RESEARCH ARTICLE

PUF-AKA-IoD: A Provably Secure PUF-Based Authentication and Key Agreement Protocol for IoD over 5G Networks

Maoxin Tang[1] and **Haonan Li**[2,*]

[1] School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2] Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur 50603, Malaysia

Abstract

The Internet of Things (IoT) is a technological ecosystem that interconnects physical devices via the Internet to enable data exchange and coordinated operations, among which the Internet of Drones (IoD) represents a specialized application integrating drone technology with IoT for remote control and data transmission. With the adoption of fifth-generation (5G) mobile communication networks in IoD environments, drones can transmit high-quality data and respond to user requests with reduced latency. In recent years, various authentication schemes using temporary credentials have been proposed to allow users and drones to establish session keys with the assistance of a control server; however, many existing schemes remain vulnerable to drone capture attacks and known session-specific temporary information attacks. To address these security challenges, this paper proposes an enhanced authentication and key agreement scheme for IoD over 5G networks that incorporates physically unclonable functions (PUFs) to strengthen drone data security. The security of the proposed scheme is validated through informal security analysis, the ROR model, and the AVISPA tool. Furthermore, comparative security and performance evaluations demonstrate that the proposed scheme achieves adequate security while maintaining competitive performance relative to existing schemes.

Keywords: IoD, 5G networks, mutual authentication, key agreement, PUF.

1 Introduction

The Internet of Drones (IoD) [1, 2] refer to the connection of Unmanned Aerial Vehicles (UAVs, also known as drones) with the Internet or other communication networks to realize data exchange, information sharing and collaborative work between drones, drones and infrastructure, and drones and people. The IoD combines the Internet of Things (IoT) and drone technologies to create smarter and efficient application scenarios, such as smart agriculture management [17], smart city surveillance

Submitted: 03 October 2025
Accepted: 27 November 2025
Published: 19 December 2025

Vol. 1, **No.** 1, 2025.
10.62762/JRSC.2025.869145

***Corresponding author:**
✉ Haonan Li
24061728@siswa.um.edu.my

Citation
Tang, M., & Li, H. (2025). PUF-AKA-IoD: A Provably Secure PUF-Based Authentication and Key Agreement Protocol for IoD over 5G Networks. *Journal of Reliable and Secure Computing*, 1(1), 25–40.

[15] and smart traffic monitoring [18]. The common communication networks in the IoD environment include wireless, wired, and cellular mobile. Although drones can use various communication networks, considering the application requirements and environmental conditions, cellular mobile network is usually used to achieve remote control and communication of drones. At present, mobile cellular technology has developed to the fifth-generation (5G) [3–5]. The 5G technology can provide advantages for the IoD environment, such as faster data transmission rates, more device connections, and larger signal coverage. This contributes to the advancement of drone technology, enabling innovative application scenarios and providing practicality and benefits across various fields.

The 5G-enabled IoD architecture is depicted in Figure 1, encompassing three primary entities: users, a control server, and drones. In the IoD environment, only legitimate users can remotely control the drone, perform flight tasks and control various functions of the drone. The control server, located in the control room, acts as a trusted authority that receives user requests and communicates with the drones in the corresponding area based on these requests. Furthermore, the control server is responsible for user and drone registration. The drone is deployed in different flight areas, and it can carry various sensors and cameras, collect data in real time and transmit it to users via the 5G network. Recently, several studies [6–8] involving authentication and key agreement (AKA) schemes apply 5G technology to IoD environment. In 2021, Wu et al. [6] combined 5G technology with the IoD environment and proposed an enhanced AKA scheme for drone communication. Feng et al. [7] applied intelligent 5G technology to the IoD environment and proposed a blockchain-based drone cross-domain AKA scheme. Ren et al. [8] devised a drone-assisted AKA scheme within a 5G-enabled satellite-ground fusion network, using PUF to guarantee data security.

Owing to the openness of mobile networks, malicious attackers have the capability to intercept messages sent over public channels and manipulate or disrupt these messages, resulting in the compromise of user privacy. Additionally, mobile drones are susceptible to capture by attackers, allowing them t o utilize power analysis [9] to access the stored data within. Ultimately, attackers employ these messages and data to launch attacks, such as drone capture [10, 11], impersonation [12, 13], and known specific-session temporary information (KSSTI) [14] attacks.

It can be mathematically represented as $R = PUF(C)$, where C represents the challenge and R is the corresponding response. Because of the influence of environmental factors and conditions, the response of a PUF to the same challenge may exhibit slight variations. As a result, it becomes imperative to account for the potential noise impact during the authentication process. Referring to the literature [15, 16], when putting forward an AKA scheme for drone communication over 5G, we utilize a fuzzy extractor to address the noise challenge present in the PUF response.

We propose an enhanced AKA scheme within the IoD over 5G networks (shown in Figure 1). The main contributions of this paper can be outlined as follows.

(1) To overcome the stated security problems, we put forward an improved AKA scheme for drone communications within 5G networks, which employs PUF technology to ensure the data integrity of the drones. Additionally, the proposed scheme enables users to successfully establish session keys with drones with the assistance of a control server.

(2) To showcase the security of our proposed scheme, we performed a comprehensive analysis, incorporating informal security analysis, Real-Or-Random (ROR), and the Automated Verification of Internet Security Protocols and Applications (AVISPA) tool.

(3) We carried out a comparative analysis, assessing the security and performance of the proposed scheme in relation to existing schemes. The comparison results revealed that our scheme not only ensures an adequate level of security but also exhibits a certain degree of performance.

The structural arrangement of this paper is as follows. Firstly, we conduct an overview of the pertinent studies concerning AKA schemes that rely on IoD in Section 2. In Section 3, presents the cryptographic preliminaries for the design of the authentication protocol. In Section 4, we propose the AKA scheme based on a 5G-enabled IoD environment using PUF. We carried out a thorough analysis by utilizing informal security analysis, the ROR model, and the AVISPA tool to show the security of our proposed scheme in Section 5. In Section 6, we evaluate the proposed scheme in relation to existing schemes, considering both security and performance aspects. Finally, a summary of this paper

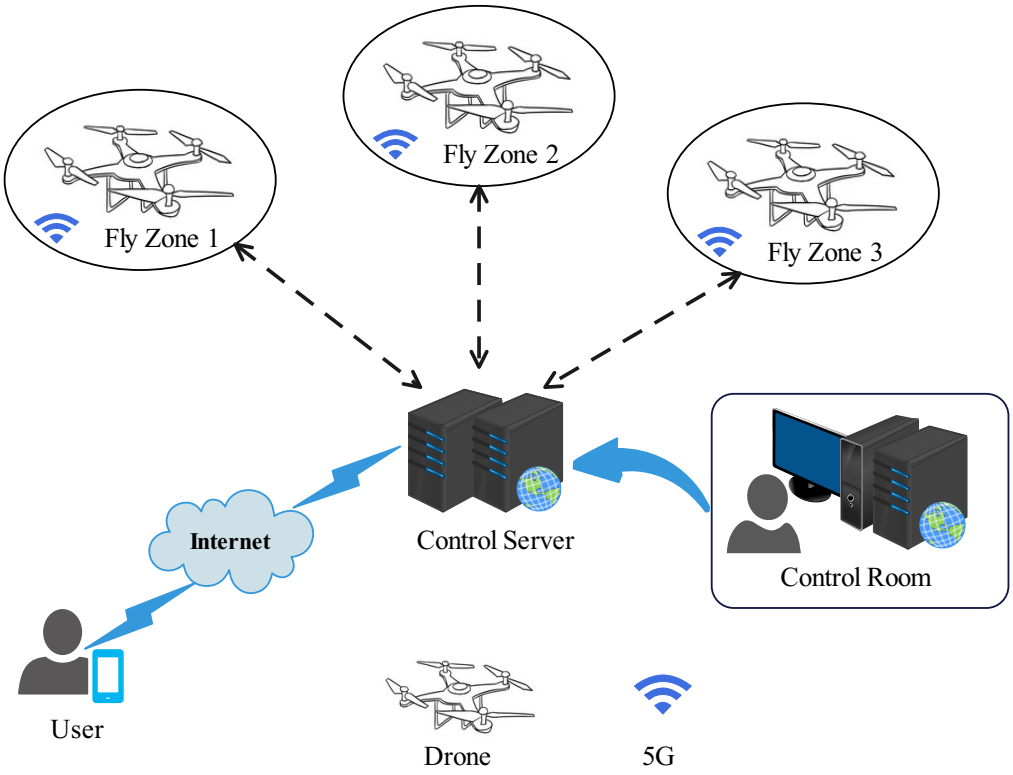


Figure 1. The 5G-enabled IoD architecture.

is provided in Section 7.

2 Related Work

Recently, many researchers have designed AKA schemes in the IoD environment to ensure the security of drone communication. Wazid et al. [12] conducted an examination and devised a secure user authentication scheme for remote access in the context of IoD deployment. However, Hussain et al. [13] found that their scheme suffered from impersonation, stolen verifier, and session key disclosure attacks. Simultaneously, they devised an AKA scheme tailored to smart city monitoring, with IoD as its underlying framework, and claimed to meet the required security features. Unfortunately, Wu et al. [6] established that their scheme exhibited vulnerabilities to privileged insider and drone capture attacks. Similarly, they devised an authentication scheme relying on 5G technology for drone communication, and claimed that the application of 5G in an IoD environment can enable drones to respond faster and transmit data faster. Srinivas et al. [19] put forward an anonymous AKA scheme for IoD using temporal certificates, and claimed that the scheme provides sufficient security. Bera et al. [20] put forward an authentication scheme for data transmission and collection relying on 5G-enabled IoD environment, incorporating elliptic curve cryptography (ECC) within their scheme. Zhang et al. [21] designed a lightweight AKA scheme within the IoD environment to guarantee the security of data transmission. However, Jan et al. [14] found that their scheme failed to deliver mutual authentication and was susceptible to replay, impersonation, and KSSTI attacks. Irshad et al. [5] put forward an authentication scheme tailored for the purpose of facilitating data delivery and collection in the context of IoD using 5G technology, and this scheme employed both blockchain and ECC. Akram et al. [22] designed an authentication scheme for controlling drone access in the context of smart city surveillance. However, Park et al. [23] illustrated the absence of perfect forward secrecy (PFS) in their scheme and suffered from drone impersonation, session key disclosure, and denial of services (DoS) attacks. Akram et al. [24] devised a blockchain-based privacy-preserving AKA scheme for drone networks, using ECC and digital signatures. Mishra et al. [25] designed a secure blockchain authentication key management architecture based on drone environments beyond 5G, and affirmed that their scheme could withstand widely recognized attacks and exhibits commendable performance.

