Design and Analysis of Secure Lightweight Remote User Authentication and Key Agreement Scheme in Internet of Drones Deployment

Mohammad Wazid, *Student Member, IEEE*, Ashok Kumar Das, *Senior Member, IEEE*, Neeraj Kumar, *Senior Member, IEEE*, Athanasios V. Vasilakos, *Senior Member, IEEE*, and Joel J. P. C. Rodrigues, *Senior Member, IEEE*

***Abstract*—The Internet of Drones (IoD) provides a coordinated access to unmanned aerial vehicles that are referred as drones. The on-going miniaturization of sensors, actuators, and processors with ubiquitous wireless connectivity makes drones to be used in a wide range of applications ranging from military to civilian. Since most of the applications involved in the IoD are real-time based, the users are generally interested in accessing real-time information from drones belonging to a particular fly zone. This happens if we allow users to directly access real-time data from flying drones inside IoD environment and not from the server. This is a serious security breach which may deteriorate performance of any implemented solution in this IoD environment. To address this important issue in IoD, we propose a novel lightweight user authentication scheme in which a user in the IoD environment needs to access data directly from a drone provided that the user is authorized to access the data from that drone. The formal security verification using the broadly accepted automated validation of Internet security protocols and applications tool along with informal security analysis show that our scheme is secure against several known attacks. The performance comparison demonstrates that our scheme is efficient with respect to various parameters, and it provides better security as compared to those for the related existing schemes. Finally, the practical demonstration of our scheme is done using the widely accepted NS2 simulation.**

***Index Terms*—Automated validation of Internet security protocols and applications (AVISPA), Internet of Drones (IoD), NS2 simulation, security, user authentication.**

Manuscript received July 21, 2017; revised October 1, 2018 and November 6, 2018; accepted December 14, 2018. Date of publication December 19, 2018; date of current version May 8, 2019. This work was supported in part by the Finep/Funttel under the Radiocommunication Reference Center Project of the National Institute of Telecommunications, Brazil, under Grant 01.14.0231.00, in part by the National Funding from the Fundação para a Ciência e a Tecnologia under Project UID/EEA/500008/2013, and in part by the Brazilian National Council for Research and Development (CNPq) under Grant 309335/2017-5. *(Corresponding author: Joel J. P. C. Rodrigues.)*

M. Wazid is with the Cyber Security and Networks Laboratory, Innopolis University, Innopolis 420500, Russia (e-mail: wazidkec2005@gmail.com).

A. K. Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad 500 032, India (e-mail: iitkgp.akdas@gmail.com; ashok.das@iiit.ac.in).

N. Kumar is with the Department of Computer Science and Engineering, Thapar University, Patiala 147 004, India (e-mail: neeraj.kumar@thapar.edu).

A. V. Vasilakos is with the Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 971 87 Luleå, Sweden (e-mail: th.vasilakos@gmail.com).

J. J. P. C. Rodrigues is with the National Institute of Telecommunications (Inatel), Santa Rita do Sapucaí 37540-000, Brazil, with the Covilhã Delegation, Instituto de Telecomunicações, 1049-001 Lisbon, Portugal, and also with PPGIA, University of Fortaleza, Fortaleza 60811-905, Brazil (e-mail: joeljr@ieee.org).

Digital Object Identifier 10.1109/JIOT.2018.2888821

I. Introduction

The Internet of Drones (IoD) is a layered network control architecture designed mainly for coordinating the access of unmanned aerial vehicles (UAVs) to controlled airspace, and also for providing navigation services. IoD provides different services for drone applications, such as traffic surveillance, package delivery, search, and rescue [5]. IoD is becoming very popular day by day as it provides different types of services and applications, which further facilitate the life of people. An architecture of IoD is given Fig. 1, in which there are different types of entities, such as drones (UAVs), server, external users, and control room (internal user).

A general physical structure of a drone is given in Fig. 2. Each drone has computing power, recorder, energy supply, communication module, sensors, and actuators. Each drone has a defined fly zone in which it can fly and send the required information to the control room. The internal user seating in the control room controls the drone remotely. The inbuilt sensors in the drone send the physical phenomena, such as temperature and concentration of hazardous gases, and the inbuilt camera in the drone takes the photographs or captures videos of the target and sends all these information to the drone box via some wireless communication technology, such as WiFi. Each drone box is further connected to the server which is then connected to the control room.

The weight, model as well as energy source of a drone are typically the major components that impact its several factors, such as maximum altitude, flight range, and flight duration along with maximum payload [6]. The sensors are treated as a crucial category of payloads. Majority of drones are now equipped with cameras [6]. While buying a drone, the cameras and microphones are the most frequently used payloads for drones and these commonly come as standard [6]. Cameras can be also infrared and thus, such types of cameras may enable night vision as well as heat sensing too. Other sensors used in the drones include biological sensors

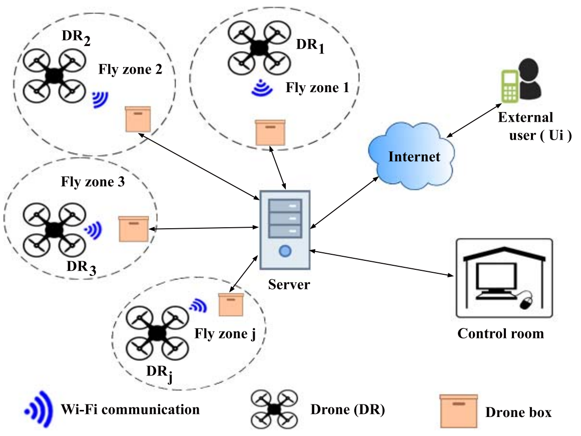


Fig. 1. IoD architecture (adapted from [1]–[4]).

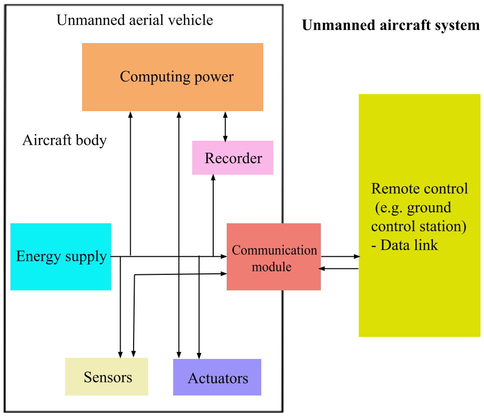


Fig. 2. General physical structure of an drone (adapted from [1], [4]).

that can detect microorganisms and meteorological sensors that can also measure various parameters, such as temperature, wind, and humidity. In addition, cameras can be utilized for payloads in order to apply for the prevention, criminal investigation, criminal prosecution, and also sentencing of criminal behavior [6]. Moreover, most applications in the IoD environment assume drones to be flying camera surveillance.

The key components of any drone are sensors. Only sensor systems with acceptable performance produce commercial drones the flight characteristics they need. In addition, the application chances of commercial UAVs are also decided by the sensors [7]. Various tasks require various sensors or rather different sensor combinations. Some of the sensor types applied in drones include the following [7].

1) Light detection and ranging that can be used for collision avoidance and navigation, and also for 3-D scanner.
2) Cameras that can be used for observation, collision avoidance, data acquisition, and navigation.
3) Pressure gauges that can be used for navigation and data collection.
4) Global positioning system that can be used for position determination.

In the given architecture in Fig. 1, suppose an external user who wants to access the data collected by the drones in a particular fly zone. Such an access is possible provided both user and accessed drone mutually authenticate each other with the help of the server. The server is considered as the trusted entity in the network, which means that the server will not be compromised by an attacker, whereas the drones can be even physically compromised by that attacker. Since the communication between the entities are wireless in nature, several security and privacy related issues arise in the IoD environment. Thus, various attacks including replay, man-in-the-middle, impersonation, privileged insider, and password guessing can be possible. Therefore, designing a secure authentication scheme is necessary in the IoD environment, while keeping the scheme to be lightweight, that is, it should be efficient in communication and computation overheads at the user, server as well as drone sides. A detailed survey on IoD can be found in [5] and [8].

A. *Motivation*

Gharibi *et al.* [5] pointed out that there are a variety of threats that must be safeguarded. Among the threats, authentication of drones and other components outside the IoD system, jamming of the broadcast messages, clogging the airspace, and hacking of the drones are the prominent ones. Most of the applications involved in the IoD are real-time based. Therefore, obviously the users (external parties) are generally interested in accessing the real-time information from the drones belonging to a particular fly zone. This happens if we allow the users to directly access the real-time data from the flying drones inside the IoD environment and not from the server. Usually, the information that is gathered by the server from the drones periodically, and as a result, the collected information may not be always real-time data. Therefore, to obtain the real-time information from the drones, the user (for example, driver of an ambulance) needs to access the data directly from an accessed drone provided that the user is authorized to access it from that drone in order to restrict unauthorized access of the data from the drones. On the basis of the received data from the accessed drones, the user can take important decision, such as the driver of the ambulance can choose the route which has less congestion in order to help that driver to save the life of a patient. This motivates the need of designing efficient and secure user authentication scheme in IoD environment, and as a result, it becomes a very important topic in research of IoD security.

B. *Related Work*

Hassanalian and Abdelkefi [9] provided a survey on classification of flying drones which ranges from unmanned air vehicles to smart dusts. Furthermore, they discussed the design and fabrication challenges of micro drones, existing methods for increasing their endurance, and various navigation and control techniques. In addition, they also discussed the limitations of the existing drones as well as the proposed solutions for the next generation of drones.

