sensors

MDPI

Article

Provably Secure Mutual Authentication and Key Agreement Scheme Using PUF in Internet of Drones Deployments

Yohan Park [1], Daeun Ryu [1], Deokkyu Kwon [2,*] and Youngho Park [2,*]

[1] School of Computer Engineering, Keimyung University, Daegu 42601, Republic of Korea;
[2] School of Electronics Engineering, Kyungpook National University, Daegu 41566, Republic of Korea
* Correspondence: kdk145@knu.ac.kr (D.K.); parkyh@knu.ac.kr (Y.P.); Tel.: +82-53-950-7842 (Y.P.)

Abstract: Internet of Drones (IoD), designed to coordinate the access of unmanned aerial vehicles (UAVs), is a specific application of the Internet of Things (IoT). Drones are used to control airspace and offer services such as rescue, traffic surveillance, environmental monitoring, delivery and so on. However, IoD continues to suffer from privacy and security issues. Firstly, messages are transmitted over public channels in IoD environments, which compromises data security. Further, sensitive data can also be extracted from stolen mobile devices of remote users. Moreover, drones are susceptible to physical capture and manipulation by adversaries, which are called drone capture attacks. Thus, the development of a secure and lightweight authentication scheme is essential to overcoming these security vulnerabilities, even on resource-constrained drones. In 2021, Akram et al. proposed a secure and lightweight user–drone authentication scheme for drone networks. However, we discovered that Akram et al.'s scheme is susceptible to user and drone impersonation, verification table leakage, and denial of service (DoS) attacks. Furthermore, their scheme cannot provide perfect forward secrecy. To overcome the aforementioned security vulnerabilities, we propose a secure mutual authentication and key agreement scheme between user and drone pairs. The proposed scheme utilizes physical unclonable function (PUF) to give drones uniqueness and resistance against drone stolen attacks. Moreover, the proposed scheme uses a fuzzy extractor to utilize the biometrics of users as secret parameters. We analyze the security of the proposed scheme using informal security analysis, Burrows–Abadi–Needham (BAN) logic, a Real-or-Random (RoR) model, and Automated Verification of Internet Security Protocols and Applications (AVISPA) simulation. We also compared the security features and performance of the proposed scheme and the existing related schemes. Therefore, we demonstrate that the proposed scheme is suitable for IoD environments that can provide users with secure and convenient wireless communications.

Keywords: AVISPA; BAN logic; Internet of Drones; mutual authentication; PUF

Citation: 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. https://doi.org/10.3390/s23042034

Academic Editor: Constantin Caruntu and Ciprian Romeo Comşa

Received: 15 December 2022
Revised: 7 February 2023
Accepted: 8 February 2023
Published: 10 February 2023

1. Introduction

Internet of Drones (IoD) [1], which is often referred to as an unmanned aerial vehicles (UAVs) network, is a layered network control architecture designed to coordinate the access of drones. Drones in IoD environments can perform various flight tasks by embedding various sensors, actuators, recorders, batteries, computations, and communication modules. Figure 1 shows the basic structure of a drone in IoD environments. With these modules, drones are used to control the airspace and offer services such as rescue, healthcare, traffic surveillance, environmental monitoring, delivery, and search to users [2]. The IoD architecture generally comprises remote users, a control server, and drones. Remote users query the information of drones to receive useful services. The control server is centrally located in the wireless communication flow, mediating and providing a seamless data exchange process between remote users and drones. Drones, located in their own flying zone, collect surrounding environment information and send it to users through the control center.

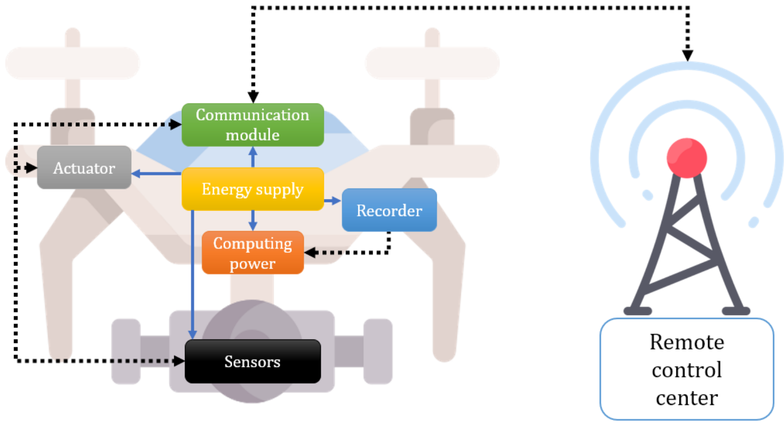

Figure 1. Basic structure of the drone in IoD environments.

Although IoD environments offer useful services to users, they can suffer from several privacy and security issues [3]. Firstly, IoD environments can be vulnerable to various security attacks, such as eavesdropping, deleting, and intercepting, because all messages are transmitted via a public channel. Moreover, the mobile devices of remote users can be stolen/lost, and the sensitive stored data of these devices can threaten the whole IoD environment. Additionally, drones can be physically captured by malicious adversaries who can try to impersonate them using secret information extracted from drones using power analysis attacks. Finally, drones in IoD environments are designed to use restricted power, computation, and storage sources because the entire energy source is preferentially devoted to flying tasks. Thus, a secure and lightweight authentication scheme is necessary, considering the above security vulnerabilities and specific features of IoD environments.

In 2021, Akram et al. [4] proposed a user–drone access scheme designed to be secure and lightweight for drone networks. The authors claimed that the scheme resists user, control center, and drone impersonation attacks and provides anonymity and untraceability. However, we find that Akram et al.'s scheme is vulnerable to drone impersonation, verification table leakage, and denial of service (DoS) attacks. In addition, their scheme cannot ensure perfect forward secrecy and fails to guarantee correctness. To improve these vulnerabilities, we propose a mutual authentication and key agreement (MAKA) scheme that can provide convenient services to users with high security and efficiency for IoD environments. In the proposed scheme, we utilize biometrics [5] to resist various security attacks, such as offline guessing attacks on user devices. Moreover, we apply physical unclonable function (PUF) [6] technology to prevent cloning and physical attacks of drones using power analysis attacks. Considering real-time communication in IoD environments and the limited computation resources of user devices and drones, we only utilize hash functions and exclusive-OR operators, which are reliable in terms of computation and communication overheads.

1.1. Research Contributions

- We review and perform a security analysis of Akram et al.'s scheme. Then, we propose a MAKA scheme designed to ensure high security using biometrics and PUF. Hash functions and exclusive-OR operations are used for lightweight architecture, making the proposed scheme suitable for drone networks. Moreover, a fuzzy extractor and PUF are applied in the proposed scheme to enhance the security level.
- We prove the security robustness of the proposed scheme using the Automated Verification of Internet Security Protocols and Applications (AVISPA) simulation tool [7,8], Real-or-Random (RoR) model [9], and Burrows–Abadi–Needham (BAN) logic [10].
- We perform an informal analysis to ensure that the proposed scheme can provide security against various attacks, including offline password guessing, session key

disclosure, verification table leakage, impersonation, and DoS attacks. Additionally, we show that the proposed scheme can achieve mutual authentication, perfect forward secrecy, untraceability, and anonymity.

- We evaluate and compare the security features, communication, and computation costs of the proposed scheme with existing authentication schemes, including Akram et al.'s scheme.

1.2. Organization

In Section 2, we introduce existing studies on IoD environments. We provide a system model as well as an adversary model, fuzzy extractor, and PUF used in the proposed scheme in Section 3. Then, we show Akram et al.'s scheme in Section 4. Section 5 describes security vulnerabilities discovered in Akram et al.'s scheme. The proposed scheme is introduced in Section 6. Security analyses, i.e., BAN logic, RoR model, AVISPA, are shown in Section 7, and performance analyses, i.e., security features, communication, computation costs, are shown in Section 8. In Section 9, we conclude our paper and describe future works.

2. Related Works

Since the basic concept of IoD environments was introduced by Gharibi et al. [1], various authentication schemes have been proposed over the past few years. In 2018, Wazid et al. [11] proposed an authentication scheme to provide remote users with drone services based on three-factor technology. To apply lightweight communication services, Wazid et al. utilize hash function and exclusive-OR operators. However, their scheme cannot prevent privileged insider and impersonation attacks. In 2019, Teng et al. [12] analyzed security vulnerabilities, named "attacker mode", which can happen in IoD environments. Thus, they proposed an authentication scheme utilizing the elliptic curve digital signature algorithm (ECDSA) to verify the legitimacy of identity signatures on drones. However, Teng et al.'s scheme was designed as an authentication scheme involving two-way authentication between drones based on ECC, which incurs a large computational overhead. Srinivas et al. [13] proposed a temporal credential-based authentication for IoD networks. Srinivas et al. argued that security and efficiency are the main requirements for the IoD environment, and a lightweight authentication protocol is essential to satisfy these requirements. In their scheme, the authors claimed that it can resist various security attacks such as a stolen mobile device, replay, MITM, ephemeral secret leakage (ESL), impersonation, password and/or biometric update, and remote drone capture attacks. In 2020, Ali et al. [14] pointed out that Srinivas et al.'s scheme [13] does not provide untraceability and resists stolen verifier attacks. To overcome that, Ali et al. suggested a lightweight authentication scheme for drones using symmetric key primitives and temporal credentials. Ever [15] suggested a framework for mobile sinks used in drones using bilinear pairing and ECC, which has a large computational cost. However, Ever's protocol cannot provide user anonymity and untraceability [16]. In 2022, Wu et al. [17] proposed a drone communication scheme for 5G networks. They argued that several existing IoD protocols have high computation overheads because of using a public key infrastructure (PKI) mechanism. Therefore, they only utilized hash functions and exclusive-OR operators. In the same year, Tanveer et al. [18] proposed an authentication mechanism for IoD environments. They used an AES-CBC-256 cipher and ECC to ensure the anonymity of users. Although the above schemes [11–15,17,18] provide useful services such as healthcare, rescue, and traffic surveillance, they can suffer from physical attacks because each drone cannot protect security parameters from power analysis attacks.

To strengthen the authentication process and access control of drones, various PUF-based authentication schemes have been proposed. Alladi et al. [19] proposed a two-stage authentication protocol that divided drone hierarchies for smart drone networks. In Alladi et al.'s scheme, each drone equipped with PUF communicates with a ground station through a leader drone, reducing network overhead. Thus, the authors claimed their scheme does not require the storage of secret keys in drones, protecting it from impersonation, drone tampering, and MITM attacks. In the same years, Pu et al. [20] proposed an authentication protocol for drone environments using PUF and chaotic systems. The authors used the challenge–response pair of the PUF as the seed value of the chaotic system to jumble the message randomly. In 2021, Zhang et al. [21] suggested a three-party authentication scheme for IoD environments. In Zhang et al.'s scheme, the head drone manages member drones and mediates the communication between the ground station and member drones. The entire process of their scheme only uses hash functions and XOR operations. Moreover, the authors introduced PUF systems to prevent physical capture attacks.

In 2021, Akram et al. [4] suggested a scheme for secure and efficient drone access in IoD networks. The authors demonstrated that various security attacks, e.g., user, control center, and drone impersonation attacks, can be prevented in their scheme. However, our security analysis indicates that their scheme is vulnerable to DoS, session key disclosure, stolen-verifier, and drone impersonation attacks and cannot provide perfect forward secrecy.

We summarize the cryptographic techniques and the advantages and limitations of the existing related schemes [4,11–15,17–21] in Table 1. Although previous authentication schemes can provide convenient services to users, they still have high computational and communication overhead and security drawback problems. Therefore, we propose a secure drone-access scheme to improve these security flaws considering lightweight communication characteristics of IoD environments. The proposed scheme can provide stolen mobile device and drone impersonation attacks using biometric and PUF technologies, respectively. Moreover, the proposed scheme can support efficient communications using only hash functions and exclusive-OR operators.

Table 1. Cryptographic technologies and properties of the related schemes for IoD environments.

