sensors

MDPI

Article

A PUF-Based Secure Authentication and Key Agreement Scheme for the Internet of Drones

Jihye Choi, Seunghwan Son, Deokkyu Kwon * and Youngho Park *

School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea; jihye@knu.ac.kr (J.C.); sonshawn@knu.ac.kr (S.S.)
* Correspondence: kdk145@knu.ac.kr (D.K.); parkyh@knu.ac.kr (Y.P.); Tel.: +82-53-950-7842 (Y.P.)

Abstract: The Internet of Drones (IoD) is an emerging industry that offers convenient services for humans due to the high mobility and flexibility of drones. The IoD substantially enhances human life by enabling diverse drone applications across various domains. However, a malicious adversary can attempt security attacks because communication within an IoD environment is conducted through public channels and because drones are vulnerable to physical attacks. In 2023, Sharma et al. proposed a physical unclonable function (PUF)-based authentication and key agreement (AKA) scheme for the IoD. Regrettably, we discover that their scheme cannot prevent impersonation, stolen verifier, and ephemeral secret leakage (ESL) attacks. Moreover, Sharma et al.'s scheme cannot preserve user untraceability and anonymity. In this paper, we propose a secure and lightweight AKA scheme which addresses the shortcomings of Sharma et al.'s scheme. The proposed scheme has resistance against diverse security attacks, including physical capture attacks on drones, by leveraging a PUF. Furthermore, we utilize lightweight operations such as hash function and XOR operation to accommodate the computational constraints of drones. The security of the proposed scheme is rigorously verified, utilizing "Burrows–Abadi–Needham (BAN) logic", "Real-or-Random (ROR) model", "Automated Validation of Internet Security Protocols and Application (AVISPA)", and informal analysis. Additionally, we compare the security properties, computational cost, communication cost, and energy consumption of the proposed scheme with other related works to evaluate performance. As a result, we determine that our scheme is efficient and well suited for the IoD.

Keywords: Internet of Drones; PUF; authentication; cryptanalysis; security

Academic Editors: A.S.M. Kayes, Wenny Rahayu and Ahmad Salehi Shahraki

Received: 7 January 2025
Revised: 31 January 2025
Accepted: 5 February 2025
Published: 6 February 2025

Citation: Choi, J.; Son, S.; Kwon, D.; Park, Y. A PUF-Based Secure Authentication and Key Agreement Scheme for the Internet of Drones. *Sensors* **2025**, *25*, 982. https://doi.org/10.3390/s25030982

1. Introduction

The "Internet of Drones (IoD)" [1] is considered as a prominent industry that is shaping the future of human life through the diverse applications and capabilities of drones. With their high mobility and flexibility, drones are ideally suited for performing tasks across various domains [2]. Drones provide effective alternatives for performing tasks that are labor-intensive or challenging for human operators. The IoD is a network architecture which coordinates drone access and manages their operations within the Internet [3]. Generally, the IoD architecture consists of a control station server (CSS), drones, and remote users. The CSS acts as the control center, overseeing drone operations to ensure appropriate functionality and facilitating communication between drones and remote users. Drones are equipped with various sensors, computational capabilities, and communication modules and can connect to a CSS via the Internet to execute a range of tasks [4]. Drones can be deployed in various environments and provide a wide range of services, including traffic

monitoring, aerial photography, delivery, rescue, and surveillance [5]. Drones collect the surrounding data and transmit them to a CSS or share it with remote users through CSS arbitration [6]. This interconnected structure enables drones to offer convenient services to remote users who benefit from enhanced functionality.

Although the IoD presents various advantages for enhancing human life, it still encounters several critical challenges requiring resolution. In the IoD architecture, communication between drones, remote users, and the CSS occurs through public channels [7]. This exposes the IoD system to potential attacks comprising replay, eavesdropping, insider, and man-in-the-middle (MITM) attacks [8,9]. Additionally, drones are susceptible to unauthorized physical access as they operate in open airspace [10]. A malicious attacker can hijack or physically capture a drone to obtain sensitive data and attempt to disrupt drone operation by injecting malicious payloads. Such breaches can compromise user privacy and lead to substantial security risks. To address these vulnerabilities, various security technologies have been proposed for IoD environments, such as intrusion detection system and anti-jamming [11–13]. In this paper, we focus on the authentication and key agreement (AKA) to preserve privacy, determinate identity of network participants, and establish secure communication channels between users and drones. Another pressing challenge is lightweight computation for drones. Drones have limitations of processing capabilities and database capacity [14], which makes them differ from a CSS, which operates in environments with abundant computing power and storage. Computations are completed within a constrained timeframe to eliminate time delay as the IoD services rely on real-time operation. As a result, it is indispensable to design a secure and lightweight AKA scheme for the IoD in order to guarantee efficient performance while maintaining data security and computational efficiency.

In recent years, various AKA schemes have been proposed to provide security for IoD environments [15–18]. However, such schemes suffer from challenges in lightweight operation and resistance to security vulnerabilities, including physical attacks, which are important issues in IoD environments. To overcome these vulnerabilities, Sharma et al. [19] proposed a physical unclonable function (PUF)-based AKA scheme for the IoD in 2023. Their scheme considered the computational limitations of drones by employing the hash function, exclusive-OR (XOR), and PUF. Sharma et al. argued that their scheme defends numerous adversarial attacks, including privileged insider, MITM, replay, and drone capture attacks. Unfortunately, we demonstrate that their scheme cannot prevent impersonation, stolen verifier, and ephemeral secret leakage (ESL) attacks. Specifically, the session key shared by the user and the drone is exposed by an adversary, compromising mutual authentication. Furthermore, their scheme fails to guarantee user untraceability and anonymity. Therefore, we propose a robust and secure AKA scheme that addresses the flaws of Sharma et al.'s scheme. The proposed scheme defends diverse attacks containing drone capture, impersonation, stolen verifier, and ESL attacks. Moreover, the proposed scheme adopts a PUF that is similar to the approach utilized in Sharma et al.'s scheme. Drones can generate a secret key masked with "challenge–response" pair and protect the data stored in their memory using the key. The proposed scheme achieves enhanced security mitigating the security shortcomings of Sharma et al.'s scheme. Our scheme effectively prevents various security threats including impersonation, stolen verifier, and ESL attacks while introducing additional security properties. Moreover, the proposed scheme achieves a better balance between security and cost efficiency. Compared to Sharma et al.'s scheme, our scheme offers improved security without compromising performance or practicality.

1.1. Contributions

This study offers the following major contributions:

- We analyze Sharma et al.'s scheme and indicate the security weaknesses related to impersonation, stolen verifier, and ESL attacks of their scheme. Furthermore, we demonstrate that their scheme does not guarantee mutual authentication, user untraceability, and anonymity.
- We suggest a lightweight and secure AKA scheme to mitigate the drawbacks of Sharma et al.'s scheme. The proposed scheme adopts one-way hash functions and XOR operations, which are suitable for drones with limited computing power. Additionally, we incorporate a PUF to manage the data stored in drones securely and prevent unauthorized accesses to drones.
- We demonstrate that our scheme ensures the robustness against numerous attacks by performing informal analysis. Moreover, we conduct "Burrows–Abadi–Needham (BAN) logic", "Real-or-Random (ROR) model", and "Automated Validation of Internet Security Protocols and Application (AVISPA)", which represent the resilience of our scheme formally.
- We prove that our scheme achieves cost efficiency with respect to computational cost, communication cost, and energy consumption by conducting a comparison between the proposed scheme and other relevant schemes.

1.2. Organization

We discuss associated studies for the IoD in Section 2. We provide an explanation of the IoD architecture model, adversary model, and the properties of a PUF in Section 3. We revisit Sharma et al.'s scheme in Section 4. We conduct a cryptanalysis of Sharma et al.'s scheme to verify that their scheme has security vulnerabilities in Section 5. We propose a secure and cost-effective AKA scheme for the IoD, which remedies the flaws identified in Sharma et al.'s scheme in Section 6. We assess the resilience of the proposed AKA scheme by adopting various examination methods in Section 7. We highlight the robustness and efficiency through a comparative analysis between the proposed and relevant schemes in Section 8. Finally, we wrap up our study with concluding remarks in Section 9.

2. Related Works

The IoD is a rapidly growing industry that attracts significant attention, prompting researchers to develop AKA schemes for secure IoD communication. In 2021, Nikooghadam et al. [20] devised an AKA scheme for smart city surveillance to construct secure communication between user and drone. They used elliptic curve cryptography (ECC) to enhance energy costs more than traditional public-key cryptosystems (e.g., RSA). Unfortunately, Alzahrani et al. [21] indicated that Nikooghadam et al.'s scheme cannot defend stolen verifier and insider attacks, and that it also lacks user anonymity and untraceability. They proposed an AKA scheme between a user and drone that addresses the security vulnerabilities of Nikooghadam et al.'s scheme. However, their scheme still suffers from security attacks, including drone capture and insider attacks, and cannot ensure security properties, including user anonymity, message integrity, and confidentiality [22]. Tanveer et al. [23] presented an AKA protocol for the IoD environment using ECC. They utilized AEGIS and ECC to enhance their scheme. However, the scheme cannot prevent impersonation and drone capture attacks [24]. Dwivedi et al. [25] propounded a data delivery AKA scheme for tactile Internet-enabled IoD. Their scheme employs ECC and blockchain, providing security for various attacks. It also provides user anonymity, unlinkability, and data immutability. However, previously proposed schemes [20,21,23,25] use ECC, which involves high-complexity computation unsuitable for drones. Because drones are constraint with

regard to their computing power, a lightweight authentication protocol is required for the IoD.

Therefore, many researchers have focused on designing protocols with lightweight computational overhead. Ali et al. [26] devised a biometric-based AKA scheme between user and drone for smart city surveillance. Their scheme used lightweight operations such as hash function, XOR operation, and symmetric encryption. Regrettably, the scheme has weaknesses related to server session key disclosure, spoofing, and forgery attacks [27]. Chaudhary et al. [28] designed an anonymous AKA scheme for the IoD. Their scheme uses only an XOR operation and a one-way hash function for computational efficiency. Unfortunately, their scheme is vulnerable to user impersonation attacks and cannot preserve user privacy protection. Lee et al. [29] propounded a lightweight AKA protocol for the IoD using a one-way hash function and an XOR operation. Although they assert that their scheme rectifies the vulnerabilities of Chaudhary et al.'s scheme and is resistant against numerous attacks, it is still susceptible to the physical attacks of drones. Hussain et al. [30] also presented a lightweight authentication protocol for the IoD environment using symmetric encryption, a one-way hash function, and an XOR operation. The analysis of their scheme shows that it can prevent various attacks. However, it cannot defend against impersonation attacks and physical attacks on drones. Pratap et al. [15] suggested an AKA scheme between a user and a drone for the IoD that addresses the resource limitation issue of drones by utilizing hyperelliptic curve cryptography (HECC). Unfortunately, their scheme is susceptible to drone capture attacks. Although all of these schemes [15,26,28–30] are computationally efficient, they exhibit security drawbacks, particularly a susceptibility to drone capture attacks.

To mitigate the risk of physical attack on drones, numerous researchers have carried out studies. Zhang et al. [16] propounded a key management scheme for the IoD. They considered restricted computing power and physical security issue of drones using a PUF and lightweight operations. Tanveer et al. [17] proposed a biometric-based AKA scheme securing information within the IoD infrastructure. They adopted a PUF, a hash function and symmetric encryption to provide secure communication between users and drones. Tanveer et al. [18] also devised a PUF-based authentication scheme, establishing a session key between users and drones. Using a PUF, a hash function, and AEGIS, their scheme addresses the susceptibility and resource constraints of drone communication. Sharma et al. [19] suggested a lightweight and physical attack-resistant AKA scheme for the IoD environment. Regrettably, we identified that Sharma et al.'s has limitations in defending user impersonation, stolen verifier, and ESL attacks. Moreover, user anonymity and untracability are not preserved in their scheme. Therefore, we propose a robust and lightweight AKA scheme to address the shortcomings in Sharma et al.'s scheme. Table 1 represents the summary of the related schemes.

Table 1. Summary of the proposed scheme and related schemes.

