A Lightweight and Anonymous Application-Aware Authentication and Key Agreement Protocol for the Internet of Drones

Image Bhattarai, *Graduate Student Member, IEEE*, Cong Pu, *Member, IEEE*, Kim-Kwang Raymond Choo, *Senior Member, IEEE*, and Dragan Korać

***Abstract*—The drone technology has continuously been evolving since the beginning of the first decade of the 21st century with exceptional growth over the last several years. To pave the way for an interoperable aerial–ground communication platform, the Internet of Drones (IoD) framework has emerged to systematically organize a batch of drones to collect multiple application-specific data simultaneously and report them to a close ground station. As the collected data might contain sensitive information, people become more critically aware of data security and privacy issues associated with IoD applications. Authentication and key agreement protocols are able to protect IoD data from unauthorized access. However, the recent schemes fail to distinguish between types of data during the authentication and key establishment process, which leads to data leakage that sensitive data are being accessed by unauthorized entities. To address the data leakage issue and fill the research gap, this article proposes a lightweight and anonymous application-aware authentication and key agreement protocol (also called *liteA4*) for IoD systems. The fundamental idea of *liteA4* is that the ground station and the drone perform data type-aware mutual authentication and establish separate session keys for different types of data before the drone delivers the collected data to the ground station. The major techniques, such as hash function, bitwise XOR, and physical unclonable function (PUF), are used to implement *liteA4*. We select the Automated Validation of Internet Security Protocols and Applications (AVISPAs) tool to verify the security of *liteA4* in the cyber-threat environment. We also set up a simulation framework and conduct comprehensive and comparative experiments to validate the performance of *liteA4*. Extensive experimental results demonstrate that *liteA4* not only is a safe and reliable protocol in the adversarial setting but also provides better results than its counterpart approaches in terms of communication overhead, computational time, storage cost, as well as energy consumption.**

***Index Terms*—Anonymous, application-aware, authenticated key agreement, Internet of Drones (IoD), lightweight.**

Manuscript received 17 November 2023; revised 26 January 2024 and 13 February 2024; accepted 15 February 2024. *(Corresponding author: Cong Pu.)*

Image Bhattarai and Cong Pu are with the Department of Computer Science, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: image.bhattarai@okstate.edu; congpu@acm.org).

Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: raymond.choo@fulbrightmail.org).

Dragan Korać is with the Department of Mathematics and Informatics, University of Banja Luka, 78000 Banja Luka, Bosnia and Herzegovina (e-mail: dragan.korac@pmf.unibl.org).

Digital Object Identifier 10.1109/JIOT.2024.3367799

I. Introduction

As DRONE technology continues to evolve and starts playing a critical role in modern smart cities, the civil and commercial industries have transformed and adapted as well. During the COVID-19 pandemic, drones were used in a wide array of humanitarian contexts, e.g., delivering vaccines in India [1], detecting individuals with infectious respiratory conditions in Australia [2], etc. With the innovations in lithium-ion battery technology, ultradense microchip, and carbon fiber composites, the drone industry faces a bright future ahead. According to the recently published ''Drone Market Analysis'' [3], the commercial and recreational drone markets are estimated to be valued at approximately 56 billion U.S. dollars by the end of 2030. Taking advantage of 5G & B5G and artificial intelligence & machine learning, we envision that the drone technology will open up a goodly number of new services and reshape the way we work, live and thrive in the near future.

To support the development of aerial communication technology, several international standard development organizations, including the Third Generation Partnership Project (3GPP), the Institute of Electrical and Electronics Engineers (IEEE), as well as the International Telecommunication Union (ITU) have been working on the standardization (e.g., IEEE P1936.1 [4], 3GPP TR 36.777 [5], ITU F.749.10 [6]) for the integration of drones into existing/emerging communication infrastructure [7]. With the new era of drones, the conventional Internet of Things (IoT) has evolved to the Internet of Drones (IoD). In the IoD paradigm, each drone is regarded as an aerial smart object equipped with sensing devices, computing capabilities, and storage systems, and is able to communicate with any nearby entity (i.e., other drones, ground stations, ground IoT devices, etc.) via wireless technology. Specifically, the IoD paradigm virtually partitions airspace (or geographical area) into task zones, as shown in Fig. 1. In each task zone, one or multiple ground stations can communicate with nearby drones for task-specific operations (e.g., retrieving traffic information or collecting data from ground IoT devices) through various types of connection in a way that enables effective information gathering, sharing, and processing. In summary, the IoD paradigm stands in the center of the 4th industrial revolution, and is anticipated to address the grand challenges of conventional mobile networks and elevate mobile computing to new heights.

TABLE I
NOMENCLATURES

Notation	Meaning
liteA4	Lightweight and Anonymous Application-Aware Authentication and Key Agreement Protocol
PUFs	Physical Unclonable Functions
IoT	Internet of Things
IoD	Internet of Drones
3GPP	The Third Generation Partnership Project
ITU	The International Telecommunication Union
IEEE	The Institute of Electrical and Electronics Engineers
AVISPA	Automated Validation of Internet Security Protocols and Applications
HLPSL	High-Level Protocol Specification Language
OFMC	On-the-fly Model Checker
CL-AtSe	Constraint-Logic-based Attack Searcher
SLAP-IoD	Secure and Lightweight Authentication Protocol - IoD
SAAF-IoD	Secure and Anonymous Authentication Framework - IoD
PUF-IPA	PUF-based Anonymous Authentication Protocol - IoT
5G	5th Generation Mobile Network
B5G	Beyond 5th Generation Mobile Network
AKA	Authentication and Key Agreement
XOR	Exclusive OR Operation
ECC	Elliptic Curve Cryptography
BPV	Boyko-Peinado-Venkatesan
ACE	Lightweight Hash Function and Authenticated Encryption

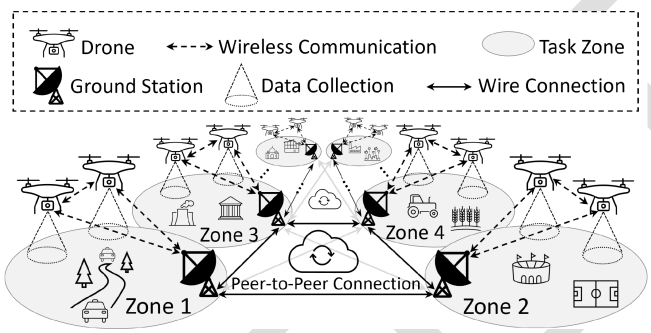

Fig. 1. IoD framework and potential applications. Zone 1: traffic surveillance and control; Zone 2: entertainment, sport, and media; Zone 3: industrial plant environmental monitoring and safety; and Zone 4: precision agriculture.

A. Research Challenges and Motivation

Although the IoD paradigm brings substantial benefits and enables an extremely large number of potentially promising applications, its generic architecture necessitates innovative solutions, ranging from security protocol to data privacy. The security and privacy challenges require engineers' full attention and scientific input from researchers because the IoD security and privacy are not built-in properties but added on as an afterthought. As a result, plenty of malicious activities attempt to take advantage of this design flaw and launch assaults on the IoD systems to achieve their adversarial objectives. Taking drone-assisted autonomous driving as an example, drones are deployed to collect information about real-time traffic conditions for traffic management authority as well as detect far-away objects for autonomous driving vehicles to operate safely [8]. Disclosing/compromising drone-collected data to/by unauthorized entities can result in car accidents or even terrorist attacks [9].

During the past years, a variety of authentication techniques [10], [11], [12], [13], [14] have been proposed to protect either IoD data from adversary's unauthorized access or similar environments, such as IoT and vehicular ad hoc networks. Unfortunately, the state-of-the-art techniques either have inherent security vulnerabilities in their designs or realize the desired security and privacy requirements with resource-hungry operations. Most importantly, none of these techniques distinguish between types of device-collected data during the authentication and key establishment process. Thus, they have to establish one secret session key for the entire communication session via which the drone will submit all collected data. However, this will lead to data leakage that sensitive data are being accessed by unauthorized entities with the same secret session key. For example, the adversary might be able to compromise previously established secure session keys. If the same secure session key is used to encrypt all types of data, the adversary who compromises the previously established secure session key can have access to all the data collected by the drone. This is because all data are encrypted with the same session key. However, if different secure session keys are used to encrypt different types of data collected by the drone, the adversary can only obtain access to the data whose secure session key has been compromised. Other types of data that are encrypted with different secure session keys are still safe. Last but not least, conventional session-based key establishment schemes will generate a large number of secret session keys if there are frequent communications between the drone and the ground station. It is immediately obvious that repeatably establishing secret session keys cause non-negligible computational overhead to IoD entities, especially to resource-constrained drones.

B. Contribution

Motivated by the above discussion, in this article we focus on a secure data type-aware authentication and key agreement protocol that takes advantage of cost-effective techniques to realize the requirements of data privacy and security. It would be unprecedented to realize such an innovative approach because the current IoD technical community does not have the similar technique, and the produced work will fill a gap in the existing body of research. We also verify the protocol's security resilience against cyber attacks with a specific security protocol verification tool, and evaluate its performance and scalability through extensive experiments. In summary, our contribution is summarized in the following.

1) We propose a lightweight and anonymous application-aware authentication and key agreement protocol (also called *liteA4*) for IoD systems. In *liteA4*, the ground station and the drone perform data type-aware mutual authentication and establish separate session keys for different types of data before the drone delivers the collected data to the ground station.
2) We set up an adversarial environment in the Automated Validation of Internet Security Protocols and Applications tool (AVISPA) [15], implement *liteA4* in the High-Level Protocol Specification Language (HLPSL) [16], and then evaluate *liteA4*'s security resilience against several cyber attacks, such as man-in-the-middle and replay attacks.

3) We set up an experimental environment and conduct comprehensive experiments to evaluate *liteA4*'s performance and scalability in terms of various metrics. In addition, we select three representative benchmark schemes, SLAP-IoD [17], SAAF-IoD [18], and PUF-IPA [19], implement them and *liteA4*[1] in Python, and compare their performance and scalability.

Extensive experimental results demonstrate that *liteA4* not only is a safe and reliable protocol in the adversarial setting but also provides superior performance than its counterparts in terms of communication overhead, computational time, storage cost, as well as energy consumption.

C. Novelty

Our work is different from the existing research in terms of three aspects: 1) investigating the promising IoD architecture; 2) developing a new application-aware authentication protocol; and 3) adopting resource-friendly functions and operations. First of all, we focus our efforts to contribute to the IoD community. The promising IoD paradigm is believed to be one of the most important subjects for scientific investigation within many commercial companies and technical groups. Our work will provide a thorough analysis about the IoD architecture and its unique security and privacy challenges and requirements. Second, this work proposes a novel application-aware authentication protocol for IoD systems. The IoD community does not lack authentication mechanisms to protect the IoD communications. However, what has been lacking in the current theory is a lightweight and anonymous application-aware authentication protocol that adopts resource-friendly computing operations to achieve the security and privacy requirements concurrently for drone communications in the IoD environment. Moreover, our work can significantly decrease the communication and computation cost through reducing the number of established secure session keys, compared to the traditional authentication approaches. This is because the drone establishes a unique secure session key for each type of data with the ground station, and each secure session key can be used to encrypt the same type of data during multiple communication sessions with the ground station. Third, we choose resource-friendly techniques, such as hash function, bitwise XOR, and PUF, to realize the proposed application-aware authentication protocol. Compared to other heavyweight techniques (i.e., elliptic curve cryptography (ECC), bilinear pairings, etc.) which are used for resource-constrained IoD systems, our solution has less computational and storage overhead while meeting the required security and privacy requirements.