Schemes	Cryptographic Technologies	Advantages and Limitations
Wazid et al. [11]	* Hash functions * Fuzzy extractor	* Presented IoD environments and utilized biometrics information to ensure the security of remote users * Vulnerable to privileged insider and impersonation attacks
Teng et al. [12]	* ECDSA	* Defined security threats in IoD environments named "attacker mode" * Requires large computation overheads
Srinivas et al. [13]	* Hash functions * Fuzzy extractor	* Used temporal credentials for mutual authentication * Vulnerable to untraceability and stolen verifier attacks
Ali et al. [14]	* Hash functions * Fuzzy extractor * Symmetric key primitives	* Anonymous and lightweight security solution using temporal credentials and symmetric key primitives * Vulnerable to ESL, physical and cloning attacks
Ever et al. [15]	* Bilinear pairings * ECC	* Analyzed studies utilized UAVs as mobile sinks * Require high computation overheads * Cannot provide anonymity and untraceability
Wu et al. [17]	* Hash functions * Fuzzy extractor	* Proposed a drone-to-user authentication scheme for 5G networks * Vulnerable to physical attacks due to the stored parameters in UAV
Tanveer et al. [18]	* Hash functions * Fuzzy extractor * ECC * Symmetric key primitives	* Provides anonymous communication to users using AES and ECC * Vulnerable to physical attacks due to the stored parameters in UAV
Alladi et al. [19]	* PUF * Message authentication code * Symmetric key primitives	* Classified drones by layer and proposed PUF-based two-stage authentication protocol * Vulnerable to replay, insider, server spoofing, DoS attacks
Pu et al. [20]	* PUF * Chaotic system	* Used PUF and chaotic map technologies to generate random key * Vulnerable to physical attacks because of a stored challenge value in the memory of UAV
Zhang et al. [21]	* Hash functions * Fuzzy extractor * FourQ * Symmetric key primitives	* Proposed authentication scheme using FourQ and BPV pre-computation technologies * Require high computation and communication overheads * Cannot provide user anonymity
Akram et al. [4]	* Hash functions * Fuzzy extractor * Symmetric key primitives	* Provide privacy of location information to remote users and drones * Vulnerable to drone impersonation, stolen verifier, and DoS attacks, and have correctness problem

3. Preliminaries

We present the system model and adversary model for IoD environments. Moreover, we introduce some relevant preliminaries to understand this paper.

3.1. System Model

As shown in Figure 2, IoD environments consist of a control center, users and drones. According to the IoD environment model, various drones collect the data in their particular zones in a target field and transmit the data to the server. External users are required to connect to the server to obtain data from the deployed drones. For access, secure authentication is necessary between the user and drone via the control center. Subsequently, the user and drone pair share a session key and begin communication. The details of this process are as follows.

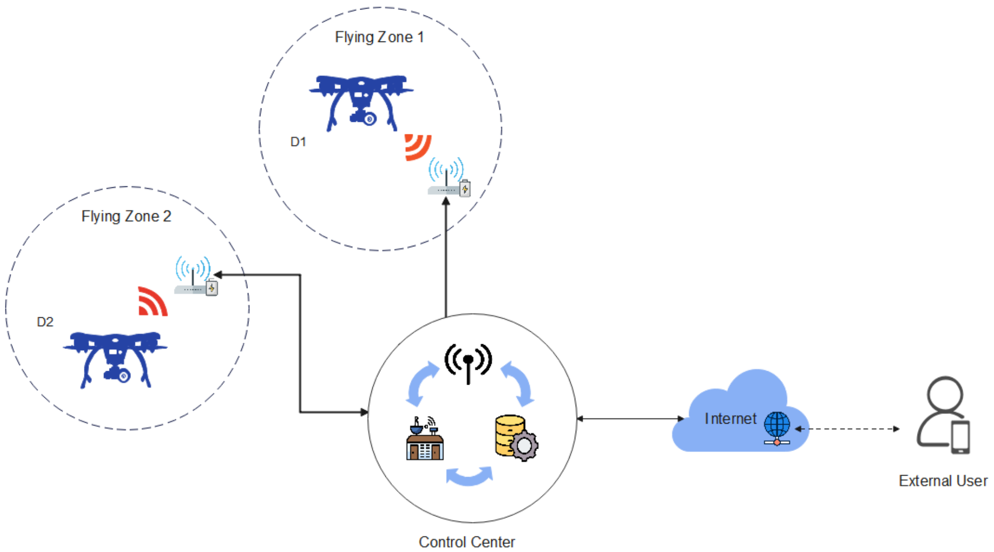

Figure 2. The general system model of IoD environments.

- Remote user (U_m): A remote user U_m owns a mobile device to receive IoD services. To communicate with a drone D_n, U_m must register with the control center. U_m utilizes biometric technology in addition to identity and password to store sensitive information safely.
- Control center: The control center is a trusted third party with enough computation and storage capacities. Therefore, the control center perform a role as the system manager of IoD environments. Furthermore, the control center authenticates with both U_m and D_n information and helps U_m to access the D_n. The control center generates secret keys for U_m and D_n against their identities.
- Drone (D_n): A drone D_n collects the data in their particular flying zone and must be registered by the control center to communicate with U_m. Then, D_n sends the data to $=U_m$ through the control center. Moreover, D_n has restricted computation and storage capacities.

3.2. Adversary Model

We follow the widely used adversary model, named the "Dolev–Yao (DY) adversary model" [22,23]. Under the DY model, the entities involved in the IoD environments, i.e., U_m and D_n, are not assumed to be trustworthy, and the communication of the channel is insecure. Therefore, an adversary $\mathcal{A}$ can modify or delete the transmitted messages and also can eavesdrop on the exchanged messages. Furthermore, drones move around in unattended hostile areas with collected sensor data. Thus, they are vulnerable to physical capture attacks [11,24], and the sensitive data stored in the drone can be extracted using the power analysis attacks.

3.3. Fuzzy Extractor

The fuzzy extractor [25] is widely accepted to verify the biometric authentication. A biometric key can be generated with a biometric template such as fingerprints, faces and irises. The fuzzy extractor is defined with the following two algorithms:

- $Gen(Bio_m) = (\alpha_m, \beta_m)$: It is a probabilistic algorithm to generate a secret key α_m. The user inputs biometric Bio_m, the output of this function is the secret parameter α_m, and the public reproduction parameter β_m.
- $Rep(Bio^*_m, \beta_m) = (\alpha_m)$: It is a deterministic algorithm to recreate the original α_m. The function accepts a noisy user biometric Bio^*_m and controls the noise using the public reproduction parameter β_m. Then, this algorithm reproduces the original biometric secret key α_m.

3.4. Physical Unclonable Function

PUF is a physical circuit that maps a bit-string pair called "challenge–response pair" [6]. When an input challenge value is entered into the PUF circuit, it produces a value that isan arbitrary string of bits. In this paper, we use PUF to generate secret values instead of stringing them in the memory of the drone and obtain a stable response good enough for security using fuzzy extractors. The property of PUF is as below.

- The PUF is a physical microstructure of the device.
- It is extremely difficult or impossible to clone the PUF circuit.
- An unpredictable response value must be output.
- It is possible to evaluate and implement a PUF circuit easily.

4. Revisit of Akram et al.'s Scheme

Akram et al. [4] suggested a drone-access authentication protocol for surveillance tasks in a smart city. Akram et al.'s scheme is composed of the following phases: (1) user registration; (2) drone registration; (3) authentication and key agreement (AKA) phases. Table 2 shows the whole notation and description in their scheme.

Table 2. Notations and descriptions.

Notation	Description
ID_m, ID_n	Identity of the user and drone
SID_c, SID_m, SID_n	Pseudonym of the control center, user and drone
Bio_m	Biometric of the user
k_m, k_n	Master private key of the user and drone
s, MSK	Secret keys of the control center
$Rep(.)$	Fuzzy biometric reproduction
$Gen(.)$	Fuzzy biometric generator
a_1, a_2, a_3	Random numbers
SK	Session key
$h(.)$	Hash function
$\|\|$	Concatenation operator
$\oplus$	Exclusive-OR operator

4.1. Registration Phase

4.1.1. Remote User Registration Phase

Step 1: The user inputs their own ID_m, PW_m and imprints Bio_m. Then, U_m calculates $Gen(Bio_m) = (\alpha_m, \beta_m)$ and sends ID_m to the control center.

Step 2: The control center calculates $SID_m = h(ID_m||s)$, $k_m = h(SID_m||MSK)$ and generates a random number a_m. After that, the control center computes $MID_m = Enc_{MSK}(SID_m||\alpha_m)$ and sends $\{k_m, SID_m, SID_n\}$ to U_m.

Step 3: U_m computes $\gamma_m = h(ID_m||PW_m||\alpha_m) \oplus k_m$, $SID^u_m = h(ID_m||PW_m) \oplus SID_m$. Then, U_m stores $\{\gamma_m, SID^u_m, SID_n\}$.

4.1.2. Drone Registration Phase

Step 1: D_n selects ID_n and sends it to the control center.

Step 2: The control center computes $SID_n = h(ID_n||s)$, $k_n = h(SID_n||MSK)$ and stores $\{ID_n, k_n, SID_n\}$ in its database. Then, the control center sends $\{k_n, SID_n\}$ to D_n.

Step 3: When D_n receives $\{k_n, SID_n\}$, D_n saves them in the memory.

4.2. AKA Phase

Step 1: U_m inputs ID_m, PW_m and also imprints Bio_m. Then, U_m computes $\alpha_m = Rep(Bio_m, \beta_m)$, $SID_m = SID_m^u \oplus h(ID_m||PW_m)$, $k_m = \gamma_m \oplus h(ID_m||PW_m||\alpha_m)$. Afterward, U_m generates a_1 and computes $A_1 = h(SID_m||SID_c||k_m) \oplus a_1$, $A_2 = h(SID_m||SID_c||k_m||a_1) \oplus SID_n$ and $A_3 = h(SID_m||SID_n||SID_c||k_m||a_1)$. Finally, U_m sends $\{MID_m, A_1, A_2, A_3\}$ to the control center.

Step 2: The control center retrieves $(SID_m||\alpha_m) = Dec_{MSK}(MID_m)$. Then, the control center computes $k_m = h(SID_m|| MSK)$, $a_1^* = A_1 \oplus h(SID_m^*||SIDc||k_m^*)$ and $SID_n^* = A_2 \oplus h(SID_m^*||SID_c||k_m^*||a_1^*)$, and verifies k_n against SID_n^*. Then, the control center computes $A_3^* = h(SID_m^*||SID_n^*||SID_c||k_m^*||a_1^*)$ and checks $A_3^* \stackrel{?}{=} A_3$. The control center generates a_2, a_m^{new} and computes $MID_m^{new} = Enc_{MSK}(SID_m||a_m^{new})$, $A_4 = h(SID_n^*||k_n) \oplus (a_1^*||a_2||MID_m^{new})$, $A_5 = h(SID_n^*||SID_c||k_n||a_1^*) \oplus SID_m^*$ and $A_6 = h(SID_m^*||SID_n^*||SID_c||k_n||a_1^*||a_2)$. Finally, the control center sends $\{A_4, A_5, A_6\}$ to the drone D_n.

Step 3: D_n computes $(a_1^{**}||a_2^*||MID_m^{new}) = A_4 \oplus h(SID_n||k_n)$, $SID_m^{**} = A_5 \oplus h(SID_n||SID_c||k_n||a_1^{**})$ and $A_6^* = h(SID_M^{**}||SID_n|| SID_c||k_n||a_1^{**}||a_2^*)$. Then, D_n checks $A_6^* \stackrel{?}{=} A_6$ and generates a_3. After that, D_n computes $A_7 = h(SID_n||SID_m^{**}||a_1^{**}) \oplus (a_2||a_3^*||MID_m^{new})$, $A_8 = h(a_1^{**}||a_2||a_3^*)$, $SK_{nm} = h(SID_m^{**}||SID_n||SID_c||A_8)$ and $A_9 = h(SID_m^{**}||SID_n|| SID_c||a_2||a_3^*||A_8)$. Finally, D_n sends $\{A_7, A_9\}$ to U_m.