Year	Scheme	Contributions	Limitations
2024	[15]	• Proposed a mutual AKA scheme for the IoD environment • Using HECC	• Cannot prevent drone capture attacks • Large computation cost
2024	[16]	• Proposed a lightweight AKA scheme for the IoD environment • Considered computation costs for drones • Using PUF and hash functions	• Cannot prevent replay and privileged insider attacks

Table 1. *Cont.*

Year	Scheme	Contributions	Limitations
2024	[17]	• Proposed a biometric-based AKA scheme for the IoD environment • Using PUF and symmetric encryption	• Large computation cost • Does not consider various security properties
2024	[18]	• Proposed a PUF-based AKA scheme for the IoD environment • Using PUF and AEGIS	• Large computation cost • Does not consider various security properties
2023	[19]	• Introduced a lightweight AKA scheme for the IoD environment • Using PUF to prevent physical attacks on the drones	• Cannot prevent impersonation, stolen verifier, ESL attacks • Cannot ensure user anonymity and untraceability
-	Proposed	• Propose a lightweight AKA scheme between user and drone for the IoD environment • Address the security vulnerabilities of Sharma et al.'s scheme using PUF and hash function • Consider resource limitations of the drones	

3. Preliminaries

In this part, we explain essential concepts and background for a comprehensive understanding of the proposed scheme. We describe the system model, adversary model and the PUF.

3.1. System Model

Figure 1 illustrates the IoD architecture. There are three entities in the proposed system model: control station server (CSS), remote users, and drones. These entities communicate through wireless channels.

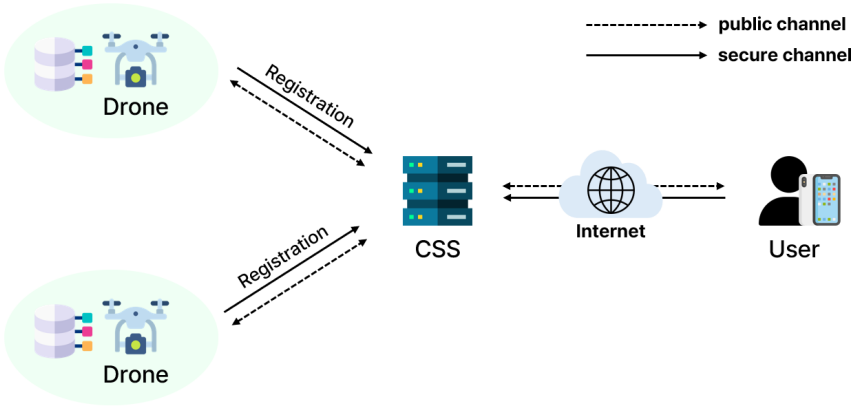

Figure 1. System model for the IoD.

- The CSS is a fully trusted entity. The CSS possesses abundant resources and extensive memory capabilities for controlling system networks. First, the CSS initializes the entire system and registers users and drones. Sensitive data related to users and drones and the information collected by drones are stored in its database. Users and drones authenticate with the mediation of the CSS.
- Remote users need to authenticate with the CSS to access the data stored in the CSS and utilize convenient services. After mutual authentication support from the CSS, users also can directly access the real-time information gathered by drones.
- Drones are deployed in open airspace and gather surrounding information. The information collected by drones is transmitted to the CSS for further processing. PUFs which are embedded in drones protect the secret parameters stored in drones. If a drone is captured, the PUF will be unusable and authentication with the user

or the CSS cannot be completed. Additionally, drones have limited resources and memory capabilities.

3.2. Adversary Model

In this paper, we evaluate the security of AKA scheme by adopting the widely operated threat models "Dolev-Yao (DY)" [31] and "Canetti-Krawczyk (CK)" [32]. The DY and CK models provide the assumptions used to characterize the potential of an adversary. A malicious adversary $\mathcal{A}$ can delete, insert, eavesdrop, revise, and re-transmit messages sent through a public channel. Moreover, $\mathcal{A}$ can obtain and expose session state and temporary session keys or the master key of the CSS. Based on following assumptions, we assess the security of the proposed scheme.

- $\mathcal{A}$ can steal a smart device of a remote user and and use power analysis attacks to retrieve secret credentials stored in the device [33,34].
- $\mathcal{A}$ can be a legitimate user of the system or an outsider and can attempt various attacks using obtained information.
- $\mathcal{A}$ can steal the verification table stored in the CSS and can attempt various attacks using obtained information.
- $\mathcal{A}$ can attempt a variety of attacks, including MITM, privileged insider, replay, and impersonation attacks.

3.3. Physical Unclonable Function

The microstructure of the hardware exhibits unique physical deviations generated by manufacturing disparities. The PUF depends on the characteristic property of the microstructure. The PUF can be considered as fingerprint of the hardware. A PUF includes a unique input–output pair called the "challenge–response" pair. We can use a unique response for authentication and key generation. In this paper, we illustrate the operation of a PUF as $R = PUF(C)$. The notation C indicates a challenge and R indicates a response. We describe the attributes of the PUF as follows:

- A PUF is an unclonable circuit. It is impossible for any $PUF'(\cdot)$ to satisfy $PUF'(C) = PUF(C)$.
- While $PUF(C) = R$ can be computed easily, determining R for a given C within polynomial time is computationally infeasible.
- The output of a PUF is unpredictable [35].

In the proposed scheme, we adopt a PUF to prevent unauthorized physical accesses on drones and protect secret information stored in their memory. Drones can use PUF responses as a secret key using its uniqueness.

4. Review of Sharma et al.'s Scheme

An overview of Sharma et al.'s scheme is provided here. Table 2 summarizes the key notations utilized in Sharma et al.'s scheme. The following outlines its details:

Table 2. Notations.

Notations	Descriptions
CSS	Control server station
U_i	i-th user
D_j	j-th drone
X_{CSS}	Master key of CSS
ID_i	Identity of U_i

Table 2. *Cont.*

PID_i	Pseudo identity of U_i
DID_j	Identity of D_j
$PDID_j$	Pseudo identity of D_j
$\oplus$	Exclusive-OR operation
h	One-way hash function
T_i	Timestamp
SK	Session key

4.1. Initialization Phase

Initially, the CSS chooses its identity ID_{CSS}, a secret key X_{CSS}, and a one-way hash function $h(\cdot)$. Then, the CSS calculates a pseudo-identity $CID_{CSS} = h(X_{CSS}||ID_{CSS})$ and publishes $h(\cdot)$ and CID_{CSS}.

4.2. Drone Registration Phase

Step 1: D_j picks its identity DID_j and a challenge C, and computes $R = PUF(C)$. Then, D_j sends $\{DID_j, C, R\}$ to the CSS through a secure channel.

Step 2: The CSS calculates $PDID_j = h(DID_j||X_{CSS})$ after receiving the message and stores $\{PDID_j, C, R\}$ in the database. Then, the CSS transmits $\{PDID_j\}$ to D_j securely.

Step 3: D_j saves $\{PDID_j, C, CID_{CSS}\}$ to a database.

4.3. User Registration Phase

Step 1: U_i selects ID_i and PW_i. Then, U_i transmits ID_i to the CSS securely.

Step 2: The CSS computes $PID_i = h(ID_i||X_{CSS})$ and $s_i = h(PID_i||X_{CSS})$ upon receiving the message. The CSS sends $\{PID_i, s_i, PDID_j, C\}$ to U_i through a secure channel after storing $\{ID_i, PID_i, s_i\}$ in the database.

Step 3: U_i calculates $s_i^{'} = s_i \oplus h(ID_i||PW_i)$ and $PID_i^{'} = PID_i \oplus h(ID_i||PW_i)$. Finally, U_i stores $\{s_i^{'}, PID_i^{'}, C, PDID_j\}$.

4.4. Authentication and Key Agreement Phase

First, the user U_i transmits an authentication request message to the CSS. The CSS mediates between the user U_i and the drone D_j, verifying whether U_i and D_j are legitimate or not. Finally, U_i and D_j share a session key for establishing secure communication. Figure 2 indicates the processes of authentication and key agreement.

Step 1: U_i inserts identity ID_i and password PW_i, and computes $s_i = s_i^{'} \oplus h(ID_i||PW_i)$, and $PID_i = PID_i^{'} \oplus h(ID_i||PW_i)$. Then, U_i generates a random number r_1 and timestamp T_1 and calculates $M_1 = PDID_j \oplus h(CID_{CSS}||T_1)$, $M_2 = r_1 \oplus h(CID_{CSS}||PID_i||s_i)$, and $V_1 = h(r_1||s_i||C)$. Further, U_i sends the message $\{PID_i, M_1, M_2, V_1, T_1\}$ to the CSS through an open channel.

Step 2: The CSS first checks whether T_1 is valid or not. If it is valid, the CSS retrieves s_i against PID_i and computes $PDID_j = M_1 \oplus h(CID_{CSS}||T_1)$, $r_1^* = M_2 \oplus h(CID_{CSS}||PID_i||s_i)$, and $V_1^* = h(r_1^*||s_i||C)$. Then, the CSS verifies that V_1^* is equal to V_1. If they are identical, the CSS generates a timestamp T_2, and calculates $M_3 = C \oplus h(PDID_j||T_2)$, $M_4 = r_1 \oplus h(CID_{CSS}||R)$, and $V_2 = h(r_1||R||CID_{CSS}||PID_i||PDID_j)$. The CSS transmits $\{PID_i, CID_{CSS}, M_3, M_4, V_2, T_2\}$ over a public channel.

Step 3: D_j verifies the legitimacy of T_2. If it is legitimate, D_j computes $C^* = M_3 \oplus h(PDID_j||T_2)$, $R^* = PUF(C^*)$, $r_1^* = M_4 \oplus h(CID_{CSS}||R^*)$, and $V_2^* = h(r_1^*||R^*||CID_{CSS}||PID_i||PDID_j)$. Then, D_j checks whether V_2^* and V_2 are equal or not. After checking the

equality, D_j generates a random number r_2 and a timestamp T_3. C_{new} is a substring of r_2. After that, D_j calculates $R_{new} = PUF(C_{new})$, $M_5 = R_{new} \oplus h(PDID_j||CID_{CSS}||r_1||T_3)$, $M_6 = R_{new} \oplus r_2$, $V_3 = h(R_{new}||r_2)$, and $SK = h(PID_i||PDID_j\ ||CID_{CSS}||r_1||r_2)$, and sends $\{PDID_j, M_5, M_6, V_3, T_3\}$ to the CSS through a public channel.

Step 4: The CSS checks the validity of T_3. If it is valid, the CSS calculates $R^*_{new} = M_5 \oplus h(PDID_j||\ CID_{CSS}||r_1||T_3)$, $r^*_2 = M_6 \oplus R^*_{new}$, and $V^*_3 = h(R^*_{new}||r^*_2)$. Further, the CSS compares V^*_4 with V_4. If they are equal, the CSS stores $\{C_{new}, R_{new}\}$ in the database and generates a timestamp T_4. The CSS computes $M_7 = r_2 \oplus h(T_4||r_1)$ and $V_4 = h(r_1||r_2)$, and transmits $\{CID_{CSS}, M_7, V_4, T_4\}$ to U_i.

Step 5: U_i verifies that T_4 is legitimate. If legitimate, U_i computes $r^*_2 = M_7 \oplus h(T_4||r_1)$ and $V^*_4 = h(r_1||r^*_2)$. Then, U_i checks that V^*_4 is equal to V_4. If they are equal, U_i stores C_{new} and establishes the session key $SK = h(PID_i||PDID_j||CID_{CSS}||r_1||r_2)$.

