A Privacy-Preserving Access Control Protocol for 6G Supported Intelligent UAV Networks

A Privacy-Preserving Access Control Protocol for 6G Supported Intelligent UAV Networks

Khalid Mahmood *Senior Member, IEEE*, Salman Shamshad, Mohammad Hossein Anisi, *Senior Member, IEEE*, Alessandro Brighente, Muhammad Asad Saleem, Ashok Kumar Das *Senior Member, IEEE*,

***Abstract*—Due to their autonomous operation, high mobility, and real-time communication capabilities, 6G-supported Unmanned Aerial Vehicles (6G-UAVs) (i.e., drones) are increasingly being utilized to enhance data collection and management in Intelligent Transportation Systems (ITSs). Despite their manifold benefits, 6G-supported UAV-based ITS (6G-U-ITS) faces unique security challenges beyond conventional cyber and physical threats. These include real-time authentication, impersonation attacks, physical tampering or cloning and protection against identity spoofing in highly dynamic environments. For instance, an attacker may steal a drone and use its identity to send authenticated malicious messages to the ITS, causing road accidents. Therefore, a secure authentication scheme must ensure resilience against UAV identity theft and unauthorized access while maintaining low-latency and computational efficiency to support the stringent real-time security requirements of 6G-U-ITS. Existing authentication schemes are not specifically designed to address these challenges, making it imperative to develop a lightweight and robust authentication mechanism tailored for 6G-U-ITS. Moreover, most of the existing protocols are vulnerable to physical tampering and impersonation attacks and also require high computation overhead. In this paper, to mitigate these limitations and satisfy the aforementioned requirements, we propose a secure access control protocol for 6G-U-ITS. To the best of our knowledge, this is the first security solution in the literature that can achieve security against UAVs physical attacks. Furthermore, we justify the robustness of the designed protocol against potential attacks through detailed formal and informal security assessment. Via testbed experiments, we show that our protocol achieves 20.66% and 22.82% higher efficiency on communication and computation overhead, respectively, compared to other contemporary competing protocols.**

***Index Terms*—Authentication Protocol, Security Protocol, Key Agreement, UAV, Intelligent Transport Systems**

I. Introduction

THE remarkable progresses in hardware, software, and Information and Communication Technology (ICT) has played a crucial role in the innovation and advance of Intelligent Transport Systems (ITSs). Presently, ITS is considered one of the key components of a smart city and one of the key components of Sixth-Generation (6G) networks [1]. ITSs are currently part of road infrastructures which can benefit from their capabilities. However, ITS technology is constantly evolving. The next generation of ITS (i.e., autonomous and connected vehicles) is heading toward the final stages for large-scale worldwide deployment [2]. The trials of these technologies have been commenced in various countries and significant attempts are ongoing to mandate and regulate such next-generation networks. Subsequently, the increase in the involvement of interconnected and autonomous vehicles will enable many open-ended paths for novel services and applications.

With the unprecedented propagation of Internet of Things (IoT) objects such as cameras, GPS sensors, and many more, Unmanned Aerial Vehicles (UAVs) (i.e., drones) have emerged as the latest breakthrough. UAVs can be deployed and operated to offer diverse industrial and commercial services such as surveillance, aerial views, delivery of goods, rescue, and many more [3], [4]. Moreover, the emergence of contemporary technologies like software-defined networking and the 6G mobile network has broadened UAVs' computing and networking capabilities to offer security, reliability, and ultra-low latency [5], [6]. Furthermore, drones can coordinate with other components (e.g., edge computing servers) to boost their storage and computing abilities [7], [8]. Eventually, the physical things (e.g., sensors, drones, and other objects) are controlled with the help of distinct computational techniques and algorithms to formulate a cyber-physical system for UAVs.

UAVs can be used to increase the automation of ITSs. In fact, automation in an ITS can not be achieved by merely automating the vehicles on the road. It is essential to automate also other end-to-end and road components such as rescue teams, road surveys, traffic police, and field support team. Automating such components can be realized using smart UAVs, where a road support team can be backed up or replaced by a swarm of UAVs flying around the roads. Additionally, 6G-supported UAVs offer users a reliable, seamless, and unprecedented connectivity of autonomous vehicles [9], [10]. For instance, the driver of an ambulance may want to access the traffic congestion information to avoid any delay, a traffic police officer may need to get surveillance information to control traffic, and the rescuers may want to have an aerial

Khalid Mahmood is with the Graduate School of Intelligent Data Science, National Yunlin University of Science and Technology, Douliu 64002, Taiwan (email: khalidm.research@gmail.com)

Salman Shamshad is with the Department of Software Engineering, The University of Lahore, Lahore 54590, Pakistan (e-mail: salmanshamshad01@gmail.com).

Mohammad Hossein Anisi is with School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, United Kingdom. (email: m.anisi@essex.ac.uk)

Alessandro Brighente is with the Department of Mathematics and HIT research center, Univ. of Padova, Italy. (emails: alessandro.brighente@unipd.it)

Muhammad Asad Saleem is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: masadsaleem123@gmail.com).

Ashok Kumar Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India (e-mail: iitkgp.akdas@gmail.com).

(Corresponding Author: Salman Shamshad)

view of a disaster site. To this aim, they can easily access the UAV deployed in that specific area through a ground station server.

Ensuring secure communication and authentication among UAVs in 6G-U-ITS is essential. In such environments, interconnected UAVs, roadside infrastructure, and vehicular networks exchange real-time data over public channels, making the system vulnerable to security threats such as impersonation, ephemeral secret leakage, physical tampering or cloning. Malicious entities can exploit these vulnerabilities to gain unauthorized access to control signals or traffic data, potentially leading to traffic disruptions, UAV hijacking, and safety hazards. To address this, researchers have proposed various authentication protocols aimed at enhancing privacy and security, as summarized in Table I. It presents an analysis of recent authentication protocols, highlighting their development techniques, security strengths, and vulnerabilities. Despite these advancements, many existing protocols remain susceptible to security threats, including physical tampering, impersonation, and ephemeral secret leakage, underscoring the need for a more resilient authentication mechanism. To mitigate these vulnerabilities, we propose a privacy-preserving access control protocol for 6G-supported intelligent UAV networks. The proposed protocol effectively resists potential security threats, ensuring lightweight, real-time authentication while protecting against impersonation, and ephemeral secret leakage attacks. Moreover, we utilize a physical uncloneable function (PUF) to mitigate physical tampering or cloning.

The use of PUFs in the proposed protocol is critical to addressing security challenges within 6G-U-ITS, particularly in ensuring the integrity and authenticity of UAVs and infrastructure components. In the proposed protocol, PUFs enable tamper detection for UAVs by leveraging their unique, hardware-intrinsic properties. Each UAV generates a PUF-based response, which serves as a device-specific identifier. If an attacker physically tampers with or clones a UAV, the PUF response changes, allowing the ITS infrastructure to detect unauthorized modifications immediately. This ensures that only legitimate UAVs can participate in the authentication process, preventing compromised devices from accessing the network. Furthermore, it ensures that each challenge-response pair is unique to a specific UAV. Since the PUF response cannot be replicated, an attacker cannot clone a UAV and use it to impersonate a legitimate entity, effectively mitigating identity spoofing threats. Another advantage of PUF-based authentication is its lightweight nature, ensuring that security mechanisms do not introduce significant computational or communication overhead. Given the resource constraints of UAVs, traditional cryptographic methods would be too computationally demanding. By integrating PUFs into the authentication process, the protocol enables secure and efficient UAV identification with minimal computational cost, making it highly suitable for large-scale, low-latency 6G-U-ITS deployments.

II. Related Work

Due to the fragile nature of the communication medium, security and privacy issues in 6G-U-ITS have attracted various researchers' attention. In this context, many researchers have contributed by designing authentication and access control protocols to secure communication among the involved entities in 6G-U-ITS. Nevertheless, it is worth noticing that the majority of the protocols in the existing literature fail to offer robust security solutions, as depicted in Table I. For instance, Wazid et al. [11] contributed to protecting the IoD environment using the symmetric key protocol. However, Hussain et al. [23] identified that the protocol in [11] has no resistance against the forgery of control centers, users and drone impersonation attacks based on stolen verifiers. Later, the authors of [23] came up with their solution to improve the security loopholes presented in [11]. Srinivas et al. [12] also presented a temporal credential-based access control protocol for the IoD environment. Later, Ali et al. [24] argued that the protocol in [12] does not preserve anonymity and is vulnerable to impersonation attacks based on stolen verifiers. Thereafter, the authors of [24] designed an extended protocol to strengthen the security of [12]. Unfortunately, Ali et al.'s [24] protocol later proved to be susceptible to Ephemeral Secret Leakage (ESL) attack. Another pairing-based access control scheme was presented by Zhou et al. [13]. Unfortunately, Chaudhry et al. [25] proved it insecure against forgery attacks. In 2019, the authors of [26] suggested a digital signature-based protocol for securing IoD infrastructure. Unfortunately, the protocol in [26] is unable to offer protection against physical and location threats. Zhang et al. [14] also contributed to offering the desired security of the IoD environment. Nevertheless, after an in-depth security evaluation of their protocol [14], we found that an attacker can easily launch a timestamp modification attack and masquerade attacks on drones. Besides, their protocol is proven defenseless against side-channel and anonymity violation threats. In [27], another security protocol was presented for IoD. Unfortunately, likewise to [14], the protocol in [27] has the same issues, i.e., insecurity against drone capture and anonymity violation attacks.