In 2020, Alladi et al. [16] devised an innovative authentication scheme founded on PUF technology,

which guaranteed secure communication between drones and ground station servers, as well as among drones themselves. Gope et al. [26] designed an AKA scheme in an edge node assisted IoD environment, which used PUF technology in drones to uphold the security of stored data. Nair et al. [27] proposed a mutual AKA scheme founded on surveillance drone networks, incorporating both PUF and positioning technologies within their scheme. Alkatheiri et al. [28] put forward a sturdy authentication scheme tailored for the IoD, using PUF to ensure low-cost identity verification and key generation. Yu et al. [15] put forward a secure AKA scheme utilizing PUF in smart cities, and claimed that this scheme ensured higher security and better performance. Karmakar et al. [29] designed an intelligent adaptive session authentication scheme for IoD, which used PUF and eliminates the noise of PUF through a fuzzy extractor. Park et al. [23] designed a provably secure AKA scheme using PUF in IoD deployment, and it was stated that this scheme can overcome known security vulnerabilities. Ren et al. [8] proposed a drone-assisted AKA scheme in a 5G-enabled satellite-ground fusion network, which used PUF to ensure data security. Furthermore, the authors asserted that their scheme boasts significant advantages in terms of performance.

3 Preliminaries

This section reviews two core primitives used in our protocol: Physically Unclonable Functions (PUFs) and Fuzzy Extractors.

3.1 Physically Unclonable Function (PUF)

A PUF is a hardware-resident one-way function induced by uncontrollable, irreproducible manufacturing variations. Given a challenge $C \in \{0,1\}^x$, the device returns a response $R \in \{0,1\}^y$, forming a challenge–response pair (CRP) [34]. Two fundamental properties are:

- *Uniqueness*: responses from different devices to the same challenge are statistically unrelated;
- *Reproducibility*: responses from the same device to the same challenge are repeatable under stable conditions.

In practice, environmental and process variations — such as temperature shifts, supply-voltage fluctuations, and device aging — can cause the measured response $\tilde{R}$ to deviate slightly from the enrolled R. Moreover, exposing large sets of challenge–response pairs may enable modeling attacks that predict unseen responses. Therefore, protocols should neither store nor transmit raw PUF responses and should strictly limit CRP exposure.

In our setting, for a drone D_j we write $R_j = \mathrm{PUF}(C_j)$, where C_j is the challenge assigned to D_j and R_j is the corresponding response.

3.2 Fuzzy Extractor

A fuzzy extractor derives a stable, nearly uniform secret from a noisy, non-uniform source such as a PUF response, and later reproduces the same secret when given another sample that is sufficiently close to the original [35]. It consists of two algorithms:

- $\mathrm{Gen}(w) \rightarrow (sk, hd)$: Given a noisy source w, output a uniformly distributed key $sk \in \{0,1\}^\ell$ and public helper data hd. The helper data reveals only negligible information about sk and does not enable recovery of w.
- $\mathrm{Rep}(w', hd) \rightarrow sk$: Given a fresh sample w' and the helper data hd, reproduce the same key sk whenever the distance between w' and w under the chosen metric does not exceed a tolerance δ.

Instantiation in our protocol. On the drone side, we use the PUF response as the input to the fuzzy extractor during enrollment and authentication:

$$(K_j, P_j) = \mathrm{Gen}(R_j), \qquad K_j = \mathrm{Rep}(\mathrm{PUF}(C_j), P_j).$$

Only the helper data P_j is stored and its integrity is protected; raw responses R_j are neither persisted nor transmitted. This converts device-unique but noisy entropy into a stable secret while mitigating response exposure.

3.3 System and Adversary Model

System model. The system comprises a user U_i (with a mobile device, MD), a drone D_j equipped with a PUF and a fuzzy extractor, and a control server CS holding a long-term secret s. Registration uses a secure channel; all subsequent communication takes place over a public channel controlled by the adversary. Clocks are loosely synchronized and timestamps T_1, T_2, T_3 are accepted if $|T_k - T_c| \leq \Delta T$, as in Section 4.

The control server (CS) denotes a logical trusted service that may be realized by a set of trusted nodes under operational load. Our threat model and proofs treat the CS as a single logical principal and remain unchanged by this deployment choice.

To avoid a single point of failure and performance bottlenecks, the CS can be deployed with stateless

front-end nodes behind a load balancer. Persistent records—identities (HID_i, UR_i) and $(HDID_j, C_j)$, helper data P_j, challenge-allocation metadata, and revocation lists—reside in a replicated datastore with integrity protection; raw PUF responses R_j are neither stored nor transmitted. Challenge issuance uses leasing to ensure uniqueness across nodes and to bound CRP exposure, while identity-based rate limiting with backoff mitigates denial-of-service attacks. For wide-area networks, an optional hierarchical layout places edge CS instances near drones for enrollment and authentication, with a regional or root CS responsible for policy, auditing, and roaming. These deployment choices eliminate the CS as a bottleneck while preserving the protocol flow, the threat model, and the security conclusions.

Adversary capabilities. We adopt a Dolev–Yao network attacker with the following abilities:

- Network control: full eavesdropping, modification, injection, blocking, replay, and reordering on the public channel;
- Offline guessing: the adversary may attempt dictionary attacks against low-entropy secrets;
- Device capture (memory read): on U_i's MD, the adversary can extract $\{UT_i, UUi, UVi, UR_i, DID_j\}$ after acceptance; on D_j, it can read $\{HDID_j, SS_j, n_j, P_j\}$;
- Bounded PUF access: the adversary may issue up to q_p challenge queries to the target PUF (CRP exposure is bounded), but raw responses R_j are not stored or transmitted by the protocol.

4 Proposed Scheme

In order to address the vulnerabilities of existing IoD authentication schemes, such as drone capture attacks and known specific session temporary information attacks, we propose an improved AKA scheme for the drone environment (as shown in Figure 1). This scheme comprises both the registration, login and authentication phases, and the communication entities include U_i, CS and D_j. A summary of the notations in our proposed scheme is presented in Table 1.

4.1 Registration Phase

The registration has two phases, including drone registration and user registration.

Drone (D_j) registration phase. D_j needs to register with the CS before deployment. The processes for D_j registration are described in detail below and are shown in Figure 2.

Table 1. Notations used in this paper

Symbol	Description
U_i	i-th User
D_j	j-th Drone
CS	Control Server
MD	User's mobile device
ID_i / DID_j	Identities of U_i and D_j
HID_i / $HDID_j$	Temporary identities of U_i and D_j
PW_i	Password of U_i
s	CS's private key
a_i, r_i, r_j, n_j	Random nonces
C_j	PUF challenge assigned to D_j
R_j	PUF response of D_j to challenge C_j
$\mathrm{Gen}(\cdot)$, $\mathrm{Rep}(\cdot)$	Fuzzy extractor algorithms: key generation and key reproduction
T_1, T_2, T_3	Timestamps for freshness check
SK	Session key derived by U_i and D_j
$h(\cdot)$	One-way hash function
$\|$, $\oplus$	Concatenation, bitwise XOR

(1) First, D_j selects DID_j, n_j and computes $HDID_j = h(DID_j \| n_j)$. Lastly, D_j transmits $\{DID_j, n_j\}$ to CS via secure channel.

(2) After CS receives the DID_j, n_j, it selects C_j for D_j and subsequently calculates $SR_j = h(DID_j \| s \| n_j)$. Finally, CS stores the $\{DID_j, n_j, C_j\}$ in database, and transmits $\{SR_j, C_j\}$ to D_j.

(3) Upon receiving $\{SR_j, C_j\}$, D_j calculates $R_j = PUF(C_j)$, $(K_j, P_j) = Gen(R_j)$, $SS_j = DID_j \oplus SR_j$. Finally, D_j stores $\{HDID_j, SS_j, n_j, P_j\}$ in its database.

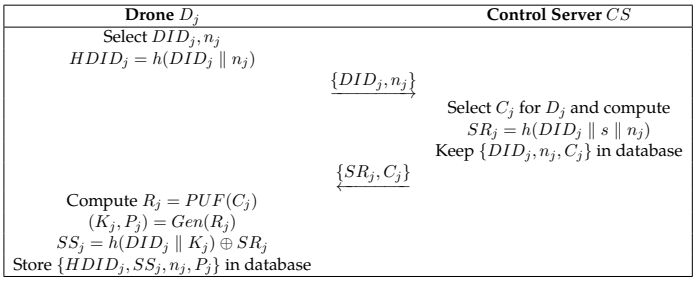


Figure 2. Registration phase for D_j.

User (U_i) registration phase. U_i need to register with the CS before they can get services. The process of U_i registration is illustrated in Figure 3, and the specific steps are elaborated below.