U_i	CSS	D_j
Inputs ID_i, PW_i		
Computes		
$s_i = s_i^{'} \oplus h(ID_i\|\|PW_i)$		
$PID_i = PID_i^{'} \oplus h(ID_i\|\|PW_i)$		
Generates r_1, T_1		
Computes		
$M_1 = PDID_j \oplus h(CID_{CSS}\|\|T_1)$		
$M_2 = r_1 \oplus h(CID_{CSS}\|\|PID_i\|\|s_i)$		
$V_1 = h(r_1\|\|s_i\|\|C)$		
	$\{PID_i, M_1, M_2, V_1, T_1\}$ →	
	Checks $\|T_1 - T^*_1\| \leq \triangle T$	
	Computes	
	$PDID_j = M_1 \oplus h(CID_{CSS}\|\|T_1)$	
	$r^*_1 = M_2 \oplus h(CID_{CSS}\|\|PID_i\|\|s_i)$	
	$V^*_1 = h(r^*_1\|\|s_i\|\|C)$	
	Verifies $V^*_1 \stackrel{?}{=} V_1$	
	Generates T_2	
	Computes	
	$M_3 = C \oplus h(PDID_j\|\|T_2)$	
	$M_4 = r_1 \oplus h(CID_{CSS}\|\|R)$	
	$V_2 = h(r_1\|\|R\|\|CID_{CSS}\|\|PID_i\|\|PDID_j)$	
		$\{PID_i, CID_{CSS}, M_3, M_4, V_2, T_2\}$ →
		Checks $\|T_2 - T^*_2\| \leq \triangle T$
		Computes
		$C^* = M_3 \oplus h(PDID_j\|\|T_2)$
		$R^* = PUF(C^*)$
		$r^*_1 = M_4 \oplus h(CID_{CSS}\|\|R^*)$
		$V^*_2 = h(r^*_1\|\|R^*\|\|CID_{CSS}\|\|PID_i\|\|PDID_j)$
		Verifies $V^*_2 \stackrel{?}{=} V_2$
		Generates r_2, T_2 (C_{new}: substring of r_2)
		Computes
		$R_{new} = PUF(C_{new})$
		$M_5 = R_{new} \oplus h(PDID_j\|\|CID_{CSS}\|\|r_1\|\|T_3)$
		$M_6 = R_{new} \oplus r_2$
		$V_3 = h(R_{new}\|\|r_2)$
		$SK = h(PID_i\|\|PDID_j\|\|CID_{CSS}\|\|r_1\|\|r_2)$
		← $\{PDID_j, M_5, M_6, V_3, T_3\}$
	Checks $\|T_3 - T^*_3\| \leq \triangle T$	
	Computes	
	$R^*_{new} = M_5 \oplus h(PDID_j\|\|CID_{CSS}\|\|r_1\|\|T_3)$	
	$r^*_2 = M_6 \oplus R^*_{new}$	
	$V^*_3 = h(R^*_{new}\|\|r^*_2)$	
	Verifies $V^*_3 \stackrel{?}{=} V_3$	
	Stores $\{C_{new}, R_{new}\}$	
	Generates T_4	
	Computes	
	$M_7 = r_2 \oplus h(T_4\|\|r_1)$	
	$V_4 = h(r_1\|\|r_2)$	
	← $\{CID_{CSS}, M_7, V_4, T_4\}$	
Checks $\|T_4 - T^*_4\| \leq \triangle T$		
Computes		
$r^*_2 = M_7 \oplus h(T_4\|\|r_1)$		
$V^*_4 = h(r_1\|\|r^*_2)$		
Verifies $V^*_4 \stackrel{?}{=} V_4$		
Stores $\{C_{new}\}$		
Computes		
$SK = h(PID_i\|\|PDID_j\|\|CID_{CSS}\|\|r_1\|\|r_2)$		

Figure 2. Authentication and key agreement phase of Sharma et al.'s scheme.

5. Cryptanalysis of Sharma et al.'s Scheme

Cryptanalysis is conducted to indicate that Sharma et al.'s scheme cannot prevent impersonation, stolen verifier, ESL attacks and cannot ensure user anonymity and untraceability. The detailed steps are outlined as follows:

5.1. User Impersonation Attack

A malicious adversary $\mathcal{A}$ impersonates a legitimate user using the secret parameters extracted from user's smart device. Then, $\mathcal{A}$ establishes a session key with a drone. The details are outlined below.

Step 1: $\mathcal{A}$ can exploit a power analysis attack to extract the secret information $\{s_i^{'}, PID_i^{'}, C, PDID_j\}$ stored on the user's smart device, under the assumptions described in Section 3.2.

Step 2: $\mathcal{A}$ eavesdrops on PID_i transmitted through a public channel and obtains $h(ID_i||PW_i) = PID_i^{'} \oplus PID_i$. Then, $\mathcal{A}$ can calculate $s_i = s_i^{'} \oplus h(ID_i||PW_i)$.

Step 3: $\mathcal{A}$ generates a number r_A randomly and a timestamp T_A, and calculates the request messages $M_1 = PDID_j \oplus h(CID_{CSS}||T_A)$, $M_2 = r_A \oplus h(CID_{CSS}||PID_i||s_i)$, and $V_1 = h(r_A||s_i||C)$.

Step 4: The CSS receives the request message and delivers the random number of $\mathcal{A}$ to D_j. Then, D_j computes a session key $SK = h(PID_i||PDID_j||CID_{CSS}||r_A||r_2)$ and transmits $\{PDID_j, M_5, M_6, V_3, T_3\}$ to the CSS.

Step 5: The CSS authenticates D_j and sends the message $M_7 = r_2 \oplus h(T_4||r_A)$ to $\mathcal{A}$. Finally, $\mathcal{A}$ obtains $r_2 = M_7 \oplus h(T_4||r_1)$ and computes $SK = h(PID_i||PDID_j||CID_{CSS}||r_A||r_2)$.

5.2. Stolen Verifier Attack

Under the CK model, $\mathcal{A}$ can access the verification table $\{ID_i, PID_i, s_i\}$ stored in the database of the CSS. Further, $\mathcal{A}$ can access the pseudo-identities of each of $\{PID_i, PDID_j$, and $CID_{CSS}\}$ entities, because they are transmitted through an open channel and not updated. To compute the session key between U_i and D_j, $\mathcal{A}$ calculates $r_1 = M_2 \oplus h(CID_{CSS}||PID_i|s_i)$ and $r_2 = M_7 \oplus h(T_4||r_1)$, where M_2 and M_7 are sent through an open channel. Finally, $\mathcal{A}$ can obtain the session key $SK = h(PID_i||PDID_j||CID_{CSS}||r_1||r_2)$.

5.3. Ephemeral Secret Leakage Attack

In Sharma et al.'s scheme, U_i and D_j establish a session key using the pseudo-identities of each $\{PID_i, PDID_j, CID_{CSS}\}$ entity and the random numbers $\{r_1, r_2\}$ generated by U_i and D_j. Therefore, if $\mathcal{A}$ gains those values, $\mathcal{A}$ can calculate the session key shared between U_i and D_j. Under the CK model, $\mathcal{A}$ can acquire the ephemeral random numbers r_1, r_2 generated during a session. Furthermore, $\mathcal{A}$ can eavesdrop on the pseudo-identities $\{PID_i, PDID_j, CID_{CSS}\}$ sent through an open channel. As a result, $\mathcal{A}$ can derive the session key $SK = h(PID_i||PDID_j||CID_{CSS}||r_1||r_2)$.

5.4. User Anonymity and Untraceability

$\mathcal{A}$ can eavesdrop the message sent through a public channel in accordance with the adversary model described in Section 3.2. In the AKA phase of Sharma et al.'s scheme, U_i and the CSS transmit PID_i through a public channel. At the end of the AKA phase, they do not update PID_i. Therefore, Sharma et al.'s scheme lacks the ability to preserve user untraceability and anonymity.

6. Proposed Scheme

Here, we detail our AKA scheme for the IoD, designed with PUF technology. The proposed scheme comprises the following phases: (1) initialization, (2) registration, (3) authentication and key agreement, and (4) password update. Users and drones register themselves to the CSS and share a session key with arbitration of the CSS. Detailed steps are outlined as follows.

6.1. Initialization

The CSS selects $h(\cdot)$ as a one-way hash function, along with a secret key X_{CSS} and an identity CID_{CSS}. Then, the CSS publishes $h(\cdot)$.

6.2. Drone Registration Phase

A drone registers itself with the CSS before authentication. Figure 3 represents the procedures of drone registration. Details are outlined below.

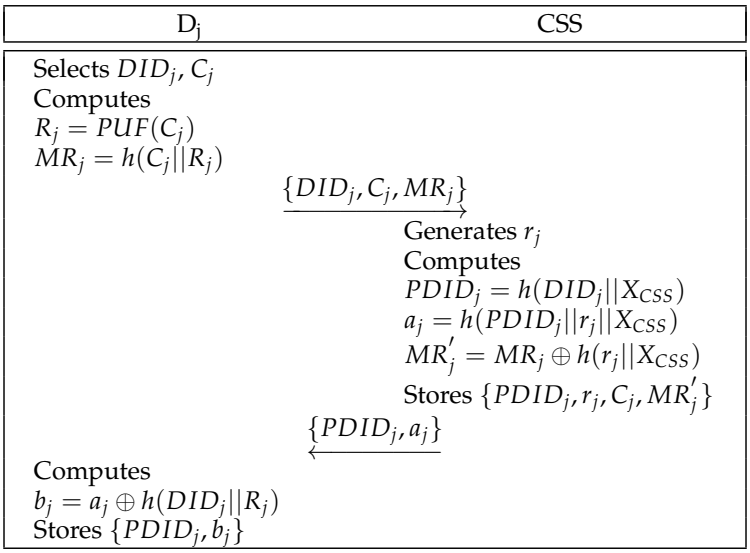

Figure 3. Drone registration of the proposed scheme.

Step 1: D_j chooses its identity DID_j and a challenge C, and computes $R = PUF(C)$ and $MR_j = h(C_j||R_j)$. Then, D_j sends $\{DID_j, MR_j\}$ to the CSS securely.

Step 2: The CSS generates a random number r_j, and calculates $PDID_j = h(DID_j||X_{CSS})$, $a_j = h(PDID_j||r_j||X_{CSS})$, and $MR_j^{'} = MR_j \oplus h(r_j||X_{CSS})$ after receiving the message. Then, the CSS stores $\{PDID_j, r_j, C_j, MR_j^{'}\}$ in a database and transmits $\{PDID_j, a_j\}$ to D_j securely.

Step 3: D_j computes $b_j = a_j \oplus h(DID_j||R_j)$, and saves $\{PDID_j, b_j\}$ to a database.

6.3. User Registration Phase

A user registers themselves with the CSS before authentication. Figure 4 shows the comprehensive steps of user registration. The following steps outline the details of this process.

Step 1: First, U_i selects an identity ID_i and a password PW_i. Further, U_i generates a number e_i randomly and transmits ID_i to the CSS securely.

Step 2: Upon receiving the message, the CSS generates a number r_i randomly and calculates $PID_i = h(ID_i||X_{CSS})$, $RID_i = h(CID_{CSS}||r_i||X_{CSS})$, and $s_i = h(PID_i||X_{CSS})$. The CSS sends $\{PID_i, RID_i, s_i, PDID_j\}$ to U_i through secure channel after it stores $\{PID_i, r_i\}$ in the database.

Step 3: U_i calculates $f_i = e_i \oplus h(ID_i||PW_i)$, $H_i = h(ID_i \oplus e_i||PW_i \oplus e_i)$, $RID_i^{'} = RID_i \oplus h(ID_i||PW_i||e_i)$, $PDID_j^{'} = PDID_j \oplus h(RID_i||ID_i||PW_i)$, and $s_i^{'} = s_i \oplus h(RID_i||PW_i||e_i)$. Finally, U_i stores $\{PID_i, f_i, H_i, RID_i^{'}, PDID_j^{'}, s_i^{'}\}$ in the database.

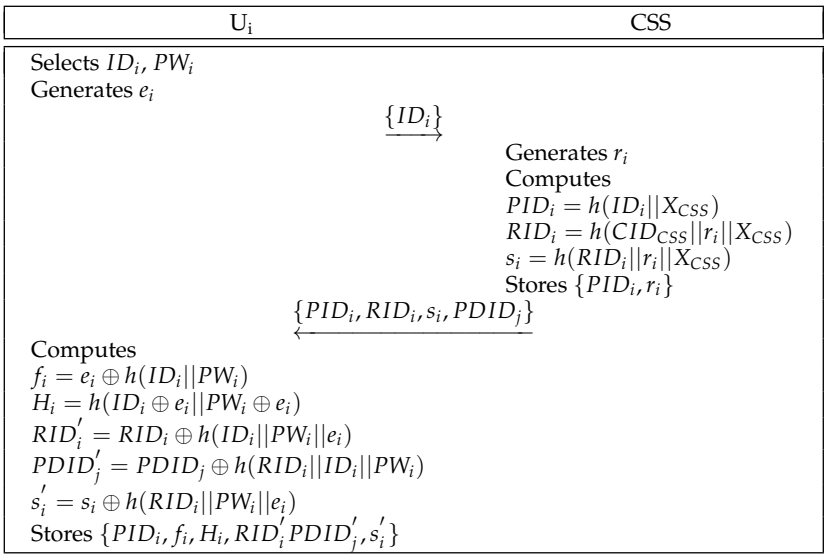

Figure 4. User registration of the proposed scheme.