Recently, Bera et al. [15] in 2020 proposed a certificate-based blockchain-assisted protocol using Elliptic Curve Cryptography (ECC) for the IoD network. However, the authors in [28] illustrated that the protocol in [15] has several weaknesses, including impersonations, replay, and Man-in-the-Middle (MITM) attacks. Besides, [28] also debated that [15] does not preserve anonymity. Bera et al. [16] presented another blockchain-based protocol in the same year (i.e., 2020) with ECC primitives. Unfortunately, [16] was also proven insecure by [29] since it lacks users and drone anonymity. Furthermore, [29] argued that the protocol in [16] does not authenticate the signatures of the ground station and, ultimately, is defenseless against ground station masquerade threats. Quite recently, in 2021, Nikooghadam et al. [17] devised an ECC-based access control scheme for IoD using the symmetric key. Alzahrani et al. [19] also introduced a resource-efficient protocol for massive crowd management in IoD. Kirsal et al. [30] presented a secure framework for mobile sinks in IoD. Likewise, Tanveer et al. [20] suggested a privacy-preserving protocol for the IoD environment. In the same year, Hussain et al. [18] also proposed a user access protocol for IoD-enabled smart city surveillance systems. Unfortunately, [18] protocols

TABLE I: Summary of Existing Studies

Authors	Year	Technique	Demerits
Wazid et al. [11]	2018	∗ Symmetric key	Does not preserve drone anonymity and Defenseless against, impersonation and physically drone cloning/ tampering attacks
Srinivas et al. [12]	2019	∗ Three-factor	Does not preserves forward & backward secrecy, and drone's anonymity
Zhou et al. [13]	2019	∗ Bilinear Pairing	Defenseless against forgery and physically drone cloning/ tampering attacks
Zhang et al. [14]	2020	∗ Symmetric key	Weakness against MITH, timestamps modification, impersonation and physically drone cloning/ tampering attacks
Bera et al.-I [15]	2020	∗ ECC	Does not ensures drone anonymity and susceptible to impersonation and physically drone cloning/ tampering attacks
Bera et al.-II [16]	2020	∗ ECC	Vulnerable to stolen verifier, drone anonymity violation and physically drone cloning/ tampering attacks
Nikooghadam et al. [17]	2021	∗ ECC	Vulnerable to stolen verifier, drone anonymity violation and physically drone cloning/ tampering attacks
Hussain et al. [18]	2021	∗ Symmetric key	Susceptible to MITH, desynchronization and physically drone cloning/ tampering attacks
Alzahrani et al. [19]	2021	∗ ECC	Defenseless against stolen verifier and physically drone cloning/ tampering attacks
Tanveer et al. [20]	2021	∗ Three-Factor	Insecure against drone anonymity violation, stolen verifier and physically drone cloning/ tampering attacks
Pu et al. [21]	2022	∗ Bilinear Pairing	Prone to stolen verifier and lacks privacy
Tanveer et al. [10]	2023	∗ Three-Factor	Prone to physically drone cloning/ tampering attacks
Cui et al. [22]	2023	∗ Chaotic Map	Insecure against stolen verifier and physically drone cloning/ tampering attacks

also suffer from various concerns ranging from physical attacks (e.g., drone capture attack) to cyber-attacks (e.g., MITM, desynchronization, ESL and impersonation attacks, etc.).

Pu et al. [31] introduced a lightweight and anonymous application-aware authentication protocol for IoD, focusing on data type-aware authentication and key agreement but not addressing security challenges unique to UAV-assisted ITS environments. Similarly, Umar et al. [32] presented a physical-layer authentication approach for IoV, which enhances authentication performance through multiple attributes-based propagation scenario identification, but lacks a robust cryptographic-based authentication scheme. Miao et al. [33] proposed a UAV-assisted authentication protocol for IoV using ECC, which improves authentication efficiency but does not consider advanced security features such as resistance against machine learning-based attacks or PUF-based device authentication. Compared to these approaches, our proposed protocol is specifically designed for 6G-U-ITS, incorporating PUFs for device authentication for enhanced key agreement. Unlike previous works, our solution provides strong resistance against physical tampering or cloning and cyber threats, including impersonation and ephemeral secret leakage attacks.

A. Motivation and Contributions

In 6G-U-ITS, users are assumed to acquire sensitive real-time information directly from distinct critical infrastructures. However, due to the connection of the collected information with users' safety, it is vital to employ some access control mechanisms to consider the privacy and security guarantees. Moreover, drones are usually resource-constrained; therefore, the designed solution must be efficient enough to be implemented in practical applications. No doubt, plenty of access control protocols have been designed for 6G-U-ITS. Nevertheless, the analysis in Table I shows that no work has been done till now that can offer concrete security features to the 6G-U-ITS environment. Additionally, no solution in the existing literature can offer protection against tempering or drone capturing attacks (i.e., resistance against physical attack). Motivated by these facts, we design a security solution that can guarantee the desired security to 6G-U-ITS. The main contributions of this article are as follows:

- We devise a novel lightweight key agreement and authentication protocol for 6G-U-ITS. Our protocol utilizes efficient cryptographic primitives like the hash function and bit-wise XoR operation to provide reduced computational complexity and communication overhead compared to other available schemes.
- We exploit a physical unclonable function to provide a scheme secure against physical/ cloning attacks.
- Our protocol allows us to dynamically add a new drone in a network in case some older drone is malfunctioned/ exhausted, therefore increasing the scalability of the network.
- We formally verify the security of our protocol through the broadly accepted Real-Or-Random (ROR) oracle model, showing significant advantages over state-of-the-art protocols.
- We compare our protocol with contemporary related protocols in terms of computation and communication costs. Via testbed evaluation, we show that our protocol reduces the communication overhead by 20.66% and the computation overhead by 22.82% compared to the other relevant state of the art protocols.

The rest of the research work is organized as follows: Section III describes the essential preliminaries that are considered for developing the scheme. The proposed protocol is presented in Section IV. The detailed security analysis of our scheme is described in Section V. Section VI demonstrates the performance analysis of our protocol with various relevant protocols. In the end, the research work is concluded in Section VII, along with our future directions.

III. Preliminaries

In this section, we provide some elementary knowledge related to our presented protocol including the threat model, system model, physically unclonable function, fuzzy extractor and security requirements. Moreover, Table II lists the notation used throughout the paper.

TABLE II: Notation Guide

Notation	*Description*	*Notation*	*Description*
$\mathcal{GSS}_k$	Ground Station Server	$\mathbb{K}$	Master key of $\mathcal{GSS}_k$
$\mathcal{U}_i$	i^{th} Mobile user	$\mathcal{D}_j$	j^{th} Drone
id_i	$\mathcal{U}_i$'s Identity	id_j	$\mathcal{D}_j$'s Identity
pid_i	Pseudonym of $\mathcal{U}_i$	pid_j	Pseudonym of $\mathcal{D}_j$
bio_i	Biometric of $\mathcal{U}_i$	sid_i	Masked identity of $\mathcal{U}_i$
$Gen(.)$	Fuzzy biometric generator	$Rep(.)$	Fuzzy biometric reproduction
$h(.)$	Hash function	pid_k	Pseudonym of $\mathcal{GSS}_k$
$Enc_{\mathbb{K}}, Dec_{\mathbb{K}}$	Encryption/ Decryption algorithm	Z_p^*	Positive set of integers
$\oplus$	Bitwise XOR Operator	$\|$	Concatenation operator
$\mathbb{A}$	Adversary	n_1, n_2, n_3	Random numbers
$\stackrel{?}{=}$	Either equal to or not		

A. Threat Model

We summarize the capabilities of $\mathbb{A}$ under the broadly accepted Dolev-Yao (DY) [34], Canetti-Krawczyk (CK) [35], and extended CK (eCK) [36] threat models. The DY model grants $\mathbb{A}$ to have full control over the communication channel. In contrast, the CK model enables $\mathbb{A}$ to actively attack ongoing sessions (CK), while the eCK (eCK) possesses even stronger capabilities to compromise multiple parties or break cryptographic primitives. $\mathbb{A}$ is empowered to do any of the following:

- $\mathbb{A}$ has full control over public channels.
- $\mathbb{A}$ can eavesdrop, modify, or delete the transmitted messages.
- $\mathbb{A}$ can physically capture a drone $\mathcal{D}_j$ and can extract the information stored in it.
- $\mathbb{A}$ can be a distrustful insider or outsider but $\mathcal{GSS}_k$'s secret key $\mathbb{K}$ assumed to be protected from $\mathbb{A}$.
- $\mathbb{A}$ can mount a Key Compromise Attack (KCI) on the proposed protocol using long-term secret parameters.

B. Fuzzy Extractor

A fuzzy extractor is a cryptographic mechanism designed to derive a stable and reproducible secret from noisy biometric data. It consists of two functions:

- Fuzzy Generator $Gen(.)$: Extracts a stable secret key and a helper string from a biometric input.
- Fuzzy Reproduction $Rep(.)$: Recovers the secret key from a slightly different biometric input using the helper string.

Formally, a fuzzy extractor is defined as a tuple of algorithms:

$$(Gen, Rep)$$

where:

- $Gen : \mathcal{X} \rightarrow (\mathcal{R}, P)$ is a probabilistic function that takes a biometric sample $x \in \mathcal{X}$ from the biometric domain $\mathcal{X}$ and outputs a secret key $r \in \mathcal{R}$ along with a public helper string P:
$$(r, P) \leftarrow Gen(x)$$
- $Rep : \mathcal{X} \times \mathcal{P} \rightarrow \mathcal{R}$ is a deterministic function that takes a noisy version x' of the biometric input and the helper string P, reconstructing the secret key r if x' is sufficiently close to x:
$$Rep(x', P) = r, \quad \text{if } d(x, x') \leq \tau$$

where $d(x, x')$ is a predefined distance metric, and τ is the tolerance threshold allowing small variations in biometric data.

C. Physical Unclonable Functions

A Physical Unclonable Function (PUF) is a hardware security primitive that exploits the intrinsic physical variations of manufacturing processes to produce unique, repeatable outputs. PUFs are primarily used for device authentication and secure key generation. The definition and functionality can be encapsulated as follows:

Given a challenge C, a PUF device responds with an output R, such that:

$$R \leftarrow \text{PUF}(C)$$

Here, PUF represents the PUF instance on the device, C denotes the challenge applied to the PUF, and R is the response generated. The arrow $\leftarrow$ signifies the assignment of the response R after processing the challenge C through PUF. PUFs leverage manufacturing inconsistencies to uniquely identify devices through a set of equations that define their operational characteristics:

1) **Challenge-Response Relationship**:
$$R = f(C)$$
This equation describes the function f that maps a challenge C to a response R, specific to each PUF.
2) **Uniqueness Condition**:
$$\mathbb{P}(f_i(C) = f_j(C)) \ll 1, \quad \forall i \neq j$$
It states that the probability of two devices producing the same response to identical challenges is extremely low, underscoring their uniqueness.

D. Security Requirements

The integration of 6G in UAV networks enhances connectivity but introduces critical security and privacy challenges. A robust access control mechanism is essential to ensure secure communication, authentication, and data protection while maintaining efficiency. Below are key security requirements for a privacy-preserving access control protocol in 6G-supported UAV networks.

1) Mutual Authentication: All entities (Users, Ground Station Server and Drones) must authenticate each other to prevent unauthorized access and impersonation attacks. A secure authentication mechanism should ensure that only legitimate devices can participate in network operations, maintaining low-latency communication.

2) Data Confidentiality: Sensitive UAV data, including mission details and surveillance feeds, must be encrypted using cryptographic techniques such as homomorphic encryption (HE) and elliptic curve cryptography (ECC) to prevent unauthorized interception.

3) User Privacy Protection: The identities of UAV operators and mission data must be protected against tracking and profiling by adversaries. Using pseudonym-based authentication, an operator's identity ID should be transformed into an unlinkable pseudonym PID.