(1) To begin, U_i chooses ID_i, PW_i, and a random number a_i to calculate $HID_i = h(ID_i \| a_i)$, and then U_i uses a secure channel to deliver $\{HID_i, a_i\}$ to CS.

(2) CS computes $UR_i = h(a_i \| s), US_i =$

$h(HID_i \parallel UR_i \parallel s)$ after receiving the $\{HID_i, a_i\}$. Subsequently, CS saves $\{HID_i, UR_i\}$ in its database, whiling retrieves $\{DID_j\}$ and delivers the $\{UR_i, US_i, DID_j\}$ to U_i.

(3) When the $\{UR_i, US_i, DID_j\}$ is received, U_i computes $UU_i = h(HID_i \parallel PW_i \parallel a_i)$, $UT_i = a_i \oplus h(ID_i \parallel PW_i)$. $UV_i = h(a_i \parallel PW_i) \oplus US_i$. After that, U_i stores $\{UT_i, UU_i, UV_i, UR_i, DID_j\}$ in MD.

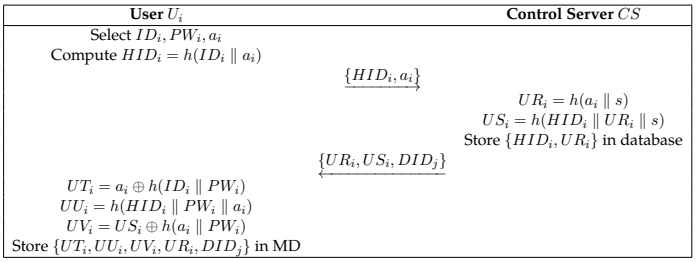


Figure 3. Registration phase for U_i.

4.2 Login and Authentication Phase

In this section, U_i completes login and achieves mutual authentication with CS, D_j, respectively. Furthermore, U_i and D_j establish the SK through the assistance of CS. The process of login and authentication is illustrated in Figure 4, with the precise steps elaborated as follows.

(1) To begin, U_i needs to input ID_i and PW_i into MD, and then MD begins calculating $a_i = UT_i \oplus h(ID_i \parallel PW_i)$, $HID_i = h(ID_i \parallel a_i)$, $UU_i^* = h(HID_i \parallel PW_i \parallel a_i)$, and verifies $UU_i^* \stackrel{?}{=} UU_i$. If this condition is met, U_i selects a random number r_i and DID_j to calculate $US_i = UV_i \oplus h(a_i \parallel PW_i)$, $I_1 = r_i \oplus h(UR_i \parallel US_i)$, $I_2 = DID_j \oplus h(r_i \parallel US_i)$. Then, U_i acquires T_1 to calculate $V_1 = h(HID_i \parallel DID_j \parallel r_i \parallel T_1)$. Lastly, U_i transmits the message $M_1 = \{HID_i, I_1, I_2, V_1, T_1\}$ to CS.

(2) On receiving the M_1, CS checks freshness of T_1. Then, CS uses HID_i to retrieve UR_i in the database, and computes $US_i = h(HID_i \parallel UR_i \parallel s)$, $r_i = I_1 \oplus h(UR_i \parallel US_i)$, $DID_j = I_2 \oplus h(r_i \parallel US_i)$, $V_1^* = h(HID_i \parallel DID_j \parallel r_i \parallel T_1)$, and checks $V_1^* \stackrel{?}{=} V_1$. If it holds, CS uses DID_j to retrieve n_j, and calculates $SR_j = h(DID_j \parallel s \parallel n_j)$, $I_3 = C_j \oplus h(n_j \parallel DID_j)$, $I_4 = (r_i \parallel US_i) \oplus h(DID_j \parallel SR_j)$. Then, CS retrieves T_2 to compute $V_2 = h(DID_j \parallel r_i \parallel SR_j \parallel T_2)$. At last, CS transmits the message $M_2 = \{HID_i, I_3, V_2, T_2\}$ to D_j.

(3) The initial stage when D_j receives M_2 is to validate T_2. Next, D_j computes $C_j = I_3 \oplus h(n_j \parallel DID_j)$, $K_j = Rep(PUF(C_j), P_j)$,$SR_j = SS_j \oplus h(DID_j \parallel K_j$, $(r_i \parallel US_i) = I_4 \oplus h(DID_j \parallel SR_j)$, $V_2^* = h(DID_j \parallel r_i \parallel SR_j \parallel T_2)$, and verifies $V_2^* \stackrel{?}{=} V_2$. If it is valid, D_j selects r_j to compute $SK = h(HID_i \parallel r_i \parallel HDID_j \parallel r_j)$, $I_5 = (HDID_j \parallel r_j) \oplus h(r_i \parallel US_i)$, $V_3 = h(HDID_j \parallel SK \parallel US_i \parallel T_3)$. Then, D_j transmits the message $M_3 = \{I_5, V_3, T_3\}$ to U_i.

(4) Upon receipt of the M_3, U_i confirms the validity of T_3 and calculates $(HDID_j \parallel r_j) = I_5 \oplus h(r_i \parallel US_i)$, $SK = h(HID_i \parallel r_i \parallel HDID_j \parallel r_j)$, $V_3^* = h(HDID_j \parallel SK \parallel US_i \parallel T_3)$, and verifies $V_3^* \stackrel{?}{=} V_3$. If it holds, this indicates that U_i and D_j successfully established the SK for future communications.

5 Security Analysis

5.1 Formal Security Analysis

To confirm the semantic security of the session key, we adopt the ROR model [30] in this section. In the ROR model, $\mathcal{A}$ employs various queries to simulate real attacks. The security of the session key is determined by whether $\mathcal{A}$ succeeds in winning the game.

Security Model. In our proposed scheme, three participants are involved: $\Pi_{U_i}^x$, Π_{CS}^y, and $\Pi_{D_j}^z$, denoting the x-th user instance, y-th control server instance, and z-th sensor node instance respectively. Moreover, $\mathcal{A}$ is capable of simulating both active and passive attacks by performing the following queries.

(1) $Execute(\mathcal{O})$: $\mathcal{A}$ can intercept messages exchanged between U_i, CS, and D_j by executing this query, where $\mathcal{O} = \{\Pi_{U_i}^x, \Pi_{D_j}^y, \Pi_{CS}^z\}$.

(2) $Send(\mathcal{O}, M_i)$: This query is that $\mathcal{A}$ sends a message M_i to $\mathcal{O}$, and then obtains the feedback.

(3) $Hash(string)$: When $\mathcal{A}$ runs this query and enters a $string$, it returns its hash value.

(4) $CorruptMD(\Pi_{U_i}^x)$: This query allows $\mathcal{A}$ to extract secret parameters from MD.

(5) $Test(\mathcal{O})$: The game starts by turning coin c. $\mathcal{A}$ may get the session key if $c = 1$. $\mathcal{A}$ yields a random number if $c = 0$. Otherwise, $\mathcal{A}$ gets an empty value.

We demonstrate the security of our system in the following theorem in accordance with the security model.

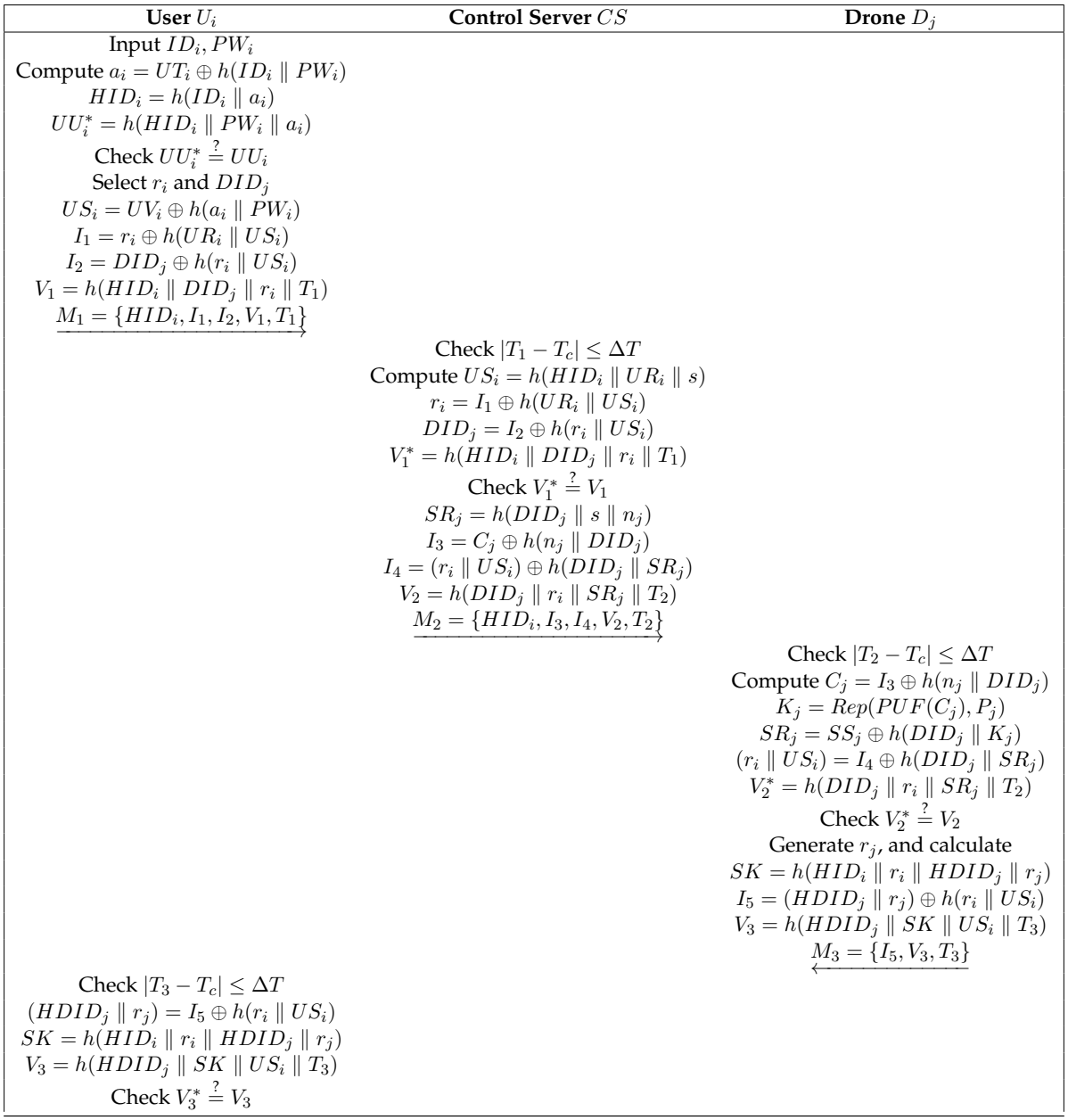


Figure 4. Login and authentication phase.