Gharibi *et al.* [5] presented a conceptual model for the architecture of IoD-based system. The key concepts of three existing large scale networks such as air traffic control network, cellular network, and Internet are explored to the novel architecture for drone traffic management. Hall [8] discussed about the opportunities available to improve public and commercial drone operations. The classes of drones such as military drones and noncompliant drones are also provided in [8].

Later on, Won *et al.* [10], [11] proposed secure communication protocols for drones and smart objects. A suite of cryptographic protocols was proposed by them in order to deal with three different communication scenarios: 1) one-to-one; 2) one-to-many; and 3) many-to-one. For one-to-one, an efficient certificateless signcryption tag key encapsulation mechanism supports authenticated key agreement, and provides nonrepudiation and user revocation feature. For one-to-many, a certificateless multirecipient encryption scheme was presented by them in which a drone can send privacy-sensitive data to multiple smart objects. For many-to-one, a certificateless data aggregation protocol was proposed, which allows drones to collect data from hundreds of smart objects. Won *et al.* [11] also implemented the real drone application for the smart parking management system. The performance of their system was evaluated in a testbed consisting of the commercially available devices, such as AR.Drone2.0 and TelosB. Thus, for such kind of application, the real-time data is very much essential to analyze by the authorized authority. As a result, the authenticated key management is very crucial security protocol for securing the real-time data access.

Interestingly from the technology perspective, UAVs in the IoD are foreseen as an important component of an advanced cyber-physical Internet of Things (IoT) ecosystem [12]. Motlagh *et al.* [12] presented a comprehensive survey on UAVs. Furthermore, they highlighted the potential of UAVs for the delivery of IoT services from height and also addressed the relevant challenges.

Wireless sensor networks (WSNs) have several potential application including traffic monitoring, landslide detection, pipeline monitoring, border patrol, rehabilitation applications, precision agriculture, laboratory tutoring, real-time soccer playing monitoring, asset tracking, real-time healthcare monitoring, and military applications [13]. For all these critical applications, the real-time data access is needed by an authorized user (external party) from some designated sensor nodes directly. Thus, the user authentication is needed for securing WSNs. Turkanović *et al.* [13] proposed a user authentication scheme in WSNs for the IoT environment, which negotiates a session key with a general sensor node. Their scheme provides mutual authentication between the user, sensor node, and the gateway node. Their scheme is suitable for the resource-constrained sensor nodes as it uses only simple hash and bitwise XOR computations. However, Farash *et al.* [14] pointed out several security pitfalls in Turkanović *et al.*'s [13] scheme, such as it does not provide sensor node anonymity and user traceability, and it is not also secure against man-in-the-middle attack, session key security, sensor node impersonation attack, and stolen smart card attacks.

Challa *et al.* [15] proposed a new signature-based authenticated key establishment scheme in IoT environment, where the real-time data access from the IoT sensing devices by an authorized user is needed. Their scheme is efficient in computation and communication, and these are comparable with other related existing approaches. Furthermore, they demonstrated the practicality of their scheme using the widely accepted NS2 simulation.

As the majority of the applications using the drones in the IoD environment are based on real-time data access, a user is typically interested in obtaining the real-time services from the deployed drones, which fall under a particular fly zone. Wazid *et al.* [4] also emphasized that there is a potential requirement to deploy efficient as well as secure user authentication mechanism which should permit only an authorized user, such as a driver of an ambulance, in the IoD environment to access the data directly from some designated accessed drones in the network. Wazid *et al.* [4] suggested an authentication model which can be used in the IoD environment. They discussed several security challenges along with security requirements for the IoD environment. They also provided a taxonomy consisting of several security protocols in the IoD environment.

C. Research Contributions

The main contributions of this paper are listed below.

1) We propose a novel lightweight user authentication and key agreement scheme for IoD deployment. The proposed scheme only uses the efficient one-way cryptographic hash functions and bitwise XOR operations, apart from the fuzzy extractor method for the user biometric verification at the login phase discussed in Section III-C.
2) The proposed scheme is shown to be resistant against various known attacks through the formal security verification using the widely accepted AVISPA tool [16] and also through informal security analysis.
3) The proposed scheme is compared with related existing schemes and it is shown that the scheme provides better tradeoff between the security and functionality features, and communication and computation overheads as compared to those for other schemes.
4) Finally, the practical demonstration of the proposed scheme is performed through the broadly used NS2 simulation [17].

D. Structure of This Paper

The rest of this paper is organized as follows. In Section II, we provide the network as well as threat models used in the proposed scheme. The various phases of the proposed scheme are then discussed in Section III. The rigorous security analysis along with the formal security verification using the widely accepted AVISPA tool of the proposed scheme is given in Section IV. The performance comparison among the related existing schemes and the proposed scheme is provided in Section V. The practical demonstration of the proposed scheme using the widely used NS2 simulation tool is also provided in Section VI. Finally, this paper is concluded in Section VII.

II. System Models

In the proposed scheme, we follow the following two models to explain its working and usability.

A. Network Model

The network model of the proposed remote user authentication scheme for the IoD environment is given in Fig. 1. According to the network model, various drones are deployed in the different zones of a target field (e.g., a city) which can send data to the server (control room). Suppose there is an external user (U_i) (i.e., some ambulance driver) who wants to know the traffic condition in some particular area of the city. U_i can obtain easily this information from the deployed drones. U_i is connected to the server through the Internet. For accessing the real-time information, a secure remote user authentication is needed between an accessed drone (DR_j) and user (U_i). This authentication between U_i and DR_j happens via the server (S). After mutual authentication, both U_i and DR_j can establish a session key and start communication securely.

B. Threat Model

We follow the widely used Dolev–Yao (DY) [18] threat model in the proposed scheme. According to the DY model, any two communicating parties communicate over an insecure channel (open channel). Under this model, the communication channel is public, and the end-point entities such as U_i and DR_j are assumed to be untrustworthy. Thus, an attacker, say $\mathcal{A}$ can eavesdrop the exchanged messages and can also delete or modify the transmitted messages. As pointed out in [11], the drones may move around in unattended hostile areas with collected sensor data and hence, there are possibilities of physical capturing of drones by $\mathcal{A}$. Therefore, $\mathcal{A}$ can extract data from the captured drones' memory using the power analysis attacks [19]. However, the server S is considered as a fully trusted entity and it will not be compromised by $\mathcal{A}$.

III. Proposed Scheme

The proposed scheme explained in this section consists of seven phases: 1) predeployment; 2) user registration; 3) login; 4) authentication and key agreement; 5) password and biometric update; 6) dynamic drone addition; and 7) drone key management. In the proposed scheme, three factors used are: 1) mobile device MD_i of a user U_i; 2) password of U_i; and 3) biometrics of U_i. The notations used in this paper are given in Table I. We use the random nonces and current timestamps to protect against replay attack. For this purpose, we assume that all the network entities involved in the IoD environment are synchronized with their clocks. The proposed scheme is lightweight as it uses the efficient cryptographic one-way hash function and bitwise XOR operations, apart from the fuzzy extractor technique that is only needed for biometric verification at the user side.

A. Predeployment Phase

In this phase, the server S is responsible for registering each drone DR_j prior to its deployment in the IoD environment (for example, over various roads in a city). For this purpose, S first selects a unique 160-bit secret number k and also a unique identity $\mathrm{ID}_{\mathrm{DR}_j}$ for each DR_j, and computes its pseudo identity as $\mathrm{RID}_{\mathrm{DR}_j} = h(\mathrm{ID}_{\mathrm{DR}_j}||k)$. S further chooses 160-bit master key $\mathrm{MK}_{\mathrm{DR}_j}$ corresponding to DR_j and calculates its temporal credential as $\mathrm{TC}_{\mathrm{DR}_j} = h(\mathrm{ID}_{\mathrm{DR}_j}||\mathrm{MK}_{\mathrm{DR}_j}||\mathrm{RTS}_{\mathrm{DR}_j})$, where $\mathrm{RTS}_{\mathrm{DR}_j}$ is the registration timestamp of DR_j.

TABLE I
Notations Used in This Paper

Symbol	Description
U_i, MD_i	i^{th} user and his/her mobile device, respectively
DR_j	j^{th} drone
S	Server (trusted authority)
ID_i, PW_i, BIO_i	U_i's identity, password and biometric information, respectively
ID_s, ID_{DR_j}	Identities of S and DR_j, respectively
RID_i, RID_s, RID_{DR_j}	Pseudo identities of U_i, S and DR_j, respectively
k, n	160-bit secret numbers of S and U_i, respectively
r_1, r_2, r_3	160-bit random nonces of U_i, S and DR_j, respectively
RTS_{U_i}, RTS_{DR_j}	Registration timestamps of U_i and DR_j, respectively
MK_{U_i}, MK_{DR_j}	160-bit master keys of U_i and DR_j, respectively
T_1, T_2, T_3	Current timestamps
ΔT	Maximum transmission delay associated with a message
$Gen(\cdot)$	Fuzzy extractor generation procedure
$Rep(\cdot)$	Fuzzy extractor reproduction procedure
BIO_i	Personal biometrics of U_i
σ_i	Biometric secret key of U_i
τ_i	Public reproduction parameter of U_i
t	Error tolerance threshold used in fuzzy extractor
$h(\cdot)$	Collision-resistant cryptographic one-way hash function
$\|\|$, $\oplus$	Concatenation and bitwise XOR operations, respectively