Step 4: The U_m computes $(a_2^*|| a_3^{**}||MID_m^{new}) = A_7 \oplus h(SID_n||SID_m||a_1)$, $A_8^* = h(a_1||a_2^*||a_3^{**})$ and $A_9^* = h(SID_m||SID_n||SID_c|| a_2^*||a_3^{**}||A_8^*)$. Then, it validates $A_9^* \stackrel{?}{=} A_9$ and computes $SK_{nm} = h(SID_m^{**}||SID_n|| SID_c||A_8^*)$.

5. Cryptanalysis of Akram et al.'s Scheme

According to Section 3.2, an adversary $\mathcal{A}$ can obtain a $\{\gamma_m, SID_m^u, SID_n\}$ from legitimate user's mobile device. Moreover, $\mathcal{A}$ can obtain $\{k_n, SID_n\}$ from a captured drone using a power analysis attack. With this information, various security attacks, i.e., session key disclosure, drone impersonation, stolen-verifier, DoS attacks, and perfect forward secrecy, can be executed by $\mathcal{A}$. The details are shown below.

5.1. Session Key Disclosure Attack

For $\mathcal{A}$ to generate a session key $SK_{nm} = h(SID_m||SID_n||SID_c||A_8)$, $\mathcal{A}$ has to obtain SID_m, SID_n and $A_8 = h(a_1||a_2||a_3)$. The procedures are as follows.

Step 1: $\mathcal{A}$ computes $(a_1||a_2||MID_m^{new}) = A_4 \oplus h(SID_n|| k_n)$, $SID_m = A_5 \oplus h(SID_n|| SID_c||k_n||a_1)$, and $(a_2||a_3||MID_m^{new}) = A_7 \oplus h(SID_n||SID_m||a_1)$.

Step 2: $\mathcal{A}$ calculates $SK_{nm} = h(SID_m||SID_n||SID_c||A_8)$.

Thus, Akram et al.'s scheme is insecure against session key disclosure attacks.

5.2. Drone Impersonation Attack

In this attack, we assume that $\mathcal{A}$ can capture drones D_n physically and obtain the value $\{SID_n, k_n\}$ stored in the memory of D_n. In order to be able to forward message $\{A_7, A_9\}$ on behalf of legal D_n, then $\mathcal{A}$ has to calculate the value of $A_7 = h(SID_n||SID_m||a_1) \oplus (a_2||a_3||MID_m^{new})$, $A_9 = h(SID_m||SID_n||SID_c||a_2||a_3||A_8)$. $\mathcal{A}$ can compute the A_7 and A_9 through the following below:

Step 1: The adversary $\mathcal{A}$ first intercepts $\{A_4, A_5, A_6\}$ transmitted by the public channel.

Step 2: $\mathcal{A}$ can obtain a_1, a_2, MID_m^{new} by computing $(a_1||a_2||MID_m^{new}) = A_4 \oplus h(SID_n||k_n)$.

Step 3: $\mathcal{A}$ can compute SID_m through $SID_m = A_5 \oplus h(SID_n||SID_c||k_n||a_1)$.

Step 4: $\mathcal{A}$ generates random a_3^* and computes $A_8^* = h(a_1||a_2||a_3^*)$.

Step 5: $\mathcal{A}$ can successfully compute $A_7^* = h(SID_n||SID_m|| a_1) \oplus (a_2||a_3^*||MID_m^{new})$, $A_9^* = h(SID_m||SID_n||SID_c|| a_2||a_3^*||A_8^*)$.

Therefore, Akram et al.'s scheme cannot resist drone impersonation attacks.

5.3. Stolen-Verifier Attack

When $\mathcal{A}$ obtains the table information $\{k_n, SID_n\}$ of the control center, $\mathcal{A}$ can calculate $SK_{nm} = h(SID_m||SID_n||SID_c||A_8)$. The steps are the same as Section 5.1. Therefore, Akram et al.'s scheme is vulnerable to stolen-verifier attacks.

5.4. Perfect Forward Secrecy

Let us suppose that the control center's long-term secret key MSK is compromised by the adversary $\mathcal{A}$, and $\mathcal{A}$ has captured all the previously transmitted messages MID_m, A_1, A_2 and A_4 through the public channel. $\mathcal{A}$ can retrieve SID_m through $(SID_m||a_m) = Dec_{MSK}$ (MID_m), compute $k_m = h(SID_m||MSK)$, $a_1 = A_1 \oplus h(SID_m||SID_c||k_m)$, $SID_n = A_2 \oplus h(SID_m||SID_c||k_m||a_1)$, and $k_n = h(SID_n||MSK)$. Furthermore, $\mathcal{A}$ can retrieve a_1 and a_2 through $(a_1||a_2||MID_m^{new}) = A_4 \oplus h(SID_n||k_n)$ and compute $A_8 = h(a_1||a_2||a_3)$. Finally, $\mathcal{A}$ computes the session key $SK_{nm} = h(SID_m||SID_n||SID_c||A_8)$. Thus, Akram et al.'s scheme does not provide perfect forward secrecy.

5.5. DoS Attack

In the AKA phase, the login process is not executed normally in the remote user (U_m) side. Afterward, the inputs ID_m, PW_m, and Bio_m, U_m compute α_m, SID_m, and k_m. Then, U_m immediately generates a random nonce and computes an authentication request message $\{MID_m, A_1, A_3\}$. Therefore, the adversary $\mathcal{A}$ can send unlimited amounts of login authentication request messages to the control center if $\mathcal{A}$ obtains a stolen/lost mobile device of U_m and inputs a randomly selected identity, password, and biometrics. These messages can threaten the load on the control center. Thus, Akram et al.'s scheme is vulnerable to DoS attacks.

5.6. Correctness

In the user registration phase, the control center calculates the value of MID_m. After that, the MID_m is not transmitted to U_m, and U_m cannot compute it because the MID_m is masked with MSK, which is the control center's secret key. However, in the AKA phase, U_m sends the MID_m to the control center as the first transmitted message. Thus, Akram et al.'s scheme has a correctness problem.

6. Proposed Scheme

The proposed scheme consists of the following phases: (1) initialization; (2) user registration; (3) drone registration; (4) MAKA. We show the flowchart of the proposed scheme in Figure 3. The proposed scheme is lightweight as it uses only the cryptographic one-way hash function and exclusive-OR operations, apart from the fuzzy extractor and PUF technique that is needed for verification at the user side and drone side, respectively.

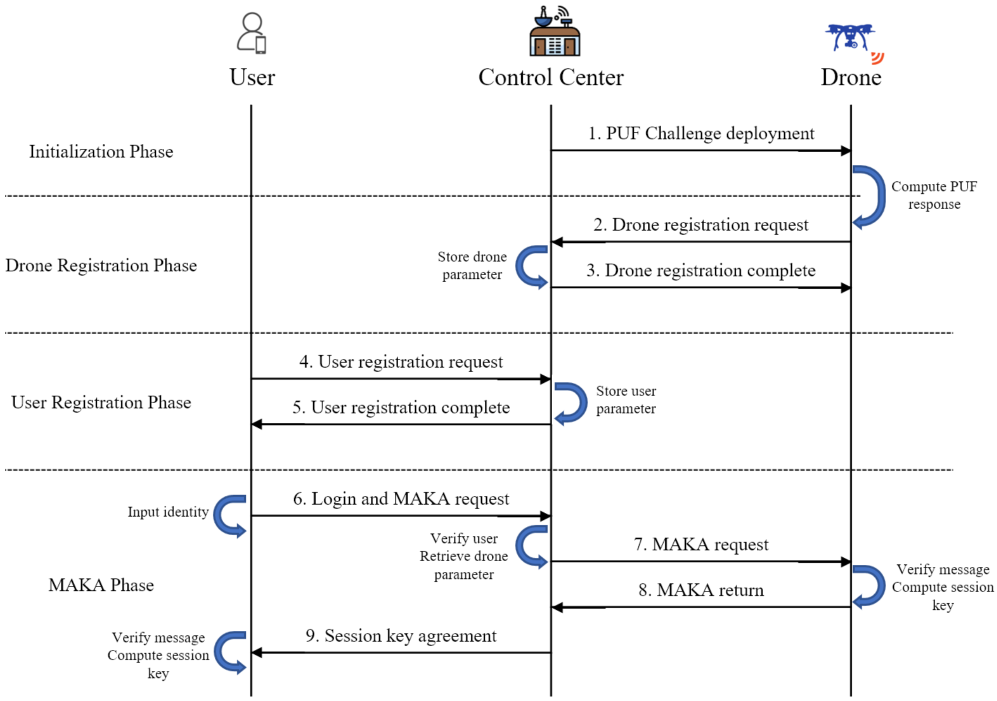

Figure 3. The overall flowchart of the proposed scheme.

6.1. Initialization Phase

This phase describes that the control center selects an identity and a challenge for the drone D_n before the registration phase. Detailed steps are illustrated in Figure 4. Additionally, this phase is performed via a secure channel.

Control Center		**Drone D_n**
Selects ID_n Generates a challenge CH_n	$\xrightarrow{\{ID_n, CH_n\}}$	
		Stores $\{ID_n, CH_n\}$ in the memory

Figure 4. Initialization phase of the proposed scheme.

Step 1: The control center selects an identity ID_n and a challenge CH_n and sends $\{ID_n, CH_n\}$ to the drone D_n.

Step 2: The drone stores $\{ID_n, CH_n\}$ in the memory.

6.2. Drone Registration Phase

In this phase, a drone D_n is registered at the control center to its deployment in the IoD environments through a secure channel. Detailed steps are illustrated in Figure 5.

Step 1: The drone D_n retrieves the challenge CH_n stored in the memory and computes $RE_n = PUF(CH_n)$, and $Gen(RE_n) = (\alpha_n, \beta_n)$. After that, the D_n sends $\{ID_n, CH_n\}$ to the control center.

Step 2: The control center generates a random number a_n and computes $SID_n = h(ID_n||s)$, $k_n = h(SID_n||s||a_n)$, and saves $\{ID_n, SID_n, a_n, CH_n\}$ in the database. Then, the control center sends $\{SID_n, k_n\}$ to the D_n.

Step 3: Finally, the D_n deletes the CH_n and computes $\gamma_n = h(ID_n||\alpha_n) \oplus k_n$, $SID_n^D = h(ID_n||\alpha_n||k_n) \oplus SID_n$, and stores $\{\gamma_n\}$ in its memory.

Drone D_n		**Control Center D_n**
Computes $RE_n = PUF(CH_n)$ $Gen(RE_n) = (\alpha_n, \beta_n)$		
	$\xrightarrow{\{ID_n, CH_n\}}$	
		Random number a_n Computes $SID_n = h(ID_n \vert\vert s)$ $k_n = h(SID_n \vert\vert s \vert\vert a_n)$ Saves $\{ID_n, SID_n, a_n, CH_n\}$ in database
	$\xleftarrow{\{SID_n, k_n\}}$	
Delete CH_n from the memory $\gamma_n = h(ID_n \vert\vert \alpha_n) \oplus k_n$ $SID_n^D = h(ID_n \vert\vert \alpha_n \vert\vert k_n) \oplus SID_n$ Stores $\{\gamma_n\}$		

Figure 5. Drone registration phase of the proposed scheme.

6.3. User Registration Phase

In the user registration phase, a remote user U_m has to register at the control center to access the real-time information from an accessed drone D_n in IoD environments. This procure performs via a secure channel with the following steps. Figure 6 shows the details.

User U_m		**Control Cetner**
Selects ID_m, PW_m Imprint Bio_m Calculate $Gen(Bio_m) = (\alpha_m, \beta_m)$		
	$\xrightarrow{\{ID_m\}}$	
		Random number a_m Computes $SID_m = h(ID_m \vert\vert s)$ $k_m = h(SID_m \vert\vert s \vert\vert a_m)$ $SID_m^* = SID_m \oplus h(s \vert\vert a_m)$ $MID_m = h(SID_m \vert\vert a_m)$ Stores $\{MID_m, SID_m^*, a_m\}$ in database
	$\xleftarrow{\{k_m, SID_m, SID_n, MID_m\}}$	
Computes $\gamma_m = h(ID_m \vert\vert PW_m \vert\vert \alpha_m) \oplus k_m$ $\delta_m = h(\alpha_m \vert\vert k_m \vert\vert SID_m)$ $SID_m^u = h(ID_m \vert\vert PW_m) \oplus SID_m$ $SID_n^u = h(PW_m \vert\vert \alpha_m) \oplus SID_n$ Stores $\{\gamma_m, \delta_m, SID_m^u, SID_n^u, MID_m\}$		

Figure 6. User registration phase of the proposed scheme.