6.4. Authentication and Key Agreement Phase

Authentication between U_i and D_j is established in this phase. After the authentication, they share a session key with the mediation of the CSS. Figure 5 depicts the details of the AKA phase.

Step 1: U_i inserts his/her identity ID_i and password PW_i, and computes $e_i^* = f_i \oplus h(ID_i||PW_i)$ and $H_i^* = h(ID_i \oplus e_i^*||PW_i \oplus e_i^*)$. Then, U_i compares whether H_i^* and H_i are equal or not. If they are equal, login is completed. U_i calculates $RID_i = RID_i^{'} \oplus h(ID_i||PW_i||e_i)$, $PDID_j = PDID_j^{'} \oplus h(RID_i||ID_i||PW_i)$, and $s_i = s_i^{'} \oplus h(RID_i||PW_i||e_i)$. Then, U_i selects a random number r_1 and a timestamp T_1, and calculates $M_1 = PDID_j \oplus h(RID_i||s_i||T_1)$, $M_2 = r_1 \oplus h(PDID_j||RID_i||s_i||T_1)$, and $V_1 = h(PID_i||RID_i||\ PDID_j||r_1||s_i||T_1)$. Further, U_i sends a message $\{PID_i, M_1, M_2, V_1, T_1\}$ to the CSS through an open channel.

Step 2: The CSS first checks whether T_1 is valid or not. If it is valid, the CSS retrieves r_i against PID_i and computes $RID_i = h(CID_{CSS}||r_i||X_{CSS})$, $s_i = h(RID_i||r_i||X_{CSS})$, $PDID_j^* = M_1 \oplus h(RID_i||s_i||T_1)$, $r_1^* = M_2 \oplus h(PDID_j^*||RID_i||s_i||T_1)$, and $V_1^* = h(PID_i||RID_i\ ||PDID_j^*||r_1^*||s_i||T_1)$. Then, the CSS verifies that V_1^* is equal to V_1. If they are equal, the CSS generates a timestamp T_2 and retrieves r_j against $PDID_j$. Then, the CSS calculates $a_j = h(PDID_j||r_j||X_{CSS})$, $MR_j = MR_j^{'} \oplus h(r_j||X_{CSS})$, $M_3 = (PID_i||C_j) \oplus h(PDID_j||T_2)$, $M_4 = r_1 \oplus h(a_j||MR_j||C_j||PDID_j||T_2)$, and $V_2 = h(r_1||MR_j||PDID_j||PID_i||a_j|T_2)$. The CSS transmits $\{M_3, M_4, V_2, T_2\}$ over a public channel.

Step 3: D_j verifies the legitimacy of T_2. If it is legitimate, D_j computes $PID_i^*||C_j^* = M_3 \oplus h(PDID_j||T_2)$, $R_j^* = PUF(C_j^*)$, $a_j^* = b_j \oplus h(DID_j||R_j^*)$, $MR_j^* = h(C_j^*||R_j^*)$, $r_1^* = M_4 \oplus h(a_j^*||MR_j^*||C_j^*||PDID_j||T_2)$, and $V_2^* = h(r_1^*||MR_j^*||PDID_j||PID_i^*||a_j^*||T_2)$. Then, D_j checks whether V_2^* and V_2 are equal or not. If they are equal, D_j generates a random number r_2, a new challenge C_j^{new} and a timestamp T_3. After that, D_j calculates $R_j^{new} = PUF(C_j^{new})$, $MR_j^{new} = h(C_j^{new}||R_j^{new})$, $M_5 = (C_j^{new}||MR_j^{new}) \oplus h(PDID_j||MR_j||a_j||r_1||T_3)$, $M_6 = h(r_2||R_j^{new}) \oplus h(PDID_j||MR_j^{new}||a_j||T_3)$, $V_3 = h($

$PDID_j||C_j^{new}||MR_j^{new}||h(r_2||R_j^{new})||a_j||\ r_1||T_3)$, and $SK = h(PID_i||PDID_j||r_1||h(r_2||R_j^{new}))$, and sends $\{PDID_j, M_5, M_6, V_3, T_3\}$ to the CSS through a public channel.

Step 4: The CSS checks the validity of T_3. If it is valid, the CSS calculates $C_j^{new*}||MR_j^{new*} = M_5 \oplus h(PDID_j||MR_j||a_j||r_1||T_3)$, $h(r_2||R_j^{new})^* = M_6 \oplus h(PDID_j||MR_j^{new*}||a_j||T_3)$, and $V_3^* = h(PDID_j||C_j^{new*}||MR_j^{new*}||h(r_2||R_jnew)^*||a_j||r_1\ ||T_3)$. Further, the CSS compares V_4^* with V_4. After checking the equality, the CSS generates a timestamp T_4 and computes $PID_i^{new} = h(PID_i||r_1||T_4)$, $M_7 = h(r_2||R_j^{new}) \oplus h(PID_i^{new}||PDID_j||s_i||\ RID_i||T_4)$, $V_4 = h(PID_i^{new}||PDID_j||s_i||RID_i||h(r_2||R_j^{new})||T_4)$, and $MR_j^{new'} = MR_j^{new} \oplus h(r_j||X_{CSS})$. Then, the CSS transmits $\{M_7, V_4, T_4\}$ to U_i and updates $\{C_j^{new}, MR_j^{new'}, PID_i^{new}\}$.

Step 5: U_i verifies that T_4 is legitimate. If it is legitimate, U_i computes $PID_i^{new*} = h(PID_i||r_1||T_4)$, $h(r_2||R_j^{new})^* = M_7 \oplus h(PID_i^{new*}||PDID_j||s_i||RID_i||T_4)$ and $V_4^* = h(PID_i^{new*}||PDID_j||s_i||RID_i||h(r_2||R_j^{new})^*||T_4)$. Then, U_i checks whether V_4^* is equal to V_4. If they are equal, U_i updates PID_i^{new} and computes the session key $SK = h(PID_i||PDID_j||r_1||h(r_2||R_j^{new}))$.

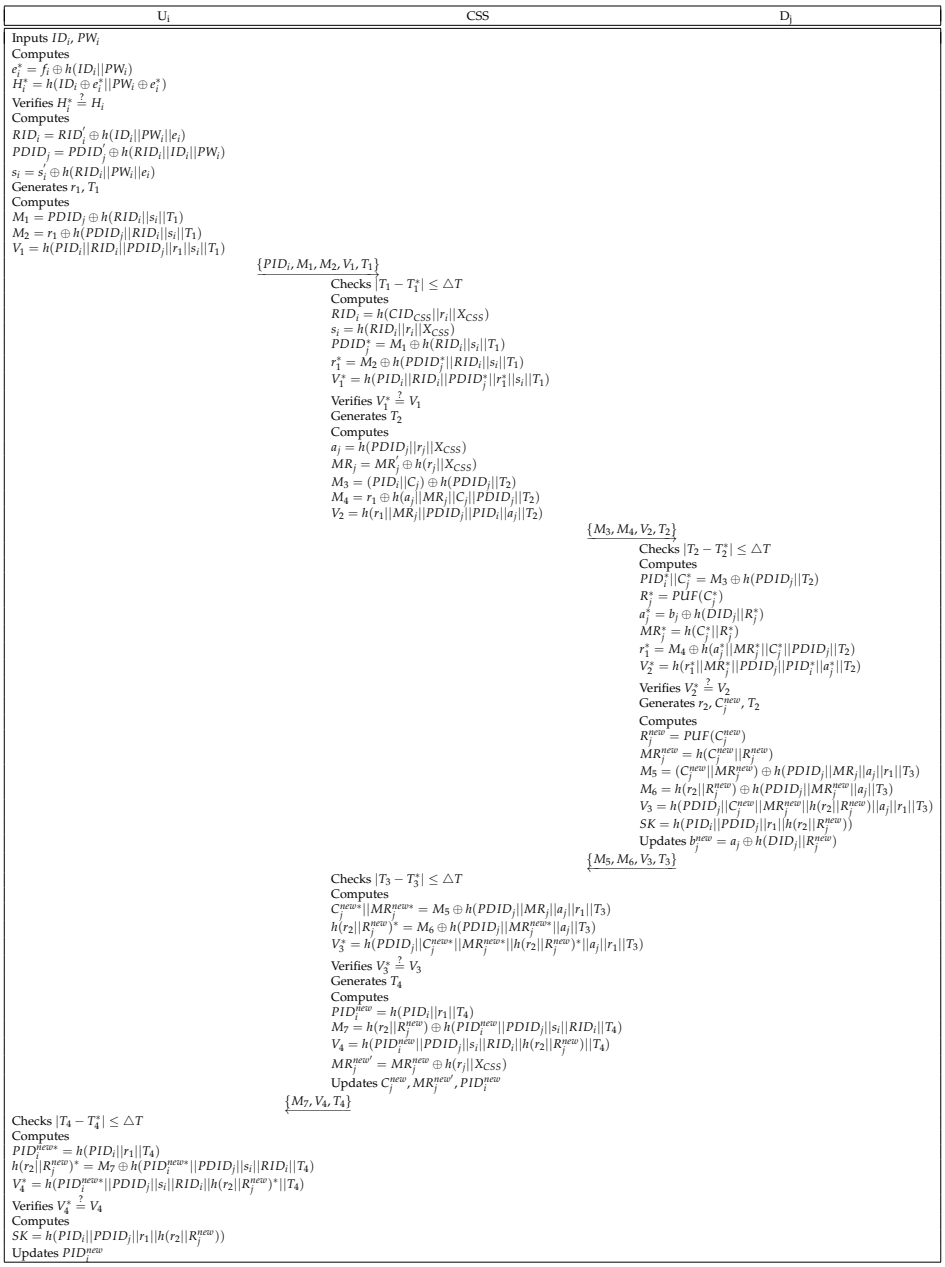

Figure 5. Authentication and key agreement phase of the proposed scheme.

6.5. Password Update Phase

Step 1: U_i inputs his/her identity ID_i and password PW_i, and computes $e_i^* = f_i \oplus h(ID_i||PW_i)$ and $H_i^* = h(ID_i \oplus e_i^*||PW_i \oplus e_i^*)$. Then, U_i compares that H_i^* and H_i are equal or not. If they are equal, login is completed.

Step 2: U_i inserts new password PW_i^{new}. Then, U_i calculates $f_i^{new} = e_i \oplus h(ID_i||PW_i^{new})$, $H_i^{new} = h(ID_i \oplus e_i||PW_i^{new} \oplus e_i)$, $RID_i'^{new} = RID_i \oplus h(ID_i||PW_i^{new}||e_i)$, $PDID_j'^{new} = PDID_j \oplus h(RID_i||ID_i||PW_i^{new})$, and $s_i'^{new} = s_i \oplus h(RID_i||PW_i^{new}||e_i)$. Finally, U_i stores $\{PID_i, f_i^{new}, H_i^{new}, RID_i'^{new}, PDID_j'^{new}, s_i'^{new}\}$ to the database.

7. Security Analysis

Here, we discuss the approach to verifying the resilience of the proposed scheme. To formally validate the robustness of our scheme, we employ "BAN logic", "RoR model", "AVISPA", and informal analysis. The results demonstrate that our scheme effectively resists various attacks while ensuring critical security requirements comprising mutual authentication, user anonymity, and untraceability. Further details are provided below.

7.1. BAN Logic

BAN logic is regarded as a standard analytical approach which is utilized to substantiate formally whether mutual authentication is achieved in AKA schemes. It has been extensively utilized by researchers to demonstrate the mutual authentication of various protocols. In this section, we first introduce the key notations and foundational rules of BAN logic. Subsequently, BAN logic analysis is applied to the proposed scheme. The primary BAN logic notations used in this study are summarized in Table 3. Further details of the analysis are as follows:

Table 3. Notations in BAN logic.