D. Paper Organization

The remainder of this article is organized as follows. The state-of-the-art techniques are reviewed in Section II. We present network and adversarial models as well as security and performance requirements in Section III. After that, we introduce the proposed protocol in Section IV. We also conduct security verification and analysis as well as experimental study, and present their results in Sections V and VI, respectively. Finally, we conclude this article with the direction of future research in Section VII.

[1]*liteA4*'s HLPSL verification programs are publicly available at https://github.com/congpu/liteA4.

II. Related Work

Even though the data type-aware authentication and key agreement protocol is still lacking in the current IoD community, conventional approaches have been studied for IoD systems in the last few years. Yu et al. [17] developed an authentication protocol, named as SLAP-IoD, to protect IoD data exchange over insecure wireless medium. The major operation that they choose to realize the protocol objectives is the PUF. Here, the PUF serves two purposes: 1) physical identity protection and 2) less computation overhead. However, the authors fail to consider the stability and error-tolerance of PUF in the harsh environment (i.e., wide swings in temperatures) where it is extremely difficult to restore the same secret information with the PUF. Some researchers argue that the state-of-the-art schemes have relatively high computation and communication cost. To improve the existing situation, they propose a lightweight authentication and key agreement approach (called AKA) with hash function and bitwise XOR operation in [21]. Unfortunately, other researchers [31] have systematically proved that AKA actually cannot protect IoD systems from harmful attacks such as compromised user anonymity, denial-of-service, and replay attacks. Lounis et al. [22] investigated how to build a secure communication channel between drones, and then design a PUF-based drone authentication protocol (known as D2D-MAP). The major drawbacks of D2D-MAP can be summarized as follows. First, they assume that drones will be operating in an ideal environment where the PUF is able to function perfectly. However, this is not exactly true in practice, e.g., drones are being deployed for search and rescue missions in the dangerous wildfire situation. Second, D2D-MAP creates one secret session key to encrypt all collected data which might contain sensitive as well as nonsensitive information. This might disclose the sensitive information to unauthorized entity, resulting in potential data leakage.

In addition to the above-mentioned work, some other solutions, such as precalculation-based [23], ECC-based [24], blockchain-based [25], smart cards-based [26], proxy signature delegation-based [27], and ACE permutation-based [28] authentication and key agreement protocols, have been designed to secure wireless communications between IoD entities. These solutions are able to achieve the desired levels of security and privacy, however, they are either realized with resource-hungry operation (i.e., Boyko–Peinado–Venkatesan (BPV)-FourQ), demanding additional hardware (i.e., smart card), or having inherent design flaws (i.e., ECC). For instance, BPV precalculation and FourQ are chosen to authenticate drone, user, and ground station in the IoD environment. While the BPV algorithm intrinsically increases the size of private key (i.e., $\geq$ 64 KB), a nonnegligible storage overhead is being added to the resource-constrained drones. Moreover, the security analysis and experimental study [32]

TABLE II
COMPARISON OF EXISTING WORKS

Feature	[17]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	*liteA4*
MU	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MI	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
DA	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓	✓	✓	✓
LO	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓	✓	✓
AS	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

✓: Provides ✗: Does Not Provide
Features: **MU:** Mutual Authentication; **MI:** Message Integrity; **DA:** Drone/Device Anonymity; **LO:** Lightweight Operations; **AS:** Application-Aware Session Key Establishment.

have demonstrated that one ECC-based approach [33] might be vulnerable to drone impersonation, and the adversary has some chance to compromise its session keys. Besides the above-mentioned weaknesses, these protocols have a common problem: implicitly assuming all drone-collected data have the same type and establishing one secret session key to encrypt all drone-collected data. As mentioned earlier, this implicit assumption will lead to data leakage that sensitive data are being accessed by unauthorized entities.

In the IoT domain, some solutions [20], [29], [30] have been proposed to protect data from unauthorized access. The work in [20] focuses on a realistic anonymous user authentication in wireless sensor networks, where the legitimate user is allowed to access data from any specific sensor node. Aman et al. [30] used PUF along with wireless link fingerprints derived from the wireless channel characteristics between two communicating entities to realize data provenance with authentication and privacy preservation in IoT systems. However, the above approaches do not consider the types of data during the authentication process. In [29], a lightweight privacy-preserving authentication protocol is proposed for RFID systems. The authors consider the ideal PUF environment, which is different from our work. In this article, we relax the assumption of the ideal PUF environment by integrating fuzzy extractor and error correction code with the PUF to deal with the scenario that the identical challenges fed to the PUF might not be able to get the same responses.

After analyzing the approaches presented above, we have identified research gaps relevant to the protection of IoD data from adversary's unauthorized access. First, the existing approaches do not distinguish between types of data during the process of authentication and key establishment. As a result, one secure session key is established to encrypt all collected data, which leads to data leakage that sensitive data are being accessed by unauthorized entities with the same secure session key. Second, conventional session-based key establishment schemes will generate a large number of secret session keys if there are frequent communications between the drone and the ground station. It is immediately obvious that repeatedly establishing secret session keys causes nonnegligible communication and computation overhead to IoD entities, especially to resource-constrained drones. Last but not least, the existing solutions either adopt resource-hungry operations or have inherent vulnerabilities in their design.

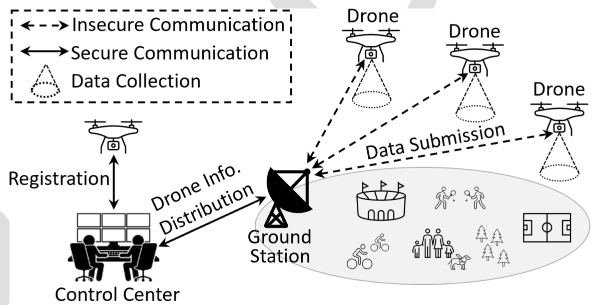

Fig. 2. System model.

In summary, the IoD paradigm has become an active research field and is of great interest to many technical communities and commercial companies, e.g., IEEE Communications Society [34], Ericsson [35], etc. However, the authentication and key agreement protocol that establishes the data type-aware secret session key with resource-friendly computing operations is still missing in the IoD community. Thus, in this article, we focus on the lightweight and anonymous application-aware authentication and key agreement protocol. It would be unprecedented to realize such an innovative approach because the current IoD technical community does not have the similar technique, and the produced work will fill a gap in the existing body of research. Finally, we compare *liteA4* with existing schemes in Table II.

III. NETWORK AND ADVERSARIAL MODELS AND THE OBJECTIVES OF PROTOCOL AND THE DESIGN OF PUF

A. Network Model

In our network there are three major participants, the control center, the ground station, and the drone, which are shown in Fig. 2. The control center is a fully trusted entity which registers each drone's identity information in the database. After completing the registration, the control center dispatches a fleet of drones to the task region, where drones will collect the information of targets and periodically report the observations to a nearby ground station. Note that the drone observations might entail multitudinous data (different data types; sensitive and nonsensitive data) about multiple targets. In order to avoid storing secret information in the memory directly, the integrated circuits of drones are produced with PUFs [36], and the secret information can be restored via

PUF when needed. After receiving the observational data from drones, the ground station will decrypt the observational data and transmit them to the control center over the secure channel. Finally, we assume that the ground station is a trusted player as well.

B. Threat Models

In the system, two well-known threat models, Canetti–Krawczyk and Dolev–Yao threat models [37], are considered for the potential adversaries. The rationale behind the adoption of the Dolev–Yao and Canetti–Krawczyk models is to establish a ‘‘strong adversary model’’ through combing the powerful adversary capabilities from the Dolev–Yao and Canetti–Krawczyk models. The Dolev–Yao threat model assumes that the wireless communication medium is unsafe. As a result, the ground station and the drone who are communicating over this unsecure platform do not proceed on the exchange of critical information before verifying each other's identities. Moreover, since the wireless medium is publicly accessible, the exchanged messages between the ground station and the drone can be eavesdropped or even captured by the nearby adversary. And on this basis the adversary might choose to fabricate the messages, and then replay them to disrupt the normal communication. The adversary also can physically capture the drone with specific types of equipment, and attempt to extract the secret information stored in the memory. However, this malicious behavior may change the physical characteristics of integrated circuit, resulting in PUF malfunctions. In addition, to extend the capabilities of adversary mentioned above, the system also considers the Canetti–Krawczyk threat model. Specifically, the adversaries are able to compromise session state specific information or previously established secure session keys. In summary, the goal of the adversary is to access the drone observations without being detected.

C. Objectives of Protocol

We identify the following security and performance objectives to be met by the proposed protocol.

1) *Authentication:* The identities of legitimate drone and ground station can be verified.
2) *Application-Aware Session Key Establishment:* A data type specific secret session key can be established between the drone and the ground station.
3) *Integrity:* The accuracy, completeness, and consistency of messages can be guaranteed.
4) *Confidentiality:* The drone's observational data is unintelligible to the external adversary.
5) *Anonymity:* The drone uses the pseudonym, rather than the real identity, for the communication with the ground station.
6) *Smaller Overhead:* Smaller computation and communication overhead should be observed.

D. Physical Unclonable Function

PUFs are universally utilized as a hardware-specific security primitive to offer cryptographic services for electronic devices [38]. The physical structure of PUF is formed in the process of manufacturing. Since it is inevitable for each integrated circuit to have slight physical differences from the manufacturing process, the PUF is believed to be impossible to replicate or clone. Thanks to its unique features, the PUF is generally considered to be the identification of an electronic device, which is analogous to a person's social security number.

Algorithm 1: Response Generation Algorithm *rGen*

Input: Modulus n; Challenge che

Function `rGen(`n, che`):`
- /* $\stackrel{\circledast}{\leftarrow}$ denotes sampling */
- /* $\oplus$ denotes exclusive OR function */
- /* $\mathbb{Z}_n$ denotes the set of remainders in arithmetic modulo n */
- $O = F_{puf}(che)$;
- $res \stackrel{\circledast}{\leftarrow} \mathbb{Z}_n$;
- $S = O \oplus ECC(res)$;
- **return** $\{res, S\}$;

Algorithm 2: Response Restore Algorithm *rRes*

Input: Challenge che; Helper string S

Function `rRes(`che, S`):`
- $O' = F_{puf}(che)$;
- $res = D_{er}(S \oplus O')$;
- **return** res;

Typically, the PUF is fed with an input and generates an output. The input and output are called *challenge* and *response*, respectively. The combination of challenge and response goes by the name challenge–response pair (CRP). A single PUF always responds to the same challenge equivalently (i.e., the same response is produced), and two distinct PUF instances should respond to the same unbiased challenges differently (i.e., different responses are produced). Generally, the PUF could be demonstrated as a math expression, denoted as $res = F_{puf}(che)$, where PUF's challenge and response are represented as che and res, respectively.