For pairwise key establishment between two neighboring drones (see Section III-G), S selects a symmetric bivariate polynomial $\mathcal{P}(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{m} g_{i,j}x^i y^j \in \mathrm{GF}(p)[x, y]$ of degree m over a finite field (Galois field) $\mathrm{GF}(p)$, where the coefficients $g_{i,j}$'s are taken from $\mathrm{GF}(p)$. The prime p is selected as a large number and m is also taken large, which is much larger than the number of drones deployed in the target field in order to preserve unconditional security and m-collusion resistant property against drones capture attack by an adversary [20]. An example of a symmetric bivariate polynomial is $\mathcal{P}(x, y) = x^4 + 3x^3 + 2x^2y^2 + 3y^3 + y^4$ over $\mathrm{GF}(5)$ as $\mathcal{P}(y, x) = y^4 + 3y^3 + 2y^2x^2 + 3x^3 + x^4 = \mathcal{P}(x, y)$. S then generates a temporary identity $\mathrm{TID}_{\mathrm{DR}_j}$ corresponding to the pseudo-identity $\mathrm{RID}_{\mathrm{DR}_j}$, and computes its polynomial share $\mathcal{P}(\mathrm{TID}_{\mathrm{DR}_j}, y)$ which is a univariate polynomial of degree m in $\mathrm{GF}(p)$. Note that to store $\mathcal{P}(\mathrm{TID}_{\mathrm{DR}_j}, y)$, the storage space required in DR_j's memory is $(m + 1)\log_2(p)$ bits as the coefficients are from $\mathrm{GF}(p)$.

S then stores the information $\{\mathrm{TID}_{\mathrm{DR}_j}, \mathrm{RID}_{\mathrm{DR}_j}, \mathrm{TC}_{\mathrm{DR}_j}, \mathcal{P}(\mathrm{TID}_{\mathrm{DR}_j}, y)\}$ in the memory of DR_j and then deploys DR_j in the deployment field, whereas S keeps the information $\{\mathrm{RID}_{\mathrm{DR}_j}, \mathrm{TC}_{\mathrm{DR}_j}, \mathcal{P}(x, y), k\}$ in its database.

B. User Registration Phase

This phase discusses the registration procedure for an external user (for example, driver of an ambulance) U_i for accessing the real-time information from an accessed drone DR_j in the IoD environment. For this purpose, U_i requires to register at the server S securely either in person or via a secure channel with the following steps.

Step R1: U_i selects an identity ID_i and sends the registration request message $\langle \text{ID}_i \rangle$ to S securely. After receiving registration request, S calculates U_i's pseudo identity as $\text{RID}_i = h(\text{ID}_i||k)$ using its corresponding 160-bit secret number k. S also computes its own pseudo identity as $\text{RID}_s = h(\text{ID}_s||k)$, $A = h(\text{RID}_s||\text{ID}_i)$ and the temporal credential of U_i as $\text{TC}_{U_i} = h(\text{ID}_i||\text{MK}_{U_i}||\text{RTS}_{U_i})$, where MK_{U_i} is the 160-bit master secret key of U_i and RTS_{U_i} is the registration timestamp generated for U_i by the server S. S then sends the registration reply message $\langle \text{RID}_i, \text{RID}_{\text{DR}_j}, \text{RID}_s, \text{TC}_{U_i}, A \rangle$ to U_i securely.

Step R2: After receiving registration reply from S, U_i chooses a password PW_i of his/her choice, and inputs his/her biometric BIO_i at the sensor of his/her mobile device MD_i. For biometric verification, we apply the widely used the fuzzy extractor method [21], [22]. A fuzzy extractor is a pair of two functions where one function generates the uniform random bits from given input while the other recovers the string from an input close to the original input within a predefined threshold. The function pair in a fuzzy extractor is given below.

1) *Gen:* It is a probabilistic generation function that takes the user personal biometrics BIO_i as input, and returns $\sigma_i \in \{0, 1\}^l$ as the biometric secret key of length l bits and τ_i as the public reproduction parameter, that is, $\text{Gen}(\text{BIO}_i) = (\sigma_i, \tau_i)$.
2) *Rep:* It is a deterministic function, whose inputs are the user biometrics, say BIO'_i and τ_i, provided the Hamming distance between BIO'_i and the original previously entered biometrics BIO_i is less than or equal to an error tolerance threshold value t. The output is the original biometric key σ_i, that is, $\sigma_i = \text{Rep}(\text{BIO}'_i, \tau_i)$.

MD_i generates the biometric secret key σ_i and its corresponding public parameter τ_i as $\text{Gen}(\text{BIO}_i) = (\sigma_i, \tau_i)$.

Step R3: MD_i generates 160-bit secret number n for U_i and calculates $\text{RID}'_i = \text{RID}_i \oplus h(\text{PW}_i||\sigma_i)$, $\text{RID}'_{\text{DR}_j} = \text{RID}_{\text{DR}_j} \oplus h(\text{ID}_i||\text{PW}_i||\sigma_i)$, $\text{TC}'_{U_i} = \text{TC}_{U_i} \oplus h(\text{ID}_i||\sigma_i)$, masked password $\text{RPW}_i = h(\text{PW}_i||n)$, $\text{RID}'_s = \text{RID}_s \oplus h(\text{RID}_i||\sigma_i)$. MD_i further computes the following:

$$A' = A \oplus h(\text{RID}_i||\sigma_i||\text{PW}_i)$$
$$B = n \oplus h(\text{PW}_i||\text{ID}_i||\sigma_i)$$
$$C = h\big(A||\text{RID}_{\text{DR}_j}||\text{RPW}_i||\sigma_i\big).$$

Finally, MD_i stores the information $\{\text{RID}'_i, \text{RID}'_{\text{DR}_j}, \text{RID}'_s, \text{TC}'_{U_i}, A', B, C, \tau_i, \text{Gen}(\cdot), \text{Rep}(\cdot), h(\cdot), t\}$ in its memory. S also stores $\{\text{ID}_i, \text{RID}_i, \text{TC}_{U_i}, \text{RID}_s\}$ in its database.

The user registration phase is briefed in Fig. 3.

User (U_i)/ Mobile device (MD_i)	Server (S)
Pick identity ID_i. $\xrightarrow{\langle ID_i \rangle}$ (via secure channel)	
	Calculate $RID_i = h(ID_i \,\|\|k)$, $RID_s = h(ID_s \,\|\|k)$, $A = h(RID_s \,\|\|ID_i)$, $TC_{U_i} = h(ID_i \,\|\|MK_{U_i} \,\|\|RTS_{U_i})$. $\xleftarrow{\langle RID_i, RID_{DR_j}, RID_s, TC_{U_i}, A \rangle}$ (via secure channel)
Pick PW_i, imprint BIO_i. Calculate $Gen(BIO_i) = (\sigma_i, \tau_i)$, $RID'_i = RID_i \oplus h(PW_i\|\| \sigma_i)$, $RID'_{DR_j} = RID_{DR_j} \oplus h(ID_i \,\|\|PW_i\|\| \sigma_i)$, $TC'_{U_i} = TC_{U_i} \oplus h(ID_i\|\| \sigma_i)$, $RPW_i = h(PW_i \,\|\|n)$, $RID'_s = RID_s \oplus h(RID_i \,\|\|\sigma_i)$, $A' = A \oplus h(RID_i\|\|\sigma_i\|\|PW_i)$, $B = n \oplus h(PW_i\|\| ID_i\|\| \sigma_i)$, $C = h(A\|\| RID_{DR_j} \,\|\|RPW_i \,\|\|\sigma_i)$. Store $\{RID'_i, RID'_{DR_j}, RID'_s, TC'_{U_i}, A'$, $B, C, \tau_i, Gen(\cdot), Rep(\cdot), h(\cdot), t\}$ in MD_i.	Store $\{ID_i, RID_i, TC_{U_i}, RID_s\}$ in S.

Fig. 3. Summary of user registration phase.

C. Login Phase

U_i needs to perform the following steps to execute the login phase.

Step L1: U_i inputs his/her identity ID_i and password PW'_i into the interface of MD_i, and also imprints his/her biometrics BIO'_i at the sensor of MD_i. MD_i then calculates biometric key $\sigma'_i = \text{Rep}(\text{BIO}'_i, \tau_i)$ provided that the Hamming distance between the original biometrics BIO_i at the time of registration and the recent entered BIO'_i is less than or equal to the error tolerance threshold value t. Next, MD_i calculates the following:

$$\text{RID}_i = \text{RID}'_i \oplus h\big(\text{PW}'_i||\sigma'_i\big)$$
$$\text{RID}_{\text{DR}_j} = \text{RID}'_{\text{DR}_j} \oplus h\big(\text{ID}_i||\text{PW}'_i||\sigma'_i\big)$$
$$\text{TC}_{U_i} = \text{TC}'_{U_i} \oplus h\big(\text{ID}_i||\sigma'_i\big)$$
$$\text{RID}_s = \text{RID}'_s \oplus h\big(\text{RID}_i||\sigma'_i\big)$$
$$n = B \oplus h\big(\text{PW}'_i||\text{ID}_i||\sigma'_i\big)$$
$$\text{RPW}'_i = h\big(\text{PW}'_i||n\big).$$

MD_i further computes $A = A' \oplus h(\text{RID}_i||\sigma'_i||\text{PW}'_i)$ and $C' = h(A||\text{RID}_{\text{DR}_j}||\text{RPW}'_i||\sigma'_i)$. After these computations, MD_i checks whether the condition $C' = C$ holds or not. If it holds, U_i passes both password and biometric verification. Otherwise, the login phase is terminated immediately.