Step 1: The user U_m selects an identity ID_m, a password PW_m, and a biometric template Bio_m. After that, the mobile device calculates $Gen(Bio_m) = (\alpha_m, \beta_m)$. The U_m sends $\{ID_m\}$ to the control center.

Step 2: The control center generates random number a_m and computes $SID_m = h(ID_m||s)$, $k_m = h(SID_m||s||a_m)$, $SID_m^* = SID_m \oplus h(s||a_m)$ and $MID_m = h(SID_m||a_m)$. Then, the control center stores $\{MID_m, SID_m^*, a_m\}$ in the database, and sends $\{k_m, SID_m, SID_n, MID_m\}$ to the U_m.

Step 3: The U_m computes $\gamma_m = h(ID_m||PW_m||\alpha_m) \oplus k_m$, $\delta_m = h(\alpha_m||k_m||SID_m)$, $SID_m^u = h(ID_m||PW_m) \oplus SID_m$, and $SID_n^u = h(PW_m||\alpha_m) \oplus SID_n$, and stores $\{\gamma_m, \delta_m, SID_m^u, SID_n^u, MID_m\}$ in the memory.

6.4. MAKA Phase

The following steps are performed among the U_m, the control center, and an accessed drone D_n through a public channel. To establish a session key for secure communication among them, they need to perform the MAKA processes. Details are illustrated in Figure 7.

User U_m	**Control Center**	**Drone D_n**
Inputs ID_m, PW_m Imprints Bio_m $\alpha_m = Rep(Bio_m, \beta_m)$ $SID_m = h(ID_m\|\|PW_m) \oplus SID_m^u$ $SID_n = h(PW_m\|\|\alpha_m) \oplus SID_n^u$ $k_m = h(ID_m\|\|PW_m\|\|\alpha_m) \oplus \gamma_m$ $\delta_m^* = h(\alpha_m\|\|k_m\|\|SID_m)$ Checks if $\delta_m^* \stackrel{?}{=} \delta_m$ Selects a_1 $A_1 = h(SID_m\|\|SID_c\|\|k_m) \oplus a_1$ $A_2 = h(SID_m\|\|SID_c) \oplus SID_n$ $V_1 = h(SID_m\|\|SID_n\|\|SID_c\|\|k_m\|\|a_1)$		
$\xrightarrow{\{MID_m, A_1, A_2, V_1\}}$		
	Checks whether $MID_m = MID_m^{old}$ or $MID_m = MID_m^{new}$ If $(MID_m == MID_m^{old})$ {Retrieves $\{SID_m^*, a_m\}$ against MID_m^{old}} If $(MID_m == MID_m^{new})$ {Retrieves $\{SID_m^*, a_m\}$ against MID_m^{new}} $SID_m = SID_m^* \oplus h(s\|\|a_m)$ $k_m = h(SID_m\|\|s\|\|a_m)$ $a_1 = A_1 \oplus h(SID_m\|\|SID_c\|\|k_m)$ $SID_n = A_2 \oplus h(SID_m\|\|SID_c)$ $V_1^* = h(SID_m\|\|SID_n\|\|SID_c\|\|k_m\|\|a_1)$ Checks if $V_1^* \stackrel{?}{=} V_1$ $MID_m^{new} = h(SID_m\|\|a_1)$ Updates MID_m^{new} Checks for ID_n, a_n, CH_n against SID_n from its database $k_n = h(SID_n\|\|s\|\|a_n)$ Selects a_2 $A_3 = h(SID_n\|\|k_n) \oplus (a_1\|\|a_2)$ $A_4 = h(SID_n\|\|k_n\|\|a_1) \oplus SID_m$ $A_5 = h(SID_c\|\|ID_n) \oplus CH_n$ $V_2 = h(SID_m\|\|SID_n\|\|SID_c\|\|k_n\|\|a_1\|\|a_2)$	
	$\xrightarrow{\{A_3, A_4, A_5, V_2\}}$	
		Computes $CH_n = A_5 \oplus h(SID_c\|\|ID_n)$ $RE_n = PUF(CH_n)$ $\alpha_n = Rep(RE_n, \beta_n)$ $k_n = \gamma_n \oplus h(ID_n\|\|\alpha_n)$ $SID_n = SID_n^D \oplus h(ID_n\|\|\alpha_n\|\|k_n)$ $(a_1\|\|a_2) = A_3 \oplus h(SID_n\|\|k_n)$ $SID_m = A_4 \oplus h(SID_n\|\|k_n\|\|a_1)$ $V_2^* = h(SID_m\|\|SID_n\|\|SID_c\|\|k_n\|\|a_1\|\|a_2)$ Checks if $V_2^* \stackrel{?}{=} V_2$ Selects a_3 $A_6 = h(SID_m\|\|SID_n\|\|a_1) \oplus (a_2\|\|a_3)$ $A_7 = h(SID_m\|\|SID_n\|\|SID_c)$ $SK = h(A_7\|\|a_1\|\|a_2\|\|a_3)$ $V_3 = h(A_7\|\|a_1\|\|a_3\|\|SK)$
	$\xleftarrow{\{A_6, V_3\}}$	
$(a_2\|\|a_3) = A_6 \oplus h(SID_m\|\|SID_n\|\|a_1)$ $A_7 = h(SID_m\|\|SID_n\|\|SID_c)$ $SK = h(A_7\|\|a_1\|\|a_2\|\|a_3)$ $V_3^* = h(A_7\|\|a_1\|\|a_3\|\|SK)$ Checks if $V_3^* \stackrel{?}{=} V_3$ Updates MID_m^{new}		

Figure 7. MAKA phase of the proposed scheme.

Step 1: The U_m inputs ID_m and PW_m, and imprints Bio_m. After that, U_m computes $\alpha_m = Rep(Bio_m, \beta_m)$, $SID_m = h(ID_m||PW_m) \oplus SID_m^u$, $SID_n = h(PW_m||\alpha_m) \oplus SID_n^u$, $k_m = h(ID_m||PW_m||\alpha_m) \oplus \gamma_m$, and $\delta_m^* = h(\alpha_m||k_m||SID_m)$, and checks $\delta_m^* \stackrel{?}{=} \delta_m$. Then, the U_m generates a random nonce a_1 and calculates $A_1 = h(SID_m||SID_c||k_m) \oplus a_1$, $A_2 = h(SID_m||SID_c) \oplus SID_n$, and $V_1 = h(SID_m||SID_n\ ||SID_c||k_m||a_1)$. The U_m sends $\{MID_m, A_1, A_2, V_1\}$ to the control center.

Step 2: The control center checks whether $MID_m = MID_m^{old}$ or $MID_m = MID_m^{new}$. If $(MID_m == MID_m^{old})$ then, retrieves $\{SID_m^*, a_m\}$ against MID_m^{old}, and if $(MID_m == MID_m^{new})$, retrieves $\{SID_m^*, a_m\}$ against MID_m^{new}. After that, the control center computes $SID_m = SID_m^* \oplus h(s||a_m)$, $k_m = h(SID_m||s||a_m)$, $a_1 = A_1 \oplus h(SID_m||SID_c||\ k_m)$, $SID_n = A_2 \oplus h(SID_m||SID_c)$, and $V_1^* = h(SID_m||SID_n||SID_c||k_m||a_1)$. If

$V_1^* \stackrel{?}{=} V_1$ is correct, the control center computes $MID_m^{new} = h(SID_m||a_1)$ and updates MID_m^{new}. Then, the control center checks for ID_n, a_n, CH_n against SID_n from its database and computes $k_n = h(SID_n||s||a_n)$. The control center calculates $A_3 = h(SID_n||k_n) \oplus (a_1||a_2)$, $A_4 = h(SID_n||k_n||a_1) \oplus SID_m$, $A_5 = h(SID_c||ID_n) \oplus CH_n$, and $V_2 = h(SID_m||SID_n||SID_c||k_n||a_1||a_2)$ and sends $\{A_3, A_4, A_5, V_2\}$ to the drone.

Step 3: The drone D_n computes $CH_n = A_5 \oplus h(SID_c||ID_n)$, $RE_n = PUF(CH_n)$, $\alpha_n = Rep(RE_n, \beta_n)$, $k_n = \gamma_n \oplus h(ID_n||\alpha_n)$, $SID_n = SID_n^D \oplus h(ID_n||\alpha_n||k_n)$, $(a_1||a_2) = A_3 \oplus h(SID_n||k_n)$, $SID_m = A_4 \oplus h(SID_n||k_n||a_1)$, and $V_2^* = h(SID_m||SID_n||SID_c ||k_n||a_1||a_2)$. If $V_2^* \stackrel{?}{=} V_2$ is correct, the D_n generates a random nonce a_3, and calculates $A_6 = h(SID_m||SID_n||a_1) \oplus (a_2||a_3)$, $A_7 = h(SID_m||SID_n||SID_c)$, $SK = h(A_7||a_1||a_2||a_3)$, and $V_3 = h(A_7||a_1||a_3||SK)$. Then, the D_n sends $\{A_6, V_3\}$ to the U_m.

Step 4: The U_m computes $(a_2||a_3) = A_6 \oplus h(SID_m||SID_n ||a_1)$, $A_7 = h(SID_m||SID_n|| SID_c)$, $SK = h(A_7||a_1||a_2||a_3)$, and $V_3^* = h(A_7||a_1||a_3||SK)$ and checks $V_3^* \stackrel{?}{=} V_3$. Then, the U_m updates MID_m^{new}.

7. Security Analysis

To prove the security robustness of the proposed scheme, BAN logic, RoR model, and AVISPA simulation are used in this section. Using informal security analysis, we analyze the theoretical security of the proposed scheme.

7.1. BAN Logic

BAN logic [10] is a widely known formal proof used by many researchers to show mutual authentication of protocols [26–28]. Therefore, we apply the proposed scheme to BAN logic proof and verify mutual authentication. We introduce notations and descriptions for BAN logic in Table 3.

Table 3. Basic notations in BAN logic.

Notation	Description
$\mathcal{PR}_1, \mathcal{PR}_2$	Principals
MSG_1, MSG_2	Statements
SK	Session key
$\mathcal{PR}_1 \mid\equiv MSG_1$	$\mathcal{PR}_1$ **believes** MSG_1
$\mathcal{PR}_1 \mid\sim MSG_1$	$\mathcal{PR}_1$ once **said** MSG_1
$\mathcal{PR}_1 \Rightarrow MSG_1$	$\mathcal{PR}_1$ **controls** MSG_1
$\mathcal{PR}_1 \triangleleft MSG_1$	$\mathcal{PR}_1$ **receives** MSG_1
$\#MSG_1$	MSG_1 is **fresh**
$(MSG_1)_{KEY}$	MSG_1 is **encrypted** with KEY
$\mathcal{PR}_1 \stackrel{KEY}{\longleftrightarrow} \mathcal{PR}_2$	$\mathcal{PR}_1$ and $\mathcal{PR}_2$ have shared key KEY

7.1.1. Rules

In BAN logic, there are five logical rules: message meaning rule (MMR), nonce verification rule (NVR), jurisdiction rule (JR), belief rule (BR), and freshness rule (FR). Details are as follows.

