session initiation protocol. *Peer Netw. Appl.* **2017**, *10*, 92–105. [CrossRef]
15. Limbasiya, T.; Soni, M.; Mishra, S.K. Advanced formal authentication protocol using smart cards for network applicants. *Comput. Electr. Eng.* **2018**, *66*, 50–63. [CrossRef]
16. Chandrakar, P.; Om, H. An extended ECC-based anonymity-preserving 3-factor remote authentication scheme usable in TMIS. *Int. J. Commun. Syst.* **2018**, *31*, e3540. [CrossRef]
17. Sharma, G.; Kalra, S. A lightweight multi-factor secure smart card based remote user authentication scheme for cloud-IoT applications. *J. Inf. Secur. Appl.* **2018**, *42*, 95–106. [CrossRef]
18. Gope, P.; Sikdar, B. Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. *IEEE Internet Things J.* **2018**, *6*, 580–589. [CrossRef]
19. Siddiqui, Z.; Gao, J.; Khan, M.K. An improved lightweight PUF–PKI digital certificate authentication scheme for the Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 19744–19756. [CrossRef]
20. Zhou, L.; Li, X.; Yeh, K.H.; Su, C.; Chiu, W. Lightweight IoT-based authentication scheme in cloud computing circumstance. *Future Gener. Comput. Syst.* **2019**, *91*, 244–251. [CrossRef]
21. Martínez-Peláez, R.; Toral-Cruz, H.; Parra-Michel, J.R.; García, V.; Mena, L.J.; Félix, V.G.; Ochoa-Brust, A. An enhanced lightweight IoT-based authentication scheme in cloud computing circumstances. *Sensors* **2019**, *19*, 2098. [CrossRef]
22. Alzahrani, B.A.; Chaudhry, S.A.; Barnawi, A.; Al-Barakati, A.; Shon, T. An anonymous device to device authentication protocol using ECC and self certified public keys usable in Internet of Things based autonomous devices. *Electronics* **2020**, *9*, 520. [CrossRef]
23. Islam, S.H.; Biswas, G. Design of two-party authenticated key agreement protocol based on ECC and self-certified public keys. *Wirel. Pers. Commun.* **2015**, *82*, 2727–2750. [CrossRef]
24. Mandal, S.; Mohanty, S.; Majhi, B. Cryptanalysis and enhancement of an anonymous self-certified key exchange protocol. *Wirel. Pers. Commun.* **2018**, *99*, 863–891. [CrossRef]
25. Chen, Y.; López, L.; Martínez, J.F.; Castillejo, P. A lightweight privacy protection user authentication and key agreement scheme tailored for the Internet of Things environment: LightPriAuth. *J. Sensors* **2018**, *2018*, 7574238. . [CrossRef]
26. Lee, J.; Yu, S.; Kim, M.; Park, Y.; Das, A.K. On the design of secure and efficient three-factor authentication protocol using honey list for wireless sensor networks. *IEEE Access* **2020**, *8*, 107046–107062. [CrossRef]
27. Yu, Y.; Hu, L.; Chu, J. A secure authentication and key agreement scheme for IoT-based cloud computing environment. *Symmetry* **2020**, *12*, 150. [CrossRef]
28. He, D.; Zeadally, S.; Kumar, N.; Wu, W. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2052–2064. [CrossRef]
29. Tsai, J.L.; Lo, N.W. A privacy-aware authentication scheme for distributed mobile cloud computing services. *IEEE Syst. J.* **2015**, *9*, 805–815. [CrossRef]
30. Kumari, A.; Kumar, V.; Abbasi, M.Y.; Kumari, S.; Chaudhary, P.; Chen, C.M. Csef: Cloud-based secure and efficient framework for smart medical system using ecc. *IEEE Access* **2020**, *8*, 107838–107852. [CrossRef]
31. Bhuarya, P.; Chandrakar, P.; Ali, R.; Sharaff, A. An enhanced authentication scheme for Internet of Things and cloud based on elliptic curve cryptography. *Int. J. Commun. Syst.* **2021**, *34*, e4834. [CrossRef]
32. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
33. Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. In Proceedings of the Advances in Cryptology—EUROCRYPT 2002: International Conference on the Theory and Applications of Cryptographic Techniques Amsterdam, The Netherlands, 28 April–2 May 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 337–351.
34. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In Proceedings of the Public Key Cryptography-PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, 23–26 January 2005; Springer: Berlin/Heidelberg, Germany, 2005, pp. 65–84.
35. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [CrossRef]
36. Kwon, D.K.; Yu, S.J.; Lee, J.Y.; Son, S.H.; Park, Y.H. WSN-SLAP: Secure and lightweight mutual authentication protocol for wireless sensor networks. *Sensors* **2021**, *21*, 936. [CrossRef]
37. Yu, S.; Lee, J.; Sutrala, A.K.; Das, A.K.; Park, Y. LAKA-UAV: Lightweight authentication and key agreement scheme for cloud-assisted Unmanned Aerial Vehicle using blockchain in flying ad hoc networks. *Comput. Netw.* **2023**, *224*, 109612. [CrossRef]
38. Kim, M.; Lee, J.; Oh, J.; Kwon, D.; Park, K.; Park, Y.; Park, K.H. A Secure Batch Authentication Scheme for Multiaccess Edge Computing in 5G-Enabled Intelligent Transportation System. *IEEE Access* **2022**, *10*, 96224–96238. [CrossRef]
39. Boyko, V.; MacKenzie, P.; Patel, S. Provably secure password-authenticated key exchange using Diffie-Hellman. In Proceedings of the Advances in Cryptology—EUROCRYPT 2000: International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000; Proceedings 19; Springer: Berlin/Heidelberg, Germany, 2000; pp. 156–171.
40. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst. (TOCS)* **1990**, *8*, 18–36. [CrossRef]
41. Kwon, D.; Son, S.; Park, Y.; Kim, H.; Park, Y.; Lee, S.; Jeon, Y. Design of secure handover authentication scheme for urban air mobility environments. *IEEE Access* **2022**, *10*, 42529–42541. [CrossRef]

42. Son, S.; Kwon, D.; Lee, S.; Jeon, Y.; Das, A.K.; Park, Y. Design of Secure and Lightweight Authentication Scheme for UAV-Enabled Intelligent Transportation Systems using Blockchain and PUF. *IEEE Access* **2023**, *11*, 60240–60253. [CrossRef]
43. Cho, Y.; Oh, J.; Kwon, D.; Son, S.; Yu, S.; Park, Y.; Park, Y. A secure three-factor authentication protocol for e-governance system based on multiserver environments. *IEEE Access* **2022**, *10*, 74351–74365. [CrossRef]
44. Wu, H.L.; Chang, C.C.; Zheng, Y.Z.; Chen, L.S.; Chen, C.C. A secure IoT-based authentication system in cloud computing environment. *Sensors* **2020**, *20*, 5604. [CrossRef]
45. Kang, B.; Han, Y.; Qian, K.; Du, J. Analysis and improvement on an authentication protocol for IoT-enabled devices in distributed cloud computing environment. *Math. Probl. Eng.* **2020**, . [CrossRef]
46. Huang, H.; Lu, S.; Wu, Z.; Wei, Q. An efficient authentication and key agreement protocol for IoT-enabled devices in distributed cloud computing architecture. *EURASIP J. Wirel. Commun. Netw.* **2021**, *2021*, 150. [CrossRef]
47. Alam, I.; Kumar, M. A novel protocol for efficient authentication in cloud-based IoT devices. *Multimed. Tools Appl.* **2022**, *81*, 13823–13843. [CrossRef]
48. Wu, T.Y.; Kong, F.; Meng, Q.; Kumari, S.; Chen, C.M. Rotating behind security: An enhanced authentication protocol for IoT-enabled devices in distributed cloud computing architecture. *EURASIP J. Wirel. Commun. Netw.* **2023**, *2023*, 36. [CrossRef]
49. Park, K.; Park, Y. IAKA-CIOT: An improved authentication and key agreement scheme for cloud enabled internet of things using physical unclonable function. *Sensors* **2022**, *22*, 6264. [CrossRef] [PubMed]