Step L2: MD_i generates the current timestamp T_1 and 160-bit random nonce r_1. MD_i computes the following:

$$M_1 = \text{RID}_i \oplus h(\text{RID}_s||T_1)$$
$$M_2 = \text{RID}_{\text{DR}_j} \oplus h\big(\text{TC}_{U_i}||\text{ID}_i||T_1\big)$$
$$M_3 = h\big(\text{RID}_s||\text{TC}_{U_i}||T_1\big) \oplus r_1$$
$$M_4 = h\big(\text{ID}_i||\text{RID}_s||\text{RID}_{\text{DR}_j}||\text{TC}_{U_i}||r_1||T_1\big).$$

Finally, MD_i sends the login request message $\text{Msg}_1 = \langle M_1, M_2, M_3, M_4, T_1 \rangle$ to S via a public channel.

D. Authentication and Key Agreement Phase

After receiving login request message $\langle M_1, M_2, M_3, M_4, T_1 \rangle$ from U_i, the following steps are performed among the user U_i, the server S and an accessed drone DR_j, and after that U_i and DR_j establish a session key for secure communication between them.

Step AK1: S first checks the timeliness of T_1 by the condition $|T_1 - T_1^*| \leq \Delta T$, where the maximum transmission delay is presented by ΔT and T_1^* is the reception time of the message $\langle M_1, M_2, M_3, M_4, T_1 \rangle$. If the condition is valid, S computes $\text{RID}_i = M_1 \oplus h(\text{RID}_s||T_1)$, and fetches ID_i and TC_{U_i} corresponding to the computed RID_i. S further computes the following:

$$\begin{aligned} \mathrm{RID}_{\mathrm{DR}_j} &= M_2 \oplus h\big(\mathrm{TC}_{U_i}||\mathrm{ID}_i||T_1\big) \\ r_1' &= M_3 \oplus h\big(\mathrm{RID}_s||\mathrm{TC}_{U_i}||T_1\big) \\ M_4' &= h\big(\mathrm{ID}_i||\mathrm{RID}_s||\mathrm{RID}_{\mathrm{DR}_j}||\mathrm{TC}_{U_i}||r_1'||T_1\big) \end{aligned}$$

and checks if the condition $M_4' = M_4$ is valid. If it is valid, U_i is authenticated by S; otherwise, S terminates the session immediately.

Step AK2: S generates a random nonce r_2 and the current timestamp T_2, and calculates the following:

$$\begin{aligned} M_5 &= h\big(\mathrm{TC}_{\mathrm{DR}_j}||\mathrm{RID}_{\mathrm{DR}_j}\big) \oplus h(\mathrm{RID}_s||r_1||r_2) \\ M_6 &= h\big(\mathrm{TC}_{\mathrm{DR}_j}||T_2\big) \oplus \mathrm{RID}_i \\ M_7 &= h\big(\mathrm{RID}_{\mathrm{DR}_j}||\mathrm{TC}_{\mathrm{DR}_j}||h(\mathrm{RID}_s||r_1||r_2)||T_2\big). \end{aligned}$$

S then sends authentication request message $\mathrm{Msg}_2 = \langle M_5, M_6, M_7, T_2\rangle$ to DR_j via a public channel.

Step AK3: After receiving the message in step AK2, DR_j first checks the timeliness of T_2 by the condition $|T_2 - T_2^*| \leq \Delta T$, where T_2^* is the reception time of the message. If the timeliness matches, DR_j proceeds to calculate the following:

$$\begin{aligned} \mathrm{RID}_i &= M_6 \oplus h\big(\mathrm{TC}_{\mathrm{DR}_j}||T_2\big) \\ M_8 &= M_5 \oplus h\big(\mathrm{TC}_{\mathrm{DR}_j}||\mathrm{RID}_{\mathrm{DR}_j}\big) \\ M_9 &= h\big(\mathrm{RID}_{\mathrm{DR}_j}||\mathrm{TC}_{\mathrm{DR}_j}||M_8||T_2\big) \end{aligned}$$

and then checks if the condition $M_9 = M_7$ is satisfied. If it is valid, S is authenticated by DR_j; otherwise, DR_j terminates the session immediately. DR_j then starts generating a random nonce r_3 and the current timestamp T_3, and calculates $M_{10} = h(\mathrm{RID}_{\mathrm{DR}_j}||\mathrm{RID}_i||T_3) \oplus r_3$, the session key $\mathrm{SK}_{ij} = h(M_8||r_3||\mathrm{RID}_i||\mathrm{RID}_{\mathrm{DR}_j})$ shared with U_i, $M_{11} = h(\mathrm{RID}_i||\mathrm{RID}_{\mathrm{DR}_j}||r_3) \oplus M_8$ and $M_{12} = h(\mathrm{SK}_{ij}||T_3)$. DR_j directly sends authentication reply message $\mathrm{Msg}_3 = \langle M_{10}, M_{11}, M_{12}, T_3\rangle$ to U_i via an open channel.

Step AK4: After receiving the message in step AK3, U_i first checks the timeliness of T_3 by the condition $|T_3 - T_3^*| \leq \Delta T$, where T_3^* is the reception time of the message. U_i calculates $r_3' = M_{10} \oplus h(\mathrm{RID}_{\mathrm{DR}_j}||\mathrm{RID}_i||T_3)$, $M_8' = M_{11} \oplus h(\mathrm{RID}_i||\mathrm{RID}_{\mathrm{DR}_j}||r_3')$, the session key $\mathrm{SK}_{ij}' = h(M_8'||r_3'||\mathrm{RID}_i||\mathrm{RID}_{\mathrm{DR}_j}) (= \mathrm{SK}_{ij})$ shared with DR_j and $M_{13} = h(\mathrm{SK}_{ij}'||T_3)$, and further checks the condition $M_{13} = M_{12}$. If it matches, DR_j is authenticated by U_i, and the computed session key SK_{ij}' by U_i is correct; otherwise, U_i terminates the session immediately. After that both U_i and DR_j maintain the same computed session key $\mathrm{SK}_{ij} (= \mathrm{SK}_{ij}')$ for future their secure communication.

The login, and authentication and key agreement phases related to the proposed scheme are summarized in Fig. 4.

E. Password and Biometric Update Phase

A secure authentication scheme should have the facility of password and biometric update procedure, so that a legal user can update his/her password and biometric information at any time for security reasons without interacting with the server S. U_i needs to perform the following steps to execute password and biometric update phase.

Step PB1: U_i first provides his/her identity ID_i and old password PW_i^o into the interface of MD_i, and also imprints his/her old biometrics BIO_i^o at the sensor of MD_i. MD_i then calculates biometric secret key $\sigma_i^o = \mathrm{Rep}(\mathrm{BIO}_i^o, \tau_i)$ provided that the Hamming distance between the original biometrics BIO_i at the time of registration and the recent entered BIO_i^o is less than or equal to the error tolerance threshold value t. MD_i further calculates the following:

$$\begin{aligned} \mathrm{RID}_i &= \mathrm{RID}_i' \oplus h\big(\mathrm{PW}_i^o||\sigma_i^o\big) \\ \mathrm{RID}_{\mathrm{DR}_j} &= \mathrm{RID}_{\mathrm{DR}_j}' \oplus h\big(\mathrm{ID}_i||\mathrm{PW}_i^o||\sigma_i^o\big) \\ \mathrm{TC}_{U_i} &= \mathrm{TC}_{U_i}' \oplus h\big(\mathrm{ID}_i||\sigma_i^o\big) \\ \mathrm{RID}_s &= \mathrm{RID}_s' \oplus h\big(\mathrm{RID}_i||\sigma_i^o\big) \\ n &= B \oplus h\big(\mathrm{PW}_i^o||\mathrm{ID}_i||\sigma_i^o\big) \\ \mathrm{RPW}_i^o &= h\big(\mathrm{PW}_i^o||n\big) \\ A &= A' \oplus h\big(\mathrm{RID}_i||\sigma_i^o||\mathrm{PW}_i^o\big) \\ C^o &= h\big(A||\mathrm{RID}_{\mathrm{DR}_j}||\mathrm{RPW}_i^o||\sigma_i^o\big). \end{aligned}$$

After these computations, MD_i checks whether the condition $C^o = C$ holds or not. If it holds, U_i passes both password and biometric verification, and he/she proceeds for password and biometric update procedure. Otherwise, this phase is terminated immediately.

Step PB2: U_i provides new password PW_i^n and imprints new biometric BIO_i^n at the sensor of his/her mobile device MD_i. Note that since the biometrics usually remains unchanged, if U_i feels not to update his/her biometrics then he/she can still keep old biometrics BIO_i^o. In that case, it is treated that BIO_i^n will be BIO_i^o. Otherwise, MD_i computes $\mathrm{Gen}(\mathrm{BIO}_i^n) = (\sigma_i^n, \tau_i^n)$.