Notations	Descriptions
P_1, P_2	Principals
M_1, M_2	Statements
SK	Session key
$P_1 \overset{K}{\leftrightarrow} P_2$	P_1 and P_2 share the key K
$P_1 \mid\equiv M_1$	P_1 believes M_1
$\#M_1$	M_1 is fresh
$P_1 \mid\sim M_1$	P_1 said M_1
$P_1 \Rightarrow M_1$	P_1 controls M_1
$P_1 \triangleleft M_1$	P_1 receives M_1
$P_1 \overset{K}{\rightleftharpoons} P_2$	K is only known to trusted principals P_1 and P_2
$\{M_1\}_K$	M_1 is masked by K

7.1.1. Rules

The fundamental BAN logic rules utilized in this paper are outlined below.
Message meaning rule (MMR):

$$\frac{P_1 \mid\equiv P_1 \overset{K}{\leftrightarrow} P_2, P_1 \triangleleft (M_1)_K}{P_1 \mid\equiv P_2 \mid\sim M_1}$$

Nonce verification rule (NVR):

$$\frac{P_1 \mid\equiv \#M_1, P_1 \mid\equiv P_2 \mid\sim M_1}{P_1 \mid\equiv P_2 \mid\equiv M_1}$$

Jurisdiction rule (JR):

$$\frac{P_1| \equiv P_2 \Rightarrow M_1, P_1| \equiv P_2| \equiv M_1}{P_1| \equiv M_1}$$

Freshness rule (FR):

$$\frac{P_1| \equiv \#M_1}{P_1| \equiv \#(M_1, M_2)}$$

Belief rule (BR):

$$\frac{P_1| \equiv (M_1, M_2)}{P_1| \equiv M_1}$$

7.1.2. Idealized Forms

Idealized forms are defined as below.

Msg_1**:** $U_i \rightarrow CSS : (PDID_j, r_1, T_1)_{s_i}$

Msg_2**:** $CSS \rightarrow D_j : (PID_i, r_1, T_2)_{a_j}$

Msg_3**:** $D_j \rightarrow CSS : (h(r_2||R_j^{new}), T_3)_{a_j}$

Msg_4**:** $CSS \rightarrow U_i : (h(r_2||R_j^{new}), T_4)_{s_i}$

7.1.3. Goals

The security goals used to verify the guarantee of mutual authentication comprise the following:

Goal 1: $U_i| \equiv U_i \overset{SK}{\leftrightarrow} D_j$

Goal 2: $D_j| \equiv U_i \overset{SK}{\leftrightarrow} D_j$

Goal 3: $U_i| \equiv D_j| \equiv U_i \overset{SK}{\leftrightarrow} D_j$

Goal 4: $D_j| \equiv U_i| \equiv U_i \overset{SK}{\leftrightarrow} D_j$

7.1.4. Assumptions

Assumptions are defined as follows:

A_1**:** $CSS| \equiv (U_i \overset{s_i}{\leftrightarrow} CSS)$

A_2**:** $CSS| \equiv \#(T_1)$

A_3**:** $D_j| \equiv D_j \overset{a_j}{\leftrightarrow} CSS$

A_4**:** $D_j| \equiv \#(T_2)$

A_5**:** $CSS| \equiv (D_j \overset{a_j}{\leftrightarrow} CSS)$

A_6**:** $CSS| \equiv \#(T_3)$

A_7**:** $U_i| \equiv (U_i \overset{s_i}{\leftarrow} CSS)$

A_8**:** $U_i| \equiv \#(T_4)$

A_9**:** $U_i| \equiv CSS \Rightarrow (U_i \overset{h(r_2||R_j^{new})}{\rightleftharpoons} D_j)$

A_{10}**:** $D_j| \equiv CSS \Rightarrow (U_i \overset{r_1}{\rightleftharpoons} D_j)$

A_{11}**:** $U_i| \equiv D_j \Rightarrow (U_i \overset{SK}{\leftrightarrow} D_j)$

A_{12}**:** $D_j| \equiv U_i \Rightarrow (U_i \overset{SK}{\leftrightarrow} D_j)$

7.1.5. Proof

The procedure for the proof is described as follows:

Step 1: According to Msg_1, we can obtain S_1.

$$S_1 : CSS \triangleleft (PDID_j, r_1, T_1)_{s_i}$$

Step 2: By applying S_1 and A_1 to the MMR, we can obtain S_2.

$$S_2 : CSS| \equiv U_i| \sim (PDID_j, r_1, T_1)$$

Step 3: By applying A_2 to the FR, we can obtain S_3.

$$S_3 : CSS| \equiv \#(PDID_j, r_1, T_1)$$

Step 4: By applying S_2 and S_3 to the NVR, we can obtain S_4.

$$S_4 : CSS| \equiv U_i| \equiv (PDID_j, r_1, T_1)$$

Step 5: According to Msg_2, we can obtain S_5.

$$S_5 : D_j \triangleleft (PID_i, r_1, T_2)_{a_j}$$

Step 6: By applying S_5 and A_3 to the MMR, we can obtain S_6.

$$S_6 : D_j| \equiv CSS| \sim (PID_i, r_1, T_2)$$

Step 7: By applying A_4 to the FR, we can obtain S_7.

$$S_7 : D_j| \equiv \#(PID_i, r_1, T_2)$$

Step 8: By applying S_6 and S_7 to the NVR, we can obtain S_8.

$$S_8 : D_j| \equiv CSS| \equiv (PID_i, r_1, T_2)$$

Step 9: According to Msg_3, we can obtain S_9.

$$S_9 : CSS \triangleleft (h(r_2||R_j^{new}), T_3)_{a_j}$$

Step 10: By applying S_9 and A_5 to the MMR, we can obtain S_{10}.

$$S_{10} : CSS| \equiv D_j| \sim (h(r_2||R_j^{new}), T_3)$$

Step 11: By applying A_6 to the FR, we can obtain S_{11}.

$$S_{11} : CSS| \equiv \#(h(r_2||R_j^{new}), T_3)$$

Step 12: By applying S_{10} and S_{11} to the NVR, we can obtain S_{12}.

$$S_{12} : CSS| \equiv D_j| \equiv (h(r_2||R_j^{new}), T_3)$$

Step 13: According to Msg_4, we can obtain S_{13}.

$$S13 : U_i \triangleleft (h(r_2||R_j^{new}), T_4)_{s_i}$$

Step 14: By applying S_{13} and A_7 to the MMR, we can obtain S_{14}.

$$S_{14} : U_i| \equiv CSS| \sim (h(r_2||R_j^{new}), T_4)$$

Step 15: By applying A_8 to the FR, we can obtain S_{15}.

$$S_{15} : U_i| \equiv \#(h(r_2||R_j^{new}), T_4)$$

Step 16: By applying S_{14} and S_{15} to the NVR, we can obtain S_{16}.

$$S_{16} : U_i| \equiv CSS| \equiv (h(r_2||R_j^{new}), T_4)$$

Step 17: We can obtain S_{17} from S_{12}, S_{16}, and A_9 because the session key is $SK = h(PID_i||PDID_j||r_1||h(r_2||R_j^{new}))$.

$$S_{17} : U_i| \equiv D_j| \equiv (U_i \overset{SK}{\leftrightarrow} D_j) \quad \textbf{(Goal 3)}$$

Step 18: By applying S_{17} and A_{11} to the JR, we can obtain S_{18}.

$$S_{18} : U_i| \equiv (U_i \overset{SK}{\leftrightarrow} D_j) \quad \textbf{(Goal 1)}$$

Step 19: We can obtain S_{19} from S_4, S_8, and A_{10} because the session key is $SK = h(PID_i||PDID_j||r_1||h(r_2||R_j^{new}))$.

$$S_{17} : D_j| \equiv U_i| \equiv (U_i \overset{SK}{\leftrightarrow} D_j) \quad \textbf{(Goal 4)}$$

Step 20: By applying S_{19} and A_{12} to the JR, we can obtain S_{20}.

$$S_{18} : D_j| \equiv (U_i \overset{SK}{\leftrightarrow} D_j) \quad \textbf{(Goal 2)}$$

7.2. RoR Model

This section demonstrates the application of the RoR model to the proposed scheme. The RoR model is a well-known formal analysis that can verify whether an authentication protocol provides the semantic security of a session key [36–38]. Before explaining the application of the RoR model to the proposed scheme, we describe its basic concepts and notations. Under the RoR model, $\mathcal{A}$ executes queries that can attempt both active and passive attacks to reveal the session key. We describe the queries executed by $\mathcal{A}$, as detailed below. We denote three participants—a user, a drone, and a CSS—as $\mathcal{P}_U^{t_1}$, $\mathcal{P}_D^{t_2}$, and $\mathcal{P}_{CSS}^{t_3}$, respectively. The notation t_k is defined as a participant instance of a user, a drone, and a CSS.

- *Execute* ($\mathcal{P}_U^{t_1}$, $\mathcal{P}_D^{t_2}$, $\mathcal{P}_{CSS}^{t_3}$): Using this query, $\mathcal{A}$ eavesdrops on messages transmitted over a public channel among $\mathcal{P}_U^{t_1}$, $\mathcal{P}_D^{t_2}$, and $\mathcal{P}_{CSS}^{t_3}$.
- *Send* ($\mathcal{P}^t$, *M*): A message M can be transmitted to participant $\mathcal{P}^t$ by $\mathcal{A}$ to receive a response message.
- *CorruptMD* ($\mathcal{P}_U^{t_1}$): This query denotes smart device stolen attacks. $\mathcal{A}$ can attempt to extract the secret parameters stored in a user's smart device.
- *Test* ($\mathcal{P}^t$): Using this query, $\mathcal{A}$ determines if the speculative session key is a real session key or a random string. A fair coin c is flipped at the beginning of this query. $\mathcal{A}$ obtains $c = 1$ when $\mathcal{P}^t$ returns a real session key and $c = 0$ when $\mathcal{P}^t$ returns a random string. Otherwise, $\mathcal{A}$ receives a null. $\mathcal{A}$ is considered the winner of the game if $\mathcal{A}$ can judge whether the value output by $\mathcal{P}^t$ is the session key or a random string.

Theorem 1. *Consider $\mathcal{A}$ to attempt to compromise the proposed scheme within polynomial time. Let Adv_A denote the advantage that $\mathcal{A}$ successfully distinguishes the session key from a random string. Consequently, we obtain the result of the advantage as follows:*

$$Adv_A \leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|PUF|} + 2max\{C \cdot q_s^{s'}, \frac{q_s}{2^l}\}$$

$|PUF|$ and $|Hash|$ are defined as the output range of the PUF $PUF(\cdot)$ and the hash function $H(\cdot)$. Additionally, q_p and q_h denote the number of PUF and Hash queries executed by $\mathcal{A}$, respectively.

Proof. The semantic security of the session key is verified as demonstrated in a series of games $G_i(i = 0, 1, 2, 3)$. $Pr[Succ_i]$ indicates the possibility that $\mathcal{A}$ correctly distinguishes c in G_i.

$Game_0$: At the start of the game, $\mathcal{A}$ selects a random bit c. Hence, we can obtain Equation (1).

$$Adv_{\mathcal{A}} = |2Pr[Succ_0] - 1| \tag{1}$$

$Game_1$: $\mathcal{A}$ attempts an eavesdropping attack by conducting an *Execute* query. Further, $\mathcal{A}$ runs *Test* queries to determine if the acquired value is a session key or not. $\mathcal{A}$ must know $PDID_j$, r_1, and $h(r_2 R_j^{new})$ to acquire the session key $SK = h(PID_i PDID_j r_1 h(r_2\ R_j^{new}))$. However, these values cannot be obtained by eavesdropping attacks. This means that $\mathcal{A}$ has no advantage to be gained through an *Execute* query. Therefore, the probability of $\mathcal{A}$ winning G_1 is equal to that of $\mathcal{A}$ winning G_0.

$$Pr[Succ_1] = Pr[Succ_0] \tag{2}$$

$Game_2$: In this game, $\mathcal{A}$ runs *Send* and *Hash* queries to expose the session key. The transmitted messages can be modified by $\mathcal{A}$. However, $\mathcal{A}$ should find a hash collision to win the game because all transmitted messages are masked by a one-way function $H(\cdot)$. Therefore, the advantage that $\mathcal{A}$ can gain at the end of G_2 is obtained based on the birthday paradox.

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{q_h^2}{2|Hash|} \tag{3}$$

$Game_3$: Similar to $Game_2$, $\mathcal{A}$ runs *Send* and *PUF* queries. Due to security properties of the PUF described in Section 3.3, $\mathcal{A}$ cannot obtain an advantage after conducting $Game_3$.