In noisy environments, the identical challenges fed to the PUF might not be able to get the same responses [39]. In other words, the PUF is sensitive to external environment changes/noise, thus, the secret data of cryptographic operations might not be regenerated by the PUF. To resolve this important issue, we decide to integrate fuzzy extractor and error correction code with the PUF. A PUF response generation algorithm (*rGen*) is first defined in Algorithm 1. The *rGen* algorithm will output a tuple $\{res, S\}$. Specifically, res is the CRP response and S is a helper string. Here, S is used to reproduce res.

The rationale behind the adoption of error correction code [40] is to reduce bit errors (up to x bit) in res. A response restore algorithm (*rRes*) is also created and shown in Algorithm 2. With *rRes*, res can be restored with the assistance of S and D_{er}, even though the PUF's output O' is different from its original output O by at most x bits.

IV. Proposed Protocol

In this section, we describe the proposed lightweight and anonymous application-aware authentication and key agreement protocol, which we refer to as *liteA4* in the following.

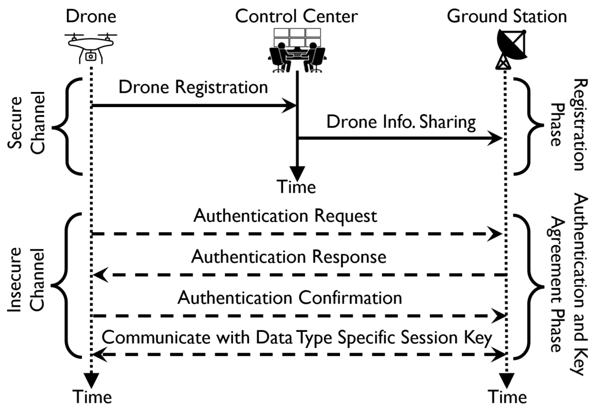

Fig. 3. *liteA4* communication sequence diagram.

The communication sequence diagram of *liteA4* is shown in Fig. 3. The basic idea of *liteA4* is that the control center first registers each drone for a set of different tasks (data types) to complete (or collect) in the task region. Then, the control center shares each drone's identity information and registered tasks (data types) with the ground station via a secure channel. Finally, the ground station and the drone perform data type-aware mutual authentication and establish separate session keys for different types of data before the drone delivers the collected data to the ground station. The major techniques such as hash function, bitwise XOR, and PUF are used to implement *liteA4*. In summary, *liteA4* consists of two major phases: 1) drone registration and 2) authentication and key establishment.

A. Drone Registration Phase

The control center registers the drone D_κ at the time t_i in the following steps.

1) The drone D_κ chooses its real identity RID_κ and initial PUF challenge $che_\kappa^{t_i}$. The drone's real identity RID_κ is used to calculate its pseudonym, rather than being used directly in the communication. It is worth mentioning that the drone's pseudonym is mainly used to guarantee no one else is getting its real identity except the legitimate ground station, even though the adversary can get intercepted transcripts.
2) The drone D_κ feeds PUF challenge $che_\kappa^{t_i}$ into its PUF $F_{puf}(\cdot)$ to compute the corresponding PUF response $res_\kappa^{t_i} = F_{puf}(che_\kappa^{t_i})$. The PUF response $res_\kappa^{t_i}$ serves as a critical component in the calculation of other information (e.g., the pseudonym of drone). Thus, the PUF response $res_\kappa^{t_i}$ is dynamically calculated with the PUF challenge $che_\kappa^{t_i}$ and the PUF function $F_{puf}(\cdot)$.
3) The drone D_κ calculates its initial pseudonym $PID_\kappa^{t_i} = H(RID_\kappa \parallel res_\kappa^{t_i})$ with RID_κ and $res_\kappa^{t_i}$, where $H{:}\{0,1\}^m$ is a set of fixed length (saying m bits) strings. The pseudonym $PID_\kappa^{t_i}$ can guarantee the drone's identity privacy. No one else can learn the drone's real identity except the control center.
4) The drone D_κ shares $\{RID_\kappa, PID_\kappa^{t_i}, che_\kappa^{t_i}, res_\kappa^{t_i}\}$ with the control center via a secure channel. The control center is assumed to be a trusted entity that has access to all drones' information. The secure channels can be realized through the time-based one-time password algorithm [41] or the physical mediums.
5) The control center assigns the drone D_κ with a set of different tasks $DT_\kappa = [dt_1, dt_2, \ldots, dt_x, \ldots, dt_n]$ to complete, and shares DT_κ via a secure channel. Here, each task indicates different data types that the drone D_κ needs to collect and dt_x represents the xth task. n is the total number of tasks assigned to the drone D_κ. In *liteA4*, the drone establishes a unique secret session key for different type of data with the ground station.
6) The control center shares the drone D_κ's information $\{RID_\kappa, PID_\kappa^{t_i}, che_\kappa^{t_i}, res_\kappa^{t_i}, DT_\kappa\}$ with the ground station G_z via a secure channel. Here, i is a notation to distinguish different timestamp t_i. With the identity and task information of the drone D_κ, the ground station G_z can negotiate data type-specific secret session keys with the drone D_κ.

When the drone registration phase is complete, the ground station G_z stores the drone D_κ's real identity, initial pseudonym, initial CRP, and registered data types, while the drone D_κ only stores its real identity, initial PUF challenge, as well as registered data types. The major operations of drone registration phase are summarized in Algorithm 3.

Algorithm 3: Drone D_κ Registration Algorithm

/* t_{cur}: the current system time */
/* $RandID(\cdot)$: random ID function */
/* $RandNum(\cdot)$: random number function */
/* $H(\cdot)$: hash function */
/* $SecureSend(\cdot)$: secure data transfer */
/* CC: control center */
Function DroneRegistration():
  $RID_\kappa \leftarrow RandID(t_{cur})$;
  $che_\kappa^{t_i} \leftarrow RandNum(RID_\kappa)$;
  $res_\kappa^{t_i} \leftarrow F_{puf}(che_\kappa^{t_i})$;
  $PID_\kappa^{t_i} \leftarrow H(RID_\kappa \parallel res_\kappa^{t_i})$;
  /* drone shares identity information with control center via secure channel */
  $SecureSend(D_\kappa, CC, \{RID_\kappa, PID_\kappa^{t_i}, che_\kappa^{t_i}, res_\kappa^{t_i}\})$;
  /* control center assigns tasks to drone */
  $DT_\kappa \leftarrow [dt_1, dt_2, \cdots, dt_x, \cdots, dt_n]$;
  /* control center shares registered data types with drone via secure channel */
  $SecureSend(CC, D_\kappa, DT_\kappa)$;

B. Authentication and Key Establishment Phase

When the drone D_κ is about to submit the type dt_x data to the ground station G_z at the time t_j, it mutually authenticates with the ground station G_z and establishes a specific secret session key for the type dt_x data according to the following steps.

1) The drone D_κ computes its PUF response $res_\kappa^{t_i} = F_{puf}(che_\kappa^{t_i})$ and pseudonym $PID_\kappa^{t_i} = H(RID_\kappa \parallel res_\kappa^{t_i})$. For security reasons, the drone does not store the PUF response and the pseudonym in the memory, but calculates them dynamically. The drone is free to cache the pseudonym for rapid access. However, in this article we assume that the drone chooses to delete the pseudonym for saving memory space.

2) The drone D_κ generates a random number r_{t_j} and calculates the following:

$$m_{1a} = r_{t_j} \oplus H\big(GID_z \| t_j \| RID_\kappa \| res_\kappa^{t_i}\big)$$
$$m_{1b} = dt_x \oplus H\big(GID_z \| t_j \| RID_\kappa \| res_\kappa^{t_i} \| r_{t_j}\big)$$
$$m_{1c} = H\big(GID_z \| t_j \| RID_\kappa \| res_\kappa^{t_i} \| r_{t_j} \| dt_x\big).$$

Here, GID_z is the identifier of the ground station G_z. m_{1a} and m_{1b} are used to share r_{t_j} and dt_x with the ground station G_z, respectively. m_{1c} can help the ground station G_z verify the integrity of r_{t_j} and dt_x.

3) The drone D_κ sends the message $M_1 = \{GID_z, t_j, PID_\kappa^{t_i}, m_{1a}, m_{1b}, m_{1c}\}$ to the ground station G_z via an insecure channel. Here, the message M_1 is regarded as the authentication request message.

4) The ground station G_z retrieves the time t_j, and compares it with the current system time t_{cur}. The timestamp verification is designed to reject the replayed messages. If the difference is larger than or equal to a threshold t^Δ, $(t_{cur} - t_j) \geq t^\Delta$, the message M_1 is rejected. Otherwise, the ground station G_z calculates the following:

$$r'_{t_j} = m'_{1a} \oplus H\big(GID_z \| t_j \| RID_\kappa \| res_\kappa^{t_i}\big)$$
$$dt'_x = m'_{1b} \oplus H\Big(GID_z \| t_j \| RID_\kappa \| res_\kappa^{t_i} \| r'_{t_j}\Big)$$
$$m'_{1c} = H\Big(GID_z \| t_j \| RID_\kappa \| res_\kappa^{t_i} \| r'_{t_j} \| dt'_x\Big).$$

If $m'_{1c} \neq m_{1c}$, the message M_1 is rejected and the authentication process fails. In liteA4, the drone is only allowed to establish a secret session key for the assigned data type with the ground station. Thus, if the drone D_κ is not registered for the type dt'_x data, the authentication request is rejected. Otherwise, the ground station G_z generates a random number s_{t_p} and calculates the following at the time t_p:

$$m_{2a} = s_{t_p} \oplus H\Big(RID_\kappa \| res_\kappa^{t_i} \| r'_{t_j} \| t_p \| GID_z\Big)$$
$$m_{2b} = H\Big(RID_\kappa \| res_\kappa^{t_i} \| r'_{t_j} \| t_p \| GID_z \| s_{t_p}\Big).$$

Here, m_{2a} is used to share s_{t_p} with the drone D_κ and m_{2b} can help the drone D_κ verify the integrity of s_{t_p}.

5) The ground station G_z sends the message $M_2 = \{PID_\kappa^{t_i}, t_p, GID_z, m_{2a}, m_{2b}\}$ to the drone D_κ via a public channel. Here, the message M_2 can be considered as the authentication response message.