1. MMR :

$$\frac{\mathcal{PR}_1 \Big| \equiv \mathcal{PR}_1 \stackrel{KEY}{\leftrightarrow} \mathcal{PR}_2, \quad \mathcal{PR}_1 \triangleleft (MSG_1)_{KEY}}{\mathcal{PR}_1 | \equiv \mathcal{PR}_2 | \sim MSG_1}$$

2. NVR :

$$\frac{\mathcal{PR}_1 | \equiv \#(MSG_1), \quad \mathcal{PR}_1 | \equiv \mathcal{PR}_2 \Big| \sim MSG_1}{\mathcal{PR}_1 | \equiv \mathcal{PR}_2 | \equiv MSG_1}$$

3. JR :

$$\frac{\mathcal{PR}_1 | \equiv \mathcal{PR}_2 \Rightarrow MSG_1, \quad \mathcal{PR}_1 | \equiv \mathcal{PR}_2 | \equiv MSG_1}{\mathcal{PR}_1 \Big| \equiv MSG_1}$$

4. BR :

$$\frac{\mathcal{PR}_1 \Big| \equiv (MSG_1, MSG_2)}{\mathcal{PR}_1 \Big| \equiv MSG_1}$$

5. FR :

$$\frac{\mathcal{PR}_1 \Big| \equiv \#(MSG_1)}{\mathcal{PR}_1 \Big| \equiv \#(MSG_1, MSG_2)}$$

7.1.2. Goals

In the proposed scheme, there are four goals for the BAN logic. Let the user, control center, and drone be U_m, CC, and D_n, respectively.

Goal 1: $D_n| \equiv D_n \stackrel{SK}{\longleftrightarrow} U_m$

Goal 2: $D_n| \equiv U_m| \equiv D_n \stackrel{SK}{\longleftrightarrow} U_m$

Goal 3: $U_m| \equiv D_n \stackrel{SK}{\longleftrightarrow} U_m$

Goal 4: $U_m| \equiv D_n| \equiv D_n \stackrel{SK}{\longleftrightarrow} U_m$

7.1.3. Idealized Forms

Three messages, i.e., $\{MID_m, A_1, A_2, V_1\}$, $\{A_3, A_4, A_5, V_2\}$, and $\{A_6, V_3\}$, are transmitted via open channels in the proposed scheme. These messages are converted to idealized forms in BAN logic as below.

Mes_1 : $U_m \rightarrow CC : \{a_1, SID_n\}_{SID_m}$
Mes_2 : $CC \rightarrow D_n : \{a_1, a_2, SID_m\}_{k_n}$
Mes_3 : $D_n \rightarrow U_m : \{a_2, a_3\}_{SID_m}$

7.1.4. Assumptions

We show the assumptions using in BAN logic as follows.

AS_1**:** $CC| \equiv \#(a_1)$
AS_2**:** $D_n| \equiv \#(a_2)$
AS_3**:** $U_m| \equiv \#(a_3)$
AS_4**:** $D_n| \equiv U_m \Rrightarrow (D_n \stackrel{SK}{\longleftrightarrow} U_m)$
AS_5**:** $U_m| \equiv D_n \Rrightarrow (D_n \stackrel{SK}{\longleftrightarrow} U_m)$
AS_6**:** $CC| \equiv CC \stackrel{SID_m}{\longleftrightarrow} U_m$
AS_7**:** $D_n| \equiv CC \stackrel{k_n}{\longleftrightarrow} D_n$
AS_8**:** $U_m| \equiv D_n \stackrel{SID_m}{\longleftrightarrow} U_m$

7.1.5. BAN Logic Proof

Step 1: We can obtain RA_1 from the message Mes_1.

$$RA_1 : CC \lhd \{a_1, SID_n\}_{SID_m}$$

Step 2: We can obtain RA_2 from the rule MMR using RA_1 and AS_6.

$$RA_2 : CC| \equiv U_m| \sim (a_1, SID_n)$$

Step 3: We can obtain RA_3 from the rule FR using S_3 and AS_1.

$$RA_3 : CC| \equiv \#(a_1, SID_n)$$

Step 4: We can obtain RA_4 from the rule NVR using RA_2 and RA_3.

$$RA_4 : CC| \equiv U_m| \equiv (a_1, SID_n)$$

Step 5: We can obtain RA_5 from the message Mes_2.

$$RA_5 : D_n \lhd \{a_1, a_2, SID_m\}_{k_n}$$

Step 6: We can obtain RA_6 from the MMR using RA_5 and AS_7.

$$RA_6 : D_n| \equiv CC| \sim (a_1, a_2, SID_m)$$

Step 7: We can obtain RA_7 from the FR using RA_6 and AS_2.

$$RA_7 : D_n| \equiv \#(a_1, a_2, SID_m)$$

Step 8: We can obtain RA_8 from the NVR using RA_6 and RA_7.

$$RA_8 : D_n| \equiv CC| \equiv (a_1, a_2, SID_m)$$

Step 9: We can obtain RA_9 from the message Mes_3.

$$RA_9 : U_m \lhd \{a_2, a_3\}_{SID_m}$$

Step 10: We can obtain RA_{10} from the MMR using RA_9 and AS_8.

$$RA_{10} : U_m| \equiv D_n| \sim (a_2, a_3)$$

Step 11: We can obtain RA_{11} from the NVR using RA_{10} and AS_3.

$$S_{11} : U_m| \equiv D_n| \equiv (a_2, a_3)$$

Step 12: We can obtain RA_{12} and RA_{13} from RA_8 and RA_{11}. Therefore, U_m and D_n can compute the session key $SK = h(A_7||a_1||a_2||a_3)$, where $A_7 = h(SID_m||SID_n||SID_c)$.

$$RA_{12} : D_n| \equiv U_m| \equiv (D_n \overset{SK}{\longleftrightarrow} U_m) \quad \textbf{(Goal 2)}$$
$$RA_{13} : U_m| \equiv D_n| \equiv (D_n \overset{SK}{\longleftrightarrow} U_m) \quad \textbf{(Goal 4)}$$

Step 13: We can obtain RA_{14} and RA_{15} from the jurisdiction rule using RA_{12} and AS_4, and RA_{13} and AS_5, respectively.

$$RA_{14} : D_n| \equiv (D_n \overset{SK}{\longleftrightarrow} U_m) \quad \textbf{(Goal 1)}$$
$$RA_{15} : U_n| \equiv (D_n \overset{SK}{\longleftrightarrow} U_m) \quad \textbf{(Goal 3)}$$

7.2. RoR Model

The Real-or-Random model [9] is a formal proof analysis that proves the session key security of the protocol. Thus, we establish a premise for applying the proposed scheme to the RoR model. There are participants, adversaries and queries in our scheme. Participants are the entities that communicate with each other in the proposed scheme. Therefore, participants are as follows: PAR_U^i, PAR_C^j, and PAR_D^k, where i, j, and k are the instances of user, control center, and drone, respectively. The adversary in RoR model can modify, delete, and eavesdrop the exchanged messages. With this ability, the adversary can perform various queries such as $Execute$, $CorruptDevice$, $Send$, and $Test$. We describe the details of these queries as below.

- $Execute(PAR_U^i, PAR_C^j, PAR_D^k)$: In this query, the adversary eavesdrop messages are transmitted via an open channel. Therefore, the adversary can obtain messages generated from PAR_U^i, PAR_C^j, and PAR_D^k. This query is a passive attack.
- $CorruptDevice(PAR_U^i)$: In this query, the adversary can obtain secret parameters from PAR_U^i using a power analysis attack. Therefore, the query $CorruptDevice$ is an active attack.
- $Send(PAR)$: In this query, the adversary can send messages to all participants PAR_U^i, PAR_C^j, and PAR_D^k. Furthermore, the adversary can obtain returned messages from these participants. Thus, this query is an active attack
- $Test(PAR)$: Before starting the game, an unbiased coin UC is flipped in this query. The adversary obtains $UC = 1$ when the session key is fresh. The adversary can also obtain $UC = 0$ when the session key of the proposed scheme cannot guarantee freshness. If not, the adversary obtains a "null value" $\perp$. To achieve a secure session key agreement, the adversary cannot discriminate between the session key and the random number.

Security Proof

Theorem 1. *The adversary AD attempts to compute the session key $SK = h(A_7||a_1||a_2\ ||a_3)$ in polynomial time. Therefore, we define the possibility that AD breaks the security of the session key as $\mathcal{MA}_{AD}(P)$. Moreover, we define that HA and PU are the range space of the function $h(.)$ and $PUF(.)$, respectively. The number of HA, PU, and Send queries are qu_{ha}, qu_{pu}, and qu_{se}, respectively. We define the secret biometric bits as B_m. At last, we define the Zipf's parameter [29] as C' and s'.*

$$\mathcal{MA}_{AD}(P) \leq \frac{qu_{ha}^2}{|HA|} + \frac{qu_{pu}^2}{|PU|} + 2max\{C'qu_{se}^{s'}, \frac{qu_{se}}{2^{B_m}}\}$$

Proof. The security proof in the proposed scheme is composed of five games GA_n $(n = 0, 1, 2, 3, 4)$. Before starting the game, we define A_{GA_n} as the probability that AD wins the game and $AD[A_{GA_k}]$ as the advantage of A_{GA_k}. We follow the security proof according to [30–32].

GA_0 **:** In GA_0, the adversary selects a random bit r. Thus, we obtain the following equation.

$$\mathcal{MA}_{AD}(P) = |2AD[A_{GA_0}] - 1| \tag{1}$$

GA_1 **:** In GA_1, the adversary eavesdrops messages $\{MID_m, A_1, A_2, V_1\}$, $\{A_3, A_4, A_5, V_2\}$, and $\{A_6, V_3\}$ using $Execute$ query. Then, the adversary performs the $Test$ query to obtain the session key $SK = h(A_7||a_1||a_2||a_3)$. To compute SK, the adversary must obtain the random nonces a_1, a_2, and a_3. Moreover, A_7 is composed of SID_m, SID_n, and SID_c, where SID_m is the secret parameter of user. Therefore, the adversary cannot calculate SK. Therefore, we can obtain the following equation.

$$|AD[A_{GA_1}]| = |AD[A_{GA_0}]| \tag{2}$$

GA_2 **:** In GA_2, the adversary utilizes $Send$ and HA to attack the network. However, all of the parameters are masked in a cryptographic hash function that can prevent the hash collision problem. For this reason, the adversary cannot obtain the session key SK. According to the birthday paradox [33], we can obtain the following inequation.

$$|AD[A_{GA_2}] - AD[A_{GA_1}]| \leq \frac{qu_{ha}^2}{|HA|} \tag{3}$$

GA_3 **:** Similar to GA_2, the adversary utilizes queries *Send* and *PU* in this game. According to Section 3.4, the PUF is extremely difficult or impossible to clone. This means the adversary has no advantage in GA_3.

$$|AD[A_{GA_3}] - AD[A_{GA_2}]| \leq \frac{qu_{pu}^2}{|PU|} \tag{4}$$

GA_4 **:** This game is the final game in which the adversary extracts secret parameters $\{\gamma_m, \delta_m, SID_m^u, SID_n^u, MID_m\}$ from the device of the user using the query *CorruptDevice*. The adversary attempts to calculate SK from these parameters. However, each parameter consists of a password and the biometrics of a user, and this means that the adversary must guess the password and biometrics at the same time. Since this task is computationally infeasible, the adversary cannot compute SK. Therefore, we can obtain the following inequation using Zipf's law [29].

$$|AD[A_{GA_4}] - AD[A_{GA_2}]| \leq max\{C'qu_{se}^{s'}, \frac{qu_{se}}{2^{B_m}}\} \tag{5}$$

After the game, the adversary guesses the result bits r, and we can make the following equation.

$$AD[A_{GA_4}] = \frac{1}{2} \tag{6}$$

We can calculate and obtain Equation (7) using (1) and (2).

$$\frac{1}{2}\mathcal{MA}_{AD}(P) = |AD[A_{GA_0}] - \frac{1}{2}| = |AD[A_{GA_1}] - \frac{1}{2}| \tag{7}$$

Then, we can calculate and obtain Equation (8) from (6) and (7).

$$\frac{1}{2}\mathcal{MA}_{AD}(P) = |AD[A_{GA_1}] - AD[A_{GA_4}]| \tag{8}$$

The result (9) can be obtained using the triangular inequality.

$$\begin{aligned}
\frac{1}{2}\mathcal{MA}_{AD}(P) &= |AD[A_{GA_1}] - AD[A_{GA_4}]| \\
&\leq |AD[A_{GA_1}] - AD[A_{GA_3}]| \\
&\quad + |AD[A_{GA_3}] - AD[A_{GA_4}]| \\
&\leq |AD[A_{GA_1}] - AD[A_{GA_2}]| \\
&\quad + |AD[A_{GA_2}] - AD[A_{GA_3}]| \\
&\quad + |AD[A_{GA_3}] - AD[A_{GA_4}]| \\
&\leq \frac{qu_{ha}^2}{2|HA|} + \frac{qu_{pu}^2}{2|PU|} + max\{C'qu_{se}^{s'}, \frac{qu_{se}}{2^{B_m}}\}
\end{aligned} \tag{9}$$

After multiplying (9) by 2, we can obtain the required result inequation.

$$\mathcal{MA}_{AD}(P) \leq \frac{qu_{ha}^2}{|HA|} + \frac{qu_{pu}^2}{|PU|} + 2max\{C'qu_{se}^{s'}, \frac{qu_{se}}{2^{B_m}}\}$$

Therefore, we can demonstrate that the proposed scheme can ensure the session key security by proving the Theorem 1. □

7.3. AVISPA Simulation

AVISPA [7,8] is a simulation tool that proves the security robustness of the proposed scheme against replay and MITM attacks. Therefore, various security protocols [23,34,35] are proved by using AVISPA. In this section, we explain the main data flow of AVISPA and show the simulation result.