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{q_p^2}{2|PUF|} \tag{4}$$

$Game_4$: In this game, $\mathcal{A}$ conducts *CorruptMD* queries to extract the secret parameters $\{PID_i, f_i, H_i, RID_i^{'}, PDID_j^{'}, s_i^{'}\}$ from a user's smart device, exploiting power analysis attacks. Further, $\mathcal{A}$ aims to derive the session key $SK = h(PID_i||PDID_j||r_1||h(r_2||R_j^{new}))$. However, each parameter consists of a user's identity ID_i and password PW_i. Therefore, $\mathcal{A}$ should guess the identity and password simultaneously. We can induce the following equation by adopting Zipf's law [39]:

$$|Pr[Succ_4] - Pr[Succ_3]| \leq max\{C \cdot q_s^{s'}, \frac{q_s}{2^l}\} \tag{5}$$

To win the game, $\mathcal{A}$ has to guess the bit c after finishing all games. Because $\mathcal{A}$ has no advantage in guessing c, we derive Equation (6).

$$Pr[Succ_4] = \frac{1}{2} \tag{6}$$

Equation (7) is obtained from Equations (1) and (2).

$$\frac{1}{2}Adv_{\mathcal{A}} = |Pr[Succ_0] - \frac{1}{2}| = |Pr[Succ_1] - \frac{1}{2}| \tag{7}$$

Equation (8) is obtained based on Equations (6) and (7).

$$\frac{1}{2}Adv_{\mathcal{A}} = |Pr[Succ_1] - Pr[Succ_4]| \tag{8}$$

Equation (9) is obtained using the triangle inequality of Equation (8).

$$\begin{aligned}\frac{1}{2}Adv_{\mathcal{A}} &= |Pr[Succ_1] - Pr[Succ_4]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| + |Pr[Succ_2] - Pr[Succ_3]| + |Pr[Succ_3] - Pr[Succ_4]| \\ &\leq \frac{q_h^2}{2|Hash|} + \frac{q_p^2}{2|PUF|} + max\{C \cdot q_s^{s'}, \frac{q_s}{2^l}\}\end{aligned} \tag{9}$$

Finally, the result is obtained by multiplying Equation (9) by 2.

$$Adv_{\mathcal{A}} \leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|PUF|} + 2max\{C \cdot q_s^{s'}, \frac{q_s}{2^l}\} \tag{10}$$

Consequently, Theorem 1 is verified. □

7.3. AVISPA Tool

This section presents the key data flow of AVISPA, highlighting the security verification of the proposed scheme. AVISPA is a widely accepted simulation tool used to prove whether a protocol is secure against replay attacks and MITM attacks. "High-Level Protocol Specification Language (HLPSL)" is a language used to execute a protocol in AVISPA based on a role. First, the HLPSL2IF translator converts the code written in HLPSL into an "Intermediate Format (IF)". Then, AVISPA executes a simulation using four back-end models: "on-the-fly model checker (OFMC)", "SAT-based model checker (SATMC)", "constraint logic-based attack searcher (CL-AtSe)", and "tree automata based on automatic approximations for the analysis of security protocols (TA4SP)". If the IF is placed into the back-end by the translator, the back-end generates and summarizes the analysis result as an "output format (OF)". An authentication protocol can resist MITM and replay attacks if the summary of OF represents "SAFE".

In this paper, we use two back-ends, "OFMC" and "CL-AtSe", for the AVISPA simulation of the proposed scheme. There are three roles (U_i, D_j, and CSS) in HLPSL, and we describe session and environment roles within those three roles. The secrecy of the secret parameter and the appropriateness of mutual authentication are checked in each session. Figure 6 represents the simulation results, showing that the summaries present "SAFE" using the "OFMC" and "CL-AtSe" back-end models. Hence, replay and MITM attacks cannot be successfully performed by $\mathcal{A}$.

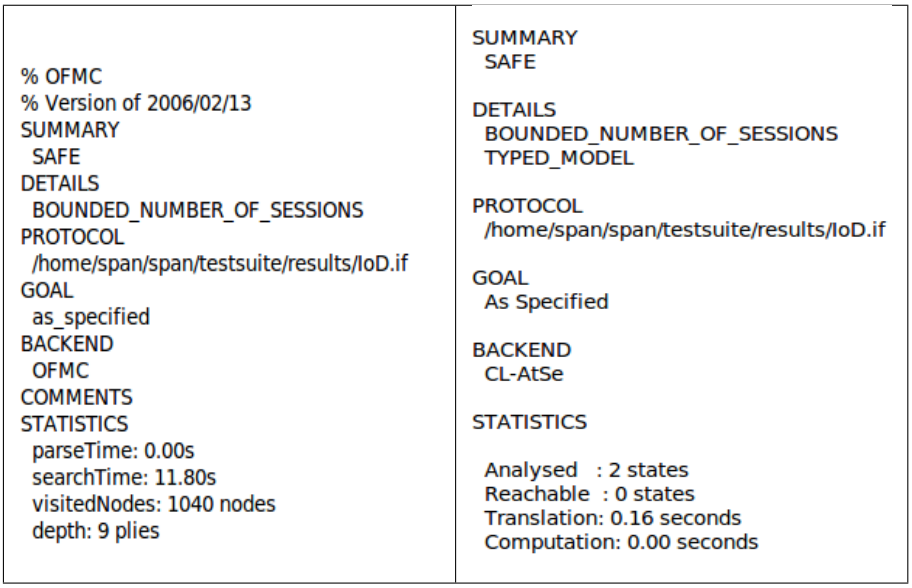

Figure 6. AVISPA simulation result under OFMC and CL-AtSe.

7.4. Informal Analysis

We analyze the proposed scheme informally to demonstrate the robustness related to numerous attacks. We also confirm that the proposed scheme achieves security requirements, including mutual authentication, perfect forward secrecy, user anonymity and untraceability.

7.4.1. Impersonation Attack

At the start of the AKA phase, U_i transmits the request message $\{PID_i, M_1, M_2, V_1, T_1\}$ to the CSS first. $\mathcal{A}$ must compute the message to impersonate U_i. Under the adversary model, $\mathcal{A}$ can obtain the secret information $\{PID_i, f_i, H_i, RID_i^{'}, PDID_j^{'}, s_i^{'}\}$ stored in the smart device of U_i. However, $\mathcal{A}$ cannot compute $\{PDID_j, RID_i, s_i\}$ because they are masked by $\{ID_i, PW_i, e_i\}$. $\mathcal{A}$ should guess ID_i and PW_i simultaneously to obtain $e_i = f_i \oplus h(ID_i||PW_i)$. It is computationally infeasible. As a result, our scheme prevents impersonation attacks.

7.4.2. Stolen Verifier Attack

The CSS stores verification table $\{PID_i, r_i\}$ in its database. According to the CK model, suppose that $\mathcal{A}$ steals the verification table. After obtaining the verification table, $\mathcal{A}$ can use the values $\{PID_i, r_i\}$ to calculate the session key $SK = h(PID_i||PDID_j||r_1||h(r_2||R_j^{new}))$. However, $\mathcal{A}$ cannot obtain the secret parameter $\{PDID_j, r_1, h(r_2||R_j^{new})\}$ without knowing the secret key $\{s_i, a_j\}$. Although $\mathcal{A}$ has $\{PID_i, r_i\}$, $\mathcal{A}$ cannot calculate s_i and a_j. Thus, the proposed scheme can defend stolen verifier attacks.

7.4.3. Ephemeral Secret Leakage Attack

$\mathcal{A}$ accesses to the ephemeral secrets r_1 and r_2, which are generated by U_i and D_j in the AKA phase. Further, $\mathcal{A}$ aims to acquire the session key $SK = h(PID_i||PDID_j||r_1||h(r_2||R_j^{new}))$. Even if $\mathcal{A}$ obtains the random secrets r_1 and r_2, $\mathcal{A}$ still does not know $PDID_j$ and $h(r_2||R_j^{new})$. $\mathcal{A}$ cannot acquire $PDID_j$ and $h(r_2||R_j^{new})$ without the secret key a_j and MR_j, which are masked by the master key of the CSS and the PUF response of D_j. Hence, our scheme can resist against ESL attacks.

7.4.4. Replay Attack

All the messages are hashed with timestamps during the AKA phase of the proposed scheme. Even if $\mathcal{A}$ intercepts a message transmitted through an open channel and tries to resend the message, $\mathcal{A}$ cannot reuse the message because each entity verifies the validity of the timestamp in every session. If a timestamp is not in a legitimate range, authentication will fail. Hence, the proposed scheme can defend replay attacks.

7.4.5. Man-in-the-Middle Attack

After intercepting the message that U_i or D_j transmit to the CSS, $\mathcal{A}$ generates a random number and a timestamp, and attempts to modify the message to send another valid message. However, $\mathcal{A}$ cannot calculate the message $\{M_1, M_2, V_1\}$ because $\mathcal{A}$ does not know the secret parameters RID_i and s_i shared between U_i and the CSS. Since RID_i and s_i are masked by the master key of the CSS and stored in a user's smart device securely, $\mathcal{A}$ cannot obtain them. In a similar way, $\mathcal{A}$ also cannot compute the message $\{M_5, M_6, V_3\}$ due to the secrecy of a_j. Therefore, our scheme is resistant to MITM attacks.

7.4.6. Privileged Insider Attack

The registration request message of U_i, $\{ID_i\}$ can be intercepted by a privileged adversary $\mathcal{A}$. Then, $\mathcal{A}$ attempts to obtain the secret values RID_i and s_i using ID_i. Even if $\mathcal{A}$ obtains ID_i, $\mathcal{A}$ cannot calculate RID_i and s_i because they are hashed with the master key of the CSS X_{CSS}. Each of the parameters necessary for calculating the session key $SK = h(PID_i||PDID_j||r_1||\ h(r_2||R_j^{new}))$ are encrypted with RID_i and s_i. Therefore, $\mathcal{A}$ cannot successfully defend against privileged insider attacks.

7.4.7. Drone Capture Attack

$\mathcal{A}$ can attempt to derive the session key $SK = h(PID_i||PDID_j||r_1||h(r_2||R_j^{new}))$ after $\mathcal{A}$ intercepts a drone D_j and extracts the information $\{PDID_j, b_j\}$. However, $\mathcal{A}$ cannot obtain the session key due to the secure property of the PUF. $\mathcal{A}$ must obtain a_j and MR_j to calculate the session key. However, these values are masked by the PUF response R_j. It is impossible to compute $R_j = PUF(C_j)$ for $\mathcal{A}$. Additionally, the proposed scheme updates R_j to R_j^{new} in every session. Thus, our scheme is robust to drone capture attacks.

7.4.8. Mutual Authentication

U_i, D_j and the CSS verify the legitimacy of the message during the AKA phase. The CSS and U_i authenticate each other by checking that V_1^* is equal to V_1 and V_4^* is equal to V_4. Similarly, the CSS and D_j authenticate each other by verifying whether V_2^* and V_2 are equal or not, and whether V_3^* and V_3 are equal or not. If the values are not identical, the authentication process is terminated. U_i and D_j mutually authenticate each other and share a session key through CSS arbitration. Hence, mutual authentication is preserved in the proposed scheme.

7.4.9. User Anonymity and Untraceability

The identity of U_i is transmitted through a secure channel one time when U_i registers itself to the CSS. Then, the CSS calculates a user's pseudo-identity PID_i and sends it to U_i. In the AKA phase, only PID_i is used during communication. After terminating the key agreement, U_i and the CSS update PID_i to new a pseudo-identity PID_i^{new}. Thus, our scheme provides user anonymity and untraceability.

7.4.10. Perfect Forward Secrecy

According to the adversarial assumptions described in Section 3.2, $\mathcal{A}$ can obtain the mater key of the CSS X_{CSS}. $\mathcal{A}$ uses X_{CSS} to calculate the session key $SK = h(PID_i||PDID_j||r_1||\ h(r_2||R_j^{new}))$. However, r_1 and $h(r_2||R_j^{new})$ are transmitted while being encrypted by secret keys s_i and a_j. Even if $\mathcal{A}$ gains X_{CSS}, $\mathcal{A}$ cannot obtain $s_i = h(RID_i||r_i||X_{CSS})$ and $a_j = h(PDID_j||r_j||X_{CSS})$. As a result, the proposed scheme guarantees perfect forward secrecy.