Step PB3: MD_i further continues to compute the following:

$$\begin{aligned} \mathrm{RID}_i^* &= \mathrm{RID}_i \oplus h\big(\mathrm{PW}_i^n||\sigma_i^n\big) \\ \mathrm{RID}_{\mathrm{DR}_j}^* &= \mathrm{RID}_{\mathrm{DR}_j} \oplus h\big(\mathrm{ID}_i||\mathrm{PW}_i^n||\sigma_i^n\big) \\ \mathrm{TC}_{U_i}^* &= \mathrm{TC}_{U_i} \oplus h\big(\mathrm{ID}_i||\sigma_i^n\big) \\ \mathrm{RPW}_i^n &= h\big(\mathrm{PW}_i^n||n\big) \\ \mathrm{RID}_s^* &= \mathrm{RID}_s \oplus h\big(\mathrm{RID}_i||\sigma_i^n\big) \\ A^* &= A \oplus h\big(\mathrm{RID}_i||\sigma_i^n||\mathrm{PW}_i^n\big) \\ B^* &= n \oplus h\big(\mathrm{PW}_i^n||\mathrm{ID}_i||\sigma_i^n\big) \\ C^* &= h\big(A||\mathrm{RID}_{\mathrm{DR}_j}||\mathrm{RPW}_i^n||\sigma_i^n\big). \end{aligned}$$

Finally, MD_i stores the information $\{\mathrm{RID}_i^*, \mathrm{RID}_{\mathrm{DR}_j}^*, \mathrm{RID}_s^*, \mathrm{TC}_{U_i}^*, A^*, B^*, C^*, \tau_i^n, \mathrm{Gen}(\cdot), \mathrm{Rep}(\cdot), h(\cdot), t\}$ in its memory, while replacing $\mathrm{RID}_i', \mathrm{RID}_{\mathrm{DR}_j}', \mathrm{RID}_s', \mathrm{TC}_{U_i}', A', B, C$, and τ_i by $\mathrm{RID}_i^*, \mathrm{RID}_{\mathrm{DR}_j}^*, \mathrm{RID}_s^*, \mathrm{TC}_{U_i}^*, A^*, B^*, C^*$, and τ_i^n, respectively.

F. Dynamic Drone Addition Phase

The proposed scheme provides the facility of addition of new drones in the network at any time.

Suppose a new drone $\mathrm{DR}_j^{\mathrm{new}}$ is needed to deploy in the IoD environment. For this purpose, the server S first generates a unique identity $\mathrm{ID}_{\mathrm{DR}_j}^{\mathrm{new}}$ for $\mathrm{DR}_j^{\mathrm{new}}$ and computes the pseudo identity as $\mathrm{RID}_{\mathrm{DR}_j}^{\mathrm{new}} = h(\mathrm{ID}_{\mathrm{DR}_j}^{\mathrm{new}}||k)$ using the secret key k of S. S further chooses 160-bit master key $\mathrm{MK}_{\mathrm{DR}_j}^{\mathrm{new}}$ corresponding to $\mathrm{DR}_j^{\mathrm{new}}$ and computes the temporal credential for $\mathrm{DR}_j^{\mathrm{new}}$ as $\mathrm{TC}_{\mathrm{DR}_j}^{\mathrm{new}} = h(\mathrm{ID}_{\mathrm{DR}_j}^{\mathrm{new}}||\mathrm{MK}_{\mathrm{DR}_j}^{\mathrm{new}}||\mathrm{RTS}_{\mathrm{DR}_j}^{\mathrm{new}})$, where $\mathrm{RTS}_{\mathrm{DR}_j}^{\mathrm{new}}$ denotes

User (U_i)/Mobile device (MD_i)	Server (S)	Drone (DR_j)
$\{RID'_i, RID'_{DR_j}, RID'_s, TC'_{U_i}, A', B, C, \tau_i,$ $Gen(\cdot), Rep(\cdot), h(\cdot), t\}$	$\{RID_{DR_j}, TC_{DR_j}, k,$ $ID_i, RID_i, TC_{U_i}, RID_s\}$	$\{(TID_{DR_j}, RID_{DR_j}), TC_{DR_j},$ $\mathcal{P}(TID_{DR_j}, y)\}$
Input ID_i, PW'_i, BIO'_i. Calculate $\sigma'_i = Rep(BIO'_i, \tau_i)$, $RID_i = RID'_i \oplus h(PW'_i\|\|\sigma'_i)$, $RID_{DR_j} = RID'_{DR_j} \oplus h(ID_i\|\|PW'_i\|\| \sigma'_i)$, $TC_{U_i} = TC'_{U_i} \oplus h(ID_i\|\|\sigma'_i)$, $RID_s = RID'_s \oplus h(RID_i\|\|\sigma'_i)$, $n = B \oplus h(PW'_i\|\|ID_i\|\|\sigma'_i)$, $RPW'_i = h(PW'_i\|\|n)$, $A = A' \oplus h(RID_i\|\| \sigma'_i\|\|PW'_i)$, $C' = h(A\|\|RID_{DR_j}\|\| RPW'_i\|\|\sigma'_i)$. Check if $C' = C$? If so, generate T_1, r_1. Compute $M_1 = RID_i \oplus h(RID_s\|\| T_1)$, $M_2 = RID_{DR_j} \oplus h(TC_{U_i} \|\|ID_i\|\|T_1)$, $M_3 = h(RID_s\|\| TC_{U_i}\|\|T_1) \oplus r_1$, $M_4 = h(ID_i\|\| RID_s\|\|RID_{DR_j}\|\| TC_{U_i}\|\|r_1\|\|T_1)$. $\xrightarrow{Msg_1 = \langle M_1, M_2, M_3, M_4, T_1\rangle}$ (via public channel)		
	Check if $\|T_1 - T_1^*\| \leq \Delta T$? If so, compute $RID_i = M_1 \oplus h(RID_s\|\| T_1)$. Check if RID_i exists in its database. If so, fetch ID_i, TC_{U_i} corresponding to RID_i. Compute $RID_{DR_j} = M_2 \oplus h(TC_{U_i} \|\|ID_i\|\|T_1)$, $r'_1 = M_3 \oplus h(RID_s\|\| TC_{U_i}\|\|T_1)$, $M'_4 = h(ID_i\|\|RID_s \|\|RID_{DR_j}\|\|TC_{U_i} \|\|r'_1\|\|T_1)$, Check if $M'_4 = M_4$? If so, generate r_2, T_2. Fetch TC_{DR_j} corresponding to RID_{DR_j}. Compute $M_5 = h(TC_{DR_j}\|\|RID_{DR_j}) \oplus h(RID_s\|\| r_1\|\|r_2)$, $M_6 = h(TC_{DR_j} \|\|T_2) \oplus RID_i$, $M_7 = h(RID_{DR_j}\|\| TC_{DR_j}\|\| h(RID_s\|\|r_1 \|\|r_2)\|\|T_2)$. $\xrightarrow{Msg_2 = \langle M_5, M_6, M_7, T_2\rangle}$ (via public channel)	
		Check if $\|T_2 - T_2^*\| \leq \Delta T$? If so, compute $RID_i = M_6 \oplus h(TC_{DR_j} \|\|T_2)$. $M_8 = M_5 \oplus h(TC_{DR_j} \|\|RID_{DR_j})$, $M_9 = h(RID_{DR_j} \|\|TC_{DR_j}\|\|M_8 \|\|T_2)$. Check if $M_9 = M_7$? If so, generate r_3, T_3. Compute $M_{10} = h(RID_{DR_j} \|\|RID_i\|\|T_3)\oplus r_3$, $SK_{ij} = h(M_8\|\|r_3\|\| RID_i\|\|RID_{DR_j})$, $M_{11} = h(RID_i\|\| RID_{DR_j}\|\|r_3) \oplus M_8$, $M_{12} = h(SK_{ij} \|\|T_3)$. $\xleftarrow{Msg_3 = \langle M_{10}, M_{11}, M_{12}, T_3\rangle}$ (to U_i) (via public channel)
Check if $\|T_3 - T_3^*\| \leq \Delta T$? If so, compute $r'_3 = M_{10} \oplus h(RID_{DR_j} \|\|RID_i\|\|T_3)$, $M'_8 = M_{11} \oplus h(RID_i\|\| RID_{DR_j}\|\|r'_3)$, $SK'_{ij} = h(M'_8\|\|r'_3\|\| RID_i\|\|RID_{DR_j})$, $M_{13} = h(SK'_{ij}\|\| T_3)$. Check if $M_{13} = M_{12}$?		
U_i and DR_j maintain the session key SK_{ij} $(= SK'_{ij})$ for future secure communication.		

Fig. 4. Summary of login and authentication and key agreement phases.

the registration timestamp of DR_j^{new}. S also generates a unique temporary identity $\text{TID}_{\text{DR}_j}^{\text{new}}$ and calculates the polynomial share $\mathcal{P}(\text{TID}_{\text{DR}_j}^{\text{new}}, y)$.

S finally stores the information $\{\text{TID}_{\text{DR}_j}^{\text{new}}, \text{RID}_{\text{DR}_j}^{\text{new}}, \text{TC}_{\text{DR}_j}^{\text{new}}, \mathcal{P}(\text{TID}_{\text{DR}_j}^{\text{new}}, y)\}$ in the memory of DR_j^{new} and deploys it in the deployment field, whereas S keeps $\{\text{RID}_{\text{DR}_j}^{\text{new}}, \text{TC}_{\text{DR}_j}^{\text{new}}\}$ corresponding to DR_j^{new} in its database. The server S also informs all the users in the network about the deployment of DR_j^{new} so that the users can access the information from DR_j^{new}, if necessary.