4) Integrity Protection: Data integrity must be ensured to prevent unauthorized modification of UAV commands, sensor data or mission logs.

E. System Model

The system model of our proposed protocol as shown in Fig 1 incorporates three primary entities: the Mobile User, the Ground Station Server, and Drones (6G-supported UAVs), all operating within a robust 6G communication network to ensure secure and efficient operations in Intelligent Transportation Systems (ITSs). Mobile Users are crucial to the ITS, actively participating in the system by sending and receiving authenticated data requests and safety messages. These users interact directly with UAVs and the Ground Station Server to carry out real-time operations, capitalizing on the ultra-low latency capabilities of the 6G network. The Ground Station Server

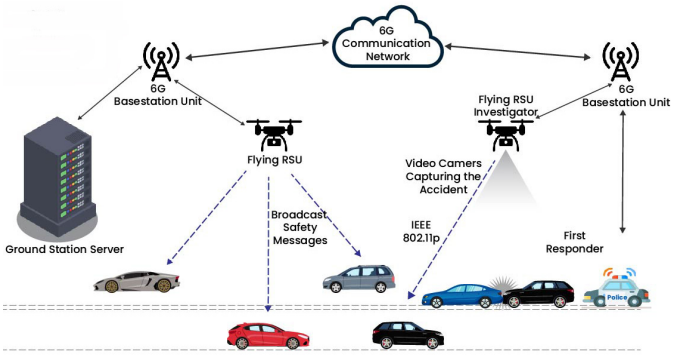

Fig. 1: Illustration of UAV-Enabled Intelligent Transportation

acts as the central command and control hub, managing UAV operations that include their deployment, authentication, and the dissemination of safety messages to Mobile Users. It maintains a secure, encrypted communication link with UAVs and ensures that all data exchanged within the network adheres to the latest security protocols to guard against identity spoofing and other cyber threats. Meanwhile, the UAVs are deployed to monitor and facilitate traffic management, accident response, and surveillance tasks within the ITS. Equipped with advanced sensors and cameras, these UAVs capture and transmit real-time data and video footage back to the Ground Station Server and Mobile Users. They utilize a dual communication mode, engaging in direct communication with the Ground Station Server via 6G base stations and establishing peer-to-peer links with each other and with Mobile Users to ensure redundant, reliable data flow, even in highly dynamic environments.

Regarding interactions, UAVs regularly authenticate themselves to the Ground Station Server using a privacy-preserving protocol that protects their identities while confirming their legitimacy. This is vital for preventing the potential misuse of UAVs by malicious entities. UAVs also directly communicate with Mobile Users, providing real-time updates and safety alerts via broadcast messages that adhere to the IEEE 802.11p standard tailored for vehicular communications. This communication strategy helps maintain a continuous flow of critical information without overloading the 6G base stations. Additionally, the Ground Station Server acts as a relay to further ensure that messages from UAVs reach Mobile Users securely and efficiently, using encryption techniques to safeguard sensitive information against potential cyber-attacks. This proposed system model exploits the capabilities of 6G technology to enhance the responsiveness and reliability of ITSs, addressing the unique security challenges posed by the high mobility and autonomous operation of UAVs.

IV. Proposed Protocol

In this section, we present our proposed protocol specifically designed for 6G-U-ITS. Our protocol mainly includes three participants that are mobile user $\mathcal{U}_i$, drone $\mathcal{D}_j$ and ground station server $\mathcal{GSS}_k$, respectively. The designed protocol allows $\mathcal{U}_i$ and $\mathcal{D}_j$ to establish a symmetric session key SK_{i-j} through $\mathcal{GSS}_k$ to securely access the information of any intended fly-zone over a public channel. We describe different phases of our designed protocol in the trailing subsections.

A. Initialization

For setting up the system, the ground station server $\mathcal{GSS}_k$ first picks pid_k as its pseudo-identity. Thereafter, $\mathcal{GSS}_k$ chooses a collision-resistance one-way hash function $h(.) : \{1,0\} \rightarrow Z_p^*$ along with its master key $\mathbb{K}$. Next, $\mathcal{GSS}_k$ picks $Gen(.)$ and $Rep(.)$ as fuzzy-biometric generator and reproduction functions, respectively. Finally, $\mathcal{GSS}_k$ publicly publishes $\{pid_k, h(.), Gen(.), Rep(.)\}$ and keeps $\mathbb{K}$ secret.

B. Pre deployment

In order to access the information from the intended fly-zone, the drones $\mathcal{D}_j$ are deployed in disjoint clusters known as fly-zones. Moreover, it is essential to register each $\mathcal{D}_j$ with the existing system. However, it is the responsibility of $\mathcal{GSS}_k$ to register each $\mathcal{D}_j$. For this purpose, $\mathcal{GSS}_k$ performs the trailing steps.

1) Initially, $\mathcal{GSS}_k$ picks id_j and pid_j as unique and pseudo identities for $\mathcal{D}_j$. $\mathcal{GSS}_k$ then computes: $x_i = h(pid_j \| \mathbb{K})$. Thereafter, $\mathcal{GSS}_k$ chooses a challenge message $\mathcal{C}_j$ and submits $\{x_j, pid_j, \mathcal{C}_j\}$ toward $\mathcal{D}_j$.
2) Upon receiving $\{x_j, pid_j, \mathcal{C}_j\}$ from $\mathcal{GSS}_k$, $\mathcal{D}_j$ uses a strong PUF function $\mathcal{PUF}_j$ embedded in its control circuit to determine the response message $\mathcal{R}_j$ corresponding the given challenge message $\mathcal{C}_j$ as follows: $\mathcal{R}_j \leftarrow \mathcal{PUF}_j(\mathcal{C}_j)$. Thereafter, $\mathcal{D}_j$ stores (k_j, pid_j) in its memory. $\mathcal{D}_j$ then forwards $\mathcal{R}_j$ to $\mathcal{GSS}_k$.
3) Whenever, $\mathcal{GSS}_k$ receives $\mathcal{R}_j$ from $\mathcal{D}_j$, $\mathcal{GSS}_k$ securely writes $(id_j, x_j, \mathcal{C}_j, \mathcal{R}_j)$ in its database corresponding to pid_j.

C. Mobile User Registration

In order to access real-time information of any particular fly-zone through $\mathcal{D}_j$, each $\mathcal{U}_i$ needs to register themselves with $\mathcal{GSS}_k$. To register $\mathcal{U}_i$, $\mathcal{GSS}_k$ performs the trailing steps:

1) Firstly, $\mathcal{U}_i$ selects unique identity id_i and password pw_i for himself. $\mathcal{U}_i$ then imprints his biometric-impression bio_i to the biometric reader and computes: $Gen(bio_i) = (\alpha_i, \beta_i)$. Thereafter, $\mathcal{U}_i$ submits $\{id_i\}$ along with a registration request toward $\mathcal{GSS}_k$.

2) On getting registration request from $\mathcal{U}_i$ along with $\{id_i\}$, $\mathcal{GSS}_k$ picks pseudo-identity pid_i for $\mathcal{U}_i$ and computes: $x_i = h(pid_i \| \mathbb{K})$. $\mathcal{GSS}_k$ generates $n_i \leftarrow Z_p^*$ and computes: $sid_i = Enc_{\mathbb{K}}(id_i \| pid_i \| n_i)$. Thereafter, $\mathcal{GSS}_k$ sends $\{x_i, pid_i, pid_j\}$ to $\mathcal{U}_i$.

3) Upon receiving $\{x_i, pid_i, pid_j\}$ from $\mathcal{GSS}_k$, $\mathcal{U}_i$ computes: $\eta = h(id_i \| pw_i \| \alpha) \oplus x_i$ and $pid_i^u = h(id_i \| pw_i) \oplus pid_i$. At the end, $\mathcal{U}_i$ stores (η, pid_i^u, pid_j) for later use.

D. Authenticated Key Establishment

Before accessing the real-time information of any particular fly-zone, $\mathcal{U}_i$ first needs to establish a session key SK_{i-j} with $\mathcal{D}_j$ via $\mathcal{GSS}_k$. A pictorial representation of our protocol's stepwise flow is illustrated in Fig. 2, with a detailed discussion as provided in the following steps:

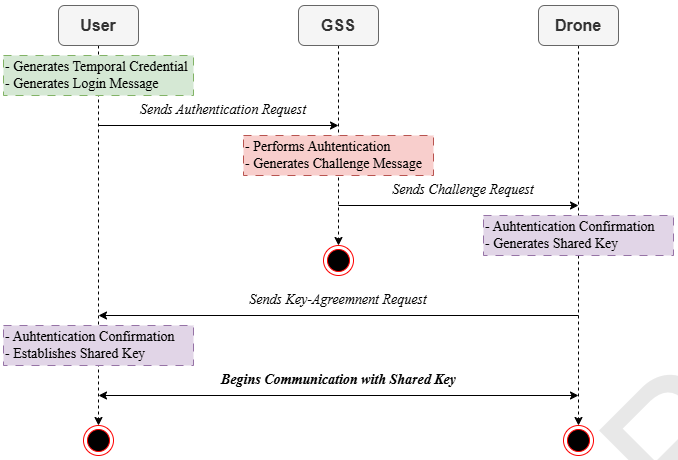

Fig. 2: Stepwise Flow of the Proposed Authentication Protocol

1) The mobile user $\mathcal{U}_i$ firstly inputs his id_i and pw_i to the interface of his mobile device. Next, $\mathcal{U}_i$ imprints his bio_i and computes: $\alpha_i = Rep(bio_i, \beta_i)$, $pid_i = pid_i^u \oplus h(id_i \| pw_i)$, $x_i = \eta_i \oplus h(id_i \| pw_i \| \alpha_i)$. Thereafter, $\mathcal{U}_i$ randomly picks $n_1 \in Z_p^*$ and further computes: $D_1 = h(pid_i \| pid_k \| x_i) \oplus n_1$, $D_2 = h(pid_i \| pid_k \| x_i \| n_1) \oplus pid_j$, and $D_3 = h(id_i \| pid_j \| pid_k \| x_i \| n_1)$. Finally, $\mathcal{U}_i$ submits $\mathbb{M}_1 \leftarrow \{sid_i, D_1, D_2, D_3\}$ to $\mathcal{GSS}_k$ as a login request message.