Firstly, we need to write the proposed scheme as a programming language named "High-Level Protocol Specification Language (HLPSL)" in AVISPA. After writing in HLPSL code, the proposed scheme is converted to "Intermediate Format (IF)". Then, the translator in AVISPA starts analyzing the IF through the four backends: "On-the-Fly Model Checker (OFMC)", "Three Automata based on Automatic Approximations for Analysis of Security Protocol (TA4SP)", "SAT-based Model Checker (SATMC)", and "Constraint Logic-based Attack Searcher (CL-AtSe)". Because OFMC and CL-AtSe only support an exclusive-OR operator, the proposed scheme is executed in these backends. The analyzed result is recorded and summarized in the "Output Format (OF)". If there is a result of "SAFE" in OF, we can demonstrate that the proposed scheme can prevent replay and MITM attacks.

In AVISPA, we define roles to be suitable for the proposed scheme. Therefore, there are three roles in the proposed scheme: the user US, control center CC, and drone DR. Moreover, we show the session and environment roles in Figure 8.

```
role session(DR, CC, US : agent, SKusdr, SKccdr, SKccus : symmetric_key, PUF,H : hash_func)
def=
local SN1, SN2, SN3, RV1, RV2, RV3 : channel(dy)
composition
user(US, CC, DR, SKusdr, SKccus, SKccdr, PUF, H, SN1, RV1)
/\ controlcenter(US, CC, DR, SKusdr, SKccus, SKccdr, PUF, H, SN2, RV2)
/\ drone(US, CC, DR, SKusdr, SKccus, SKccdr, PUF, H, SN3, RV3)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role environment()
def=
const dr, cc, us : agent,
                puf, h : hash_func,
                skusdr, skccdr, skccus : symmetric_key,
                us_cc_aa1, us_dr_aa1, cc_dr_aa2, dr_us_aa3 : protocol_id,
                sp1, sp2, sp3, sp4, sp5, sp6 : protocol_id,
                idi : text
intruder_knowledge = {h, idi}
composition
session(us, cc, dr, skusdr, skccus, skccdr, puf, h)
/\session(i, cc, dr, skusdr, skccus, skccdr, puf, h)
/\session(us, i, dr, skusdr, skccus, skccdr, puf, h)
/\session(us, cc, i, skusdr, skccus, skccdr, puf, h)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
goal
secrecy_of sp1,sp2
authentication_on us_cc_aa1
authentication_on us_dr_aa1
authentication_on cc_dr_aa2
authentication_on dr_us_aa3
end goal
environment()
```

Figure 8. Session and environment roles written in HLPSL.

Figure 9 shows the role of user US written in HLPSL code. State 1 is the user registration phase that US sends $\{ID_m\}$ to the CC through a secure channel. After receiving return message $\{k_m, SID_m, SID_n, MID_m\}$ from CC, US computes and stores γ_m, δ_m, SID_m^u, and SID_n^u in state 2. Then, US computes a login request message $\{MID_m, A_1, A_2, V_1\}$ to the CC. Note that $witness(US, CC, us\_cc\_aa1, Aa1')$ and $witness(US, DR, us\_dr\_aa1, Aa1')$ are functions to prove the freshness of random nonce a_1. Finally, US receives $\{A_6, V_3\}$ from DR and computes the session key $SK = h(A_7||a_1||a_2||\, a_3)$. The code $request(DR, US, dr\_us\_aa3, Aa3')$ means the acceptance of freshness for a_3.

```
%%%AVISPA Simulation
role user(DR, CC, US : agent, SKusdr, SKccus, SKccdr : symmetric_key, PUF,H :
hash_func, SND,RCV : channel(dy))

played_by US
def=
local State : nat,
                IDn, CHn, REn, SIDn, Kn, An, IDm, SIDm, Km, MIDm, Am, S, PWm,
BIOm, Gamma, Delta, SIDum, SIDun : text,
                A1, A2, A3, A4, A5, A6, A7, V1, V2, V3, Aa1, Aa2, Aa3, SIDc, SK:
text
                const sp1, us_cc_aa1, us_dr_aa1, cc_dr_aa2, dr_us_aa3 : protocol_id

init State := 0
transition
%%%%%%%User registration phase
1. State = 0 /\ RCV(start) =|>
State' := 1
/\ SND({IDm}_SKccus)

2. State = 1 /\ RCV({H(H(IDm. S). S. Am). H(IDm. S). H(IDn. S)}_SKccus) =|>
State' := 2
/\ Gamma' := xor(H(IDm. PWm. BIOm), H(H(IDm. S). S. Am))
/\ Delta' := H(BIOm. H(H(IDm. S). S. Am). H(IDm. S))
/\ SIDum' := xor(H(IDm. PWm), H(IDm. S))
/\ SIDun' := xor(H(PWm. BIOm), H(IDn. S))
%login and authentication phase
/\ Aa1' := new()
/\ A1' := xor(H(H(IDm. S). SIDc. H(H(IDm. S). S. Am)), Aa1')
/\ A2' := xor(H(H(IDm. S). SIDc), H(IDn. S))
/\ V1' := H(H(IDm. S). H(IDn. S). SIDc. H(H(IDm. S). S. Am). Aa1')
/\ SND(H(H(IDm. S). Am). A1'. A2'. V1')
/\ witness(US,CC,us_cc_aa1,Aa1')
/\ witness(US,DR,us_dr_aa1,Aa1')

3. State = 2 /\ RCV(xor(xor(H(H(IDm. S). H(IDn. S). Aa1'), Aa2'), Aa3'). H(H(H(IDm. S).
H(IDn. S). SIDc). Aa1'. Aa3'. H(H(H(IDm. S). H(IDn. S). SIDc). Aa1'. Aa2'. Aa3'))) =|>
State' := 3
/\SK' := H(H(H(IDm. S). H(IDn. S). SIDc). Aa1'. Aa2'. Aa3')
/\request(DR,US,dr_us_aa3,Aa3')

end role
```

Figure 9. User role written in HLPSL.

The AVISPA result is shown in Figure 10. As we mentioned before, we execute the proposed scheme in OFMC and CL-AtSe backends, and the summary of the result is "SAFE". Therefore, we prove that the proposed scheme can prevent replay and MITM attacks.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/DAPSCS.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 7.69s
  visitedNodes: 1608 nodes
  depth: 12 plies
```

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/DAPSCS.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 3 states
  Reachable  : 0 states
  Translation: 0.09 seconds
  Computation: 0.00 seconds
```

Figure 10. AVISPA result.

7.4. Informal Security Analysis

We conduct an informal analysis of the proposed scheme to demonstrate the theoretical security robustness. Details are as below.

7.4.1. Stolen/lost Mobile Device Attack

If an adversary $\mathcal{A}$ obtains a lost mobile device of U_m, it can extract secret parameters $\{\gamma_m, \delta_m, SID^u_m, SID^u_n, MID_m\}$ using power analysis attacks. However, all of secret parameters are masked in the identity ID_m, password PW_m, and biometrics Bio_m information. Therefore, $\mathcal{A}$ must guess ID_m, PW_m, and Bio_m at the same time and this process is not practical. Thus, the proposed scheme is secure against stolen/lost mobile device attacks.

7.4.2. Offline Password-Guessing Attack

An adversary $\mathcal{A}$ can attempt an offline guessing attack using $\{MID_m, A_1, A_2, V_1\}$, $\{A_3, A_4, A_5, V_2\}$ and $\{A_6, V_3\}$, and the extracted values $\{\gamma_m, \delta_m, SID^u_m, SID^u_n, MID_m\}$, $\{\gamma_n\}$ from mobile device and drone, respectively. Using a password dictionary, $\mathcal{A}$ can guess PW^*_A. However, $\mathcal{A}$ cannot know that PW^*_A is valid or not. It is because δ_m is masked with biometric secret key α_m. Therefore, the proposed scheme prevents offline password-guessing attacks.

7.4.3. Impersonation Attack

(1) User impersonation attack: In this attack, an adversary $\mathcal{A}$ tries to disguise a legitimate user U_m. $\mathcal{A}$ has to make a valid login request message $\{MID_m, A_1, A_2, V_1\}$. $\mathcal{A}$ can obtain MID_m from the mobile device. However, without having the credentials SID_m, SID_n, and k_m, it is a difficult task for $\mathcal{A}$ to calculate MID_m, A_1, A_2, V_1. Thus, $\mathcal{A}$ cannot generate a valid login request message on behalf of U_m. Hence, the proposed scheme provides protection against user impersonation attacks.

(2) Control center impersonation attack: For this attack, let us suppose that $\mathcal{A}$ tries to send the message $\{A_3, A_4, A_5, V_2\}$ to the D_n on behalf of the CC. However, without having the credentials SID_m, SID_n, k_n, ID_n, and random nonce a_1, it is computationally hard for $\mathcal{A}$ to make a valid message. Therefore, the proposed scheme is resilient against the CC impersonation attack.

(3) Drone impersonation attack: This attack is a disguise attack in which a malicious adversary $\mathcal{A}$ conceals its identity information and attempts to behave as D_n. To do this, $\mathcal{A}$ computes $CH^*_A = A_3 \oplus h(ID_n||\gamma_n)$. Since $PUF(.)$ is a physical unclonable circuit, $\mathcal{A}$ cannot compute RE_n. Therefore, it is impossible to compute $\alpha_n = Rep(RE_n, \beta_n)$, $SID_n = h(ID_n||\alpha_n)$, $k_n = \gamma_n \oplus SID_n$, $(SID_m||a_1||a_2) = A_2 \oplus h(SID_n||SID_c||k_n)$ to calculate $A_4 = h(SID_m||SID_n||a_1) \oplus (a_2||a_3)$. Thus, the proposed scheme can prevent drone impersonation attacks.

7.4.4. Replay and MITM Attacks

In the proposed scheme, all messages are masked in random nonce a_1, a_2, and a_3 to maintain the freshness. Moreover, each participant, e.g., remote user, control center, drone, checks the validity of the message by calculating and checking V^*_1, V^*_2, and V^*_3. Therefore, the proposed scheme can prevent replay and MITM attacks.

7.4.5. Physical and Cloning Attacks

For this attack, an adversary $\mathcal{A}$ intercepts a drone D_n and extracts the secret parameters $\{\gamma_n\}$ from the memory. However, $\mathcal{A}$ cannot compute the session key $SK = h(A_7||a_1||a_2||a_3)$ because each parameter in the message $\{A_3, A_4, A_5, V_2\}$ is masked in the PUF technology, which has an unclonable property. Thus, $\mathcal{A}$ cannot obtain any advantages from D_n, and this means that the proposed scheme is secure against physical or cloning attacks.

7.4.6. Privileged Insider Attack

In this attack, an adversary $\mathcal{A}$ is a privileged insider of the proposed system. Thus, $\mathcal{A}$ can obtain the registration request message $\{ID_m\}$ and secret parameters $\{\gamma_m, \delta_m, SID^u_m, SID^u_n, MID_m\}$ from the remote user U_m. However, without having PW_m and biometric secret key α_m of U_m, deriving secret credentials $SID_m = h(ID_m||PW_m) \oplus SID^u_m$ and $k_m = h(ID_m||PW_m||\alpha_m) \oplus \gamma_m$ is computationally infeasible. Thus, the proposed scheme prevents privileged insider attacks.