8. Performance Analysis

We present a performance comparison between the proposed scheme and related schemes. We estimate "security properties", "computational cost", "communication cost" and "energy consumption" of the proposed scheme and show that our scheme offers enhanced robustness and efficiency compared to others.

8.1. Security Properties

We examine the proposed scheme and comparable other schemes [15–19] regarding security features. We contemplate the following security functionalities: S_1: "resistance to impersonation attack", S_2: "resistance to stolen verifier attack", S_3: "resistance to ESL attack", S_4: "resistance to replay attack", S_5: "resistance to MITM attack", S_6: "resistance to privileged insider attack", S_7: "resistance to drone capture attack", S_8: "ensuring user anonymity and untraceability", S_9: "ensuring perfect forward secrecy", S_{10}: "performing BAN logic", S_{11}: "performing RoR model", and S_{12}: "performing AVISPA". We summarize the comparative analysis in Table 4. The proposed scheme achieves abundant security properties that are necessary for IoD communication.

Table 4. Security properties.

Security Features	[15]	[16]	[17]	[18]	[19]	Proposed
S_1	○	○	○	○	×	○
S_2	—	—	—	—	×	○
S_3	○	—	○	○	×	○
S_4	○	×	○	○	○	○
S_5	○	—	○	○	○	○
S_6	—	×	○	—	○	○
S_7	×	○	○	○	○	○
S_8	○	○	○	○	×	○
S_9	—	○	—	—	—	○
S_{10}	—	—	○	—	—	○
S_{11}	○	○	○	○	○	○
S_{12}	—	○	—	—	○	○

○: "Guarantee the security property." ×: "Do not guarantee the security property." —: "Not considered."

8.2. Computational Costs

This section focuses on analyzing the computational cost of the proposed scheme compared to other related works [15–19]. We quote the work using ubuntu 12.04.1 LTS 32-bit operating system, 2048 MB of RAM, and Intel Pentium Dual CPU E2200 2.20 GHz processor [15]. T_{HECC}, T_{fe}, T_{sym}, T_{ag}, T_{PUF} and T_h represent HECC divisor multiplication, fuzzy extractor function, symmetric encryption/decryption, AEGIS (AEAD scheme), PUF, and hash function. Table 5 depicts the execution time of the operations. We disregard the time cost of XOR and concentration operations, have extremely low computation costs [40]. In the proposed scheme, a user requires $12T_h$, a CSS requires $17T_h$, and a drone requires $2T_{PUF} + 13T_h$. Therefore, the total time overhead incurred by each entity is $2T_{PUF} + 42T_h$. Similarly, we also compute the computational costs of the related schemes and compare

them with our scheme. We represent the result of the comparison in Table 6. Although the proposed scheme incurs a slightly higher computation time than [16,19], the proposed scheme provides enhanced security. Zhang et al.'s scheme [16] is vulnerable to replay and privileged insider attacks, as outlined in Table 4. In the IoD environment, $\mathcal{A}$ can illegally control the drones to carry out malicious operations by resending intercepted authentication messages. $\mathcal{A}$ can also cause malfunctions or disruptions in drone operations to manipulate the IoD system through privileged insider attacks. Therefore, the security drawbacks of their scheme are fatal in IoD environments. Additionally, Sharma et al.'s scheme cannot withstand impersonation, stolen verifier, and ESL attacks, as demonstrated above. Therefore, our scheme has an efficient balance in terms of time cost and security.

Table 5. Execution time.

T_{HECC}	T_{fe}	T_{sym}	T_{ag}	T_{PUF}	T_h
1.113 ms	2.226 ms	0.0046 ms	0.415 ms	0.054 ms	0.0023 ms

Table 6. Computational costs.

Protocol	User	Server	Drone	Total Cost (ms)
Pratap et al. [15]	$2T_{HECC} + T_{fe} + 9T_h$	$4T_h$	$2T_{HECC} + 4T_h$	6.7171
Zhang et al. [16]	$8T_h$	$6T_h$	$2T_{PUF} + 6T_h$	0.154
Tanveer et al. [17]	$5T_{sym} + 2T_{fe} + T_{PUF} + 7T_h$	$5T_{sym} + T_{fe} + T_{PUF} + 3T_h$	$3T_{sym} + T_{fe} + T_{PUF} + 5T_h$	9.1603
Tanveer et al. [18]	$5T_{ag} + T_{fe} + 4T_h$	$5T_{ag} + 6T_h$	$2T_{ag} + T_{fe} + T_{PUF} + 3T_h$	9.5159
Sharma et al. [19]	$8T_h$	$10T_h$	$2T_{PUF} + 6T_h$	0.1632
proposed scheme	$12T_h$	$17T_h$	$2T_{PUF} + 13T_h$	0.2046

8.3. Communication Costs

We conduct a comparison of communication costs between our scheme and associated schemes [15–19]. In this paper, we consider the size of the PUF response, authentication parameter, hash function output, random number, identity, AES block, MC, HECC divisor, PUF challenge, and timestamp as 320 bits, 256 bits, 160 bits, 160 bits, 160 bits, 128 bits, 128 bits, 80 bits 32 bits, and 32 bits, respectively. In the proposed scheme, all entities transmit four messages, including $Msg1 = \{PID_i, M_1, M_2, V_1, T_1\}$, $Msg2 = \{M_3, M_4, V_2, T_2\}$, $Msg3 = \{M_5, M_6, V_3, T_3\}$, and $Msg4 = \{M_7, V_4, T_4\}$. The communication costs of the messages are $160 + 160 + 160 + 160 + 32 = 672$ bits, $(160 + 32) + 160 + 160 + 32 = 544$ bits, $(160 + 32) + 160 + 160 + 32 = 544$ bits, and $160 + 160 + 32 = 352$ bits. Therefore, the total number of bits is $672 + 672 + 512 + 352 = 2112$ bits. We also compute the communication costs of relevant approaches. Table 7 and Figure 7 represent the communication costs of the proposed scheme and relevant approaches. The comparative analysis indicates a high communication efficiency of the proposed scheme.

Table 7. Communication costs.

Protocol	Communication Cost (bits)
Pratap et al. [15]	1696
Zhang et al. [16]	2176
Tanveer et al. [17]	2272
Tanveer et al. [18]	2400
Sharma et al. [19]	2688
proposed scheme	2112

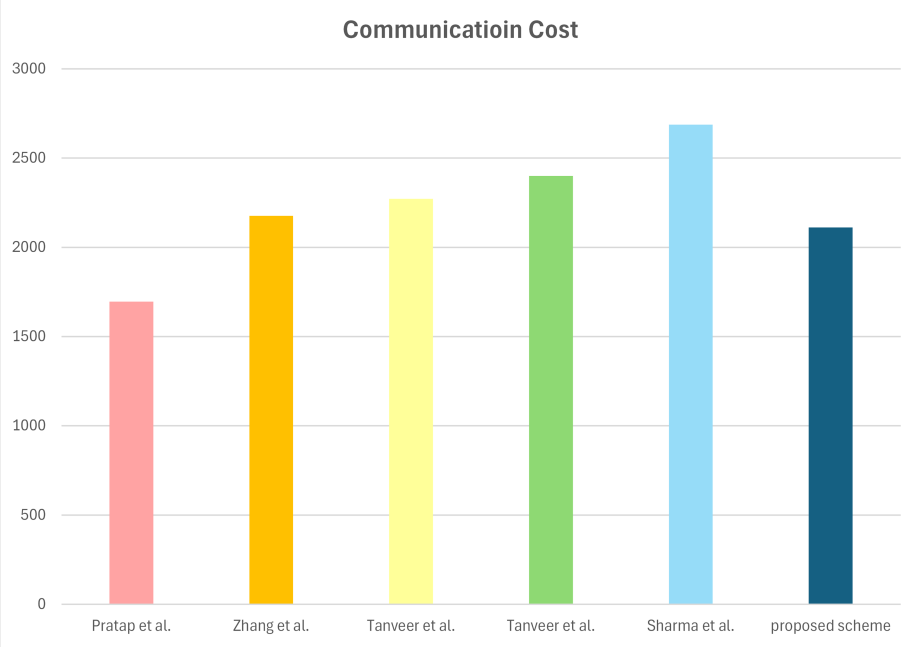

Figure 7. Communication costs [15–19].

8.4. Energy Consumption

Energy consumption can be calculated with $E = E_{comp} + E_{comm}$. Based on the equation, we estimate the energy overhead of our scheme with relevant schemes [15–19]. E_{comp} denotes the energy consumption during computation and E_{comm} denotes the energy consumption during communication [41]. According to test works conducted in [42] and the execution time in Table 5 measured by the equipment described in Section 8.2, we can compute the energy consumption for the "HECC divisor", "fuzzy extractor", "symmetric encryption/decryption", "AEGIS", "PUF", and "hash function" to be $E_{HECC} = 0.5\ \text{V} \times 0.4\ \text{A} \times 1.113\ \text{ms} = 0.2226\ \text{mJ}$, $E_{fe} = 0.4452\ \text{mJ}$, $E_{sym} = 0.00092\ \text{mJ}$, $E_{ag} = 0.083\ \text{mJ}$, $E_{PUF} = 0.0108$ and $E_h = 0.00046\ \text{mJ}$, respectively. Additionally, according to [42], communication energy consumption can be calculated as $E_{comm} = n_s E_s + n_r E_r$, where n_s denotes the number of bytes sent by the communication entity and n_r denotes the number of bytes received by the communication entity. Further, we assume that energy costs of sending and receiving message are $E_s \approx 5.9\ \mu\text{J}$, and $E_r \approx 4.7\ \mu\text{J}$ [43]. Therefore, the energy consumption of the proposed protocol during computation and communication are calculated to be $E_{comp} = 2E_{PUF} + 42E_h = 0.04092\ \text{mJ}$, and $E_{comm} = 264E_s + 264E_r = 2.7984\ \text{mJ}$. Consequently, the proposed scheme incurs the energy consumption of 2.83932 mJ. Comparison of energy consumption with associated schemes is depicted in Table 8 and Figure 8. The proposed scheme demonstrates more sustainable energy consumption compared to other related schemes.

Table 8. Energy consumption.

Protocol	Energy Consumption (mJ)
Pratap et al. [15]	3.59844
Zhang et al. [16]	2.914
Tanveer et al. [17]	4.84246
Tanveer et al. [18]	5.08318
Sharma et al. [19]	3.59424
proposed scheme	2.83932

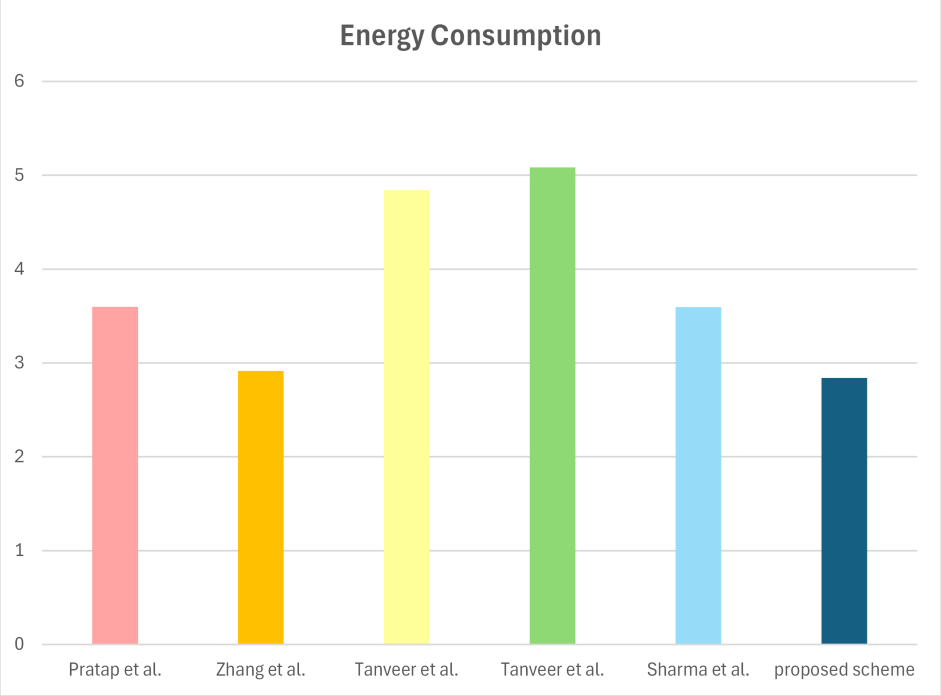

Figure 8. Energy consumption [15–19].