2) On receiving $\mathbb{M}_1$ from $\mathcal{U}_i$, $\mathcal{GSS}_k$ uses its secret key to decrypt sid_i as follows: $(id_i \| pid_i \| n_i) = Dec_{\mathbb{K}}(sid_i)$. $\mathcal{GSS}_k$ then computes $x_i' = h(pid_i \| \mathbb{K})$, $n_1' = D_1 \oplus h(pid_i' \| pid_k \| x_i')$, $pid_j' = D_2 \oplus h(pid_i' \| pid_k \| x_i' \| n_1')$, $D_3' = h(id_i' \| pid_j' \| pid_k \| x_i' \| n_1')$ and checks for $D_3' \stackrel{?}{=} D_3$ to authenticate $\mathcal{U}_i$. If the validation is successful, $\mathcal{GSS}_k$ further checks for $\{x_j, \mathcal{C}_j, \mathcal{R}_j\}$ against pid_j' from its database using $\mathbb{K}$. Thereafter, $\mathcal{GSS}_k$ generates n_2 and n_i^{new}, and computes: $sid_i^{new} = Enc_{\mathbb{K}}(id_i \| pid_i \| n_i^{new})$, $D_4 = h(pid_j' \| x_j') \oplus (n_1' \| n_2 \| \mathcal{C}_j \| sid_i^{new})$, $D_5 = h(pid_j' \| pid_k \| x_j' \| n_1') \oplus pid_i'$, and $D_6 = h(pid_i' \| pid_j' \| pid_k \| x_j' \| \mathcal{R}_j \| n_1' \| n_2)$. At the end, $\mathcal{GSS}_k$ transmits $\mathbb{M}_2 \leftarrow \{D_4, D_5, D_6\}$ toward $\mathcal{D}_j$.

3) Upon receiving $\mathbb{M}_2$ from $\mathcal{GSS}_k$, $\mathcal{D}_j$ computes $(n_1'' \| n_2' \| \mathcal{C}_j \| sid_i^{new}) = D_4 \oplus h(pid_j \| x_j)$, $pid_i'' = D_5 \oplus h(pid_j \| pid_k \| x_j \| n_1'')$, $\mathcal{R}_j \leftarrow \mathcal{PUF}_j(\mathcal{C}_j)$ and $D_6' = h(pid_i'' \| pid_j \| pid_k \| x_j \| \mathcal{R}_j \| n_1'' \| n_2')$. Next, $\mathcal{D}_j$ checks if $D_6' \stackrel{?}{=} D_6$ to verify the legitimacy of $\mathcal{GSS}_k$. In case of unsuccessful authentication, $\mathcal{D}_j$ immediately terminates the session. Elseways, $\mathcal{D}_j$ chooses $n_3 \in Z_p^*$ and calculates: $D_7 = h(pid_j \| pid_i'' \| n_1'') \oplus (n_2' \| n_3 \| sid_i^{new})$, $SK_{i-j} = h(pid_i'' \| pid_j \| pid_k \| h(n_1'' \| n_2' \| n_3))$ and $D_8 = h(pid_i'' \| pid_j \| pid_k \| n_2' \| n_3)$. Finally, $\mathcal{D}_j$ transmits $\mathbb{M}_3 \leftarrow \{D_7, D_8\}$ to $\mathcal{U}_i$.

4) On getting $\mathbb{M}_3$ from $\mathcal{D}_j$, $\mathcal{U}_i$ computes: $(n_2' \| n_3' \| sid_i^{new}) = D_7 \oplus h(pid_j \| pid_i \| n_1)$, $D_8' = h(pid_i \| pid_j \| pid_k \| n_2' \| n_3')$ and checks for $D_8' \stackrel{?}{=} D_8$. If this check returns true, then $\mathcal{U}_i$ perceives that $\mathcal{D}_j$ is legal and establishes a session key as $SK_{i-j} = h(pid_i'' \| pid_j \| pid_k \| h(n_1 \| n_2' \| n_3'))$. Finally, $\mathcal{D}_j$ updates sid_i^{new} in its mobile device.

After the successful establishment of the common session key SK_{i-j}, $\mathcal{U}_i$ can securely access the information from the intended $\mathcal{D}_j$. The overall key establishment phase is also summarized in Fig. 3. Moreover, we have presented the pseudocode of the authentication phase in Algorithm 1 and Algorithm 2.

E. Dynamic Drone Addition

Our designed protocol allows a new drone $\mathcal{D}_j^{new}$ to be deployed in the existing IoD network at any time. For this purpose, $\mathcal{GSS}_k$ executes the trailing steps in an offline manner.

1) Initially, $\mathcal{GSS}_k$ picks id_j^{new} and pid_j^{new} as unique and pseudo identities for $\mathcal{D}_j^{new}$. $\mathcal{GSS}_k$ then computes: $x_i^{new} = h(pid_j^{new} \| \mathbb{K})$. Thereafter, $\mathcal{GSS}_k$ chooses a challenge message $\mathcal{C}_j^{new}$ and submits $\{x_j^{new}, pid_j^{new}, \mathcal{C}_j^{new}\}$ toward $\mathcal{D}_j^{new}$.

2) Upon receiving $\{x_j^{new}, pid_j^{new}, \mathcal{C}_j^{new}\}$ from $\mathcal{GSS}_k$, $\mathcal{D}_j^{new}$ uses the PUF function $\mathcal{PUF}_j$ embedded in its control circuit to determine the response message $\mathcal{R}_j^{new}$ corresponding to the given challenge message $\mathcal{C}_j^{new}$ as follows: $\mathcal{R}_j^{new} \leftarrow \mathcal{PUF}_j(\mathcal{C}_j^{new})$. Thereafter, $\mathcal{D}_j^{new}$ stores (k_j^{new}, pid_j^{new}) in its memory. $\mathcal{D}_j^{new}$ then forwards $\mathcal{R}_j^{new}$ to $\mathcal{GSS}_k$.

3) Whenever, $\mathcal{GSS}_k$ receives $\mathcal{R}_j^{new}$ from $\mathcal{D}_j^{new}$, $\mathcal{GSS}_k$ writes $(id_j^{new}, x_j^{new}, \mathcal{C}_j^{new}, \mathcal{R}_j^{new})$ in its database corresponding to pid_j^{new} and encrypts with $\mathbb{K}$. At the end, $\mathcal{D}_j^{new}$ is deployed in jth cluster and its pseudonym pid_j^{new} is shared among all registered users for later communication.

F. Limitations of the Proposed Protocol

The current design supports an individual authentication scenario, where each $\mathcal{U}i$ and $\mathcal{D}j$ establish a symmetric session key SK_{i-j} through the $\mathcal{GSS}_k$ to access fly-zone information over a public channel securely. While this approach provides secure and private communication between individual participants, it may not be efficient and cost-effective in scenarios involving a large number of authentication requests. In other words, the protocol does not accommodate aggregate/batch authentication such as liteGAP [37], which could improve

Fig. 3: Authenticated Key Establishment

scalability by reducing repeated handshake request overhead in dense network environments. As part of future work, we intend to extend the current scheme to support a lightweight aggregate authentication model, allowing multiple $\mathcal{U}i$ and $\mathcal{D}j$ entities to be authenticated simultaneously through fewer interactions with $\mathcal{GSS}_k$. This enhancement would make the protocol more suitable for large-scale 6G-U-ITS deployments.

V. Security Analysis

In this section, we provide an in-depth formal and informal security analysis of our designed protocol. Firstly, in Section V-A, we measure the semantic session key security of the designed protocol with the help of a well-known Real-or-Random (RoR) formal model. Section V-B presents the formal security verification of the proposed protocol using scyther. Then, in Section V-C, we prove the resilience of the designed protocol against distinct security attacks against the broadly accepted threat model discussed in Section III-A.

A. Formal Security Evaluation

In this subsection, we present the formal security proof to test the session key security of our designed protocol under the globally accepted Real-Or-Random (ROR) model [38]. The ROR model is a fundamental cryptographic testing framework where an adversary is challenged to distinguish between a 'real' scenario (e.g., using the actual encryption of a message) and a 'random' scenario (e.g., using a random output). This model is crucial for assessing the indistinguishability properties of cryptographic protocols, especially in the context of security against chosen-ciphertext attacks. Moreover, in our protocol, we employ the ROR model to evaluate the indistinguishability of encrypted messages. The model is particularly suited to our analysis as it directly tests the protocol's ability to protect against adversaries who might exploit information from ciphertexts.

As per the RoR model, an attacker $\mathbb{A}$ interacts with the pth executing participants of the protocol. Following the designed protocol, $\mathbb{A}$ interacts with nth executing participants $\amalg^n$ (i.e., $\mathcal{U}_i$, $\mathcal{GSS}_k$ or $\mathcal{D}_j$) of the designed protocol. Therefore, we consider $\amalg_{\mathcal{U}_i}^{n1}$, $\amalg_{\mathcal{GSS}_k}^{n1}$ and $\amalg_{\mathcal{D}_j}^{n1}$ as the n_1^{th}, n_2^{th} and n_3^{th} participants

Algorithm 1 Authentication Initialization Algorithm

/* SendMessage(src, des, msg): source src sends message msg to destination des */
Function UserAuthRequest(id_i, pw_i, bio_i):
  $pid_i \leftarrow pid^u i \oplus h(idi \| pw_i)$;
  $x_i \leftarrow \eta_i \oplus h(id_i \| pw_i \| \alpha_i)$;
  $n_1 \leftarrow$ RandNum(Z_p^*);
  $D_1 \leftarrow h(pid_i \| pid_k \| x_i) \oplus n_1$;
  $D_2 \leftarrow h(pid_i \| pid_k \| x_i \| n_1) \oplus pid_j$;
  $D_3 \leftarrow h(id_i \| pid_j \| pid_k \| x_i \| n_1)$;
  SendMessage($\mathcal{U}i$, $\mathcal{GSS}k$, $\mathbb{M}_1$);
Function GroundReceiveAuth($\mathbb{M}_1$):
  $(id_i \| pid_i \| n_i) = Dec_{\mathbb{K}}(sid_i)$;
  $x_i' = h(pid_i \| \mathbb{K})$;
  $n_1' = D_1 \oplus h(pid_i' \| pid_k \| x_i')$;
  $pid_j' = D_2 \oplus h(pid_i' \| pid_k \| x_i' \| n_1')$;
  $D_3' = h(id_i' \| pid_j' \| pid_k \| x_i' \| n_1')$;
if $D_3' \neq D_3$ **then**
  reject
else
  n_2, $n^{new} i \leftarrow$ RandNum($Z^* p$);
  $sid^{new} i = Enc\mathbb{K}(id_i \| pid_i \| n_i^{new})$;
  $D_4 = h(pid_j' \| x_j') \oplus (n_1' \| n_2 \| \mathcal{C}j \| sid^{new} i)$;
  $D_5 = h(pid_j' \| pid_k \| x_j' \| n_1') \oplus pid_i'$;
  $D_6 = h(pid_i' \| pid_j' \| pid_k \| x_j' \| \mathcal{R}j \| n1' \| n_2)$;
    SendMessage($\mathcal{GSS}k$, $\mathcal{D}j$, $\mathbb{M}_2$);
end if