6) The drone D_κ retrieves the time t_p, and compares it with the current system time t_{cur}. If the difference is larger than or equal to a threshold t^Δ, $(t_{cur} - t_p) \geq t^\Delta$, the message M_2 is rejected. Otherwise, the drone D_κ calculates the following:

$$s'_{t_p} = m'_{2a} \oplus H\big(RID_\kappa \| res_\kappa^{t_i} \| r_{t_j} \| t_p \| GID_z\big)$$
$$m'_{2b} = H\Big(RID_\kappa \| res_\kappa^{t_i} \| r_{t_j} \| t_p \| GID_z \| s'_{t_p}\Big).$$

If $m'_{2b} \neq m_{2b}$, the message M_2 is rejected and the authentication process fails. Otherwise, the drone D_κ generates a random number s_{t_u} and calculates the following at the time t_u:

$$che_\kappa^{t_u} = H\Big(s_{t_u} \| s'_{t_p}\Big)$$
$$res_\kappa^{t_u} = F_{puf}\big(che_\kappa^{t_u}\big)$$
$$PID_\kappa^{t_u} = H\big(RID_\kappa \| res_\kappa^{t_u}\big)$$
$$m_{3a} = s_{t_u} \oplus H\big(GID_z \| t_u \| RID_\kappa \| res_\kappa^{t_i}\big)$$
$$m_{3b} = che_\kappa^{t_u} \oplus H\big(GID_z \| t_u \| RID_\kappa \| res_\kappa^{t_i} \| s_{t_u}\big)$$
$$m_{3c} = res_\kappa^{t_u} \oplus H\big(GID_z \| t_u \| RID_\kappa \| res_\kappa^{t_i} \| s_{t_u} \| che_\kappa^{t_u}\big)$$
$$m_{3d} = H\big(GID_z \| t_u \| RID_\kappa \| res_\kappa^{t_i} \| s_{t_u} \| che_\kappa^{t_u} \| res_\kappa^{t_u} \| PID_\kappa^{t_u}\big).$$

Here, m_{3a}, m_{3b}, and m_{3c} are used to share s_{t_u}, $che_\kappa^{t_u}$, and $res_\kappa^{t_u}$ with the ground station G_z, respectively. m_{3d} can help the ground station G_z verify the integrity of s_{t_u}, $che_\kappa^{t_u}$, and $res_\kappa^{t_u}$.

7) The drone D_κ sends the message $M_3 = \{GID_z, t_u, PID_\kappa^{t_i}, m_{3a}, m_{3b}, m_{3c}, m_{3d}\}$ to the ground station G_z via an insecure channel, updates its PUF CRP, and then calculates the secret session key $SK_{\kappa,z}^{dt_x,t_u}$ for the type dt_x data

$$SK_{\kappa,z}^{dt_x,t_u} = H\big(s_{t_u}\big) \oplus H\Big(s'_{t_p}\Big) \oplus H\big(res_\kappa^{t_u}\big) \oplus H(dt_x).$$

With two random numbers as well as the PUF response and the data type, the drone D_κ calculates a data type-specific secret session key with the ground station G_z.

8) The ground station G_z retrieves the time t_u, and compares it with the current system time t_{cur}. If the difference is larger than or equal to a threshold t^Δ, $(t_{cur} - t_u) \geq t^\Delta$, the message M_3 is rejected. Otherwise, the ground station G_z calculates the following:

$$s'_{t_u} = m'_{3a} \oplus H\big(GID_z \| t_u \| RID_\kappa \| res_\kappa^{t_i}\big)$$
$$che'^{t_u}_\kappa = m'_{3b} \oplus H\big(GID_z \| t_u \| RID_\kappa \| res_\kappa^{t_i} \| s'_{t_u}\big)$$
$$res'^{t_u}_\kappa = m'_{3c} \oplus H\big(GID_z \| t_u \| RID_\kappa \| res_\kappa^{t_i} \| s'_{t_u} \| che'^{t_u}_\kappa\big)$$
$$PID'^{t_u}_\kappa = H\big(RID_\kappa \| res'^{t_u}_\kappa\big)$$
$$m'_{3d} = H\big(GID_z \| t_u \| RID_\kappa \| res_\kappa^{t_i} \| s'_{t_u} \| che'^{t_u}_\kappa \| res'^{t_u}_\kappa \| PID'^{t_u}_\kappa.$$

Through the above calculations, the ground station G_z can restore s'_{t_u}, $che'^{t_u}_\kappa$, $res'^{t_u}_\kappa$, and $PID'^{t_u}_\kappa$, and verify their integrity accordingly. If $m'_{3d} \neq m_{3d}$, the message M_3 is rejected and the authentication process fails. Otherwise, the ground station G_z calculates the secret session key $SK_{\kappa,z}^{dt_x,t_u}$ for the type dt_x data

$$SK_{\kappa,z}^{dt_x,t_u} = H\big(s_{t_p}\big) \oplus H\big(s'_{t_u}\big) \oplus H\big(res'^{t_u}_\kappa\big) \oplus H(dt_x)$$

and updates the drone D_κ's pseudonym and PUF CRP. Using the same random numbers as well as the PUF response and assigned data type of the drone D_κ, the ground station G_z can calculate an identical data type-specific secret session key as the drone D_κ did.

By this time, the mutual authentication between the drone D_κ and the ground station G_z has finally succeeded and the

Algorithm 4: Authentication Initialization Algorithm

/* $SendMessage(src, des, msg)$: source src sends message msg to destination des */

Function DroneRequestAuth(RID_κ, $che_\kappa^{t_i}$, dt_x):
- $res_\kappa^{t_i} \leftarrow F_{puf}(che_\kappa^{t_i})$;
- $PID_\kappa^{t_i} \leftarrow H(RID_\kappa \parallel res_\kappa^{t_i})$;
- $r_{t_j} \leftarrow RandNum(t_j)$;
- $m_{1a} \leftarrow r_{t_j} \oplus H(GID_z \parallel t_j \parallel RID_\kappa \parallel res_\kappa^{t_i})$;
- $m_{1b} \leftarrow dt_x \oplus H(GID_z \parallel t_j \parallel RID_\kappa \parallel res_\kappa^{t_i} \parallel r_{t_j})$;
- $m_{1c} \leftarrow H(GID_z \parallel t_j \parallel RID_\kappa \parallel res_\kappa^{t_i} \parallel r_{t_j} \parallel dt_x)$;
- $M_1 \leftarrow \{GID_z, t_j, PID_\kappa^{t_i}, m_{1a}, m_{1b}, m_{1c}\}$;
- $SendMessage(D_\kappa, CC, M_1)$;

Function GoundReceiveAuth(M_1):
- **if** $(t_{cur} - t_j) \geq t^\Delta$ **then**
  - reject;
- **else**
  - $r'_{t_j} \leftarrow m'_{1a} \oplus H(GID_z \parallel t_j \parallel RID_\kappa \parallel res_\kappa^{t_i})$;
  - $dt'_x \leftarrow m'_{1b} \oplus H(GID_z \parallel t_j \parallel RID_\kappa \parallel res_\kappa^{t_i} \parallel r'_{t_j})$;
  - $m'_{1c} \leftarrow H(GID_z \parallel t_j \parallel RID_\kappa \parallel res_\kappa^{t_i} \parallel r'_{t_j} \parallel dt'_x)$;
  - **if** $(m'_{1c} \neq m_{1c})$ **then**
    - reject;
  - **else**
    - **if** $(dt'_x \notin DT_\kappa)$ **then**
      - reject;
    - **else**
      - $s_{t_p} \leftarrow RandNum(t_p)$;
      - $m_{2a} \leftarrow s_{t_p} \oplus H(RID_\kappa \parallel res_\kappa^{t_i} \parallel r'_{t_j} \parallel t_p \parallel GID_z)$;
      - $m_{2b} \leftarrow H(RID_\kappa \parallel res_\kappa^{t_i} \parallel r'_{t_j} \parallel t_p \parallel GID_z \parallel s_{t_p})$;
      - $M_2 \leftarrow \{PID_\kappa^{t_i}, t_p, GID_z, m_{2a}, m_{2b}\}$;
      - $SendMessage(CC, D_\kappa, M_2)$;
    - **end**
  - **end**
- **end**

Algorithm 5: Authentication Completion Algorithm

/* $update(\cdots)$: update stored information */

Function DroneCompleteAuth(M_2):
- **if** $(t_{cur} - t_p) \geq t^\Delta$ **then**
  - reject;
- **else**
  - $s'_{t_p} \leftarrow m'_{2a} \oplus H(RID_\kappa \parallel res_\kappa^{t_i} \parallel r_{t_j} \parallel t_p \parallel GID_z)$;
  - $m'_{2b} \leftarrow H(RID_\kappa \parallel res_\kappa^{t_i} \parallel r_{t_j} \parallel t_p \parallel GID_z \parallel s'_{t_p})$;
  - **if** $(m'_{2b} \neq m_{2b})$ **then**
    - reject;
  - **else**
    - $s_{t_u} \leftarrow RandNum(t_u)$;
    - $che_\kappa^{t_u} \leftarrow H(s_{t_u} \parallel s'_{t_p})$;
    - $res_\kappa^{t_u} \leftarrow F_{puf}(che_\kappa^{t_u})$;
    - $PID_\kappa^{t_u} \leftarrow H(RID_\kappa \parallel res_\kappa^{t_u})$;
    - $m_{3a} \leftarrow s_{t_u} \oplus H(GID_z \parallel t_u \parallel RID_\kappa \parallel res_\kappa^{t_i})$;
    - $m_{3b} \leftarrow che_\kappa^{t_u} \oplus H(GID_z \parallel t_u \parallel RID_\kappa \parallel res_\kappa^{t_i} \parallel s_{t_u})$;
    - $m_{3c} \leftarrow res_\kappa^{t_u} \oplus H(GID_z \parallel t_u \parallel RID_\kappa \parallel res_\kappa^{t_i} \parallel s_{t_u} \parallel che_\kappa^{t_u})$;
    - $m_{3d} \leftarrow H(GID_z \parallel t_u \parallel RID_\kappa \parallel res_\kappa^{t_i} \parallel s_{t_u} \parallel che_\kappa^{t_u} \parallel res_\kappa^{t_u} \parallel PID_\kappa^{t_u})$;
    - $M_3 \leftarrow \{GID_z, t_u, PID_\kappa^{t_i}, m_{3a}, m_{3b}, m_{3c}, m_{3d}\}$;
    - $SendMessage(D_\kappa, CC, M_3)$;
    - $update(che_\kappa^{t_u})$;
    - $SK_{\kappa,z}^{dt_x,t_u} \leftarrow H(s_{t_u}) \oplus H(s'_{t_p}) \oplus H(res_\kappa^{t_u}) \oplus H(dt_x)$;
  - **end**
- **end**