Theorem 1. *In the context of polynomial time ξ, the advantage of success for $\mathcal{A}$ in compromising our scheme is $Adv_{\mathcal{A}}^{\mathcal{P}}(\xi) \leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|PUF|} + 2C' \cdot q_s^{s'}$. Here, q_h, q_s and q_p refer to the number of hash functions, send queries, and PUF queries, respectively. The spaces of the hash function and PUF function are denoted by the variables $|Hash|$ and $|PUF|$. Lastly, there are two constants: C' and s'.*

Proof. We established five distinct games denoted as GM_i $(i = 0, 1, 2, 3, 4)$ in order to demonstrate that $\mathcal{A}$ is unable to acquire the session key. In the proof process, $Succ_A^{GM_i}(\xi)$ represents the likelihood of $\mathcal{A}$ achieving victory in GM_i, $Adv_{\mathcal{A}}^{\mathcal{P}}$ denotes the advantage of $\mathcal{A}$ in undermining the scheme. The detailed description of the process for $\mathcal{A}$ simulating queries are provided in Table 2, and the precise proof unfolds as follows.

GM_0: $\mathcal{A}$ only needs to choose coin c to start the game, refraining from making any queries. Hence, we obtain

$$Adv_{\mathcal{A}}^{\mathcal{P}}(\xi) = |2Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)] - 1|. \quad (1)$$

GM_1: On the basis of GM_0, $\mathcal{A}$ executes the $Execute()$ query to eavesdrop messages $M_1 = \{HID_i, I_1, I_2, V_1, T_1\}$, $M_2 = \{HID_i, I_3, I_4, V_2, T_2\}$, and $M_3 = \{I_5, V_3, T_3\}$. Subsequently, $\mathcal{A}$ computes SK through the execution of the $Test()$, where $SK = h(HID_i \| r_i \| HDID_j \| r_j)$. Since $\mathcal{A}$ cannot get $HDID_j, r_i$ and r_j, the SK cannot be calculated. Thus, the result of GM_1 does not increase compared to GM_0.

$$Pr[Succ_{\mathcal{A}}^{GM_1}(\xi)] = Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)]. \quad (2)$$

GM_2: Within this game, the queries $Send()$ and $Hash()$ are incorporated. As the values V_1, V_2, V_3 are safeguarded by a one-way hash function, $\mathcal{A}$ cannot manipulate or alter them. In addition, the r_i, r_j are different in each session, no hash collisions can occur. Hence, we can derive the subsequent outcomes based on the birthday paradox.

$$|Pr[Succ_{\mathcal{A}}^{GM_2}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_1}(\xi)]| \leq \frac{q_h^2}{2|Hash|}. \quad (3)$$

GM_3: This game is an expansion of GM_2, which adds the $PUF()$ query. Based on analogous argument provided by GM_2 and the security function of $PUF(\cdot)$, we have

$$|Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_2}(\xi)]| \leq \frac{q_p^2}{2|PUF|}. \quad (4)$$

GM_4: Through executing the $CorruptMD(\Pi_{U_i}^x)$ query, $\mathcal{A}$ obtains the $\{UT_i, UU_i, UV_i, UR_i, DID_j\}$ from MD, where $UT_i = a_i \oplus h(ID_i \parallel PW_i)$, $UU_i = h(HID_i \parallel PW_i \parallel a_i)$, $UV_i = US_i \oplus h(HID_i \parallel a_i)$. Since $\mathcal{A}$ does not have the correct ID_i and PW_i, $\mathcal{A}$ cannot compute a_i and US_i. Simultaneously guessing both ID_i and PW_i is a computationally infeasible endeavor for $\mathcal{A}$. Consequently, following Zipf's law [31], we can deduce that

$$|Pr[Succ_{\mathcal{A}}^{GM_4}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)]| \leq C' \cdot q_{send}^{s'}. \quad (5)$$

Ultimately, $\mathcal{A}$ aims to win the game by making a conjecture regarding c. Hence, we obtain the following

$$Pr[Succ_{\mathcal{A}}^{GM_4}(\xi)] = \frac{1}{2}. \quad (6)$$

Based on the four games described above, we can derive

$$\begin{aligned}
\frac{Adv_{\mathcal{A}}^{\mathcal{P}}(\xi)}{2} &= |Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)] - \frac{1}{2} \\
&= |Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_4}(\xi)]| \\
&= |Pr[Succ_{\mathcal{A}}^{GM_1}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_4}(\xi)]| \\
&\leq \sum_{i=0}^{2} |Pr[Succ_{\mathcal{A}}^{GM_{i+1}}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_i}(\xi)]| \\
&= \frac{q_h^2}{2|Hash|} + \frac{q_p^2}{2|PUF|} + C' \cdot q_{send}^{s'}.
\end{aligned} \quad (7)$$

Finally, we get the result is

$$Adv_{\mathcal{A}}^{\mathcal{P}}(\xi) \leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|PUF|} + 2C' \cdot q_s^{s'}. \quad (8)$$

5.2 Informal Security Analysis

5.2.1 *Drone Capture Attack*

Suppose the $\mathcal{A}$ captures the drone and uses power analysis to get data $\{HDID_j, SS_j, n_j, P_j\}$. Then, the $\mathcal{A}$ combines the intercepted HID_i, I_3 and I_4 to try to compute $C_j = I_3 \oplus h(n_j \parallel DID_j)$, $K_j = Rep(PUF(C_j), P_j)$, $SR_j = SS_j \oplus h(DID_j \parallel K_j)$. Since the security attributes of PUF and $\mathcal{A}$ cannot obtain DID_j, the $\mathcal{A}$ cannot obtain C_j, K_j and SR_j, and cannot continue to compute the correct SK. Hence, the proposed scheme can withstand drone capture attack.

5.2.2 *Known Specific-Session Temporary Information (KSSTI) Attack*

Supposing that the $\mathcal{A}$ can gain access to the random number r_i and use intercepted I_1 and I_1 to attempt to compute $h(UR_i \parallel US_i) = I_1 \oplus r_i, (r_i \parallel US_i) = I_4 \oplus h(DID_j \parallel SR_j)$. Since the $\mathcal{A}$ does not know DID_j and SR_j, they cannot obtain US_i. Similarly, assuming that the $\mathcal{A}$ obtains r_j generated by D_j and uses the intercepted I_5 to calculate $(HDID_j \parallel r_j) = I_5 \oplus h(r_i \parallel US_i)$. As r_i and US_i are confidential to $\mathcal{A}$, the $\mathcal{A}$ cannot obtain $HDID_j$. It is known that $\mathcal{A}$ is unable to calculate SK, thus indicating our scheme can resist the KSSTI attack.

5.2.3 *Perfect Forward Secrecy (PFS)*

We employ Ge et al.'s method [32] to illustrate that our proposed scheme guarantees PFS. The specific steps are outlined below.

(1) Initially, the variables $HID_i, r_i, HDID_j$, and r_j are required for the SK composition, which $SK = h(HID_i \parallel r_i \parallel HDID_j \parallel r_j)$. Then, we define SK as a node, and add the required variables around it, and point to it with an arrow. By analogy, list the required nodes in the proposed scheme completely. Here, we take r_i as an example. The computation of r_i can be achieved using either $\{SR_j, I_4, DID_j\}$ or $\{US_i, UR_i, I_1\}$.

(2) After listing all required nodes, mark all nodes as long-term keys or conveyed on public channels. This indicates that $\mathcal{A}$ can gain access to these variables, which include $HID_i, I_1, I_2, I_3, I_4$ and s.

Table 2. Simulation of various queries.