Algorithm 2 Authentication Completion Algorithm

/* SendMessage(src, des, msg): source src sends message msg to destination des */
Function DroneCompleteAuth($\mathbb{M}_2$):
  $(n_1'' \| n_2' \| \mathcal{C}j \| sid^{new} i) = D_4 \oplus h(pid_j \| x_j)$;
  $pid_i'' = D_5 \oplus h(pid_j \| pid_k \| x_j \| n_1'')$;
  $\mathcal{R}j \leftarrow \mathcal{PUF}j(\mathcal{C}_j)$;
  $D_6' = h(pid_i'' \| pid_j \| pid_k \| x_j \| \mathcal{R}j \| n1'' \| n_2')$;
if $D_6' \neq D_6$ **then**
  reject
else
  $n_3 \leftarrow$ RandNum(Z_p^*);
  $D_7 = h(pid_j \| pid_i'' \| n_1'') \oplus (n_2' \| n_3 \| sid_i^{new})$;
  $SK_{i-j} = h(pid_i'' \| pid_j \| pid_k \| h(n_1'' \| n_2' \| n_3))$;
  $D_8 = h(pid_i'' \| pid_j \| pid_k \| n_2' \| n_3)$;
  SendMessage($\mathcal{D}j$, $\mathcal{U}i$, $\mathbb{M}_3$);
end if
Function UserCompleteAuth($\mathbb{M}_3$):
  $D_8' = h(pid_i \| pid_j \| pid_k \| n_2' \| n_3')$ Checks if $D_8' \stackrel{?}{=} D_8$;
if $D_8' \neq D_8$ **then**
  reject
else
  $SK_{i-j} = h(pid_i'' \| pid_j \| pid_k \| h(n_1 \| n'2 \| n3'))$;
  Updates(sid_i^{new})
end if

for $\mathcal{U}_i$, $\mathcal{GSS}_k$ and $\mathcal{D}_j$, respectively. The RoR model employs distinct queries such as $Reveal$, $CorruptDevice$, $Execute$, $Send$, and $Test$ to execute real attack scenarios as follows.

- $Reveal(\amalg^n)$: Under this query, $\mathbb{A}$ can disclose SK_{i-j} between $\amalg^n$ and its associated partner.
- $CorruptDevice(\amalg^n)$: Following this query, $\mathbb{A}$ can model an active attack to retain secret credentials.
- $Execute(\amalg_{\mathcal{U}_i}^{n1}, \amalg_{\mathcal{GSS}_k}^{n1}, \amalg_{\mathcal{D}_j}^{n1})$: $\mathbb{A}$ can model this query to eavesdrop the messages transmitted among $\mathcal{U}_i$, $\mathcal{GSS}_k$ and $\mathcal{D}_j$.
- $Send(\amalg^n, \mathbb{M})$: $\mathbb{A}$ can execute this query to model an active attack so that he can transmit message $\mathbb{M}$ to $\amalg^n$.
- $Test(\amalg^n)$: Simulating this query, $\mathbb{A}$ can request $\amalg^n$ to check the derived session key SK_{i-j} through a probabilistic output based on a hidden bit or unbiased flipped coin τ.

Moreover, we use protected ideal PUF $\mathcal{PUF}_j$ and a pseudo-random one-way hash function $h(.)$ as random oracles.

Theorem 1: Let $\mathbb{A}$ symbolizes a polynomial-time attacker running over time ti against our designed protocol $\mathbb{P}$. Thus, the advantage of $\mathbb{A}$ in cracking the semantic security to attain the session key of our designed protocol is given by:

$$\mathcal{ADV}_{\mathbb{A}}^{\mathbb{P}}(ti) \leq \frac{q_{hash}^2}{|\mathbb{H}|} + \frac{q_p^2}{|\mathcal{PUF}|} + \frac{2q_{send}}{|\mathbb{D}|},$$

where q_{hash}^2, q_{send} and q_p^2 symbolize the number of hash, send and PUF queries, respectively. Moreover, $\mathbb{H}$, $\mathbb{D}$ and $\mathcal{PUF}$ represent the lengths of hash output, output of algorithm solving a specific problem, and output of PUF function simulating by $\mathbb{A}$, respectively.

Proof: Let $\mathbb{G}_s$ be the sequence of games where $s = 1, 2, 3, 4, 5$. Let $SUCC_{\mathbb{A}}^{\mathbb{G}_s}$ be the event that denotes the advantage of $\mathbb{A}$ to estimate τ. Thus, the advantage of $\mathbb{A}$ on winning the game can be approximated as $\mathcal{ADV}_{\mathbb{A},\mathbb{G}_s}^{\mathbb{P}} = \mathcal{PRO}[SUCC_{\mathbb{A}}^{\mathbb{G}_s}]$. The proof is identical to the proof given in [12], [39] consisting of a series of games, where it starts from a real attack $\mathbb{G}_1$ over the designed protocol $\mathbb{P}$ and ends with the game having 0 advantage. Also, we can restrain the variation in $\mathbb{A}$'s advantage among any two successive games. For every $\mathbb{G}_s$, we denote an event $\mathcal{ADV}_{\mathbb{A},\mathbb{G}_s}^{\mathbb{P}}$ against the condition where $\mathbb{A}$ rightly estimates τ produced by the $Test(.)$ queries.

• $\mathbb{G}_1$: $\mathbb{A}$ can model this game to execute a real attack against our designed protocol $\mathbb{P}$. As τ was picked arbitrarily at the begging of $\mathbb{G}_1$, thus, the trailing is obtained:

$$\mathcal{ADV}_{\mathbb{A}}^{\mathbb{P}} = |2\mathcal{ADV}_{\mathbb{A},\mathbb{G}_1}^{\mathbb{P}} - 1|. \quad (1)$$

• $\mathbb{G}_2$: Following this game, $\mathbb{A}$ can mount an eavesdropping attack to intercept all the messages $\mathbb{M}_1 \leftarrow \{sid_i, D_1, D_2, D_3\}$, $\mathbb{M}_2 \leftarrow \{D_4, D_5, D_6\}$ and $\mathbb{M}_3 \leftarrow \{D_7, D_8\}$ which are transmitted over the public channel during the key establishment phase of the designed protocol by executing $Execute(.)$ query. $\mathbb{A}$ models $Test(.)$ and $Reveal(.)$ queries at the end of this game to verify whether the disclosed session key SK_{i-j} is the random or real key. The session key SK_{i-j} established between $\mathcal{U}_i$ and $\mathcal{D}_j$ is $SK_{i-j} = h(pid_i \| pid_j \| pid_k \| h(n_1 \| n_2 \| n_3))$. It is to be noted that the construction of SK_{i-j} includes both long-term pid_k, pid_i as

well as ephemeral secrets n_1, n_2 and n_3 which is unavailable to $\mathbb{A}$. Subsequently, solely eavesdropping the communicated messages $\mathbb{M}_1$, $\mathbb{M}_2$ and $\mathbb{M}_3$ will not increase the chances of $\mathbb{A}$ on winning the game $\mathbb{G}_2$. Thus, $\mathbb{G}_1$ and $\mathbb{G}_2$ are indistinguishable, and it is clear that.

$$\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_2} = \mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_1}. \tag{2}$$

• $\mathbb{G}_3$: This game involves $\mathbb{H}$ and $Send(.)$ queries to simulate a real attack. Following the transmitted messages $\mathbb{M}_1$, $\mathbb{M}_2$ and $\mathbb{M}_3$, each $\mathcal{U}_i$ are protected from the collision-resistant $h(.)$. As all $\mathcal{U}_i$ are assigned secret credentials, identities and random nonce; thus, there will be no collision between $\mathbb{H}$ and $Send(.)$ queries when simulated by $\mathbb{A}$. Here, both $\mathbb{G}_2$ and $\mathbb{G}_3$ are identical except the inclusion of $\mathbb{H}$ and $Send(.)$ quires in $\mathbb{G}_3$. Thus, from the birthday paradox of the hash function, it follows that.

$$\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_2} - \mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_3} \leq \frac{q^2_{hash}}{2|\mathbb{H}|}. \tag{3}$$

• $\mathbb{G}_4$: The difference between $\mathbb{G}_3$ and $\mathbb{G}_4$ is that $\mathbb{G}_4$ includes the modeling of $\mathcal{PUF}$ and $Send(.)$ queries. Following the properties of an ideal PUF function, it is assumed that $\mathcal{PUF}_j$ employed at $\mathcal{D}_j$ is secure and as in $\mathbb{G}_3$, we can write as:

$$\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_3} - \mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_4} \leq \frac{q^2_p}{2|\mathcal{PUF}|}. \tag{4}$$

• $\mathbb{G}_5$: This is the final game, which includes the simulation of $CorruptDevice(.)$ query. Here, $\mathbb{A}$ can model $CorruptDevice(.)$ query to obtain the credentials $\{\eta, pid^u_i, pid_j\}$ from $\mathcal{U}_i$'s mobile device. However, it is worth noticing that the mobile device of $\mathcal{U}_i$ does not maintain any secret information (i.e., password or other secret information). Consequently, the attempt of $\mathbb{A}$ to obtain secret information through $CorruptDevice(.)$ remains useless and $\mathbb{A}$ can never get the advantage of it. Here, the algorithm $\mathbb{D}$ can be determined when resolving $CorruptDevice(.)$ by running $\mathbb{A}$ against the designed protocol. Thus, $\mathbb{G}_4$ and $\mathbb{G}_5$ are identical and we can write:

$$\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_4} - \mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_5} \leq \frac{q_{send}}{|\mathbb{D}|}. \tag{5}$$

When all the games (i.e., 1, 2, ..., 5) are modeled, then $\mathbb{A}$ makes an attempt to estimate τ to win the game through simulating $Test(.)$ query. Thus,

$$\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_5} = \frac{1}{2}. \tag{6}$$

Combining (1), (2) and (5), we obtain:

$$\begin{aligned}\tfrac{1}{2}\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A}} &= |\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_1} - \tfrac{1}{2}| \\ &= |\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_2} - \tfrac{1}{2}| \\ &= |\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_2} - \mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_4}|.\end{aligned}$$