Function GroundCompleteAuth(M_3):
- **if** $(t_{cur} - t_u) \geq t^\Delta$ **then**
  - reject;
- **else**
  - $s'_{t_u} \leftarrow m'_{3a} \oplus H(GID_z \parallel t_u \parallel RID_\kappa \parallel res_\kappa^{t_i})$;
  - $che'^{t_u}_\kappa \leftarrow m'_{3b} \oplus H(GID_z \parallel t_u \parallel RID_\kappa \parallel res_\kappa^{t_i} \parallel s'_{t_u})$;
  - $res'^{t_u}_\kappa \leftarrow m'_{3c} \oplus H(GID_z \parallel t_u \parallel RID_\kappa \parallel res_\kappa^{t_i} \parallel s'_{t_u} \parallel che'^{t_u}_\kappa)$;
  - $PID'^{t_u}_\kappa \leftarrow H(RID_\kappa \parallel res'^{t_u}_\kappa)$;
  - $m'_{3d} \leftarrow H(GID_z \parallel t_u \parallel RID_\kappa \parallel res_\kappa^{t_i} \parallel s'_{t_u} \parallel che'^{t_u}_\kappa \parallel res'^{t_u}_\kappa \parallel PID'^{t_u}_\kappa$;
  - **if** $(m'_{3d} \neq m_{3d})$ **then**
    - reject;
  - **else**
    - $update(che'^{t_u}_\kappa, res'^{t_u}_\kappa, PID'^{t_u}_\kappa)$;
    - $SK_{\kappa,z}^{dt_x,t_u} \leftarrow H(s_{t_p}) \oplus H(s'_{t_u}) \oplus H(res'^{t_u}_\kappa) \oplus H(dt_x)$;
  - **end**
- **end**

secret session key $SK_{\kappa,z}^{dt_x,t_u}$ for the type dt_x data has been successfully established for the subsequent communications. It is worth mentioning that the drone D_κ's CRP (as well as its pseudonym) has been updated after the establishment of authenticated session to reduce the risk of the adversary compromising the CRP through brute force. The major operations of authentication and key establishment phase are summarized in Algorithms 4 and 5, respectively.

V. Security Verification and Analysis

In this section, we mainly focus on the security verification of *liteA4*, and intend to prove that *liteA4* can safely operate in an adversarial environment. In addition, we demonstrate formally and informally that the secret information of *liteA4* can be securely exchanged between communication entities, and *liteA4* is immune against cyber attacks.

A. Security Verification

In this section, AVISPA [15], which is a widely used Internet security protocol verification tool, is adopted to assess the security properties of *liteA4*. The objective of this security verification is to prove that *liteA4* has no design flaws related to security operations, and can be executed properly in adversarial environments. In order to evaluate security protocols on AVISPA, *liteA4* has to be first implemented in HLPSL, which is known as HLPSL. In addition, AVISPA offers us verification components, On-the-fly Model Checker (OFMC) and Constraint-Logic-based Attack Searcher (CL-AtSe), with which we can test the security performance and features of *liteA4*. Here, OFMC is useful for examining security features of *liteA4*, namely, authenticity, confidentiality, and integrity, while CL-AtSe is appropriate for vulnerability assessment along with threat modeling. In the HLPSL implementation of *liteA4*, communication and message exchange are realized between two roles which are drone and ground station. Moreover, four auxiliary roles which are required by AVISPA are also implemented; they are intruder, goal, session, and environment. We build up an experimental environment on Ubuntu 10.04, where AVISPA [42] is properly installed and configured in Virtual Box [43]. The results of security verification obtained through HLPSL program execution on AVISPA are given in Fig. 4. As expected, *liteA4* is a safe security protocol without design flaws or

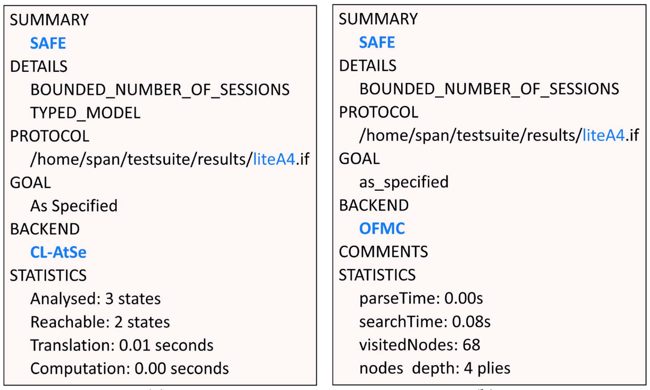

Fig. 4. Security verification results of *liteA4* from AVISPA.

vulnerabilities which can be exploited by adversary. The HLPSL security verification programs are publicly available at https://github.com/congpu/liteA4.

B. Formal Security Analysis

In this section, we exhibit the process of formal security analysis of *liteA4* based on Mao's and Boyd's logic [44]. The objective of this formal security analysis is to show that the secret information cannot be compromised by adversary, and access to these secret information is only authorized and granted to drone D_κ and ground station G_z. In other words, we attempt to theoretically affirm that $res_\kappa^{t_i}$ is presented to be a good shared secret between drone D_κ and ground station G_z, and cannot be accessed, acquired, or manipulated by an adversary in any fashion whatsoever. First, according to Mao's and Boyd's logic a group of inference rules for reasoning about logical formulas are presented. Second, we describe a sequence of initial assumptions which are reasonable beliefs, whereas communication events required by *liteA4* can be satisfied.

1) $D_\kappa \models D_\kappa \overset{(che_\kappa^{t_i}, res_\kappa^{t_i})}{\longleftrightarrow} G_z$ *and* $G_z \models G_z \overset{(che_\kappa^{t_i}, res_\kappa^{t_i})}{\longleftrightarrow} D_k$*:* The initial CRP $(che_\kappa^{t_i}, res_\kappa^{t_i})$ of drone D_k is securely shared between drone D_k and ground station G_z.
2) $D_\kappa \models G_z \lhd \| D_\kappa$*:* The real identify of drone D_κ is known to the ground station G_z.
3) $D_\kappa \models D_\kappa \overset{PID_\kappa^{t_i}}{\longleftrightarrow} G_z$ *and* $G_z \models G_z \overset{PID_\kappa^{t_i}}{\longleftrightarrow} D_\kappa$: Ground station G_z saves drone D_κ's pseudonym in its database, whereas drone D_κ is able to compute its $PID_\kappa^{t_i}$ using its real identify and CRP $(che_\kappa^{t_i}, res_\kappa^{t_i})$.
4) $D_\kappa \models G_z \lhd \| res_\kappa^{t_i}$ *and* $G_z \models D_\kappa \models \{G_z\} \lhd \| res_\kappa^{t_i}$*:* Drone D_κ generates a new $res_\kappa^{t_i}$ each time.
5) $G_z \models$ *sup* (D_κ)*:* Drone D_κ is the super-principal to ground station G_z.
6) $D_\kappa \models \#\ (res_\kappa^{t_i})$*:* Drone D_κ generates a fresh $res_\kappa^{t_i}$ each time.
7) $D_\kappa \models \#\ (r'_{t_j})$: Drone ID_i generates a fresh r'_{t_j} each time.
8) $D_\kappa \models \#\ (s'_{t_u})$*:* Drone ID_i generates a fresh s'_{t_u} each time.
9) $G_z \models \#\ (s'_{t_p})$*:* Ground station G_z generates a fresh s'_{t_p} each time.
10) $D_\kappa \overset{(che_\kappa^{t_i}, res_\kappa^{t_i})}{\boxplus} r'_{t_j}$*:* Drone D_κ encrypts the message M_1 piggybacked with r'_{t_j} using its CRP $(che_\kappa^{t_i}, res_\kappa^{t_i})$.
11) $G_z \overset{(che_\kappa^{t_i}, res_\kappa^{t_i})}{\lhd} r'_{t_j}$*:* Ground station G_z decrypts the encrypted message M_1 using drone D_κ's CRP $(che_\kappa^{t_i}, res_\kappa^{t_i})$.
12) $G_z \overset{(che_\kappa^{t_i}, res_\kappa^{t_i})}{\boxplus} s'_{t_p}$*:* Ground station G_z encrypts the message M_2 piggybacked with s'_{t_p} using drone D_κ's CRP $(che_\kappa^{t_i}, res_\kappa^{t_i})$.
13) $D_\kappa \overset{(che_\kappa^{t_i}, res_\kappa^{t_i})}{\lhd} res_\kappa^{t_i} \Re s'_{t_p}$*:* Drone D_κ decrypts the encrypted message M_2 using its CRP $(che_\kappa^{t_i}, res_\kappa^{t_i})$.
14) $D_\kappa \overset{(che_\kappa^{t_i}, res_\kappa^{t_i})}{\boxplus} s'_{t_u}$*:* Drone D_κ encrypts the message M_3 piggybacked with s'_{t_p} using its CRP $(che_\kappa^{t_i}, res_\kappa^{t_i})$.
15) $G_z \overset{(che_\kappa^{t_i}, res_\kappa^{t_i})}{\lhd} s'_{t_p} \Re res_\kappa^{t_i}$*:* Ground station G_z decrypts the encrypted message M_3 using drone D_κ's CRP $((che_\kappa^{t_i}, res_\kappa^{t_i})$, respectively.

Fig. 5 provides a detailed view of formal security analysis of *liteA4*. Our initial assertion that drone D_κ and ground station G_z are the only two communication entities who are authorized to access secret information $res_\kappa^{t_i}$, is formally proved via continuously applying inference rules. For example, Fig. 5(b) shows that secret information $res_\kappa^{t_i}$ is a good shared value between drone D_κ and ground station G_z, where we first place the statement $D_\kappa \models D_\kappa \overset{\mathrm{res}_\kappa^{t_i}}{\longleftrightarrow} G_z$ at the end of the logical construct. Thereafter, we apply the Good Key rule to the specified statement indicating whether D_κ believes that secret information $res_\kappa^{t_i}$ is only available to drone D_κ and ground station G_z (i.e., $D_\kappa \models \{D_\kappa, G_z\} \lhd\| res_\kappa^{t_i}$). Since drone D_κ knows that secret information $res_\kappa^{t_i}$ is fresh (i.e., $D_k \models \#(res_\kappa^{t_i})$, as a result, it believes that secret information $res_\kappa^{t_i}$ is a good shared secret between itself and ground station G_z. Next, the Confidentiality rule is applied to prove $D_\kappa \models \{D_\kappa, G_z\} \lhd \| res_\kappa^{t_i}$, which further demonstrates that $(che_\kappa^{t_i}, res_\kappa^{t_i})$ is only shared between drone D_κ and ground station G_z (i.e., $D_\kappa \models D_\kappa \overset{(\mathrm{che}_\kappa^{t_i}, \mathrm{res}_\kappa^{t_i})}{\longleftrightarrow} G_z$). Moreover, we can easily observe the fact that drone D_κ sends $(che_\kappa^{t_i}, res_\kappa^{t_i})$ to ground station G_z without sharing with anyone else (i.e., $D_\kappa \models G_z \lhd \| res_\kappa^{t_i}$), and drone D_κ perform encryption with $res_\kappa^{t_i}$ (i.e., $D_\kappa \overset{(\mathrm{che}_\kappa^{t_i}, \mathrm{res}_\kappa^{t_i})}{\boxplus} res_\kappa^{t_i}$). These statements are clearly defined in the initial assumptions, so the claim that secret information $res_\kappa^{t_i}$ is only shared between drone D_κ and ground station G_z is proved. Likewise, the security claim in Fig. 5(a), which states that ground station G_z believes secret information $res_\kappa^{t_i}$ is only shared between ground station G_z and drone D_k, is proved by following a similar approach.