Query	Description
$Send(\mathcal{O}, M_i)$	On $Send(\Pi^x_{U_i}$, start). Let's assume that $\Pi^x_{U_i}$ is in a normal condition and picks r_i, DID_j, and T_1 to compute $US_i = UV_i \oplus h(a_i \parallel PW_i)$, $I_1 = r_i \oplus h(UR_i \parallel US_i)$, $I_2 = DID_j \oplus h(r_i \parallel US_i)$. Subsequently, the query yields $M_1 = \{HID_i, I_1, I_2, V_1, T_1\}$.
	For $Send(\Pi^y_{CS}, (HID_i, I_1, I_2, V_1, T_1))$. Let's assume that Π^y_{CS} computes US_i, r_i, DID_j, and validates V_1 under normal conditions. Next, Π^y_{CS} computes SR_j, I_3, I_4, V_2, and then chooses T_2. The query results in $M_2 = \{HID_i, I_3, I_4, V_2, T_2\}$.
	On $Send(I^z_{D_j}, (HID_i, I_3, I_4, V_2, T_2))$. On receiving the M_2, $\Pi^z_{D_j}$ computes $C_j, K_j, SR_j, (r_i \parallel US_i), V_2$ and checks V_2. Then, $\Pi^z_{D_j}$ chooses r_j, T_3 to compute SK, I_5, V_3. Finally, $\Pi^z_{D_j}$ returns the output $M_3 = \{I_5, V_3, T_3\}$.
	For $Send(\Pi^x_{U_i}, I_5, V_3, T_3)$. On receiving the M_3, $\Pi^x_{U_i}$ calculates $(HDID_j \parallel r_j), SK, V_3$ and verifies V_3.If V_3 is correct, it indicates that $\Pi^x_{U_i}$ accepts, and the entire process terminates.
$Execute(\mathcal{O})$	Continue simulating the $Execute(\mathcal{E})$ process based on the $Send()$ queries. $(HID_i, I_1, I_2, V_1, T_1) \longleftarrow Send(\Pi^x_{U_i}$, start), $(HID_i, I_3, I_4, V_2, T_2) \longleftarrow Send(\Pi^y_{CS}, (HID_i, I_1, I_2, V_1, T_1))$, $(I_5, V_3, T_3) \longleftarrow Send(\Pi^z_{D_j}, (HID_i, I_3, I_4, V_2, T_2))$. The query returns $(HID_i, I_1, I_2, V_1, T_1)$, $(HID_i, I_3, I_4, V_2, T_2)$, and (I_5, V_3, T_3).
$CorruptMD((\Pi^x_{U_i})$	The query returns data $\{UT_i, UU_i, UV_i, UR_i, DID_j\}$ from the mobile device on the condition that the $\Pi^x_{U_i}$ is accepted.
$Test(\mathcal{O})$	A coin toss determines the value of c. If the result is 1, the secret key SK will be provided. Otherwise, a randomly generated string of the same length as SK will be returned.

(3) Lastly, eliminate all incoming edges to the marked nodes and assess whether the $\mathcal{A}$ can calculate the SK based on the remaining nodes. According to Figure 5, the $\mathcal{A}$ cannot calculate the next required variable based on the obtained variables, and thus unable to calculate the SK.

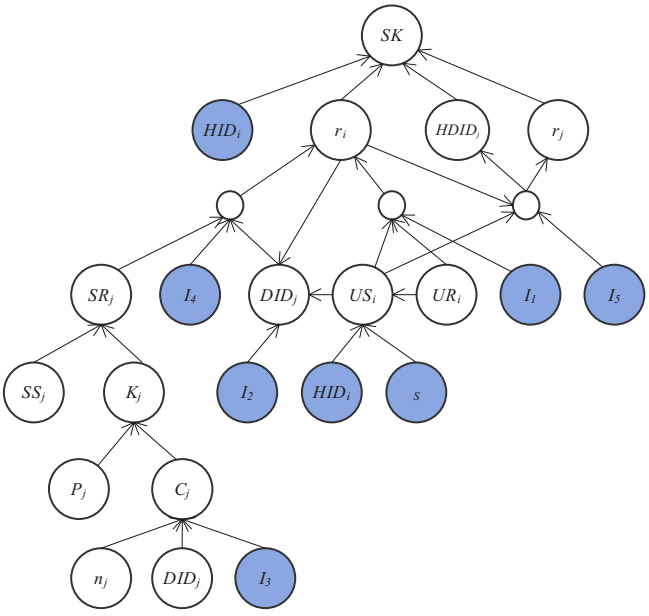


Figure 5. The result of proposed scheme for PFS.

5.2.4 *Privileged Insider Attack*

Assuming that the $\mathcal{A}$ is a malicious individual within CS, they can obtain data $\{HID_i, UR_i\}$ and $\{DID_j, n_j, C_j\}$. Then, the $\mathcal{A}$ uses this data and messages intercepted on the public channel to calculate the SK. The $\mathcal{A}$ intercepts I_4 and attempts to calculate $(r_i \parallel US_i) = I_4 \oplus h(DID_j \parallel SR_j)$, but DID_j and SR_j are confidential to the $\mathcal{A}$, so the $\mathcal{A}$ cannot calculate $(r_i \parallel US_i)$. Similarly, the $\mathcal{A}$ intercepts I_5 and calculates $(HDID_j \parallel r_j) = I_5 \oplus h(r_i \parallel US_i)$. However, without obtaining the value $(r_i \parallel US_i)$, the $\mathcal{A}$ cannot compute $(HDID_j \parallel r_j)$. Therefore, the $\mathcal{A}$ cannot calculate the SK of the proposed scheme, which show that the scheme can resist privileged insider attack.

5.2.5 Session Key Disclose Attack

Assuming the $\mathcal{A}$ intercepts messages $M_1 - M_3$, and attempts to calculate $(HDID_j \parallel r_j) = I_5 \oplus h(r_i \parallel US_i), (r_i \parallel US_i) = I_4 \oplus h(DID_j \parallel SR_j)$. However, without DID_j and SR_j, the $\mathcal{A}$ cannot calculate $(r_i \parallel US_i)$, and $(HDID_j \parallel r_j)$. This indicates that the $\mathcal{A}$ cannot obtain the SK through intercepted messages. Hence, the proposed scheme can effectively withstand session key disclosure attack.

5.2.6 Post-quantum Considerations

We briefly consider a quantum-capable adversary. Our protocol relies solely on symmetric primitives (hash/XOR) and a PUF-anchored fuzzy extractor; there are no public-key operations in the protocol flow. Against quantum attacks, Grover's algorithm provides only a quadratic speed-up over exhaustive search. With 256-bit hash outputs and 256-bit derived keys, the effective post-quantum security remains at least 128 bits. When exporting the session key to upper layers, the scheme can be combined with PSK-only or hybrid post-quantum transports. These observations do not alter the protocol messages, threat model, or the security results established in this paper.

5.3 AVISPA

AVISPA [33] is a widely known tool for analyzing schemes, which can be used to formally evaluate and verify Internet security schemes. This tool is grounded in the DY model and utilizes the High-Level Protocol Specification Language (HLPSL) for the description and definition of security schemes. Additionally, AVISPA can assist researchers in identifying and addressing vulnerabilities in authentication schemes, thereby enhancing the security of the proposed scheme.

Here, we utilize the AVISPA tool to assess the security and validity of our scheme. The detailed descriptions of the user, control server, and drone roles are shown in Figures 6(a), 6(b), and 6(c), respectively. The definitions in Figure 6(d) include session and environmental roles, and security goals. In the simulation, we used widely recognized On-the-Fly Model-Checker (OFMC) and Constraint Logic based Attack Searcher (CL-AtSe) backends to validate the proposed scheme, and the verification results are shown in Figure 7. From the results, it can be seen that the summary displays "SAFE", indicating that our proposed scheme can resist man-in-the-middle and replay attacks.

6 Security and Performance Comparisons

In this section, we enumerate the security properties satisfied by the proposed scheme and compare it with similar schemes [13, 21–23]. At the same time, we also evaluate and compare the performance of each scheme.

6.1 Security Comparisons

We evaluate the security of the proposed scheme by conducting a comparative analysis with other schemes. The main security features include S1, privileged insider attack; S2, DoS attack; S3, replay attack; S4, KSSTI attack; S5, impersonation attack; S6, session key disclosure attack; S7, drone capture attack; S8, mutual authentication; S9, PFS. Here, ✓ means that this scheme can ensure the feature or can resist the attack, × means that this scheme violates the feature or cannot resist the attack.

The results of the security comparison are displayed in Table 3. It is evident that our proposed scheme, as well as Park et al.'s scheme [23], align with the listed security features. In contrast, Zhang et al.'s scheme [21] suffered from replay, KSSTI, impersonation attacks, and failed to provide mutual authentication. Hussain et al.'s scheme [13] was susceptible to privileged insider, impersonation, and drone capture attacks. Akram et al.'s scheme [22] suffered from DoS, impersonation, session key disclosure attacks, while also violated PFS.

Table 3. Comparisons of security.

Security Features	Zhang et al. [21]	Hussain et al. [13]	Akram et al. [22]	Park et al. [23]	Ours
S1	✓	× [6]	✓	✓	✓
S2	✓	✓	× [23]	✓	✓
S3	× [14]	✓	✓	✓	✓
S4	× [14]	✓	✓	✓	✓
S5	× [14]	× [6]	× [23]	✓	✓
S6	✓	✓	× [23]	✓	✓
S7	✓	× [6]	✓	✓	✓
S8	× [14]	✓	✓	✓	✓
S9	✓	✓	× [23]	✓	✓

6.2 Performance Comparisons

Each scheme compares communication and computational costs in performance comparison, and only involves the login and authentication phases of the scheme in comparison.