Following the triangular inequality with (4), (5), and (6), we obtain:

$$\begin{aligned}\tfrac{1}{2}\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A}} &= |\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_2} - \mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_4}| \\ &\leq |\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_2} - \mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_3}| + |\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_3} - \mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_4}| \\ &\quad + |\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_4} - \mathcal{ADV}^{\mathbb{P}}_{\mathbb{A},\mathbb{G}_5}| \\ &\leq \frac{q^2_{hash}}{2|\mathbb{H}|} + \frac{q^2_p}{2|\mathcal{PUF}|} + \frac{q_{send}}{|\mathbb{D}|}.\end{aligned} \tag{7}$$

Finally, multiplying either side of (7) by 2, we obtain the required outcome:

$$\mathcal{ADV}^{\mathbb{P}}_{\mathbb{A}}(ti) \leq \frac{q^2_{hash}}{|\mathbb{H}|} + \frac{q^2_p}{|\mathcal{PUF}|} + \frac{2q_{send}}{|\mathbb{D}|}.$$

B. Security Verification using Scyther

Apart from the formal security evaluation based on the RoR model, the proposed protocol undergoes formal security verification using the Scyther. The protocol is specified in the Security Protocol Description Language (SPDL) and tested within Scyther to assess its robustness against established security threats, as outlined in [40]. The verification process assumes that the confidential data of participating entities remains secure against $\mathcal{A}$, ensuring that any vulnerabilities relying on this premise remain undetected. In the SPDL-based modeling, entities such as $\mathcal{U}_i$, $\mathcal{GSS}_k$, and $\mathcal{D}_j$ are defined within distinct roles, each incorporating the necessary authentication operations. These role-specific processes include timestamps and nonces, declared using the t and n statements. To streamline the protocol structure, macros are employed, while message transmission is articulated through $send$ and get events. Furthermore, security attributes are validated using claim events, which define key security assertions such as $Niagree$ and $Weakagree$. Scyther autonomously evaluates these security claims and generates verification results. The results, illustrated in Figure 4, affirm the protocol's security resilience.

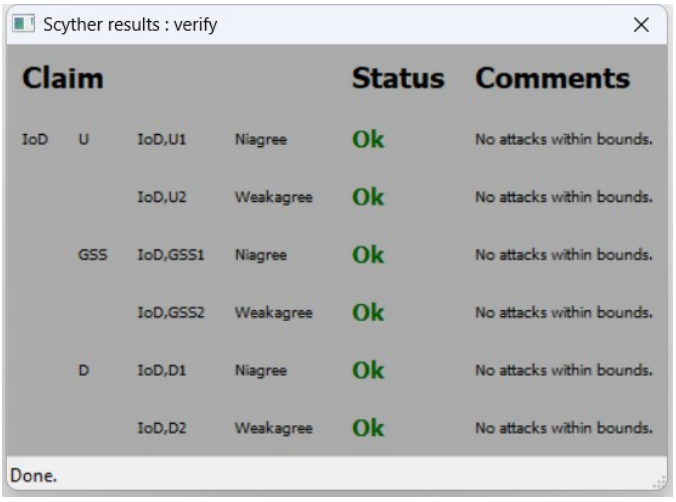

Fig. 4: Scyther Simulation Results

C. Informal Security Evaluation

The informal security discussion about the designed protocol is given as follows.

1) Authentication: Our designed protocol strictly ensures authentication where each entity firstly authenticates the other entity to verify whether it is communicating with legal entities or not. In case of unsuccessful authentication, the session is immediately terminated. Whereas, the session key is established only after successful authentication. The authentication of each entity is discussed as follows.

- $\mathcal{U}_i \rightarrow \mathcal{GSS}_k : \mathbb{M}_1 \leftarrow \{sid_i, D_1, D_2, D_3\}$: Here, $\mathcal{GSS}_k$ checks $D_3' \stackrel{?}{=} D_3$ to authenticate $\mathcal{U}_i$.
- $\mathcal{GSS}_k \rightarrow \mathcal{D}_j : \mathbb{M}_2 \leftarrow \{D_4, D_5, D_6\}$: Here, $\mathcal{D}_j$ checks $D_6' \stackrel{?}{=} D_6$ to authenticate $\mathcal{GSS}_k$.
- $\mathcal{D}_j \rightarrow \mathcal{U}_i : \mathbb{M}_3 \leftarrow \{D_7, D_8\}$: Here, $\mathcal{U}_i$ checks $D_8' \stackrel{?}{=} D_8$ to authenticate $\mathcal{D}_j$.

The above-discussed statements show how each entity first authenticates the other entity before further processing. Once the authentication is passed by all entities, then both $\mathcal{U}_i$ and $\mathcal{D}_j$ agree on the common session key $SK_{i-j} = h(pid_i \| pid_j \| pid_k \| h(n_1 \| n_2 \| n_3))$.

2) Anonymity and Privacy: During the key establishment phase of our designed protocol, the drone $\mathcal{D}_j$ communicates with other entities using its temporary credentials. In this context, the real identity id_j of $\mathcal{D}_j$ is not exchanged over a public communication channel so that an adversary $\mathbb{A}$ can never identify who is the sender or receiver. Consequently, our protocol achieves anonymity for $\mathcal{D}_j$. Moreover, the parameters involved in the public communicated messages $\mathbb{M}_1 \leftarrow \{sid_i, D_1, D_2, D_3\}$, $\mathbb{M}_2 \leftarrow \{D_4, D_5, D_6\}$ and $\mathbb{M}_3 \leftarrow \{D_7, D_8\}$ are purely dynamic in nature due to the involvement of session specific random nonce in the calculation of each parameter. Therefore, these parameters are distinct for every session and by analyzing the communicated messages of two distinct sessions, $\mathbb{A}$ can never trace whether they are initiated by the same entity. Subsequently, the designed protocol also provides untraceability property to $\mathcal{D}_j$.

3) Drone Capturing/ Physical Attack Resistance: In an IoD environment, the drones are flown in a hostile environment where the probability of hijacking or physically capturing a drone is always high. Moreover, it is quite possible that $\mathbb{A}$ may shoot down $\mathcal{D}_j$ to physically capture it. Subsequently, $\mathbb{A}$ may try to physically tamper $\mathcal{D}_j$ to obtain secret credentials stored in it so that he can share session keys using those credentials and can mislead $\mathcal{U}_i$ by sending false information. However, it is worth noticing that in our designed protocol, each $\mathcal{D}_j$ is equipped with PUF and such an attempt of tampering $\mathcal{D}_j$'s hardware will change the behavior of PUF. Consequently, PUF will never produce the right response message $\mathcal{R}_j$ [41]. Moreover, the calculation of $D_6' = h(pid_i'' \| pid_j \| pid_k \| x_j \| \mathcal{R}_j \| n_1'' \| n_2')$ always requires the real value of $\mathcal{R}_j$ and with tampered hardware, $\mathbb{A}$ can never pass the check $D_6' \stackrel{?}{=} D_6$. In a nutshell, it provides layer protection against drone capturing/ physical attacks.

4) Drone Impersonation Attack Resistance: To successfully launch this attack, $\mathbb{A}$ first needs to generate $\mathbb{M}_3 \leftarrow \{D_7, D_8\}$. For this purpose, $\mathbb{A}$ can generate n_3 on behalf of $\mathcal{D}_j$ to compute D_7 and D_8. However, it is worth noticing that $\mathbb{A}$ still needs x_j to determine $(n_1'' \| n_2' \| \mathcal{C}_j)$ which is necessary to compute D_7 and D_8. Since $\mathbb{A}$ has no access to x_j and can never construct the real output of $\mathcal{PUF}_j$. Thus, $\mathbb{A}$ is not able to compute $\mathbb{M}_3$ on $\mathcal{D}_j$'s behalf. As a result, our designed protocol offers resilience against $\mathcal{D}_j$ impersonation threat.

5) Mobile User Impersonation Resistance: Suppose $\mathbb{A}$ wants to masquerade a valid $\mathcal{U}_i$ in order to initiate a login request message $\mathbb{M}_1 \leftarrow \{sid_i, D_1, D_2, D_3\}$ to $\mathcal{GSS}_k$. In this context, $\mathbb{A}$ can generate n_1 to compute $\mathbb{M}_1$. However, it is to be noted that the computation of $D_1 = h(pid_i \| pid_k \| x_i) \oplus n_1$, $D_2 = h(pid_i \| pid_k \| x_i \| n_1) \oplus pid_j$ and $D_3 = h(id_i \| pid_j \| pid_k \| x_i \| n_1)$ requires $\mathcal{U}_i$'s secret credentials (i.e., id_i, pw_i and bio_i). Since these secret information are solely available to $\mathcal{U}_i$ and there is no clue from which $\mathbb{A}$ can get advantage to obtain them. $\mathbb{A}$ can never generate real $\mathbb{M}_1$ and it is clear that $\mathbb{A}$ can not impersonate $\mathcal{U}_i$.

6) Ground Station Server Impersonation Resistance: In order to launch this attack, $\mathbb{A}$ can make believe to registered entities that they are communicating with legal $\mathcal{GSS}_k$. Moreover, $\mathbb{A}$ may attempt to reproduce intercepted or tampered messages to prove their authenticity. In this context, $\mathbb{A}$ requires to generate $\mathbb{M}_2 \leftarrow \{D_4, D_5, D_6\}$. However, without having the secret key $\mathbb{K}$ of $\mathcal{GSS}_k$, $\mathbb{A}$ can neither access the values from the database (i.e., encrypted with $\mathbb{K}$) nor generate D_4, D_5 and D_6. Since $\mathbb{A}$ is incapable of computing the desired values of message $\mathbb{M}_2$ as a valid $\mathcal{GSS}_k$. Hence, the designed protocol resists $\mathcal{GSS}_k$ impersonation attack.

7) Preserves Perfect Forward and Backward Secrecy: The session key SK_{i-j} in our protocol is computed as $SK_{i-j} = h(pid_i \| pid_j \| pid_k \| h(n_1 \| n_2' \| n_3'))$. It is to be noted that SK_{i-j} is constructed based on short-term (i.e., random nonce) and long-term (i.e., pseudo identities) secrets, respectively. In addition, these credentials are unavailable to $\mathbb{A}$. Therefore, it is impracticable for $\mathbb{A}$ to construct SK_{i-j}. Moreover, the nature of session key SK_{i-j} for each session is dynamic due to the involvement of random nonce. Therefore, it is infeasible for $\mathbb{A}$ to estimate upcoming or previous session keys even if succeeded in compromising the SK_{i-j} of any present session. In a nutshell, the designed protocol achieves forward and backward secrecy.