Hence, the formal security analysis given in Fig. 5 assures that without prior knowledge of PUF CRP $(che_\kappa^{t_i}, res_\kappa^{t_i})$ an adversary would not be able to decipher messages and obtain secret information $res_\kappa^{t_i}$. Moreover, in the unlikely event when drone D_k is physically captured, the adversary would still not be able to obtain its PUF CRP $(che_\kappa^{t_i}, res_\kappa^{t_i})$, as drone D_k does not store its PUF CRP in the memory. Last but not least, any physical attack that attempts to alter drone D_k's circuit to retrieve the initial PUF CRP would only lead to the destruction of PUF. In conclusion, the secret information in *liteA4* is secure and protected.

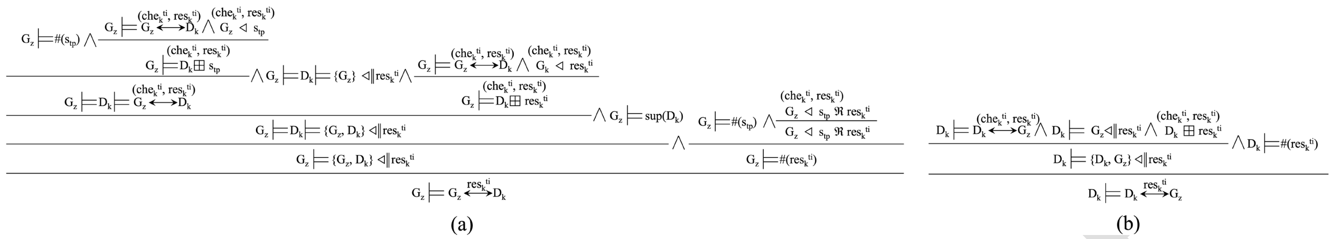

Fig. 5. Formal security analysis of *liteA4*. (a) Proof that ground station G_z believes that secure information $res_{\kappa}^{t_i}$ is only shared between drone D_{κ} and itself. (b) Proof that drone D_{κ} believes that only ground station G_z and itself can access secret information $res_{\kappa}^{t_i}$.

C. Informal Security Analysis

In this section, we analyze the operations of *liteA4* with the consideration of various cyber attacks such as replay attack, known session key attack, physical capture attack, message fabrication attack, ground station, and drone impersonation attacks, and demonstrate that *liteA4* is immune against them.

1) Replay Attack: In *liteA4*, both ground station and drone piggyback current system time (e.g., t_j, t_p, and t_u) in the messages (e.g., M_1, M_2, and M_3). Upon receiving a message, the receiver first verifies the freshness of message through checking the piggybacked system timestamp. If the piggybacked timestamp is indeed obsolete, the receiver will directly discard the message. Otherwise, the receiver will proceed with the following operations, e.g., verifying the authenticity of the message. Hence, *liteA4* is resilient against replay attacks.

2) Known Session Key Attack: We assume that the adversary is aware of the session key $SK_{\kappa,z}^{dt_x,t_u}$ negotiated between drone D_{κ} and ground station G_z for a past communication session. The session key $SK_{\kappa,z}^{dt_x,t_u}$ is calculated through the exclusive OR operations among four values, which are two random numbers (e.g., s_{t_p}, s_{t_u}), PUF response (e.g., $res_{\kappa}^{t_u}$), and data type (e.g., dt_x). Even though the adversary has a copy of session key $SK_{\kappa,z}^{dt_x,t_u}$, it cannot retrieve either of these four values and predict any future session keys. This is because it is infeasible to regenerate the same hash value without knowing the valid input. Thus, *liteA4* is protected against known session key attack.

3) Physical Capture Attack: Suppose that the adversary has successfully seized drone D_{κ} that had established a session key with ground station G_z before. Through power analysis attack, the adversary might retrieve the information stored in drone D_{κ}'s memory, e.g., identification, PUF challenge, registered data type, and session key. However, when the adversary attempts to restore drone D_{κ}'s PUF response, its effort leads to no end. This is because the power analysis attack will cause a slightest modification to the integrated circuit of drone D_{κ}, which will change or even destroy drone D_{κ}'s PUF. In addition, the adversary can only jeopardize the current communication session between drone D_{κ} and ground station G_z. Nevertheless, the data exchange between other drones and ground station G_z is still safe because other drones will negotiate session keys with ground station G_z with their unique cryptographic information. As a result, other noncaptured drones are still safe from the adversary. Therefore, *liteA4* is not impacted by physical capture attack.

4) Message Fabrication Attack: In *liteA4*, the receiver always verifies the authenticity of message through comparing the recalculated message with the received message (e.g., $m'_{1c} = m_{1c}$). If the received message passes the verification, it is believed to be authentic and the following operations of *liteA4* continues as normal. Otherwise, the receiver will directly destroy the message. Hence, *liteA4* is secure against message fabrication attack.

5) Ground Station/Drone Impersonation Attacks: Suppose that the adversary pretends to be ground station G_z. In order to establish communication with a legitimate drone, the adversary needs to generate a random number s_{t_p}, calculate message M_2 piggybacked with random number r_{t_j} from message M_1, and then send it to drone D_{κ}. However, the adversary cannot decrypt message M_1 to retrieve random number r_{t_j}. Thus, the adversary has to arbitrarily generate random number r'_{t_j}. Upon receiving message M_2, drone D_{κ} recalculates m'_{2b} and checks if $m'_{2b} = m_{2b}$. Since the adversary randomly generate random number r'_{t_j}, drone D_{κ} can easily notice that message M_2 is fabricated, coming from an untrusted entity. Therefore, *liteA4* is resilient against ground station impersonation attack. The similar idea can be applied to prove that *liteA4* is also protected from drone impersonation attack.

D. Comparison of Security Requirements

The comparison of security requirements among *liteA4*, SLAP-IoD, and SAAF-IoD is provided in Table III. In essence, *liteA4* meets every predefined security requirement, outperforming its counterpart approaches.

VI. Performance Evaluation

A. Experimental Environment and Benchmarks

To conduct experimental study, we set up a Windows-based computing environment to evaluate and analyze the performance our approach *liteA4* and three benchmark schemes in terms of different tasks. The experimental machine has 16-GB memory and a 12th generation processor of 2.10 GHz, and runs Windows 11 operating system. Our approach *liteA4* and other three benchmark schemes, SLAP-IoD [17], SAAF-IoD [18], and PUF-IPA [19] are implemented in Python language within Visual Studio Code [45] programming environment. A brief summary highlighting the central idea of SLAP-IoD, SAAF-IoD, and PUF-IPA are given below:

1) SLAP-IoD: SLAP-IoD proposes an authentication scheme that is comprised of three entities: 1) a mobile user (MU_i); 2) a drone (D_j); and 3) a control server (CS). It has five phases: 1) initialization; 2) drone registration; 3) mobile user registration; 4) authentication and key agreement; and

TABLE III
COMPARISON OF SECURITY REQUIREMENTS

Security Requirements	liteA4	SLAP*	SAAF‡	IPA†
Auth. Between Drone and User◇	✓	✓	✓	✓
Integrity	✓	✓	✓	✓
Application Aware Authentication	✓	✘	✘	✘
Anonymity	✓	✓	✓	✓
Message Modification Attack	✓	✓	✓	✓
Session Key Agreement	✓	✓	✓	✘
Drone Capture Attack	✓	✓	✓	–
Impersonation Attack	✓	✓	✓	✓
Replay Attack	✓	✓	✓	✘
Ground Station Spoofing Attack	✓	–	✓	–
Known Session Key Attack	✓	✓	✓	–
Man-In-The-Middle Attack	✓	✓	✓	✓
Desynchronization Attack	✓	✓	✓	✘

*: SLAP represents SLAP-IoD. ‡: SAAF represents SAAF-IoD.
†: IPA represents PUF-IPA.
◇: In *liteA4*, ground station G_z is equivalent to user.
✓indicates security requirement is met.
✘ indicates security requirement is not met.

TABLE IV
COMPARISON OF COMMUNICATION OVERHEAD⋆

Metrics	liteA4	SLAP*	SAAF‡	IPA†
Number of Msg.‖	150	200	150	200
Size of Msg. (KB)‡	24	27.20	24.4	9.8
Energy Cons. (J)◇	17×10^{-3}	23×10^{-3}	17×10^{-3}	23×10^{-3}

*: SLAP represents SLAP-IoD. ‡: SAAF represents SAAF-IoD.
†: IPA represents PUF-IPA.
⋆: In this experiment, we consider 50 drones in the network.
‖: The number of exchanged messages are retrieved from the communication sequence diagrams provided by *liteA4*, SLAP-IoD, SAAF-IoD, and PUF-IPA .
‡: The cumulative size of exchanged messages are calculated based on the real implementation of *liteA4*, SLAP-IoD, SAAF-IoD, and PUF-IPA.
◇: The energy consumption of communication is calculated as multiplying the number of exchanged messages by the energy consumption of exchanging one message [46].

5) password and biometric update. During the registration process, control server *CS* chooses a master key and assigns parameters to authenticate drone D_j before being positioned in its task zone. Control server *CS* also publishes necessary public parameters like fuzzy extractors and PUF. In the drone registration phase, drone D_j receives its credentials and registers with control server *CS*. Likewise, in the mobile user registration phase, mobile user MU_i receives its credentials and registers with control server *CS*. Then, mobile user MU_i and drone D_j mutually authenticate each other and establish a session key in the authentication and key agreement phase. In addition, mobile user MU_i can update his/her biometric credentials in the password update phase.

2) SAAF-IoD: SAAF-IoD proposes an authentication scheme which adopts chaotic mapping along with symmetric AES encryption. It comprises of five phases: 1) ground station enrollment; 2) drone enrollment; 3) user enrollment; 4) drone access; and 5) secret credential update. During the ground station enrollment phase, the drone service provider selects a secret key and an identifier for the ground station. Similarly, the drone service provider chooses an identifier and a secret key for a given drone in the drone enrollment phase. In the user enrollment phase, user U_i is registered with the ground station via a two-step approach: 1) the smart reader device sends secret credentials to the ground station and receives parameters in return and 2) the smart reader device performs computations with the received information and stores results in its memory. In the drone access phase, user U_i mutually authenticates with drone D_j and sets up a session key. In the last phase, user U_i can change his/her secret credentials such as biometric information.

3) PUF-IPA: PUF-IPA proposes an authentication scheme for the IoT environment, aiming to improve the PUF response accuracy without using any error correction codes. It is comprised of two phases: 1) enrollment phase and 2) authentication phase. During the enrollment phase, various cryptographically secure random numbers are generated, and different hashed values are encrypted to be stored in a database. In the authentication phase, the server initiates the authentication request, to validate every device in the network. Moreover, PUF-IPA offers shuffling and deshuffling operations that is performed during enrollment and authentication, respectively, for added security.