6.2.1 Computational Cost Comparisons

We utilize various devices to obtain execution times for cryptographic primitives and employ the acquired results to contrast the computational costs

```
role user(UA,CS,DN: agent, SKuacs: symmetric_key,H: hash_func,SND,
RCV: channel(dy))
played_by UA
def=
local State: nat,
IDi,PWi,Ai,S,HIDi,URi,USi,UTi,UUi,UVi,DIDj,Nj,
HDIDj,Cj,SRj,SSj,Kj,Ri: text,
I1,I2,V1,I3,I4,V2,Rj,I5,V3,T1,T2,T3: text,
SKij,SKji: text
const sp1,sp2,sp3,sp4,sp5,ua_cs_ri,cs_ua_ri,cs_dn_ri,dn_ua_rj: protocol_id
init State:=0
transition
%%%%Registration phase%%%%
1.State =0/\RCV(start)=|>
  State':= 1/\Ai':= new()
      /\HIDi':= H(IDi.Ai')
      /\SND({HIDi'.Ai'}_SKuacs)
      /\secret({HIDi'},sp1,{UA})
      /\secret({Ai'},sp2,{UA,CS})

2.State = 1/\RCV({H(Ai'.S).H(H(IDi.Ai').H(Ai'.S).S)}_SKuacs)=|>
  State':= 2/\ UTi':= xor(Ai',H(IDi.PWi))
      /\UUi':= H(H(IDi.Ai').PWi.Ai')
      /\UVi':=xor(H(H(IDi.Ai').H(Ai'.S).S),H(Ai'.PWi))
      /\ Ri':= new()
      /\ T1':= new()
      /\ I1':= xor(Ri',H(H(Ai'.S).H(H(H(IDi.Ai').H(Ai'.S).S))))
      /\ I2':= xor(DIDj,H(Ri'.H(H(H(IDi.Ai').H(Ai'.S).S))))
      /\ V1':= H(H(IDi.Ai').DIDj.Ri'.T1')
      /\ SND(H(IDi.Ai').I1'.I2'.V1'.T1')
      /\ witness(UA,CS,ua_cs_ri,Ri')

3.State = 2/\RCV(xor((H(DIDj.Nj').Nj'),H(Ri'.H(H(IDi.Ai').H(Ai'.S).S))))=|>
State':=3/\Rj':=new()
      /\SKij':= H(H(H(IDi.Ai').Ri').H(H(DIDj.Nj').Rj'))
      /\request(DN,UA,dn_ua_rj,Rj')
end role
```

(a) User role.

```
role server(UA,CS,DN: agent, SKuacs,SKdncs: symmetric_key,H:
hash_func,SND, RCV: channel(dy))
played_by CS
def=
local State: nat,
IDi,PWi,Ai,S,HIDi,URi,USi,UTi,UUi,UVi,DIDj,
Nj,HDIDj,Cj,SRj,SSj,Kj,Ri: text,
I1,I2,V1,I3,I4,V2,Rj,I5,V3,T1,T2,T3: text,
SKij,SKji: text
const sp1,sp2,sp3,sp4,sp5,ua_cs_ri,cs_ua_ri,cs_dn_ri,dn_ua_rj: protocol_id
init State:=0
transition
1.State =0/\RCV({H(IDi.Ai').Ai'}_SKuacs)=|>
   State':= 1 /\URi' :=H(Ai'.S)
      /\USi':= H(H(IDi.Ai').H(Ai'.S).S)
      /\SND({URi'.USi'}_SKuacs)
      /\secret({S},sp3,{CS})

2.State = 1/\RCV({DIDj.Nj'}_SKdncs)=|>
   State':= 2/\Cj':=new()
      /\SRj':= H(DIDj.S.Nj')
      /\SND({SRj'.Cj'}_SKdncs)
      /\secret({S},sp4,{CS})
      /\secret({Cj,SRj'},sp5,{DN,CS})

3.State = 2/\ RCV(H(IDi.Ai').xor(Ri',H(H(Ai'.S).H(H(H(IDi.Ai').H(Ai'.S).S)))).
xor(DIDj,H(Ri'.H(H(H(IDi.Ai').H(Ai'.S).S)))).H(H(IDi.Ai').DIDj.Ri'.T1').T1')
=|>State':= 3 /\T2' :=new()
      /\Nj':=new()
      /\Cj':=new()
      /\I3':=xor(Cj',H(Nj'.DIDj))
      /\I4':=xor((Ri'.H(H(IDi.Ai').H(Ai'.S).S)),H(DIDj.H(DIDj.S.Nj')))
      /\V2':=H(DIDj.Ri'.H(DIDj.S.Nj').T2')
      /\ SND(H(IDi.Ai').I3'.I4'.V2'.T2')
      /\ witness(CS,UA, cs_ua_ri, Ri')
      /\ request(UA,CS, ua_cs_ri, Ri')
end role
```

(b) Control server role.

```
role drone(UA,CS,DN: agent, SKdncs: symmetric_key,H: hash_func,SND,
RCV: channel(dy))
played_by DN
def=
local State: nat,
IDi,PWi,Ai,S,HIDi,URi,USi,UTi,UUi,UVi,DIDj,
Nj,HDIDj,Cj,SRj,SSj,Kj,Ri: text,
I1,I2,V1,I3,I4,V2,Rj,I5,V3,T1,T2,T3: text,
SKij,SKji: text
const sp1,sp2,sp3,sp4,sp5,ua_cs_ri,cs_ua_ri,cs_dn_ri,dn_ua_rj: protocol_id
init State:=0
transition
1.State=0/\ RCV(start)=|>
   State':= 1/\ Nj':=new()
      /\HDIDj':=H(DIDj.Nj')
      /\SND({DIDj.Nj'}_SKdncs)

2.State = 1/\RCV({(H(DIDj.S.Nj').Cj')}_SKdncs)=|>State':= 2
      /\ SSj':= xor(H(DIDj.Kj),H(DIDj.S.Nj'))

3.State = 2/\ RCV(H(IDi.Ai').xor(Cj',H(Nj'.DIDj)).xor((Ri'.H(H(IDi.Ai').
H(Ai'.S).S)),H(DIDj.H(DIDj.S.Nj'))).H(DIDj.Ri'.H(DIDj.S.Nj').T2').T2')
=|>State':= 3
      /\ Rj':= new()
      /\ T3':= new()
      /\ SKji':= H(H(H(IDi.Ai').Ri').H(H(DIDj.Nj').Rj'))
      /\ I5':= xor((H(DIDj.Nj').Rj'),(Ri'.H(H(IDi.Ai').H(Ai'.S).S)))
      /\ V3':=H(H(DIDj.Nj').SKji'.H(H(IDi.Ai').H(Ai'.S).S).T3')
      /\ SND(I5',V3',T3')
      /\ witness(DN,UA, dn_ua_rj,Rj')
      /\ request(CS,DN, cs_dn_ri,Ri')
end role
```

(c) Drone role.

```
%%%%Role for the session%%%%
role session(UA,CS,DN : agent, SKuacs,SKdncs: symmetric_key,
H:hash_func)
def=
local SN1,SN2,SN3, RV1, RV2,RV3: channel(dy)
composition
user(UA,CS,DN,SKuacs, H,SN1,RV1)
      /\server(UA,CS,DN,SKuacs,SKdncs, H, SN2,RV2)
      /\drone(UA,CS,DN,SKdncs,H,SN3,RV3)
end role
%%%%Role for the environment%%%%
role environment()
def=
const ua, cs, dn: agent,
skuacs,skdncs: symmetric_key,
h: hash_func,
idi,didj: text,
ua_cs_ri,cs_ua_ri,cs_dn_ri,dn_ua_rj: protocol_id,
sp1,sp2,sp3,sp4,sp5: protocol_id
intruder_knowledge = {ua,cs,dn,idi,didj,h}
composition
session(ua,cs,dn,skuacs,skdncs,h)/\session(i,cs,dn,skuacs,skdncs,h)
      /\session(ua,i,dn,skuacs,skdncs,h)
      /\session(ua,cs,i, skuacs,skdncs,h)
end role
%%%%Role for the goal%%%%
goal
secrecy_of sp1, sp2, sp3, sp4, sp5
authentication_on ua_cs_ri
authentication_on cs_ua_ri
authentication_on cs_dn_ri
authentication_on dn_ua_rj
end goal
environment()
```

(d) Session, environment roles and security goal.

Figure 6. Proof of AVISPA.

Table 4. The configuration of equipment and operations time.

	IQOO9	Legion R9000P	Pi3 B+ [13]
Operating System	Android system	Windows 11	Ubuntu 16.0 LTS
CPU	Qualcomm Snapdragon 8 Gen 1	AMD Ryzen 7 6800H with Radeon Graphics 3.20 GHz	Cortex-A53(ARMv8) 64-bit SoC @ 1.4GHz
Running Memory	12 GB	16 GB	1 GB
Symmetric key encryption/decryption	0.2136ms	0.1358 ms	0.013 ms
Hash function	0.0053 ms	0.0031 ms	0.006 ms

Table 5. Computational cost comparison.

Schemes	U_i (ms)	CS (ms)	D_j (ms)
Zhang et al. [21]	$10T_h = 0.053$	$7T_h = 0.0217$	$7T_h = 0.042$
Hussain et al. [13]	$15T_h + T_f = 0.0848$	$9T_h + 2T_s = 0.2995$	$7T_h = 0.042$
Akram et al. [22]	$9T_h + T_f = 0.053$	$7T_h + 2T_s = 0.2933$	$7T_h = 0.042$
Park et al. [23]	$11T_h + T_f = 0.0636$	$11T_h = 0.0341$	$10T_h + T_f = 0.066$
Ours	$10T_h = 0.053$	$8T_h = 0.0248$	$7T_h + T_f = 0.048$

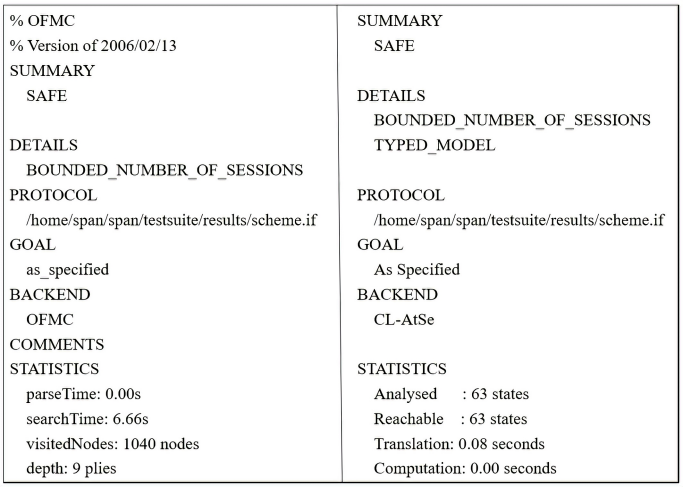


Figure 7. The OFMC and CL-AtSe results.

of each scheme. All timings were collected with single-threaded Java code using standard JCA/JCE implementations. Here, we use IQOO9 mobile phones to simulate the U_i, Lenovo computer to simulate the CS, and Pi3 B+ to simulate the D_j [13]. The development software is IntelliJ IDEA 2021.3, and we use the Java language to implement the measurement program. Hash operations are instantiated as SHA-256 via `java.security.MessageDigest` with fixed-length inputs matching protocol-level fields (e.g., $|ID| = 160$ bits, $|Z_p^*| = 128$ bits, $|T| = 32$ bits), and the reported value is the per-call latency of a single digest. Symmetric encryption/decryption times in Table 4 are obtained with AES-256-GCM using `javax.crypto.Cipher`, where plaintexts are fixed-size 128-byte buffers representative of protocol messages. The program is based on the cryptography library JPBC-2.0.0 [36] to support baselines that require ECC/pairing; our proposed scheme itself uses no public-key operations and relies only on hash/XOR.