8) KCI Attack Resilience: Under the eCK's adversarial capabilities, $\mathbb{A}$ can impersonate $\mathcal{U}_i$ and $\mathcal{D}_j$ using their long-term secrets $\{\eta, pid_i^u, pid_j\}$, $\{id_j, pid_j\}$, respectively. However, our protocol defends this vulnerability by using $\mathcal{U}_i$'s biometric, $\mathcal{D}_j$'s PUFs information to generate valid request messages. These values are not stored anywhere and are unique to each entity. Thus, $\mathbb{A}$ can not execute KCI on our protocol.

VI. Performance and Security Comparison

This section summarizes the performance analysis of our designed protocol with contemporary State-of-the-Art (SoTA) protocols Srinivas et al. [12] (SoTA1), Tanveer et al. [20] (SoTA2), Hussain et al. [23] (SoTA3), Ali et al. [24] (SoTA4), Azeem et al. [42] (SoTA5), Jan et al. [43] (SoTA6). To ensure a fair and relevant comparative analysis, we have selected benchmark protocols that are specifically designed for the network model outlined in this article. These protocols address key security aspects of UAV networks, including authentication and key management. Additionally, we apply

a further filter to prioritize the most recent protocols, ensuring the analysis reflects the latest advancements in UAV network security. By including these protocols, we aim to evaluate our proposed protocol against existing approaches and highlight its advantages in terms of robustness and efficiency. The comparative analysis ensures a thorough assessment of security performance, making our findings more comprehensive and reliable.

A. Testbed Environment

The IoD environment basically comprises three participants, including $\mathcal{U}_i$, $\mathcal{GSS}_k$ and $\mathcal{D}_j$. In order to determine the real execution time, we simulate the cryptographic primitives for $\mathcal{U}_i$, $\mathcal{GSS}_k$ and $\mathcal{D}_j$ on real-world devices such as a mobile device, desktop system, and Arduino, respectively. The specification of each device is displayed in Table III, whereas the notations and average execution time in milliseconds (ms) of each cryptographic primitive are summarized in Table IV. These simulations provide a realistic assessment of computational cost, enabling a precise evaluation of the efficiency and feasibility of the proposed protocol. The benchmark protocols [12], [20], [23], [24], [42], [43] have been implemented on the same devices to ensure an unbiased performance evaluation, providing a fair comparison with the proposed protocol.

TABLE III: Implementation Environment

Attribute	Desktop System	Mobile	Arduino
System	Intel Core i7	ViVo S5	Microcontroller:ATmega328
Platform	Ubuntu OS	Android OS	–
RAM	16 GB	8 GB	SRAM: 1 KB
Processing Power	3.9 GHZ	2.3 GHZ	16 MHz
Language	Python	Python	Pyhton
Library	PyCryptodome	PyCryptodome	PyCryptodome

TABLE IV: Execution Time of Cryptographic Primitives

Cryptographic Primitives	Notation	Execution Time		
		Arduino [ms]	MD [ms]	DS [ms]
Symmetric Enc/ Decryption	$\mathbb{T}_{E/C}$	0.796	0.541	0.0019
Point Multiplication	$\mathbb{T}_{PM}$	0.938	0.642	0.0028
One-way Hash Function	$\mathbb{T}_{H}$	1.812	0.883	0.0039
PUF	$\mathbb{T}_{PUF}$	0.510	0.409	0.0021
Fuzzy Extractor	$\mathbb{T}_{FE}$	0.311	0.215	0.0010

Note: DS=Desktop System, MD=Mobile Device

B. Computation Cost Evaluation

In our designed protocol, we employed nine hash functions E_h at $\mathcal{U}_i$'s side. Therefore, the execution time on $\mathcal{U}_i$ side is $(9\times 0.883) \approx 7.947$ ms. In contrast, seven hash functions and two encryption/ decryption have been performed on $\mathcal{GSS}_k$ side. Consequently, $\mathcal{GSS}_k$ requires $(7 \times 0.0039) + (2 \times 0.0019) \approx 0.0311$ ms to complete the authentication phase. Likewise, the computation cost of $\mathcal{D}_j$ side is $(6 \times 1.812) \approx 10.872$ ms due to the use of seven hash functions. Eventually, the designed protocol overall requires $(7.947{+}0.0311{+}10.872) \approx 18.86$ ms to complete the authentication phase. The computation cost of contemporary competing protocols [12], [20], [23], [24], [42], [43] is also determined in the same way and summarized in Table V.

C. Analysis of Communication Cost

The communication cost is computed by aggregating the number of bits transmitted in each message from different participants during the authenticated key establishment phase. For the analysis, we consider the size of distinct parameters, such as hash output, elliptic curve point, timestamp, identity, random number and symmetric key cryptographic algorithm (AES with 128-bit key size and 256-bit block cipher) as 256, 160, 160, 160, 160, 128 bits, respectively. Using these values, the number of messages transferred from/ to each entity is deduced to estimate their associated communication cost. For instance, the messages $\mathbb{M}_1 \leftarrow \{sid_i, D_1, D_2, D_3\}$, $\mathbb{M}_2 \leftarrow \{D_4, D_5, D_6\}$ and $\mathbb{M}_3 \leftarrow \{D_7, D_8\}$ are transmitted during authenticated key establishment phase. In accordance with the assumed values, the total number of bits incurred by each message is as follows: $\mathbb{M}_1 \leftarrow \{sid_i, D_1, D_2, D_3\} : 128 + 256 + 256 + 256 = 896$, $\mathbb{M}_2 \leftarrow \{D_4, D_5, D_6\} : 256 + 256 + 256 = 768$ and $\mathbb{M}_3 \leftarrow \{D_7, D_8\} : 256 + 256 = 512$, respectively. Thus, the accumulative number of bits incurred by the designed protocol is $896 + 768 + 512 = 2176$ in bits. Likewise, the communication cost of competing protocols [12], [20], [23], [24], [42], [43] is estimated in the same way and summarized in Table V.

D. Energy Consumption Evaluation

The energy consumption is determined using the formula $E = Power \times CT$, where CT represents the total computation time, as outlined in Table V, $Power$ denotes the CPU power, which is 10.88 W, and E signifies the energy consumption. A comparative assessment of the energy usage between the proposed and SoTA protocols [12], [20], [23], [24], [42], [43] is presented in Table VI. The results in Table VI demonstrate that our proposed protocol exhibits the lowest energy consumption among the compared protocols.

E. Storage Overhead Analysis

This section evaluates the storage cost of the proposed protocol in comparison with SoTA protocols. To determine the overall storage cost, we considered all the parameters that each entity within the authentication protocol must retain. Additionally, we considered the bit-length needed for each parameter, as specified in VI-C. In our proposed protocol, the $\mathcal{GSS}_k$ writes $(id_j, x_j, \mathcal{C}_j, \mathcal{R}_j)$ in database, requiring (160+128+160+160) = 608 $bits$. Likewise, $\mathcal{U}_i$ stores (η, pid_i^u, pid_j), which takes (128+160+160) = 448 $bits$. Therefore, the accumulative storage overhead of the proposed protocol is (604+448) = 1052 $bits$. The storage overhead for the SoTA protocols [12], [20], [23], [24], [42], [43] is computed in same way and is presented in Table VI.

F. Security Features Analysis

The comparative analysis on the security features of our designed and contemporary competing protocol [12], [20], [23], [24], [42], [43] against essential security features is summarized in Table VII. Referring to the facts of Table VII, it is justified that the designed protocol offers better security

TABLE V: Analysis of Computation and Communication Costs

Protocols	$\mathcal{U}_i$ [ms]	$\mathcal{GSS}_k$ [ms]	$\mathcal{D}_j$ [ms]	Aggregated Cost [ms]	Communication Cost [bits]
SoTA1	$14\mathbb{T}_H \approx 12.362$	$10\mathbb{T}_H \approx 0.039$	$9\mathbb{T}_H \approx 16.308$	28.71	2694
SoTA2	$7\mathbb{T}_H + 3\mathbb{T}_{E/D} + 3\mathbb{T}_{PM} \approx 10.192$	$2\mathbb{T}_H + 3\mathbb{T}_{E/D} + 1\mathbb{T}_{PM} \approx 0.0163$	$3\mathbb{T}_H + 2\mathbb{T}_{E/D} + 2\mathbb{T}_{PM} \approx 10.514$	20.72	2250
SoTA3	$9\mathbb{T}_H + 1\mathbb{T}_{PM} \approx 8.589$	$6\mathbb{T}_H + 1\mathbb{T}_{PM} \approx 0.0262$	$5\mathbb{T}_H \approx 11.56$	24.79	3344
SoTA4	$11\mathbb{T}_H + 1\mathbb{T}_{E/D} \approx 10.254$	$8\mathbb{T}_H + 2\mathbb{T}_{E/D} \approx 0.035$	$8\mathbb{T}_H \approx 14.496$	24.79	2720
SoTA5	$15\mathbb{T}_H + 1\mathbb{T}_{FE} \approx 13.46$	$12\mathbb{T}_H \approx 0.047$	$4\mathbb{T}_H + 1\mathbb{T}_{PUF} \approx 7.758$	21.27	4288
SoTA6	$3\mathbb{T}_H + 2\mathbb{T}_{E/D} \approx 3.731$	$4\mathbb{T}_H + 1\mathbb{T}_{E/D} \approx 0.0194$	$9\mathbb{T}_H + 2\mathbb{T}_H \approx 17.90$	21.65	3808
Proposed	$9\mathbb{T}_H \approx 7.947$	$7\mathbb{T}_H + 2\mathbb{T}_{E/D} \approx 0.0311$	$6\mathbb{T}_H \approx 10.872$	18.76	2176

TABLE VI: Energy and Storage Costs Comparison

Protocol	**Aggregated Costs**	
	Energy (mJ)	Storage $(bits)$
SoTA1	312.36	1184
SoTA2	225.44	2272
SoTA3	269.72	1152
SoTA4	269.72	2144
SoTA5	231.42	2592
SoTA6	235.55	1888
Proposed	204.11	1052

features than all competing protocol [12], [20], [23], [24], [42], [43]. Most importantly, no other protocol in the existing protocols can survive against physical/ cloning attacks except [42].

TABLE VII: Analysis of Security Features

Protocols → **Security Features ↓**	Our	SoTA1	SoTA2	SoTA3	SoTA4	SoTA5	SoTA6
Resist Physical and Cloning Attacks	✓	✗	✗	✗	✗	✓	✗
Anonymity and Untraceability	✓	✗	✗	✓	✗	✓	✓
Ensures Mutual Authentication	✓	✓	✓	✓	✓	✓	✓
Resist Mobile User Impersonation Attack	✓	✗	✗	✓	✗	✓	✓
Resist GSS Impersonation Attack	✓	✗	✓	✓	✗	✓	✓
Resist Drone Impersonation Attack	✓	✗	✗	✓	✗	✓	✓
Resist Clock-Synch. Attack	✓	✗	✗	✗	✗	✗	✗
Perfect Forward and Backward Secrecy	✓	✗	✗	✗	✗	✓	✗
Resists ESL Attack	✓	✗	✗	✗	✗	✗	✗
Resists DoS Attack	✓	✓	✗	✓	✗	✓	✓

Note: ✓Provided; ✗Not Provided

In summary, the results of Section VI demonstrate that our proposed protocol outperforms competing schemes in terms of computational, communication, energy, and storage efficiency while ensuring the highest level of security. Compared to existing protocols [12], [20], [23], [24], [42], [43], our approach significantly reduces resource consumption without compromising security guarantees, making it a more practical and robust solution for real-world applications.