G. Drone Key Management Phase

Drones are deployed in the target field and they fly over various zones. Suppose a drone wants to share its information to some other drone. In that situation, we need a secure communication between these drones. For this purpose, we need a pairwise key establishment between two neighboring communicating drones. For the pairwise key establishment between two neighboring deployed drones, say DR_j and DR_k, we can use the existing polynomial-based key distribution scheme proposed by Blundo *et al.* [23]. DR_j sends its own temporary identity TID_{DR_j} to DR_k. Similarly, DR_k also exchanges its own temporary identity TID_{DR_k} to DR_j. DR_j then computes the secret key shared with DR_k using its own polynomial share as

$$\text{SK}_{\text{DR}_j,\text{DR}_k} = \mathcal{P}\big(\text{TID}_{\text{DR}_j}, \text{TID}_{\text{DR}_k}\big).$$

In a similar way, DR_k also computes the same secret key shared with DR_j using its own polynomial share as

$$\begin{aligned}\text{SK}_{\text{DR}_k,\text{DR}_j} &= \mathcal{P}\big(\text{TID}_{\text{DR}_k}, \text{TID}_{\text{DR}_j}\big)\\ &= \mathcal{P}\big(\text{TID}_{\text{DR}_j}, \text{TID}_{\text{DR}_k}\big)\\ &= \text{SK}_{\text{DR}_j,\text{DR}_k}\end{aligned}$$

since the polynomial $\mathcal{P}(x, y)$ is symmetric. Hence, both DR_j and DR_k can communicate securely using the common established shared secret key $\text{SK}_{\text{DR}_j,\text{DR}_k} (= \text{SK}_{\text{DR}_k,\text{DR}_j})$.

Remark 1: In the proposed scheme, a drone can be dynamically added in the network at any time. Assume that an adversary $\mathcal{A}$ physically captures a drone DR_j or a drone DR_j can be also stolen by $\mathcal{A}$. $\mathcal{A}$ can extract all the credentials $\{\text{TID}_{\text{DR}_j}, \text{RID}_{\text{DR}_j}, \text{TC}_{\text{DR}_j}, \mathcal{P}(\text{TID}_{\text{DR}_j}, y)\}$ from its memory using power analysis attacks [19]. It is worth noting that $\text{TID}_{\text{DR}_j}, \text{RID}_{\text{DR}_j}, \text{TC}_{\text{DR}_j}$, and $\mathcal{P}(\text{TID}_{\text{DR}_j}, y)$ are distinct for all drones, and these credentials are generated by the server S. Suppose $\mathcal{A}$ wishes to utilize the extracted credentials for a newly deployed drone DR_j^{new}. For this purpose, $\mathcal{A}$ can generate identity $\text{ID}_{\text{DR}_j}^{\text{new}}$, master key $\text{MK}_{\text{DR}_j}^{\text{new}}$ and registration timestamp $\text{RTS}_{\text{DR}_j}^{\text{new}}$ for DR_j^{new}, but he/she can not compute $\text{RID}_{\text{DR}_j}^{\text{new}} = h(\text{ID}_{\text{DR}_j}^{\text{new}} || k)$ and polynomial share $\mathcal{P}(\text{TID}_{\text{DR}_j}^{\text{new}}, y)$ as the secret key k and the original polynomial $\mathcal{P}(x, y)$ are unknown to $\mathcal{A}$. This will restrict $\mathcal{A}$ to use the extracted information for deploying a malicious drone in the network, and thus, a deployed malicious drone can not establish secure communication with other existing deployed drones in the network. As mentioned in Section IV-B7, by capturing a drone DR_j, $\mathcal{A}$ can only compromise the session key between a registered legal user U_i and the compromised DR_j. However, the session keys between that user U_i and other noncompromised drones can not be compromised by $\mathcal{A}$ as these are all distinct. This means that compromise of a drone does not result in compromising secure communications among a user and other noncompromised drones. In this way, the proposed scheme also preserves unconditional security against drone capture attack.

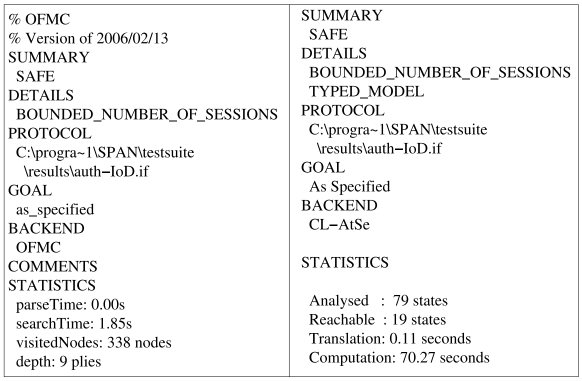
IV. Security Analysis

This section shows the ability of the proposed scheme to resist various well-known attacks.

A. Formal Security Verification Using AVISPA

This section simulates the proposed scheme for the formal security verification using the broadly accepted AVISPA tool [16] to check whether the scheme is secure against replay and man-in-the-middle attacks.

AVISPA is considered as a push-button tool that provides an expressive and modular formal language to specify protocols and their security properties. AVISPA implements various state-of-the-art techniques in order to perform automatic analysis by integrating four backends. The four backends include: 1) on-the-fly model-checker (OFMC); 2) constraint-logic-based attack searcher (CL-AtSe); 3) SAT-based model-checker (SATMC); and 4) tree automata based on automatic approximations for the analysis of security protocols (TA4SP) [16]. HLPSL integrates these backends and abstraction methods in AVISPA [24]. The executability of a protocol is verified by performing a static analysis. The HLPSL code specifying the protocol and intruder actions are translated into an intermediate format (IF) with the help of a translator, known as HLPSL2IF. The IF is then fed as input to one of the four backends for automated analysis. The detailed description of AVISPA tool and the implementation details using the HLPSL can be found in [16] and [24].



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\progra~1\SPAN\testsuite
   \results\auth−IoD.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 1.85s
  visitedNodes: 338 nodes
  depth: 9 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  C:\progra~1\SPAN\testsuite
   \results\auth−IoD.if
GOAL
  As Specified
BACKEND
  CL−AtSe