Given the execution time of the hash operation is comparable to that of the fuzzy extraction operation, we utilize the time of hash operation for comparison. Additionally, Alladi et al. [16] stated that the response time of the PUF is only $0.4\,\mu s$, and the operation times for $\|$ and $\oplus$ are extremely short, so we ignored these three operations in the comparison. The computational costs for each scheme are detailed in Table 5, while Figure 8 provides a clearer visualization of the same data.

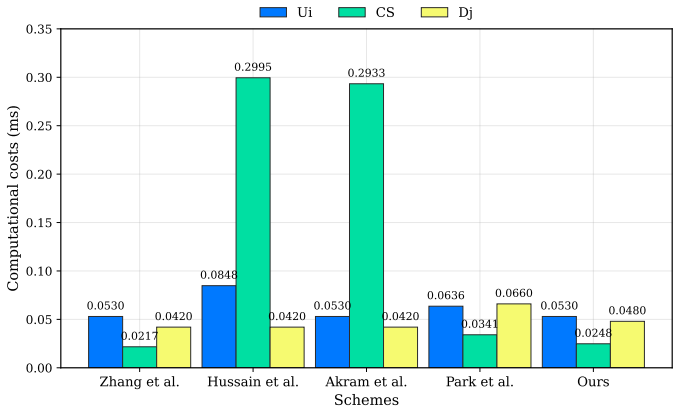
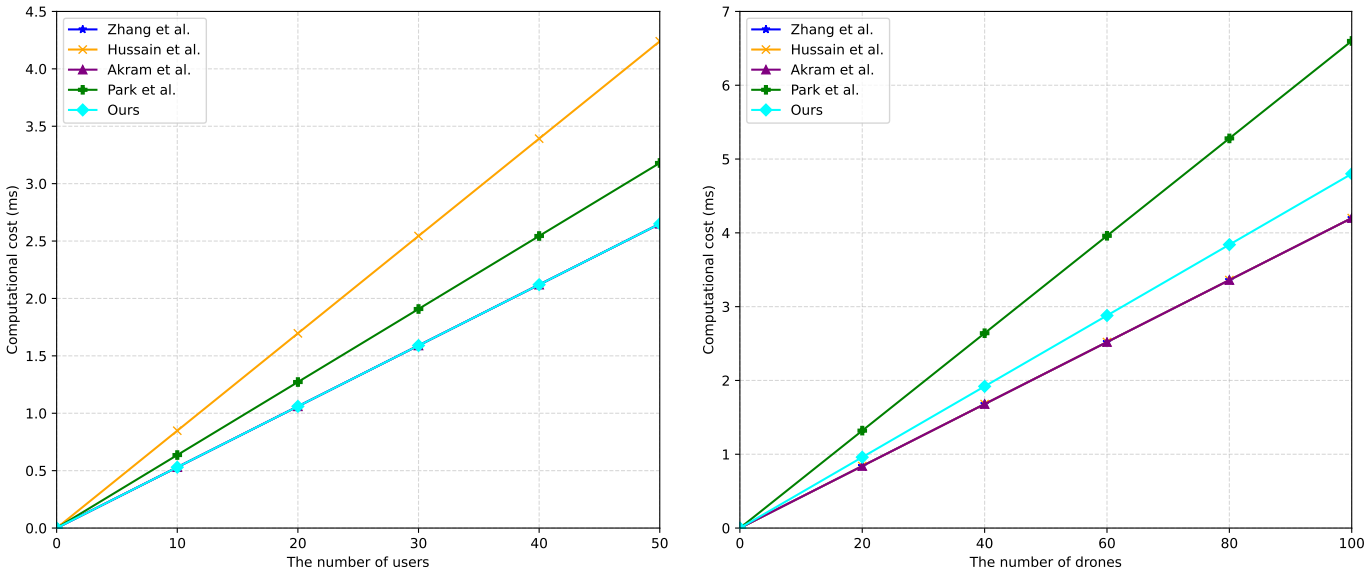


Figure 8. Computational cost comparison.

The computational costs as the number of U_i gradually increases are shown in the Figure 9(a). It can be observed that the U_i in Hussain et al.'s scheme [13] have the highest computational costs. Additionally, our scheme has the same computational costs for the



(a) Computational costs for users. (b) Computational costs for drones.

Figure 9. The computational costs for users and drones.

U_i as Zhang et al. [21] and Akram et al. [22]. As the number of D_j increases, the computational cost is depicted in Figure 9(b). In our proposed scheme, the computational cost of D_j is lower compared to the Park et al.'s scheme [23], yet slightly higher than the other schemes. It should be noted that the computational costs of D_j for the schemes proposed by Zhang et al. [21], Hussain et al. [13], and Akram et al. [22] are the same.

6.2.2 *Communication Cost Comparisons*

We evaluate the communication cost of each scheme for the messages sent over the public channel. Here, we specify the bit lengths of various elements as follows: symmetric encrypted ciphertext $|E|$ is 256 bits, the hash function $|H|$ is 256 bits, identity $|ID|$ is 160 bits, timestamp $|T|$ is 32 bits, and random number $|Z_p^*|$ is 128 bits.

In this context, our proposed scheme's overall communication costs need $2|ID|+5|Z_p^*|+3|H|+3|T| = 2 \times 160 + 5 \times 128 + 3 \times 256 + 3 \times 32 = 1824$ bits, Zhang et al.'s scheme [21] needs $6|Z_p^*| + 3|H| + |T| = 6 \times 128 + 3 \times 256 + 32 = 1568$ bits, Hussain et al.'s scheme [13] needs $|ID| + 4|Z_p^*| + 5|H| + 2|E| + 3|T| = 160 + 4 \times 128 + 5 \times 256 + 2 \times 256 + 3 \times 32 = 2560$ bits, Akram et al.'s scheme [22] needs $4|Z_p^*| + 3|H| + 2|E| = 4 \times 128 + 3 \times 256 + 2 \times 256 = 1792$ bits, and Park et al.'s scheme [23] needs $5|Z_p^*| + 5|H| = 5 \times 128 + 5 \times 256 = 1920$ bits.

Table 6 and Figure 10 collectively demonstrate that the communication costs required by our scheme are higher than those of the schemes in [21] and [22]. Nevertheless, when compared to the schemes in [13] and [23], our proposed scheme consistently maintains the lowest communication costs.

Table 6. Communication cost comparison.

Schemes	Rounds	Communication cost (bits)
Zhang et al. [21]	3	1568
Hussain et al. [13]	3	2560
Akram et al. [22]	3	1792
Park et al. [23]	3	1920
Ours	3	1824

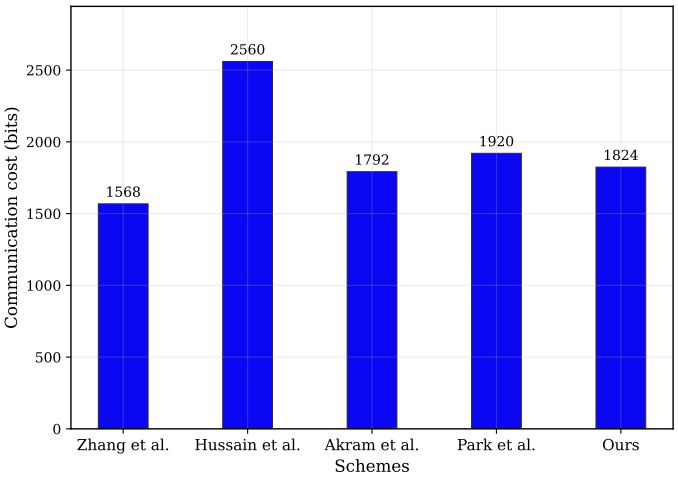


Figure 10. Communication cost comparison.

Analyzing the security comparison, we can infer that

our proposed scheme, along with the scheme in [23], satisfies all the listed security properties. In contrast, the other schemes exhibit varying degrees of security vulnerabilities. Although this scheme provides comparable security to our proposed scheme, our scheme outperforms it in terms of both computational and communication costs.