VII. Conclusion

Due to the fragile nature of communication channels, the IoD architecture 6G-U-ITS faces severe cyber and physical threats. In this paper, we propose a PUF based provably secure authentication protocol for 6G-U-ITS to overcome such issues. We formally and informally analyzed the designed solution to highlight its robustness against a variety of attacks. Moreover, we showed that the performance comparison against different matrices (i.e., communication and computation overheads) are evident that the designed protocol is efficient than all competing solutions. In a nutshell, the better performance and security strength of the designed protocol make it the best candidate to be deployed in a resource-contained 6G-U-ITS.

In future, we will simulate out protocol with the help of a broadly-recognized NS3-simulator tool to measure its performance from distinct network parameters (i.e., end-to-end delay, throughput, etc.). Additionally, we plan to extend our current design to support batch (aggregate) authentication, enabling simultaneous verification of multiple mobile users to improve system scalability and efficiency further.

References

[1] Z. Xiong, H. Sheng, W. Rong, and D. E. Cooper, "Intelligent transportation systems for smart cities: a progress review," *Science China Information Sciences*, vol. 55, no. 12, pp. 2908–2914, 2012.

[2] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "Uav-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 22–28, 2017.

[3] S. H. Alsamhi, O. Ma, M. S. Ansari, and F. A. Almalki, "Survey on collaborative smart drones and internet of things for improving smartness of smart cities," *Ieee Access*, vol. 7, pp. 128 125–128 152, 2019.

[4] A. Shahidinejad and J. Abawajy, "An all-inclusive taxonomy and critical review of blockchain-assisted authentication and session key generation protocols for iot," *ACM Computing Surveys*, vol. 56, no. 7, pp. 1–38, 2024.

[5] R. Karmakar, G. Kaddoum, and O. Akhrif, "A blockchain-based distributed and intelligent clustering-enabled authentication protocol for uav swarms," *IEEE Transactions on Mobile Computing*, 2023.

[6] D. Li, D. Liu, Y. Ren, Y. Sun, Z. Guan, Q. Wu, J. Hu, and J. Liu, "Cpaka: Mutual authentication and key agreement scheme based on conditional puf in space-air-ground integrated network," *IEEE Transactions on Dependable and Secure Computing*, no. 01, pp. 1–14, 2023.

[7] Z. Zhang, C. Hsu, M. H. Au, L. Harn, J. Cui, Z. Xia, and Z. Zhao, "Prlap-iod: A puf-based robust and lightweight authentication protocol for internet of drones," *Computer Networks*, p. 110118, 2023.

[8] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6g: Ai empowered wireless networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[9] G. Liu, Z. Yan, D. Wang, H. Wang, and T. Li, "Deptvm: Decentralized pseudonym and trust value management for integrated networks," *IEEE Transactions on Dependable and Secure Computing*, 2023.

[10] M. Tanveer, H. Alasmary, N. Kumar, and A. Nayak, "Saaf-iod: Secure and anonymous authentication framework for the internet of drones," *IEEE Transactions on Vehicular Technology*, 2023.

[11] M. Wazid, A. K. Das, N. Kumar, A. V. Vasilakos, and J. J. Rodrigues, "Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3572–3584, 2018.

[12] J. Srinivas, A. K. Das, N. Kumar, and J. J. Rodrigues, "Tcalas: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6903–6916, 2019.

[13] Y. Zhou, T. Liu, F. Tang, and M. Tinashe, "An unlinkable authentication scheme for distributed iot application," *IEEE Access*, vol. 7, pp. 14 757–14 766, 2019.
[14] Y. Zhang, D. He, L. Li, and B. Chen, "A lightweight authentication and key agreement scheme for internet of drones," *Computer Communications*, vol. 154, pp. 455–464, 2020.
[15] B. Bera, D. Chattaraj, and A. K. Das, "Designing secure blockchain-based access control scheme in iot-enabled internet of drones deployment," *Computer Communications*, vol. 153, pp. 229–249, 2020.
[16] B. Bera, S. Saha, A. K. Das, N. Kumar, P. Lorenz, and M. Alazab, "Blockchain-envisioned secure data delivery and collection scheme for 5g-based iot-enabled internet of drones environment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9097–9111, 2020.
[17] M. Nikooghadam, H. Amintoosi, S. H. Islam, and M. F. Moghadam, "A provably secure and lightweight authentication scheme for internet of drones for smart city surveillance," *Journal of Systems Architecture*, vol. 115, p. 101955, 2021.
[18] S. Hussain, K. Mahmood, M. K. Khan, C.-M. Chen, B. A. Alzahrani, and S. A. Chaudhry, "Designing secure and lightweight user access to drone for smart city surveillance," *Computer Standards & Interfaces*, vol. 80, p. 103566, 2022.
[19] B. A. Alzahrani, A. Barnawi, and S. A. Chaudhry, "A resource-friendly authentication protocol for uav-based massive crowd management systems," *Security and Communication Networks*, vol. 2021, 2021.
[20] M. Tanveer, N. Kumar, M. M. Hassan *et al.*, "Ramp-iod: A robust authenticated key management protocol for the internet of drones," *IEEE Internet of Things Journal*, 2021.
[21] C. Pu, A. Wall, and K.-K. R. Choo, "Bilinear pairing and puf based lightweight authentication protocol for iod environment," in *2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2022, pp. 115–121.
[22] J. Cui, X. Liu, H. Zhong, J. Zhang, L. Wei, I. Bolodurina, and D. He, "A practical and provably secure authentication and key agreement scheme for uav-assisted vanets for emergency rescue," *IEEE Transactions on Network Science and Engineering*, 2023.
[23] S. Hussain, S. A. Chaudhry, O. A. Alomari, M. H. Alsharif, M. K. Khan, and N. Kumar, "Amassing the security: An ecc-based authentication scheme for internet of drones," *IEEE Systems Journal*, 2021.
[24] Z. Ali, S. A. Chaudhry, M. S. Ramzan, and F. Al-Turjman, "Securing smart city surveillance: A lightweight authentication mechanism for unmanned vehicles," *IEEE Access*, vol. 8, pp. 43 711–43 724, 2020.
[25] S. A. Chaudhry, M. S. Farash, N. Kumar, and M. H. Alsharif, "Pflua-diot: A pairing free lightweight and unlinkable user access control scheme for distributed iot environments," *IEEE Systems Journal*, 2020.
[26] Y. Tian, J. Yuan, and H. Song, "Efficient privacy-preserving authentication framework for edge-assisted internet of drones," *Journal of Information Security and Applications*, vol. 48, p. 102354, 2019.
[27] G. Cho, J. Cho, S. Hyun, and H. Kim, "Sentinel: A secure and efficient authentication framework for unmanned aerial vehicles," *Applied Sciences*, vol. 10, no. 9, p. 3149, 2020.
[28] S. A. Chaudhry, K. Yahya, M. Karuppiah, R. Kharel, A. K. Bashir, and Y. B. Zikria, "Gcacs-iod: A certificate based generic access control scheme for internet of drones," *Computer Networks*, vol. 191, p. 107999, 2021.
[29] A. Irshad, S. A. Chaudhry, A. Ghani, and M. Bilal, "A secure blockchain-oriented data delivery and collection scheme for 5g-enabled iod environment," *Computer Networks*, p. 108219, 2021.
[30] Y. K. Ever, "A secure authentication scheme framework for mobile-sinks used in the internet of drones applications," *Computer Communications*, vol. 155, pp. 143–149, 2020.
[31] C. Pu, K.-K. R. Choo, and D. Korać, "A lightweight and anonymous application-aware authentication and key agreement protocol for the internet of drones," *IEEE Internet of Things Journal*, 2024.
[32] M. Umar, J. Wang, H. K. Ahmad, S. Zhao, F. Li, S. Wang, M. Zheng, Y. Shen, Z. Zhang, and X. Guo, "Multiple attributes based physical layer authentication through propagation scenario identification in the internet of vehicles," *Vehicular Communications*, vol. 45, p. 100708, 2024.
[33] J. Miao, Z. Wang, X. Ning, A. Shankar, C. Maple, and J. J. Rodrigues, "A uav-assisted authentication protocol for internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2024.
[34] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.
[35] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2001, pp. 453–474.
[36] D. Abbasinezhad-Mood, S. M. Mazinani, M. Nikooghadam, and A. Ostad-Sharif, "Efficient provably-secure dynamic id-based authenticated key agreement scheme with enhanced security provision," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 1227–1238, 2020.
[37] C. Pu, C. Warner, K.-K. R. Choo, S. Lim, and I. Ahmed, "litegap: Lightweight group authentication protocol for internet of drones systems," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 4, pp. 5849–5860, 2023.
[38] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *International Workshop on Public Key Cryptography*. Springer, 2005, pp. 65–84.
[39] M. El-Zawawy, A. Brighente, and M. Conti, "Setcap: Service-based energy-efficient temporal credential authentication protocol for internet of drones," *Computer Networks*, 2022.
[40] M. Eldefrawy, I. Butun, N. Pereira, and M. Gidlund, "Formal security analysis of lorawan," *Computer Networks*, vol. 148, pp. 328–339, 2019.
[41] C.-I. Fan, A. Karati, and S.-L. Wu, "A privacy-aware provably secure smart card authentication protocol based on physically unclonable functions," *IEEE Transactions on Dependable and Secure Computing*, 2023.
[42] A. Irshad, B. A. Alzahrani, A. Albeshri, K. Alsubhi, A. Nayyar, and S. A. Chaudhry, "Spake-dc: A secure puf enabled authenticated key exchange for 5g-based drone communications," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 4, pp. 5770–5780, 2023.
[43] S. U. Jan, M. Bilal, A. Ghani, S. Khan, R. Ahmad, and D.-H. Kim, "Robust and lightweight authentication for securing communication in the internet-of-drones (iod) environment," in *IEEE INFOCOM 2024-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2024, pp. 01–06.

A Privacy-Preserving Access Control Protocol for 6G Supported Intelligent UAV Networks