9. Conclusions

In this paper, we provided the overview of Sharma et al.'s AKA scheme and conducted a security analysis of it. We verified that their scheme is susceptible against user impersonation, stolen verifier, and ESL attacks. Then, we proposed a lightweight and secure AKA scheme for the IoD to rectify the vulnerabilities of Sharma et al.'s scheme. Fundamental necessities required for IoD communication are guaranteed through our scheme. The proposed scheme is robust to numerous adversarial attacks comprising impersonation, stolen verifier, ESL, MITM, replay, drone physical attacks. We consider the resilience of the scheme as well as the resource limitations of drones. The proposed scheme utilizes lightweight operations such as the hash function, XOR operation, and PUF. We verified the secureness of our scheme with informal analysis. We also demonstrated the security of our scheme by formally employing "BAN logic", "RoR model", and "AVISPA". We represented the efficiency of the proposed scheme, comparing it with other associated schemes. The result of our comparison showed that our scheme is highly cost-effective with robustness regarding its computational cost, communication cost, and energy consumption. Therefore, the proposed scheme allows the IoD to provide improved services. It also involves a higher number of message exchanges in the authentication phase compared with other related schemes. However, the overall communication costs remain efficient because each message has a lower cost in comparison to the compared schemes. Moreover, the proposed scheme considers a wide range of security properties and provides robust protection against various security threats. In our future work, we will implement the proposed scheme, optimizing and confirming its scalability and energy efficiency in a practical large-scale IoD environment.

Author Contributions: Conceptualization, J.C.; methodology, J.C.; software, D.K. and S.S.; validation, D.K., S.S. and Y.P.; formal analysis, J.C. and D.K.; writing—original draft preparation, J.C.; writing—review and editing, D.K., S.S. and Y.P.; supervision, Y.P.; project administration, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (Ministry of Science and ICT) (RS-2024-00450915).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within this article.

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

References

1. Gharibi, M.; Boutaba, R.; Waslander, S.L. Internet of drones. *IEEE Access* **2016**, *4*, 1148–1162. [CrossRef]
2. Yang, W.; Wang, S.; Yin, X.; Wang, X.; Hu, J. A review on security issues and solutions of the internet of drones. *IEEE Open J. Comput. Soc.* **2022**, *3*, 96–110. [CrossRef]
3. 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]
4. Mahmood, K.; Ghaffar, Z.; Nautiyal, L.; Akram, M.W.; Das, A.K.; Alenazi, M.J. A Privacy-Preserving Access Control Protocol for Consumer Flying Vehicles in Smart City Applications. *IEEE Internet Things J.* **2024**, *12*, 978–985. [CrossRef]
5. Alzahrani, A.A. VSKAP-IoD: A Verifiably Secure Key Agreement Protocol for Securing IoD Environment. *IEEE Access* **2024**, *12*, 58039–58056. [CrossRef]
6. Mishra, D.; Singh, M.; Rewal, P.; Pursharthi, K.; Kumar, N.; Barnawi, A.; Rathore, R.S. Quantum-safe secure and authorized communication protocol for internet of drones. *IEEE Trans. Veh. Technol.* **2023**, *72*, 16499–16507. [CrossRef]
7. Yahuza, M.; Idris, M.Y.I.; Ahmedy, I.B.; Wahab, A.W.A.; Nandy, T.; Noor, N.M.; Bala, A. Internet of drones security and privacy issues: Taxonomy and open challenges. *IEEE Access* **2021**, *9*, 57243–57270. [CrossRef]
8. 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]
9. Prajapat, S.; Gautam, D.; Kumar, P.; Jangirala, S.; Das, A.K.; Park, Y.; Lorenz, P. Secure lattice-based aggregate signature scheme for vehicular Ad Hoc networks. *IEEE Trans. Veh. Technol.* **2024**, *73*, 12370–12384. [CrossRef]
10. Irshad, A.; Alzahrani, B.A.; Albeshri, A.; Alsubhi, K.; Nayyar, A.; Chaudhry, S.A. SPAKE-DC: A secure PUF enabled authenticated key exchange for 5G-based drone communications. *IEEE Trans. Veh. Technol.* **2024**, *73*, 5770–5780. [CrossRef]
11. Subbarayalu, V.; Vensuslaus, M.A. An intrusion detection system for drone swarming utilizing timed probabilistic automata. *Drones* **2023**, *7*, 248. [CrossRef]
12. Ghelani, J.; Gharia, P.; El-Ocla, H. Gradient Monitored Reinforcement Learning for Jamming Attack Detection in FANETs. *IEEE Access* **2024**, *12*, 23081–23095. [CrossRef]
13. Cibecchini, S.; Chiti, F.; Pierucci, L. A Lightweight AI-Based Approach for Drone Jamming Detection. *Future Internet* **2025**, *17*, 14. [CrossRef]
14. Rezaee, M.R.; Hamid, N.A.W.A.; Hussin, M.; Zukarnain, Z.A. Comprehensive Review of Drones Collision Avoidance Schemes: Challenges and Open Issues. *IEEE Trans. Intell. Transport. Syst.* **2024**, *25*, 6397–6426. [CrossRef]
15. Pratap, B.; Singh, A.; Mehra, P.S. REHAS: Robust and Efficient Hyperelliptic Curve-Based Authentication Scheme for Internet of Drones. *Concurr. Comput. Pract. Exp.* **2024**, *37*, e8333. [CrossRef]
16. Zhang, Z.; Hsu, C.; Au, M.H.; Harn, L.; Cui, J.; Xia, Z.; Zhao, Z. PRLAP-IoD: A PUF-based robust and lightweight authentication protocol for Internet of Drones. *Comput. Netw.* **2024**, *238*, 110118. [CrossRef]
17. Tanveer, M.; Aldosary, A.; Kumar, N.; Aldossari, S.A. SEAF-IoD: Secure and efficient user authentication framework for the Internet of Drones. *Comput. Netw.* **2024**, *247*, 110449. [CrossRef]
18. Tanveer, M.; Aldosary, A.; Khokhar, S.u.d.; Das, A.K.; Aldossari, S.A.; Chaudhry, S.A. PAF-IoD: PUF-Enabled Authentication Framework for the Internet of Drones. *IEEE Trans. Veh. Technol.* **2024**, *73*, 9560–9574. [CrossRef]
19. Sharma, M.; Narwal, B.; Anand, R.; Mohapatra, A.K.; Yadav, R. PSECAS: A physical unclonable function based secure authentication scheme for Internet of Drones. *Comput. Electr. Eng.* **2023**, *108*, 108662. [CrossRef]
20. Nikooghadam, M.; Amintoosi, H.; Islam, S.H.; Moghadam, M.F. A provably secure and lightweight authentication scheme for Internet of Drones for smart city surveillance. *J. Syst. Archit.* **2021**, *115*, 101955. [CrossRef]
21. Alzahrani, B.A.; Barnawi, A.; Chaudhry, S.A. A Resource-Friendly Authentication Protocol for UAV-Based Massive Crowd Management Systems. *Secur. Commun. Netw.* **2021**, *2021*, 3437373. [CrossRef]
22. Khalid, H.; Hashim, S.J.; Hashim, F.; Ahamed, S.M.S.; Chaudhary, M.A.; Altarturi, H.H.; Saadoon, M. HOOPOE: High performance and efficient anonymous handover authentication protocol for flying out of zone UAVs. *IEEE Trans. Veh. Technol.* **2023**, *72*, 10906–10920. [CrossRef]

23. Tanveer, M.; Khan, A.U.; Kumar, N.; Hassan, M.M. RAMP-IoD: A robust authenticated key management protocol for the Internet of Drones. *IEEE Internet Things J.* **2022**, *9*, 1339–1353. [CrossRef]
24. Badshah, A.; Abbas, G.; Waqas, M.; Tu, S.; Abbas, Z.H.; Muhammad, F.; Chen, S. USAF-IoD: Ultralightweight and Secure Authenticated Key Agreement Framework for Internet of Drones Environment. *IEEE Trans. Veh. Technol.* **2024**, *73*, 10963–10977. [CrossRef]
25. Dwivedi, S.K.; Abdussami, M.; Amin, R.; Khan, M.K. D3APTS: Design of ECC Based Authentication Protocol and Data Storage for Tactile Internet enabled IoD System With Blockchain. *IEEE Trans. Consum. Electron.* **2024**, *70*, 4239–4248. [CrossRef]
26. 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]
27. Chaudhary, D.; Soni, T.; Singh, S.; Gupta, S.M.C. A Construction of Secure and Efficient Authenticated Key Exchange Protocol for Deploying Internet of Drones in Smart City. In Proceedings of the International Conference on Artificial Intelligence of Things, Ho Chi Minh City, Vietnam, 25–27 October 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 136–150.
28. Chaudhary, D.; Soni, T.; Vasudev, K.L.; Saleem, K. A modified lightweight authenticated key agreement protocol for Internet of Drones. *Internet Things* **2023**, *21*, 100669. [CrossRef]
29. Lee, T.F.; Lou, D.C.; Chang, C.H. Enhancing lightweight authenticated key agreement with privacy protection using dynamic identities for Internet of Drones. *Internet Things* **2023**, *23*, 100877. [CrossRef]
30. Hussain, S.; Farooq, M.; Alzahrani, B.A.; Albeshri, A.; Alsubhi, K.; Chaudhry, S.A. An efficient and reliable user access protocol for Internet of Drones. *IEEE Access* **2023**, *11*, 59688–59700. [CrossRef]
31. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
32. Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), Amsterdam, The Netherlands, 28 April–2 May 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 337–351.
33. Kocher, P. Differential power analysis. In Proceedings of the Advances in Cryptology (CRYPTO'99), Santa Barbara, CA, USA, 15–19 August 1999.
34. Ryu, J.; Son, S.; Lee, J.; Park, Y.; Park, Y. Design of secure mutual authentication scheme for metaverse environments using blockchain. *IEEE Access* **2022**, *10*, 98944–98958. [CrossRef]
35. Kwon, D.; Son, S.; Park, K.; Das, A.K.; Park, Y. Design of Blockchain-Based Multi-Domain Authentication Protocol for Secure EV Charging Services in V2G Environments. *IEEE Trans. Intell. Transport. Syst.* **2024**, *25*, 21783–21795. [CrossRef]
36. 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]
37. Yu, S.; Park, Y. A robust authentication protocol for wireless medical sensor networks using blockchain and physically unclonable functions. *IEEE Internet Things J.* **2022**, *9*, 20214–20228. [CrossRef]
38. Kwon, D.; Son, S.; Kim, M.; Lee, J.; Kumar Das, A.; Park, Y. A Secure Self-Certified Broadcast Authentication Protocol for Intelligent Transportation Systems in UAV-Assisted Mobile Edge Computing Environments. *IEEE Trans. Intell. Transport. Syst.* **2024**, *25*, 19004–19017. [CrossRef]
39. 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]
40. Zhou, X.; Wang, S.; Wen, K.; Hu, B.; Tan, X.; Xie, Q. Security-Enhanced Lightweight and Anonymity-Preserving User Authentication Scheme for IoT-Based Healthcare. *IEEE Internet Things J.* **2024**, *11*, 9599–9609. [CrossRef]
41. Li, F.; He, Y.; Niu, B.; Li, H.; Wang, H. Match-MORE: An efficient private matching scheme using friends-of-friends' recommendation. In Proceedings of the International Conference on Computing, Networking and Communications (ICNC), Kauai, HI, USA, 15–18 February 2016; pp. 1–6.
42. Sun, Y.; Cao, J.; Ma, M.; Zhang, Y.; Li, H.; Niu, B. EAP-DDBA: Efficient anonymity proximity device discovery and batch authentication mechanism for massive D2D communication devices in 3GPP 5G HetNet. *IEEE Trans. Depend. Secur. Comput.* **2020**, *19*, 370–387. [CrossRef]
43. Rahmati, A.; Zhong, L. Context-for-wireless: Context-sensitive energy-efficient wireless data transfer. In Proceedings of the International Conference on Mobile Systems, Applications and Services, San Juan, PR, USA, 11–14 June 2007; pp. 165–178.