STATISTICS

  Analysed   :  79 states
  Reachable  : 19 states
  Translation: 0.11 seconds
  Computation: 70.27 seconds
```

Fig. 5. Analysis of results using OFMC and CL-AtSe backends.

We have implemented the proposed scheme for the user registration, login, and authentication and key agreement phases in HLPSL. In our implementation, we have three basic roles for a user, called *user*, for the server S, called *server* and for a drone DR_j, called *drone*. Apart from three basic roles, the roles for the session, goal and environment are also implemented, which are mandatory roles for any security protocol to be analyzed in AVISPA. Note that the intruder (always denoted by i) is also one of the participants through a concrete session in the protocol execution.

We have selected the broadly used OFMC and CL-AtSe backends for the execution test to find whether there are any attacks on the proposed scheme [16]. Note that other backends, such as SATMC and TA4SP do not support bitwise XOR operations. This is why we have omitted the simulation results of SATMC and TA4SP backends in this paper. To check for the possibility of a replay attack, the backends check if the specified protocol can be executed by the legitimate agents to search for a passive intruder. The backends then supply the intruder (i) with information about a few normal sessions between the legitimate agents. To verify the DY model (also mentioned in Section II-B), the backends verify if there is any possibility of a man-in-the-middle attack. Finally, we have simulated the proposed scheme under the widely accepted Security Protocol Animator for AVISPA Web tool [25]. The simulation results provided in Fig. 5 clearly show that the proposed scheme is secure against the replay and man-in-the-middle attacks.

B. Discussion on Other Attacks

In this section, we show informally that the proposed scheme has the ability to protect the following other attacks.

1) Privileged-Insider and Offline Password Guessing Attacks: Suppose a privileged-insider user (for example, an internal user of control room), being an adversary $\mathcal{A}$, knows the registration information $\{\text{ID}_i\}$ during the user registration phase, which was sent by U_i to S. Assume that $\mathcal{A}$ has lost/stolen mobile device MD_i of the registered user U_i after the registration process is completed. $\mathcal{A}$ can then extract the important information $\{\text{RID}_i', \text{RID}_{\text{DR}_j}', \text{RID}_s', \text{TC}_{U_i}', A', B, C, \tau_i, \text{Gen}(\cdot), \text{Rep}(\cdot), h(\cdot), t\}$

stored in MD_i's memory by applying the power analysis attacks [19]. However, without having the biometric key σ_i of U_i, $\mathcal{A}$ can not derive the secret credential $n = B \oplus h(\text{PW}_i||\text{ID}_i||\sigma_i)$, and hence, he/she can not also verify a guessed password with the help of C through the offline password guessing attack. Moreover, without σ_i and n, it is hard for $\mathcal{A}$ to derive RID_i, RID_{DR_j}, RID_s, TC_{U_i}, and RPW_i. As compared to low-entropy passwords, the biometric keys have various advantages [26], [27], such as: 1) biometric keys cannot be lost or forgotten; 2) biometric keys are hard to forge or distribute; and 3) biometric keys are difficult to copy or share. Therefore, guessing the biometric keys is relatively a hard problem [22]. The proposed scheme thus provides protection against privileged-insider attack as well as offline password guessing attack.

2) User Impersonation Attack: Suppose an adversary $\mathcal{A}$ tries to impersonate a user U_i (mobile device MD_i) in order to send a valid login request to the server S. In order to make a valid login request message, say $\text{Msg}_1' = \langle M_1', M_2', M_3', M_4', T_1' \rangle$ on behalf of U_i, $\mathcal{A}$ can generate the current timestamp T_1' and a random nonce r_1'. However, without having the secret credentials RID_i, RID_s, RID_{DR_j}, and σ_i and master secret key MK_{U_i} of U_i, it is a difficult task for $\mathcal{A}$ to calculate TC_{U_i}, M_1', M_2', M_3', and M_4'. Therefore, $\mathcal{A}$ can not generate the valid Msg_1' on behalf of U_i. From the above explanation, it is clear that the proposed scheme is resilient against user impersonation attack.

3) Server Impersonation Attack: To launch this attack, suppose $\mathcal{A}$ generates the current timestamp T_2', and the random nonces r_1' and r_2'. $\mathcal{A}$ can try to send the message $\text{Msg}_2' = \langle M_5', M_6', M_7', T_2' \rangle$ to DR_j on behalf of S. However, without having the knowledge of RID_i, RID_s, and RID_{DR_j}, and master secret key MK_{DR_j} of DR_j, it is computationally hard for $\mathcal{A}$ to calculate TC_{DR_j}, M_5', M_6', and M_7'. Therefore, $\mathcal{A}$ is not able to create Msg_2' on behalf of S. Hence, the proposed scheme provides protection against server impersonation attack.

4) Drone Impersonation Attack: For this attack, suppose $\mathcal{A}$ generates the current timestamps T_2' and T_3', and random nonces r_1', r_2', and r_3'. $\mathcal{A}$ then tries to create and send a valid message, say $\text{Msg}_3' = \langle M_{10}', M_{11}', M_{12}' T_3' \rangle$ to U_i on behalf of DR_j. However, without the secret credentials RID_i, RID_s, RID_{DR_j}, and master secret key MK_{DR_j} of DR_j, it is also difficult task for $\mathcal{A}$ to calculate TC_{DR_j}, M_{10}', M_{11}', and M_{12}'. Therefore, $\mathcal{A}$ is not able to make the message Msg_3' on behalf of DR_j. As a result, the proposed scheme also provides protection against drone impersonation attack.

5) Anonymity and Untraceability: Random nonces and current timestamps are used in various exchanged messages Msg_1, Msg_2, and Msg_3 during the login, and authentication and key agreement phases. Due to this reason, the messages Msg_1, Msg_2, and Msg_3 are distinct for each session. Therefore, an adversary $\mathcal{A}$ can not trace the user, server as well as drone. Moreover, these messages do not involve directly identities or pseudo-identities RID_i, RID_s, and RID_{DR_j}, and these are embedded in the collision-resistant cryptographic one-way hash function $h(\cdot)$. Therefore, our scheme provides both anonymity and untraceability properties.

6) Password Change Attack: Suppose an adversary $\mathcal{A}$ has lost/stolen mobile device MD_i of a registered user U_i. $\mathcal{A}$ can then extract the information $\{\text{RID}_i', \text{RID}_{\text{DR}_j}', \text{RID}_s', \text{TC}_{U_i}', A', B, C, \tau_i, \text{Gen}(\cdot), \text{Rep}(\cdot), h(\cdot), t\}$ stored in MD_i's memory by applying the power analysis attacks [19]. However, $\mathcal{A}$ requires to input correct ID_i, PW_i and BIO_i of U_i in order to update password PW_i to another fake password PW_i'. Without these valid user credentials, local password and biometric verification will fail at MD_i. Therefore, $\mathcal{A}$ will not be able to provide any other fake password PW_i' and fake biometrics in order to change the password PW_i to PW_i'. Hence, our scheme is free against password change attack.

7) Resilience Against Drone Capture Attack: As discussed in [28], we also measure the resilience against drone capture attack of the proposed scheme in IoD environment as follows. Assume that c drones are physically captured by an attacker $\mathcal{A}$. It is then measured as the total secure communications compromised by a capture of c drones *not including* the communication in which the compromised drones are directly involved. Let $P_e(c)$ denote the probability that $\mathcal{A}$ can decrypt the secure communication between a user U_i and a noncompromised drone DR_j' when c drones are already compromised. If $P_e(c) = 0$, a user authentication scheme is known as unconditionally secure against drone capture attack. By physically capturing a drone DR_j, $\mathcal{A}$ can extract the valuable information $\{\text{TID}_{\text{DR}_j}, \text{RID}_{\text{DR}_j}, \text{TC}_{\text{DR}_j}, \mathcal{P}(\text{TID}_{\text{DR}_j}, y)\}$ from its memory with the help of power analysis attacks [19]. Note that TID_{DR_j}, RID_{DR_j}, TC_{DR_j}, and $\mathcal{P}(\text{TID}_{\text{DR}_j}, y)$ are different for all drones, and these are generated by S. Therefore, by capturing DR_j, $\mathcal{A}$ can only compromise the session key between that a user U_i and DR_j. Furthermore, the session keys between that user U_i and other noncompromised drones can not be compromised by $\mathcal{A}$. Then, compromise of a drone does not result in compromising secure communications among a user and other noncompromised drones. As a result, our scheme becomes unconditionally secure against drone capture attack.

8) Denial-of-Service Attack: In our scheme, during the login phase as well as password and biometric update phase, if a legal user U_i enters his/her incorrect ID_i and/or PW_i, it is locally verified by checking the condition $C^* = C$ (step L1 in Section III-C) or $C^o = C$ (step PB1 in Section III-E). The login request of the user U_i is sent to the server S only after successful verification. Also, the password and biometric update takes place only after successful verification of old password and biometrics in password and biometric update phase. As a result, the proposed scheme is secure against such kinds of denial-of-service attacks.

9) Stolen Mobile Device Attack: Suppose the mobile device MD_i of a genuine user U_i is lost or stolen by an adversary $\mathcal{A}$. $\mathcal{A}$ can then extract all information $\{\text{RID}_i', \text{RID}_{\text{DR}_j}', \text{RID}_s', \text{TC}_{U_i}', A', B, C, \tau_i, \text{Gen}(\cdot), \text{Rep}(\cdot), h(\cdot), t\}$ stored in the memory of MD_i using the power analysis attacks. To correctly guess ID_i and PW_i from the extracted information B and C, $\mathcal{A}$ needs to know both secrets n and σ_i. Therefore, it is computationally infeasible for $\mathcal{A}$ to correctly guess both ID_i and PW_i. Hence, the proposed scheme is secure against stolen mobile device attack.

TABLE II
COMPARISON OF COMMUNICATION OVERHEADS

Protocol	No. of messages	No. of bits
Our	3	1696
Challa *et al.*[15]	3	2528
Turkanovic *et al.* [13]	4	2720

TABLE III
COMPARISON OF STORAGE OVERHEADS

Protocol	User side	Server side	Sensing device/ Sensor/Drone side
Our	1288 bits	$320n + (m+1)\log_2(p)$ +800 bits	$480 + (m+1)\log_2(p)$ bits
[15]	488 bits	$1288 + 640n$ bits	1600 bits