7 Conclusion

In this paper, we presented an overview of the Internet of Drones (IoD) and its integration with 5G, and introduced a 5G-based communication architecture for drones. We surveyed AKA schemes in IoD settings, identified security gaps, and proposed a PUF-assisted AKA protocol to address them. We evaluated security by informal analysis, the ROR model, and AVISPA verification, and compared our scheme with representative baselines, showing stronger security with competitive performance. In addition, the protocol uses only lightweight operations—hashing and XOR—performs one PUF evaluation and one fuzzy-extractor reproduction per session, and completes in a small constant number of messages; as a result, both computation and communication costs remain low, enabling seamless integration at IoT scale.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Lee, T. F., Lou, D. C., & Chang, C. H. (2023). Enhancing lightweight authenticated key agreement with privacy protection using dynamic identities for Internet of Drones. *Internet of Things, 23*, 100877. [CrossRef]

[2] Pu, C., Wall, A., Choo, K. K. R., Ahmed, I., & Lim, S. (2022). A lightweight and privacy-preserving mutual authentication and key agreement protocol for Internet of Drones environment. *IEEE Internet of Things Journal, 9*(12), 9918-9933. [CrossRef]

[3] Javaid, N., Sher, A., Nasir, H., & Guizani, N. (2018). Intelligence in IoT-based 5G networks: Opportunities and challenges. *IEEE Communications Magazine, 56*(10), 94–100. [CrossRef]

[4] Ying, B., & Nayak, A. (2019). Lightweight remote user authentication protocol for multi-server 5G networks using self-certified public key cryptography. *Journal of network and computer applications, 131*, 66–74. [CrossRef]

[5] Irshad, A., Chaudhry, S. A., Ghani, A., & Bilal, M. (2021). A secure blockchain-oriented data delivery and collection scheme for 5G-enabled IoD environment. *Computer Networks, 195*, 108219. [CrossRef]

[6] Wu, T., Guo, X., Chen, Y., Kumari, S., & Chen, C. (2021). Amassing the security: An enhanced authentication protocol for drone communications over 5G networks. *Drones, 6*(1), 10. [CrossRef]

[7] Feng, C., Liu, B., Guo, Z., Yu, K., Qin, Z., & Choo, K.-K. R. (2021). Blockchain-based cross-domain authentication for intelligent 5G-enabled internet of drones. *IEEE Internet of Things Journal, 9*(8), 6224–6238. [CrossRef]

[8] Ren, X., Cao, J., Ma, R., Luo, Y., Guan, J., Zhang, Y., & Li, H. (2023). A Novel Access and Handover Authentication Scheme in UAV-Aided Satellite-Terrestrial Integration Networks Enabling 5G. *IEEE Transactions on Network and Service Management, 20*(3), 3880-3899. [CrossRef]

[9] Messerges, T. S., Dabbish, E. A., & Sloan, R. H. (2002). Examining smart-card security under the threat of power analysis attacks. *IEEE transactions on computers, 51*(5), 541–552. [CrossRef]

[10] Liu, S., & Chen, C.-M. (2022). Comments on "a secure and lightweight drones-access protocol for smart city surveillance". *IEEE Transactions on Intelligent Transportation Systems, 23*(12), 25054–25058. [CrossRef]

[11] Ever, Y. K. (2020). A secure authentication scheme framework for mobile-sinks used in the internet of drones applications. *Computer Communications, 155*, 143–149. [CrossRef]

[12] Wazid, M., Das, A. K., Kumar, N., Vasilakos, A. V., & Rodrigues, J. J. P. C. (2018). Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment. *IEEE Internet of Things Journal, 6*(2), 3572–3584. [CrossRef]

[13] Hussain, S., Mahmood, K., Khan, M. K., Chen, C.-M., Alzahrani, B. A., & Chaudhry, S. A. (2022). Designing secure and lightweight user access to drone for smart city surveillance. *Computer Standards & Interfaces, 80*, 103566. [CrossRef]

[14] Jan, S. U., Abbasi, I. A., & Algarni, F. (2022). A

Mutual Authentication and Cross Verification Protocol for Securing Internet-of-Drones (IoD). *Computers, Materials and Continua, 72*(3), 5845-5869. [CrossRef]

[15] Yu, S., Das, A. K., Park, Y., & Lorenz, P. (2022). SLAP-IoD: Secure and lightweight authentication protocol using physical unclonable functions for internet of drones in smart city environments. *IEEE Transactions on Vehicular Technology, 71*(10), 10374–10388. [CrossRef]

[16] Alladi, T., Bansal, G., Chamola, V., & Guizani, M. (2020). SecAuthUAV: A novel authentication scheme for UAV-ground station and UAV-UAV communication. *IEEE Transactions on Vehicular Technology, 69*(12), 15068–15077. [CrossRef]

[17] Bera, B., Vangala, A., Das, A. K., Lorenz, P., & Khan, M. K. (2022). Private blockchain-envisioned drones-assisted authentication scheme in IoT-enabled agricultural environment. *Computer Standards & Interfaces, 80*, 103567. [CrossRef]

[18] Kumar, A., Yadav, A. S., Gill, S. S., Pervaiz, H., Ni, Q., & Buyya, R. (2022). A secure drone-to-drone communication and software defined drone network-enabled traffic monitoring system. *Simulation Modelling Practice and Theory, 120*, 102621. [CrossRef]

[19] Srinivas, J., Das, A. K., Kumar, N., & Rodrigues, J. J. P. C. (2019). TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of drones environment. *IEEE Transactions on Vehicular Technology, 68*(7), 6903–6916. [CrossRef]

[20] Bera, B., Saha, S., Das, A. K., Kumar, N., Lorenz, P., & Alazab, M. (2020). Blockchain-envisioned secure data delivery and collection scheme for 5g-based iot-enabled internet of drones environment. *IEEE Transactions on Vehicular Technology, 69*(8), 9097–9111. [CrossRef]

[21] Zhang, Y., He, D., Li, L., & Chen, B. (2020). A lightweight authentication and key agreement scheme for Internet of Drones. *Computer Communications, 154*, 455–464. [CrossRef]

[22] Akram, M. W., Bashir, A. K., Shamshad, S., Saleem, M. A., AlZubi, A. A., Chaudhry, S. A., Alzahrani, B. A., & Zikria, Y. B. (2021). A secure and lightweight drones-access protocol for smart city surveillance. *IEEE Transactions on Intelligent Transportation Systems, 23*(10), 19634–19643. [CrossRef]

[23] Park, Y., Ryu, D., Kwon, D., & Park, Y. (2023). Provably secure mutual authentication and key agreement scheme using PUF in internet of drones deployments. *Sensors, 23*(4), 2034. [CrossRef]

[24] Akram, M. A., Ahmad, H., Mian, A. N., Jurcut, A. D., & Kumari, S. (2023). Blockchain-based privacy-preserving authentication protocol for UAV networks. *Computer Networks, 224*, 109638. [CrossRef]

[25] Mishra, A. K., Wazid, M., Singh, D. P., Das, A. K., Singh, J., & Vasilakos, A. V. (2023). Secure Blockchain-Enabled Authentication Key Management Framework with Big Data Analytics for Drones in Networks Beyond 5G Applications. *Drones, 7*(8), 508. [CrossRef]

[26] Gope, P., & Sikdar, B. (2020). An efficient privacy-preserving authenticated key agreement scheme for edge-assisted internet of drones. *IEEE Transactions on Vehicular Technology, 69*(11), 13621–13630. [CrossRef]

[27] Nair, A. S., & Thampi, S. M. (2022). PUFloc: PUF and location based hierarchical mutual authentication protocol for surveillance drone networks. In *International Conference on Ubiquitous Security* (pp. 66–89). Springer. [CrossRef]

[28] Alkatheiri, M. S., Saleem, S., Alqarni, M. A., Aseeri, A. O., Chauhdary, S. H., & Zhuang, Y. (2022). A lightweight authentication scheme for a network of unmanned aerial vehicles (UAVs) by using physical unclonable functions. *Electronics, 11*(18), 2921. [CrossRef]

[29] Karmakar, R., Kaddoum, G., & Akhrif, O. (2023). A PUF and fuzzy extractor-based UAV-ground station and UAV-UAV authentication mechanism with intelligent adaptation of secure sessions. *IEEE Transactions on Mobile Computing, 23*(5), 3858-3875. [CrossRef]

[30] Abdalla, M., Fouque, P. A., & Pointcheval, D. (2005, January). Password-based authenticated key exchange in the three-party setting. In *International workshop on public key cryptography* (pp. 65-84). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[31] Wang, D., Cheng, H., Wang, P., Huang, X., & Jian, G. (2017). Zipf's law in passwords. *IEEE Transactions on Information Forensics and Security, 12*(11), 2776–2791. [CrossRef]

[32] Ge, M., Kumari, S., & Chen, C. M. (2022). AuthPFS: A method to verify perfect forward secrecy in authentication protocols. *J. Netw. Intell, 7*(3), 734-750.

[33] Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuéllar, J., ... & Vigneron, L. (2005, July). The AVISPA tool for the automated validation of internet security protocols and applications. In *International conference on computer aided verification* (pp. 281-285). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[34] Liang, Y., Luo, E., & Liu, Y. (2023). Physically Secure and Conditional-Privacy Authenticated Key Agreement for VANETs. *IEEE Transactions on Vehicular Technology, 72*(6), 7914-7925. [CrossRef]

[35] Zhu, L., & Wang, D. (2024). Robust Multi-Factor Authentication for WSNs With Dynamic Password Recovery. *IEEE Transactions on Information Forensics and Security, 19*, 8398-8413. [CrossRef]

[36] De Caro, A., & Iovino, V. (2011). jPBC: Java pairing based cryptography. In *2011 IEEE Symposium on Computers and Communications (ISCC)* (pp. 850-855). IEEE. [CrossRef]

Maoxin Tang received the B.Eng. degree in Internet of Things Engineering from Nanjing University of Information Science and Technology, Nanjing, 210044, China, in 2021. He is currently pursuing the M.E. degree with the School of Computer Science, Nanjing University of Information Science and Technology, Nanjing, China. (Email: tangmaoxin@nuist.edu.cn)

Haonan Li received the M.S. degree in computer technology from Shandong University of Science and Technology, China, in 2024. He is currently pursuing a Ph.D. degree in Information and Communication Technologies (Computer Science) at Universiti Malaya, Malaysia. (Email: 24061728@siswa.um.edu.my)