We analyze the performance of *liteA4*, SLAP-IoD, SAAF-IoD, and PUF-IPA, and gather results on their associated communication overhead, running time, CPU time, storage overhead, as well as energy consumption by altering the number of executed algorithms and the number of drones in the system. The communication overhead gives information regarding the number of exchanged messages, the size of exchanged messages, and the amount of energy consumed by exchanging those messages. The running time measures the real elapsed time from when a protocol starts running to when it stops running. Likewise, the CPU time measures the amount of time spent by CPU executing all operations of each protocol. The storage overhead is the amount of memory space (RAM) required by the machine to run the protocol. Finally, the energy consumption denotes the amount of energy consumed due to the execution of protocol.

B. Experimental Results and Analysis

First, we measure the communication efficiency of *liteA4*, SLAP-IoD, SAAF-IoD, and PUF-IPA in terms of the number of exchanged messages, the size of exchanged messages, and the energy consumption of exchanging those messages in Table IV. Taking into consideration the communication sequence diagrams provided by *liteA4*, SLAP-IoD, SAAF-IoD, and PUF-IPA, we directly count the number of exchanged messages needed for a single drone scenario, and then calculate the total number of exchanged messages for 50 drones in the network. For instance, *liteA4* requires an authentication request message to be sent from a drone to a ground station. Next, the ground station sends an authentication response message to the drone. Finally, the drone responds by sending an authentication confirmation message. In total, three messages are needed by *liteA4* for a single drone scenario. For 50 drones in the network, *liteA4* would require a total of 150 messages. In SLAP-IoD, the first message piggybacked with drone's

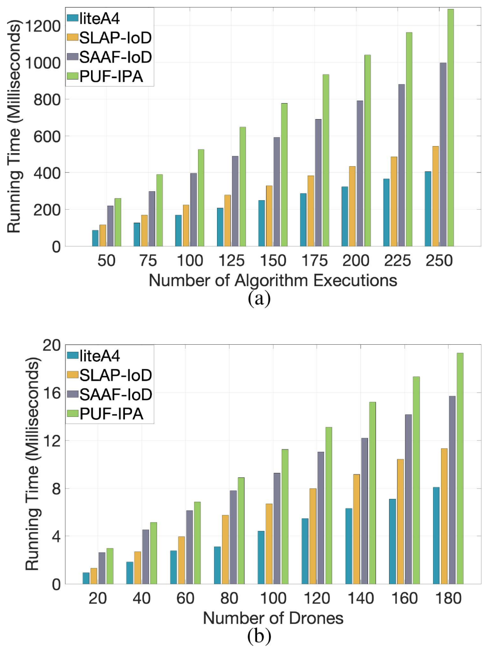

Fig. 6. Running time versus the number of algorithm executions and the number of drones.

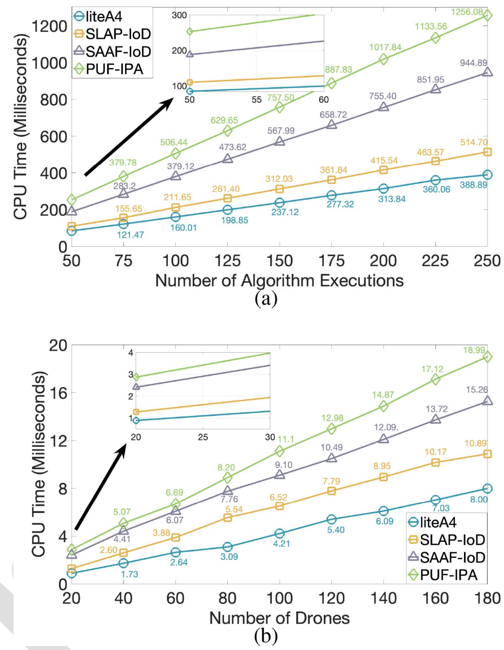

Fig. 7. CPU time versus the number of algorithm executions and the number of drones.

real identity and timestamp is sent to the CS. The CS then checks for the freshness of the message and replies a message back to the drone. After receiving the response from the CS, the drone validates the message and sends the third message to the CS. Finally, the CS receives the message, checks for the freshness, and sends the last message to the mobile user. Thus, a total of four messages are required by SLAP-IoD to authenticate a single drone and a mobile user. If there are 50 drones, 200 messages would be generated and exchanged in the network. Similarly, SAAF-IoD would require a total of 150 messages, since it requires three messages for a single drone scenario. Lastly, PUF-IPA requires four messages for a single authentication session. Hence, it would need a total of 200 messages for 50 devices. Moreover, the size of exchanged messages are 24 kB, 27.2 kB, 24.4 kB, 9.8 kB for liteA4, SLAP-IoD, SAAF-IoD, and PUF-IPA, respectively. The reason PUF-IPA has such a small size for exchanged messages is because it sends a minimal amount of message but stores all relevant values in its database. The results are obtained from the real implementation of each protocol. Finally, the energy consumption is calculated based on the number of exchanged messages and the energy consumption of exchanging one message [46]. SLAP-IoD, and PUF-IPA consume more energy than liteA4 and SAAF-IoD because they exchange a larger number of messages. liteA4 and SAAF-IoD consume the same amount of energy because they exchange the same number of messages for 50 drones in the network.

Second, we obtain the running time of *liteA4*, SLAP-IoD, SAAF-IoD, and PUF-IPA by varying the number of algorithm executions in Fig. 6(a). Overall, the running time of all protocols increase in a linear fashion when the number of algorithm executions is increased from 50 to 250. The running time for our protocol *liteA4* is the least because it employs lightweight techniques such as bitwise XOR in conjunction with PUF and hash function. SLAP-IoD also utilizes bitwise XOR along with one-way hash function. However, it has to retrieve its stored secret credentials after each message to verify the authenticity of messages. In addition, SLAP-IoD also requires supplementary steps involving the usage of cryptographic operations before generating its session key. These operations result in a higher running time in SLAP-IoD. SAAF-IoD has a higher running time compared to two protocols. This is because SAAF-IoD applies AES encryption after calculating its secret key with chaotic map. Subsequently each message has to be decrypted by the receiver to ensure integrity. As a result, this will cause a longer running time as seen in Fig. 6(a). PUF-IPA has the highest running time out of all the protocols. Similar to SAAF-IoD, it utilizes AES encryption, and has to decrypt multiple values stored in its database. This involves retrieving the entire row stored in the database, significantly increasing overall run time. Likewise, the running time of *liteA4*, SLAP-IoD, SAAF-IoD, and PUF-IPA against varying number of drones ranging from 20 to 180 are shown in Fig. 6(b). It is obvious that the running time of all three protocols increase progressively as the number of drones is increased in the network. However, our protocol *liteA4* still outperforms SLAP-IoD, SAAF-IoD, and PUF-IPA.

Third, we evaluate the CPU time of *liteA4*, SLAP-IoD, SAAF-IoD, and PUF-IPA by changing the number of algorithm executions and the number of drones in the network in Fig. 7. The CPU time represents the amount of time taken by the CPU to execute the algorithm. When increasing the number of algorithm executions from 50 to 250, the CPU time of all three protocols increase linearly. This is because multiple algorithm executions result in a longer CPU time. The CPU time of PUF-IPA is observed to be the highest. This is because the scheme has to retrieve a row of stored secret values, and then

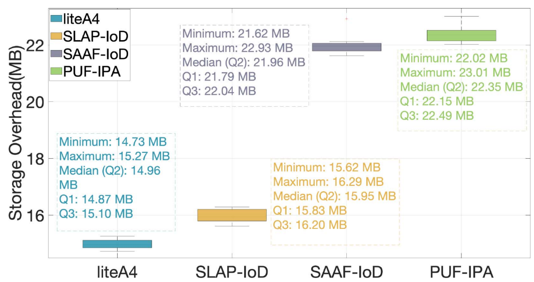

Fig. 8. Storage overhead.

decrypt them to send it to the receiving entity. SAAF-IoD also has a higher CPU time since decrypting each received cipher message and calculating encryption key require a considerable amount of CPU time, especially during multiple algorithm iterations. SLAP-IoD has a comparatively lower CPU time because of its lightweight operations, nonetheless it requires the retrieval of secret credentials which adds to its CPU time. *liteA4* outperforms other three protocols and achieves the lowest CPU time because of its optimized cryptographic operations. Similarly, the CPU time with a variable number of drones from 20 to 180 is observed in Fig. 7(b). *liteA4* attains the lowest CPU time due to its careful use of lightweight operations such as bitwise XOR, PUF, and hash functions. It shows to be a well-optimized protocol with good scalability when the number of drones is increased in the network.

Fourth, we examine the storage overhead associated with *liteA4*, SLAP-IoD, SAAF-IoD, and PUF-IPA in Fig. 8. The storage overhead represents the memory storage (RAM) allocated to each protocol. As observed in Fig. 8, PUF-IPA utilizes the largest amount of storage to run, while *liteA4* requires the least amount of storage to function. PUF-IPA encrypts the message, and then retrieves the stored secret while performing the necessary decryption, which consumes a lot of storage. Similarly, SAAF-IoD encrypts and decrypts each message, thus, it ends up consuming a significant amount of storage as well. On the other hand, drones in SLAP-IoD store their private secret credentials and retrieve them during authenticity check, which require more storage space. *liteA4* has the least amount of storage usage because it does not rely on storing secret credentials to verify message authenticity.

Finally, we inspect the energy consumption of *liteA4*, SLAP-IoD, SAAF-IoD, and PUF-IPA by varying the number of algorithm executions and the number of drones in Fig. 9. PUF-IPA is the most complex protocol as it utilizes AES encryption along with shuffling and deshuffling algorithms. Likewise, SAAF-IoD employs convoluted techniques as well as biometric updates and chaotic mapping mechanisms. Thus, it consumes more energy to execute all operations compared to *liteA4* and SLAP-IoD. Our protocol *liteA4* consumes the least amount of energy since it adopts recourse-friendly techniques such as bitwise XOR along with PUF and hash function. We also measure the running time of PUF with and without error by changing the number of algorithm executions in Fig. 10. When there are PUF errors, the running time for our protocol *liteA4* increases. The shaded area represents the difference in terms of running time incurred from unreliableness of PUF.

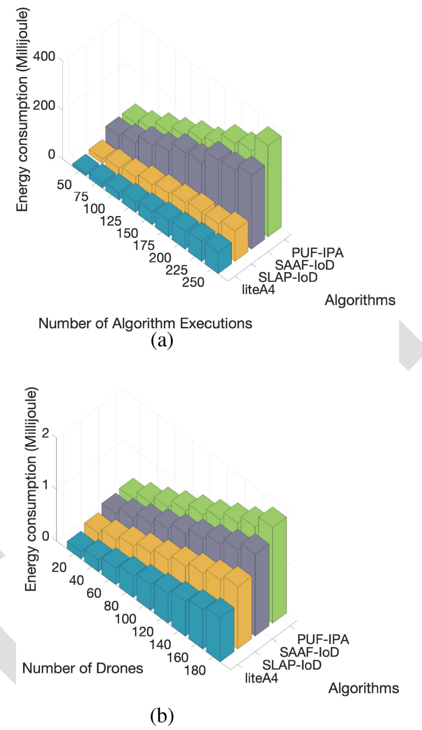

Fig. 9. Energy consumption versus the number of algorithm executions and the number of drones.