7.4.7. Ephemeral Security Leakage Attack

To prevent this security attack, the proposed scheme must maintain security even if random numbers are leaked. Thus, $\mathcal{A}$ obtains a_1, a_2, a_3, which are used during the AKA phase. However, $\mathcal{A}$ cannot calculate SID_m, k_m, and k_n without knowing the secret key s to the control center. Additionally, $\mathcal{A}$ cannot obtain any advantages to impersonate as a legitimate user U_m. Thus, the proposed scheme prevents ephemeral secret leakage (ESL) attacks.

7.4.8. Stolen-Verifier Attack

We can assume that an adversary $\mathcal{A}$ obtains table data $\{ID_n, SID_n, a_n, CH_n\}$ and $\{MID_m, SID^*_m, a_m\}$ from the database of the control center and attempts to calculate the session key $SK = h(A_7||a_1||a_2||a_3)$ or impersonate the control center. However, $\mathcal{A}$ cannot calculate the secret parameter SID_m, k_m and k_n without the secret keys of the control center and also cannot obtain random number a_1, a_2, a_3. Thus, $\mathcal{A}$ cannot compute SK or impersonate the control center. This means that the proposed scheme is resilient to stolen-verifier attacks.

7.4.9. User Anonymity and Untraceability

An adversary $\mathcal{A}$ cannot reveal the real identity ID_m of a legitimate user because of a cryptographic one-way hash function $h(.)$ masks ID_m with the secret key of the control center. Therefore, the proposed scheme provides the user's anonymity.

7.4.10. Perfect Forward Secrecy

If the master key s of the control center is leaked to an adversary $\mathcal{A}$, it can attempt to compute SK to attack the previous session. However, $\mathcal{A}$ cannot obtain the SK because $SK = h(A_7||a_1||a_2||a_3)$ does not include s. Moreover, if master secret key s of the control center is compromised, $\mathcal{A}$ cannot obtain $SID_m, SID_n, a_1, a_2, a_3$ because $\mathcal{A}$ cannot compute $SID_m = h(ID_m||s)$ without the real identity of the U_m, $SID_n = h(ID_n||\alpha_n)$ and without the secret key α_n. Therefore, $\mathcal{A}$ does not obtain any advantages over SK. This means that the proposed scheme guarantees perfect forward secrecy.

7.4.11. Mutual Authentication

In the MAKA phase, there are three messages $\{MID_m, A_1, A_2, V_1\}$, $\{A_3, A_4, A_5, V_2\}$, $\{A_6, V_3\}$ transmitted via public channels. Thus, each participant checks the legitimacy of the other participants and messages using V_1, V_2, and V_3 in the proposed scheme. If this process is successful, we can ensure authentication. Thus, the proposed scheme guarantees mutual authentication.

7.4.12. DoS Attack

If an adversary $\mathcal{A}$ tries to transmit $\{MID_m, A_1, A_2, V_1\}$ to the control center as a replay message, $\mathcal{A}$ has to pass the login phase by verifying the values of $\delta_m = h(\alpha_m||k_m||SID_m)$. However, $\mathcal{A}$ cannot construct a valid δ_m because $\mathcal{A}$ cannot obtain α_m, k_m, SID_m. Therefore, the replay message would not be sent to the control center. Thus, this proposed scheme can resist DoS attacks.

7.4.13. Drone Capture Attack

If an adversary $\mathcal{A}$ captures a drone D_n and obtains $\{\gamma_n\}$, $\mathcal{A}$ can try to threaten another legitimate drone D_{n1}. However, all of the drones are secure in PUF technology according to Section 7.4.5, and $\gamma_n = h(ID_n||\alpha_n) \oplus k_n$ is an independent parameter. Therefore, the proposed scheme can prevent drone capture attacks.

7.4.14. Session Key Disclosure Attack

To compute the session key $SK = h(A_7||a_1||a_2||a_3)$, an adversary $\mathcal{A}$ has to obtain SID_m, SID_n, a_1, a_2 and a_3. However, $\mathcal{A}$ cannot obtain any of these values because SID_m and SID_n are masked with secret key s and a_1, a_2 and a_3 are random numbers that are temporarily used in a session. Therefore, the proposed scheme is secure against session key disclosure attacks.

8. Performance Analysis

We demonstrate the security features of the proposed scheme with a related scheme [4,14,18,21,24] in terms of "security functionalities", "communication costs", and "computation costs".

8.1. Security Features Comparison

In order to provide visualized information, we offer comprehensive security properties of the proposed scheme and related schemes [4,14,17,18,21,24] in a table. As shown in Table 4, we consider various security functionalities and attacks, including "stolen smart card/mobile device", "offline password guessing ", "impersonation", "replay", "privileged-insider", "physical and cloning", "ESL", "verification table leakage", "user anonymity", "perfect forward secrecy", "mutual authentication", "DoS", "untraceability", "device/drone capture", and "correctness". Thus, our scheme offers secure and functional features as compared to the related schemes [4,14,18,21,24].

8.2. Communication Costs Comparison

We demonstrate the comparison analysis for communication costs of the proposed scheme with the other related schemes [4,14,17,18,21,24]. We refer to [4] and assume that the bit lengths for the hash function, random number, identity, PUF challenge, ECC point, and enc-decryption are 256, random, 160, 32, 160, and 128 bits, respectively. Thus, during the MAKA process of our scheme, the exchanged messages $\{MID_m, A_1, A_2, V_1\}$ require ($256 + 256 + 256 + 256 = 1024$bits), the message $\{A_3, A_4, A_5, V_2\}$ requires ($256 + 256 + 256 + 256 = 1024$bits), and the message $\{A_6, V_3\}$ requires ($256 + 256 = 512$bits), respectively. Table 5 shows the total communication costs of the proposed scheme and the related schemes.

Table 4. Security and functionality features (SFF) comparison.

SFF	[14]	[17]	[18]	[21]	[24]	[4]	Proposed
$SP1$	✓	✓	✓	✓	✓	✓	✓
$SP2$	✓	✓	✓	✓	✓	✓	✓
$SP3$	✓	✓	✓	✓	✓	✓	✓
$SP4$	✓	✓	✓	✓	✓	✓	✓
$SP5$	✓	✓	✓	✓	×	✓	✓
$SP6$	×	×	×	×	×	×	✓
$SP7$	×	✓	✓	✓	✓	✓	✓
$SP8$	✓	✓	✓	✓	×	×	✓
$SP9$	✓	✓	✓	✓	✓	✓	✓
$SP10$	×	✓	✓	✓	✓	✓	✓
$SP11$	✓	✓	✓	✓	✓	✓	✓
$SP12$	✓	✓	✓	✓	✓	×	✓

Table 4. *Cont.*

SFF	[14]	[17]	[18]	[21]	[24]	[4]	Proposed
$SP13$	✓	✓	✓	✓	✓	✓	✓
$SP14$	✓	✓	✓	✓	✓	×	✓
$SP15$	✓	✓	✓	✓	✓	×	✓

Note: $SP1$: stolen smart card/mobile device attack; $SP2$: offline password guessing attack; $SP3$: impersonation attack; $SP4$: replay attack; $SP5$: privileged-insider attack; $SP6$: physical and cloning attack; $SP7$: ESL attack; $SP8$: stolen-verifier attack; $SP9$: user anonymity; $SP10$: perfect forward secrecy; $SP11$: mutual authentication; $SP12$: DoS attack; $SP13$: untraceability; $SP14$: device/drone capture attack; $SP15$: correctness; ✓: Provide or support SFF. ×: Do not provide or support SFF.

Table 5. Comparison study of communication costs.

Schemes	Total Costs	Number of Messages
Ali et al. [14]	1696 bits	3 messages
Wu et al. [17]	3360 bits	3 messages
Tanveer et al. [18]	2240 bits	3 messages
Zhang et al. [21]	5760 bits	4 messages
Tanveer et al. [24]	1856 bits	3 messages
Akram et al. [4]	2304 bits	3 messages
Proposed	2560 bits	3 messages

Although our scheme has slightly higher communication costs than Akram et al.'s scheme [4], we offer better security functionalities and efficient computation costs compared to the related schemes [14,17,18,21,24]. Figure 11 illustrates the total communication costs of the proposed scheme and the related schemes.

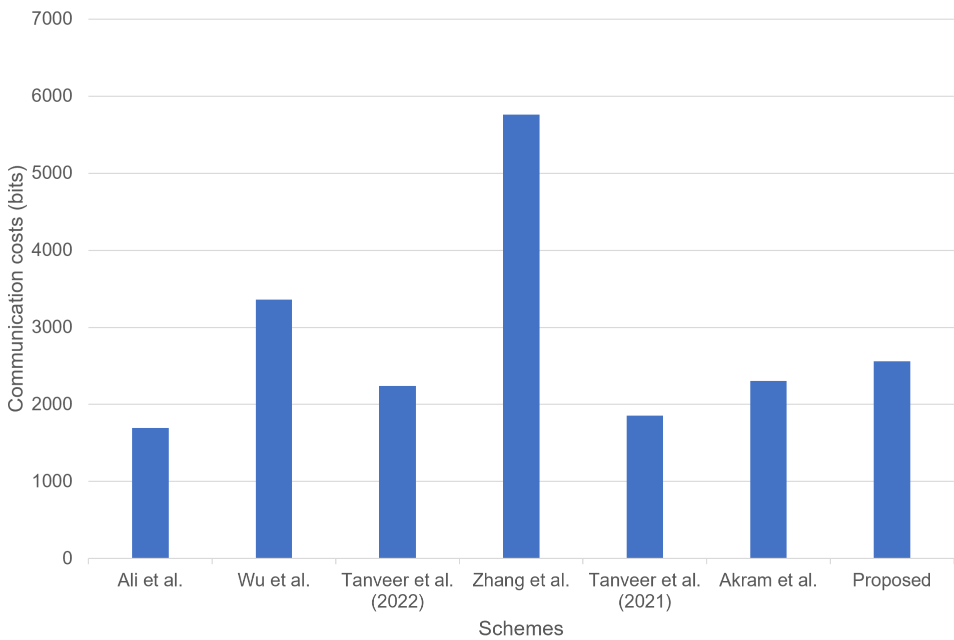

Figure 11. Communication costs comparison [4,14,17,18,21,24].

8.3. Computation Costs Comparison

We estimate the computation costs of the proposed scheme and [4,14,17,18,21,24] in the AKA phase. Referring to [18,21,24], we define that T_H, T_{ECC}, T_{ENC}, T_{FE}, T_{AC}, $T_{pm_{FourQ}}$, T_M, and T_O denote the hash function($\approx$0.029 ms), ECC multiplication($\approx$0.605 ms), enc-decryption time($\approx$0.036 ms), fuzzy extractor($\approx$0.605 ms), AEGIS($\approx$0.07 ms), FourQ point multiplication($\approx$1.199 ms), HMAC($\approx$0.053 ms), and BPV-online function($\approx$2.117 ms), respectively. Table 6 shows the total computation costs of the proposed scheme and the related schemes.

Table 6. Comparison study of computation costs.