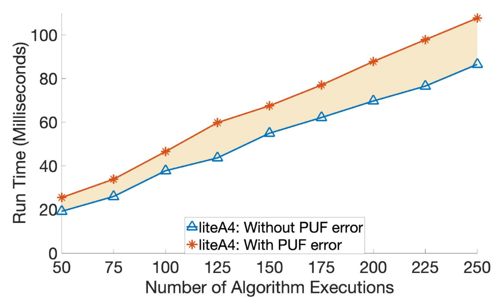

Fig. 10. Running time of PUF with and without error versus the number of algorithm executions.

VII. Conclusion

In this article, a lightweight and anonymous application-aware authentication and key agreement scheme (*liteA4*) was proposed for IoD systems, wherein a drone and a ground station perform data type-aware authentication and establish specific session key for the exchange of application-specific data. *liteA4* differentiates between different types of data, resulting in a more secure data exchange for drones being involved in multiple IoD applications concurrently. We evaluated *liteA4*'s security and resiliency by using AVISPA, and also demonstrated a formal and informal security analysis. Additionally, we conducted extensive experiments to evaluate the performance of *liteA4* in comparison with other three benchmark schemes. The experimental outcomes revealed that our protocol *liteA4* outperforms its peers without sacrificing any security prerequisites. As future work, we plan to integrate *liteA4* with consortium blockchain technique so that the ground stations can competitively and timely store the drone-collected data in the distributed data storage system.

References

[1] A. Ilangovan, S. Rajasekar, and V. Perumal, "CoVacciDrone: An algorithmic-drone-based COVID-19 vaccine distribution strategy," in *Internet of Drones*. Boca Raton, FL, USA: 2023, pp. 75–86.

[2] "UniSA working on pandemic drone' to detect coronavirus." Accessed: Nov. 2, 2023. [Online]. Available: https://www.unisa.edu.au/unisanews/2020/autumn/story11/

[3] (Drone Ind. Insights Co., Hamburg, Germany). *Drone Market Analysis 2022-2030*. Accessed: Nov. 1, 2023. [Online]. Available: https://droneii.com/drone-market-analysis-2022-2030

[4] J. Santulli, *IEEE Standard for Drone Applications Framework*, IEEE Standard 1936.1-2021, 2021.

[5] A. Zaki-Hindi, I. Kovács, R. Amorim, and J. Wigard, "Measurement reporting enhancement for 5G cellular-connected aerial vehicles," in *Proc. IEEE 34th Ann. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, 2023, pp. 1–6.

[6] *Requirements for Communication Services of Civilian Unmanned Aerial Vehicles*, ITU-Rec. F.749.10, Int. Telecommun. Union, Geneva, Switzerland, 2019.

[7] A. S. Abdalla and V. Marojevic, "Communications standards for unmanned aircraft systems: The 3GPP perspective and research drivers," *IEEE Commun. Stand. Mag.*, vol. 5, no. 1, pp. 70–77, Mar. 2021.

[8] J. Shin, M. J. Piran, H.-K. Song, and H. Moon, "UAV-assisted and deep learning-driven object detection and tracking for autonomous driving," in *Proc. 5th Int. ACM Mobicom Workshop Drone Assist. Wireless Commun. 5G Beyond*, 2022, pp. 7–12.

[9] A. Kriebitz, R. Max, and C. Lütge, "The German act on autonomous driving: Why ethics still matters," *Philosophy Technol.*, vol. 35, no. 2, pp. 1–13, 2022.

[10] E. Wisse, P. Tedeschi, S. Sciancalepore, and R. Di Pietro, "A 2RID—Anonymous direct authentication and remote identification of commercial drones," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10587–10604, Jun. 2023.

[11] B. D. Deebak and S. O. Hwang, "Intelligent drone-assisted robust lightweight multi-factor authentication for military zone surveillance in the 6G era," *Comput. Netw.*, vol. 225, Apr. 2023, Art. no. 109664.

[12] J. García, A. Benslimane, A. Braeken, and Z. Su, "μTesla-based authentication for reliable and secure broadcast communications in IoD using Blockchain," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 18400–18413, Oct. 2023.

[13] R. Vinoth, L. J. Deborah, P. Vijayakumar, and N. Kumar, "Secure multifactor authenticated key agreement scheme for industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3801–3811, Mar. 2021.

[14] C. Wang, R. Huang, J. Shen, J. Liu, P. Vijayakumar, and N. Kumar, "A novel lightweight authentication protocol for emergency vehicle avoidance in VANETs," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 14248–14257, Sep. 2021.

[15] "Automated validation of Internet security protocols and applications." 2006. [Online]. Available: http://www.avispa-project.org

[16] Y. Chevalier et al., "A high level protocol specification language for industrial security-sensitive protocols," in *Proc. SAPS*, 2004, pp. 1–13.

[17] S. Yu, A. K. Das, Y. Park, and P. Lorenz, "SLAP-IoD: Secure and lightweight authentication protocol using physical Unclonable functions for Internet of Drones in smart city environments," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10374–10388, Oct. 2022.

[18] M. Tanveer, H. Alasmary, N. Kumar, and A. Nayak, "SAAF-IoD: Secure and anonymous authentication framework for the Internet of Drones," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 232–244, Jan. 2024.

[19] M. A. Qureshi and A. Munir, "PUF-IPA: A PUF-based identity preserving protocol for Internet of Things authentication," in *Proc. 17th Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2020, pp. 1–7.

[20] P. Gope and T. Hwang, "A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7124–7132, Nov. 2016.

[21] Y. Zhang, D. He, L. Li, and B. Chen, "A lightweight authentication and key agreement scheme for Internet of Drones," *Comput. Commun.*, vol. 154, pp. 455–464, Mar. 2020.

[22] K. Lounis, S. H. H. Ding, and M. Zulkernine, "D2D-MAP: A drone to drone authentication protocol using physical Unclonable functions," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 5079–5093, Apr. 2023.

[23] N. Zhang, Q. Jiang, L. Li, X. Ma, and J. Ma, "An efficient three-factor remote user authentication protocol based on BPV-FourQ for internet of drones," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 5, pp. 3319–3332, 2021.

[24] M. El-Zawawy, A. Brighente, and M. Conti, "Authenticating drone-assisted Internet of Vehicles using elliptic curve cryptography and blockchain," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1775–1789, Jun. 2023.

[25] Y. Tan, J. Liu, and N. Kato, "Blockchain-based lightweight authentication for resilient UAV communications: Architecture, scheme, and future directions," *IEEE Wireless Commun.*, vol. 29, no. 3, pp. 24–31, Jun. 2022.

[26] J. Wang, L. Wu, H. Wang, K.-K. R. Choo, L. Wang, and D. He, "A secure and efficient multi-server authentication and key agreement protocol for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24398–24416, Dec. 2022.

[27] M. A. Abdel-Malek, K. Akkaya, A. Bhuyan, and A. S. Ibrahim, "A proxy signature-based swarm drone authentication with leader selection in 5G networks," *IEEE Access*, vol. 10, pp. 57485–57498, 2022.

[28] M. Tanveer, A. U. Khan, T. N. Nguyen, M. Ahmad, and A. A. A. El-Latif, "Towards a secure and computational framework for Internet of Drones enabled aerial computing," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 3058–3070, Sep./Oct. 2023.

[29] P. Gope, J. Lee, and T. Q. S. Quek, "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions," *IEEE Trans. Inf. Forensics Security*, vol. 13, pp. 2831–2843, 2018.

[30] M. N. Aman, M. H. Basheer, and B. Sikdar, "Data provenance for IoT with light weight authentication and privacy preservation," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10441–10457, Dec. 2019.

[31] Y. Chang, S. Huang, G. Chen, and W. Tai, "A critique of a lightweight authentication and key agreement scheme for Internet of Drones," in *Proc. SITAIBA*, 2023, pp. 337–346.

[32] M. Zhang, C. Xu, S. Li, and C. Jiang, "On the security of an ECC-based authentication scheme for Internet of Drones," *IEEE Syst. J.*, vol. 16, no. 4, pp. 6425–6428, Dec. 2022.

[33] 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 Syst. J.*, vol. 15, no. 3, pp. 4431–4438, Sep. 2021.

[34] "Internet-of-Drones: Novel applications, recent deployments and integration." Accessed: Oct. 30, 2023. [Online].Available: https://www.comsoc.org/publications/magazines/ieee-internet-things-magazine/cfp/internet-drones-novel-applications-recent

[35] (Ericsson, Stockholm, Sweden). *The Sky Is Not the Limit: The Past, Present, and Future of the Internet of Drones*. Accessed: Oct. 30, 2023. [Online].Available: https://www.ericsson.com/en/blog/2021/6/internet-of-drones-sky-is-not-the-limit

[36] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th ACM/IEEE DAC*, 2007, pp. 9–14.

[37] Q. Do, B. Martini, and K. K. R. Choo, "The role of the adversary model in applied security research," *Comput. Secur.*, vol. 81, pp. 156–181, Mar. 2019.

[38] A. Al-Meer and S. Al-Kuwari, "Physical unclonable functions (PUF) for IoT devices," *ACM Comput. Surv.*, vol. 55, no. 14, pp. 1–31, 2023.

[39] J. R. Wallrabenstein, "Practical and secure IoT device authentication using physical unclonable functions," in *Proc. IEEE FiCloud*, 2016, pp. 99–106.

[40] F. MacWilliams and N. Sloane, *The Theory of Error Correcting Codes*. Amsterdam, The Netherlands: Elsevier, 1977.

[41] D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "TOTP: Time-based one-time password algorithm," Internet Eng. Task Force, RFC 6238, 2011.

[42] "SPAN." Accessed: Oct. 8, 2023. [Online]. Available: http://people.irisa.fr/Thomas.Genet/span/

[43] "VirtualBox." Accessed: Oct. 8, 2023. [Online]. Available: https://www.virtualbox.org/

[44] W. Mao and C. Boyd, "Towards formal analysis of security protocols," in *Proc. Comput. Security Found. Workshop VI*, 1993, pp. 147–158.

[45] "VisualStudio." Accessed: Nov. 2, 2023. [Online]. Available: https://code.visualstudio.com/

[46] C. Pu and S. Lim, "A light-weight countermeasure to forwarding misbehavior in wireless sensor networks: Design, analysis, and evaluation," *IEEE Syst. J.*, vol. 12, no. 1, pp. 834–842, Mar. 2018.