Schemes	Remote User Side	Control Center Side	Drone Side	Total	Total Costs (s)
[14]	$10T_H + 1T_{FE}$	$7T_H$	$7T_H$	$24T_H + 1T_{FE}$	≈1.301 ms
[17]	$12T_H + 1T_{FE}$	$9T_H$	$8T_H$	$29T_H + 1T_{FE}$	≈1.446 ms
[18]	$9T_H + 4T_{ENC}$ $+3T_{ECC}$	$4T_H + 3T_{ENC} +$ $1T_{ECC}$	$7T_H + 2T_{ENC}$ $+2T_{ECC}$	$20T_H + 9T_{ENC} + 6T_{ECC}$	≈4.534 ms
[21]	$7T_H + 3T_{pmFourQ}+$ $1T_{ENC} + 1T_O + 1T_M$	$5T_H + 1T_{pmFourQ}$ $+2T_{ENC} + 1T_M$	$4T_H + 1T_{pmFourQ}$ $+1T_{ENC} + 1T_O$	$16T_H + 5T_{pmFourQ}$ $+4T_{ENC} + 2T_O + 2T_M$	≈10.943 ms
[24]	$6T_H + 3T_{AC}$ $+3T_{ECC} + 1T_{FE}$	$2T_H + 1T_{ECC} +$ $3T_{AC}$	$3T_H + 2T_{ECC} +$ $2T_{AC}$	$11T_H + 6T_{ECC}$ $+8T_{AC} + 1T_{FE}$	≈5.114 ms
[4]	$9T_H$	$7T_H + 2T_{ENC}$	$7T_H$	$23T_H + 2T_{ENC}$	≈0.739 ms
Ours	$11T_H + 1T_{FE}$	$11T_H$	$10T_H + 1T_{FE}$	$32T_H + 2T_{FE}$	≈2.138 ms

Compared with the proposed scheme and Akram et al.'s scheme, the proposed scheme consumes more computation costs. However, the proposed scheme utilizes the fuzzy extractor and PUF technologies and, therefore, provides much higher security to the entire IoD network systems than [4]. Figure 12 illustrates that the computational cost (delay) increases at the control center with an increasing number of users.

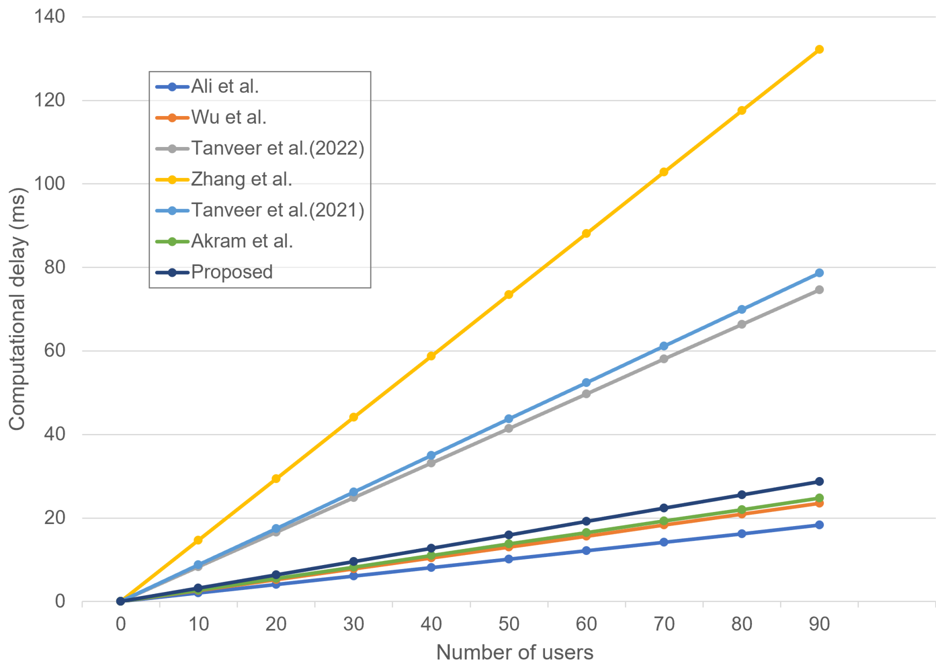

Figure 12. Computational delay at the control center with increasing the AKA requests [4,14,17,18,21,24].

9. Conclusions

In this study, we reviewed Akram et al.'s scheme, which was proposed for secure authentication between users and drones in IoD networks. In Akram et al.'s scheme, there are several security vulnerabilities, such as session key disclosure, drone impersonation, and stolen-verifier attacks. In addition, their scheme cannot ensure perfect forward secrecy and has correctness problems. To overcome the security flaws of their scheme and provide various functional features, we proposed a secure MAKA scheme using biometrics and PUF technologies. The proposed scheme can provide robustness to withstand various attacks, including session key disclosure, verification table leakage, impersonation, ESL, and privileged insider attacks. Moreover, the proposed scheme can achieve mutual authentication, perfect forward secrecy, and anonymity. To prove the session key security and mutual authentication, we analyzed the proposed scheme using an RoR model and BAN logic, respectively. Furthermore, we simulated the proposed scheme using AVISPA and showed that the proposed scheme is resilient against replay and MITM attacks. A comparative study of functionality features, efficiency, and security shows the effectiveness of the proposed scheme. Therefore, we can demonstrate that the proposed scheme has security robustness compared to existing user authentication protocols for IoD environments with reasonable computation and communication overheads. These characteristics show that the proposed scheme can provide users with high security reliability and high-speed communication in IoD environments. In future work, we intend to implement the proposed scheme in real environments using the mobile device as a user, a desktop as a server, and Raspberry PI 4 as a drone.

Author Contributions: Conceptualization, Y.P. (Yohan Park) and D.R.; Formal analysis, D.R. and D.K.; Methodology, Y.P. (Yohan Park) and D.K.; Software, D.K.; Validation, Y.P. (Yohan Park) and Y.P. (Youngho Park); Formal Proof, D.K.; Writing—original draft, and Y.P. (Yohan Park) and D.R.; Writing—review and editing, Y.P. (Yohan Park) and D.K.; Supervision, Y.P. (Yohan Park). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Bisa Research Grant of Keimyung University in 2019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Gharibi, M.; Boutaba, R.; Waslander, S.L. Internet of drones. *IEEE Access* **2016**, *4*, 1148–1162. [CrossRef]
2. Abualigah, L.; Diabat, A.; Sumari, P.; Gandomi, A.H. Applications, deployments, and integration of internet of drones (iod): A review. *IEEE Sens. J.* **2021**, *21*, 25532–25546. [CrossRef]
3. Lin, C.; He, D.; Kumar, N.; Choo, K.K.R.; Vinel, A.; Huang, X. Security and privacy for the internet of drones: Challenges and solutions. *IEEE Commun. Mag.* **2018**, *56*, 64–69. [CrossRef]
4. Akram, M.W.; Bashir, A.K.; Shamshad, S.; Saleem, M.A.; AlZubi, A.A.; Chaudhry, S.A.; Alzahrani, B.A.; Zikria, Y.B. A secure and lightweight drones-access protocol for smart city surveillance. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 19634–19643. [CrossRef]
5. Umar, M.; Islam, S.H.; Mahmood, K.; Ahmed, S.; Ghaffar, Z.; Saleem, M.A. Provable secure identity-based anonymous and privacy-preserving inter-vehicular authentication protocol for VANETS using PUF. *IEEE Trans. Veh. Technol.* **2021**, *70*, 12158–12167. [CrossRef]
6. Herder, C.; Yu, M.D.; Koushanfar, F.; Devadas, S. Physical unclonable functions and applications: A tutorial. *Proc. IEEE* **2014**, *102*, 1126–1141. [CrossRef]
7. AVISPA, T. Automated Validation of Internet Security Protocols and Applications. 2015. Available online: https://www.avispa-project.org/ (accessed on 10 February 2023).
8. Glouche, Y.; Genet, T.; Heen, O.; Courtay, O. A security protocol animator tool for AVISPA. In Proceedings of the ARTIST2 Workshop on Security Specification and Verification of Embedded Systems, Pisa, Italy, 18–20 May 2006; pp. 1–7.
9. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In *Proceedings of the International Workshop on Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 65–84.
10. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst. (TOCS)* **1990**, *8*, 18–36. [CrossRef]
11. Wazid, M.; Das, A.K.; Kumar, N.; Vasilakos, A.V.; Rodrigues, J.J. Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of drones deployment. *IEEE Internet Things J.* **2018**, *6*, 3572–3584. [CrossRef]

12. Teng, L.; Jianfeng, M.; Pengbin, F.; Yue, M.; Xindi, M.; Jiawei, Z.; Gao, C.; Di, L. Lightweight security authentication mechanism towards UAV networks. In Proceedings of the 2019 International Conference on Networking and Network Applications (NaNA), Daegu City, Republic of Korea, 10–13 October 2019; pp. 379–384.
13. Srinivas, J.; Das, A.K.; Kumar, N.; Rodrigues, J.J. TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of drones environment. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6903–6916. [CrossRef]
14. Ali, Z.; Chaudhry, S.A.; Ramzan, M.S.; Al-Turjman, F. Securing smart city surveillance: A lightweight authentication mechanism for unmanned vehicles. *IEEE Access* **2020**, *8*, 43711–43724. [CrossRef]
15. Ever, Y.K. A secure authentication scheme framework for mobile-sinks used in the internet of drones applications. *Comput. Commun.* **2020**, *155*, 143–149. [CrossRef]
16. Deebak, B.D.; AI-Turjman, F. A smart lightweight privacy preservation scheme for IoT-based UAV communication systems. *Comput. Commun.* **2020**, *162*, 102–117. [CrossRef]
17. Wu, T.; Guo, X.; Chen, Y.; Kumari, S.; Chen, C. Amassing the security: An enhanced authentication protocol for drone communications over 5G networks. *Drones* **2022**, *6*, 10–29. [CrossRef]
18. Tanveer, M.; Alkhayyat, A.; Naushad, A.; Kumar, N.; Alharbi, A.G.; et al. RUAM-IoD: A Robust User Authentication Mechanism for the Internet of Drones. *IEEE Access* **2022**, *10*, 19836–19851. [CrossRef]
19. Alladi, T.; Chamola, V.; Kumar, N.; et al. PARTH: A two-stage lightweight mutual authentication protocol for UAV surveillance networks. *Comput. Commun.* **2020**, *160*, 81–90. [CrossRef]
20. Pu, C.; Li, Y. Lightweight authentication protocol for unmanned aerial vehicles using physical unclonable function and chaotic system. In Proceedings of the 2020 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), Orlando, FL, USA, 13–15 July 2020; pp. 1–6.
21. Zhang, N.; Jiang, Q.; Li, L.; Ma, X.; Ma, J. An efficient three-factor remote user authentication protocol based on BPV-FourQ for internet of drones. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 3319–3332. [CrossRef]
22. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
23. Chattaraj, D.; Bera, B.; Das, A.K.; Rodrigues, J.J.; Park, Y. Designing Fine-Grained Access Control for Software-Defined Networks Using Private Blockchain. *IEEE Internet Things J.* **2021**, *9*, 1542–1559. [CrossRef]
24. Tanveer, M.; Kumar, N.; Hassan, M.M.; et al. RAMP-IoD: A robust authenticated key management protocol for the Internet of Drones. *IEEE Internet Things J.* **2021**, *9*, 1339–1353. [CrossRef]
25. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 523–540.
26. Kim, M.; Lee, J.; Park, K.; Park, Y.; Park, K.H.; Park, Y. Design of secure decentralized car-sharing system using blockchain. *IEEE Access* **2021**, *9*, 54796–54810. [CrossRef]
27. 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]
28. Shashidhara, R.; Nayak, S.K.; Das, A.K.; Park, Y. On the design of lightweight and secure mutual authentication system for global roaming in resource-limited mobility networks. *IEEE Access* **2021**, *9*, 12879–12895. [CrossRef]
29. 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]
30. Bagga, P.; Das, A.K.; Wazid, M.; Rodrigues, J.J.; Choo, K.K.R.; Park, Y. On the design of mutual authentication and key agreement protocol in internet of vehicles-enabled intelligent transportation system. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1736–1751. [CrossRef]
31. Son, S.; Lee, J.; Park, Y.; Park, Y.; Das, A.K. Design of blockchain-based lightweight V2I handover authentication protocol for VANET. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1346–1358. [CrossRef]
32. Wazid, M.; Bagga, P.; Das, A.K.; Shetty, S.; Rodrigues, J.J.; Park, Y. AKM-IoV: Authenticated key management protocol in fog computing-based Internet of vehicles deployment. *IEEE Internet Things J.* **2019**, *6*, 8804–8817. [CrossRef]
33. Boyko, V.; MacKenzie, P.; Patel, S. Provably secure password-authenticated key exchange using Diffie-Hellman. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 156–171.
34. 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]
35. Ryu, J.; Oh, J.; Kwon, D.; Son, S.; Lee, J.; Park, Y.; Park, Y. Secure ECC-based three-factor mutual authentication protocol for telecare medical information system. *IEEE Access* **2022**, *10*, 11511–11526. [CrossRef]