[13]	768 bits	$320n + 160$ bits	320 bits

V. PERFORMANCE COMPARISON

This section compares the performance of the proposed scheme with the related existing IoT-based schemes, such as Challa *et al.*'s [15] scheme and Turkanović *et al.*'s [13] scheme.

A. Communication Overheads Comparison

For communication overhead comparison, the identity, random number/nonce, hash output (message digest), and timestamp are considered as 160, 128, 160 (if we apply the secure hash algorithm (SHA-1) [29]), and 32 bits, respectively. We further assume that 160-bit elliptic curve cryptography (ECC) security is same as that for RSA public key cryptosystem [30]. Thus, an ECC point $P = (P_x, P_y)$ requires $(160 + 160) = 320$ bits in Challa *et al.*'s [15] scheme. The communication costs among the proposed scheme and other schemes provided in Table II clearly shows that our scheme outperforms in terms of communication cost as compared to Challa *et al.*'s [15] scheme and Turkanović *et al.*'s [13] scheme.

B. Storage Overheads Comparison

For storage overheads comparison, we consider the pre-deployment and user registration phases for the storage needed to store the credentials in user's smart card/mobile device, server/gateway node and drone/sensing node. In the proposed scheme, a drone DR_j needs to store the credentials $\{\mathrm{TID}_{\mathrm{DR}_j}, \mathrm{RID}_{\mathrm{DR}_j}, \mathrm{TC}_{\mathrm{DR}_j}, \mathcal{P}(\mathrm{TID}_{\mathrm{DR}_j}, y)\}$ which require $(160 + 160 + 160 + (m+1)\log_2(p)) = 480 + (m+1)\log_2(p)$ bits, the server S needs to store the credentials $\{\{(\mathrm{RID}_{\mathrm{DR}_j}, \mathrm{TC}_{\mathrm{DR}_j}) | 1 \leq j \leq n\}, \mathcal{P}(x, y), k, \mathrm{ID}_i, \mathrm{RID}_i, \mathrm{TC}_{U_i}, \mathrm{RID}_s\}$ which require $320n + (m + 1)\log_2(p) + 800$ bits and a user U_i's mobile device MD_i requires to store the credentials $\{\mathrm{RID}'_i, \mathrm{RID}'_{\mathrm{DR}_j}, \mathrm{RID}'_s, \mathrm{TC}'_{U_i}, A', B, C, \tau_i, t\}$ which need 1288 bits, where n is the number of drones (in the proposed scheme) or the number of sensing devices (in other schemes) deployed in the network, and biometric public reproduction parameter τ_i and error tolerance threshold t are 160 and 8 bits, respectively, and m is the degree of symmetric bivariate polynomial whose coefficients are chosen from GF(p). Table III shows the comparison of storage costs among the proposed scheme and other schemes. It is noted that though the proposed scheme needs more storage costs for user's mobile device and server, it is justified because the proposed scheme provides more security and functionality features as compared to those for Turkanović *et al.*'s [13] scheme (see Table V).

TABLE IV
COMPARISON OF COMPUTATION OVERHEADS

Protocol	User side	Server side	Sensing device/ Sensor/Drone side	Total cost
Our	$1T_{fe} + 16T_h \approx 22.22ms$	$8T_h \approx 2.56ms$	$7T_h \approx 2.24ms$	$1T_{fe} + 31T_h \approx 27.02ms$
[15]	$1T_{fe} + 5T_{ecm} + 5T_h \approx 104.20ms$	$5T_{ecm} + 4T_h \approx 86.78ms$	$4T_{ecm} + 3T_h \approx 69.36ms$	$1T_{fe} + 14T_{ecm} + 12T_h \approx 260.34ms$
[13]	$7T_h \approx 2.24ms$	$5T_h \approx 1.6ms$	$7T_h \approx 2.24ms$	$19T_h \approx 6.08ms$

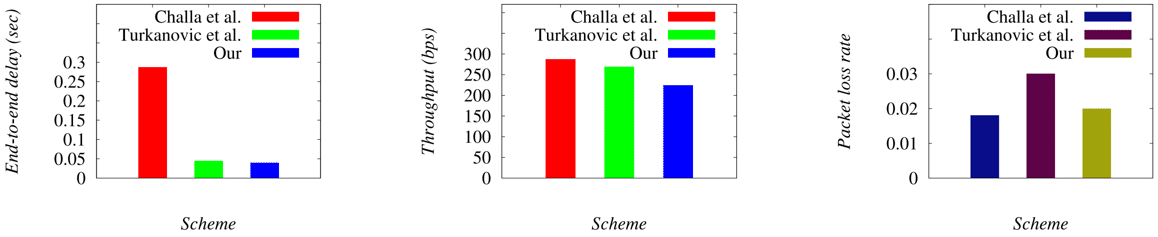
TABLE V
COMPARISON OF FUNCTIONALITY AND SECURITY FEATURES

Feature	Turkanovic *et al.* [13]	Challa *et al.*[15]	Our
FSF_1	✓	✓	✓
FSF_2	×	✓	✓
FSF_3	×	✓	✓
FSF_4	×	✓	✓
FSF_5	✓	✓	✓
FSF_6	×	✓	✓
FSF_7	✓	✓	✓
FSF_8	✓	✓	✓
FSF_9	✓	✓	✓
FSF_{10}	✓	✓	✓
FSF_{11}	×	✓	✓
FSF_{12}	✓	✓	✓
FSF_{13}	✓	✓	✓
FSF_{14}	×	✓	✓
FSF_{15}	×	✓	✓
FSF_{16}	×	×	✓
FSF_{17}	×	✓	✓

Note: FSF_1: user anonymity property; FSF_2: privileged-insider attack; FSF_3: off-line password guessing attack; FSF_4: stolen smart card/mobile device attack; FSF_5: denial-of-service attack; FSF_6: user impersonation attack; FSF_7: replay attack; FSF_8: man-in-the middle attack; FSF_9: mutual authentication; FSF_{10}: session key agreement; FSF_{11}: untraceability property; FSF_{12}: resilience against sensor node/sensing device/drone capture attack; FSF_{13}: server independent password update phase; FSF_{14}: sensor node/sensing device/drone impersonation attack; FSF_{15}: support biometric update phase; FSF_{16}: key management phase; FSF_{17}: provide formal security verification using AVISPA tool.
×: insecure against a particular attack or does not support a particular feature;
✓: secure against a particular attack or supports a particular feature.

C. Computation Overheads Comparison

Let T_h, T_{ecm}, and T_{fe} denote the time needed for executing hash function, ECC point multiplication, and fuzzy extractor function (Gen(·)/Rep(·)). Based on the results used in [31], we have $T_h \approx 0.00032$ s, $T_{ecm} \approx 0.0171$ s and $T_{fe} \approx T_{ecm}$, that is, $T_{fe} \approx 0.0171$ s. The comparison results of computation overheads among different schemes reported in Table IV shows that our scheme performs better than that for Challa *et al.*'s [15]. Our scheme requires more computation cost as compared to that for Turkanović *et al.*'s [13] scheme. However, computation cost needed for a drone in our scheme remains same as that for Turkanović *et al.*'s [13] scheme. This is also justified as our scheme provides more security and functionality features as compared to those for Turkanović *et al.*'s [13] scheme (see Table V).



(a) (b) (c)

Fig. 6. Network performance comparison. (a) EED. (b) Throughput. (c) Packet loss rate.

TABLE VI
SIMULATION PARAMETERS

Parameter	Description
Platform	Ubuntu 14.04 LTS
Tool used	NS2 2.35
Area	$400m \times 200m \times 15m$
Number of gateway nodes (GWN)	1
Number of users (U_i)	3
Number of $DR_j/SD_j/S_j$s	50
Mobility of U_is	2 mps, 15 mps
Mobility of DR_js	25 mps, 30 mps, 35 mps, 40 mps
Communication range of DR_js	200m
Simulation time	1800 seconds

TABLE VII
EXCHANGED MESSAGES BETWEEN ENTITIES USED IN SIMULATION

Exchanged messages between entities	Challa *et al.* [15]	Turkanovic *et al.* [13]	Our
$U_i \rightarrow GWN$	992 bits	672 bits	672 bits
$GWN \rightarrow SD_j/S_j/DR_j$	1024 bits	1024 bits	512 bits
$SD_j/S_j/DR_j \rightarrow GWN$	–	576 bits	–
$GWN \rightarrow U_i$	–	448 bits	–
$SD_j/DR_j \rightarrow U_i$	512 bits	–	512 bits

D. Security and Functionality Features Comparison

Finally, the security and functionality features of our scheme are also compared to those for other schemes in Table V. Turkanović *et al.*'s [13] scheme does not support the features FSF$_2$–FSF$_4$, FSF$_6$, FSF$_{11}$, and FSF$_{14}$–FSF$_{17}$, whereas Challa *et al.*'s [15] scheme does not support FSF$_{16}$. On the other hand, our scheme is significantly better than other schemes, which is evident from Table V.

VI. Practical Perspective: NS2 Simulation Study

In this section, we discuss the practical perspective of the proposed scheme, and other recently proposed related schemes, such as Challa *et al.*'s [15] scheme and Turkanović *et al.*'s [13] scheme [13] using widely used NS2 simulation.

A. Simulation Parameters

We have done the simulation of the proposed scheme, Challa *et al.*'s [15] scheme and Turkanović *et al.*'s [13] scheme on Ubuntu 14.04 LTS platform using the NS2 2.35 simulator [17], [32]. The network parameters used in the simulation are provided in Table VI. The simulation time is taken as 1800 s (30 min). DR$_j$, SD_j, and S_j represent jth drone, jth smart device, and jth sensor node in the existing schemes [13], [15]. We have considered three different types of mobility, i.e., 25, 30, 35, and 40 mps for SD_j/S_j/DR$_j$ whenever it is applicable. The users U_i move with different mobility, i.e., 2 and 15 mps. Moreover, we have taken one gateway node for all schemes. The messages exchanged between various entities and their communication costs in bits in different schemes are shown in Table VII.

B. Discussion on Simulation Results

During the experimentation, we have computed different network performance parameters, such as throughput (in bps), EED (in seconds), and packet loss rate. The impacts on these network parameters are discussed below.

1) Impact on Throughput: Throughput is computed as the number of bits transmitted per unit time, which is mathematically represented as $(n_r \times |pkt|)/T_d$, where T_d is the total time (in seconds), $|pkt|$ the size of a packet, and n_r the total number of received packets. Note that we have considered the simulation time as 1800 s, which is the actual total time. In Fig. 6(b), the throughput values for the proposed scheme, Challa *et al.*'s [15] scheme and Turkanović *et al.*'s [13] scheme are 223.89, 286.84, and 268.73 bps, respectively. The throughput of our scheme is less than that for other schemes [13], [15]. This is because the proposed scheme needs less communication cost due to small sized messages used for authentication as compared to other schemes (see Table VII).

2) Impact on End-to-End Delay: The EED is the average time taken by the data packets to arrive at a destination from a source. EED can be mathematically represented as $\sum_{i=1}^{n_p}(T_{\text{rec}_i} - T_{\text{send}_i})/n_p$, where T_{rec_i} and T_{send_i} are the receiving and sending time of a packet i, respectively, and n_p the total number of packets. From the results shown in Fig. 6(a), it is noticed that EEDs are 0.03985, 0.28683, and 0.04436 s for the proposed scheme, Challa *et al.*'s [15] scheme and Turkanović *et al.*'s [13] scheme, respectively. EED of the proposed scheme is less than that for other schemes [13], [15]. This is because the proposed scheme uses small sized messages for authentication and as a result, it requires less EED as compared to the other schemes.

3) Impact on Packet Loss Rate: Packet loss rate is another important network parameter that is measured by the number of packets loss per unit time and it can be estimated as (n_{lp}/T_d), where T_d is the total time (in seconds) and n_{lp}

the total number of lost packets. It is expected for a reliable network communication that the packet loss rate should be as less as possible to the extent. Fig. 6(c) illustrates the packet loss rates under different scenarios among the proposed scheme and other existing schemes of Challa *et al.* [15] and Turkanović *et al.* [13]. It is observed that the proposed scheme has low packet loss rate as compared to that for Turkanović *et al.*'s [13] scheme, whereas it has the similar packet loss rate as compared to that for Challa *et al.*'s [15] scheme.

VII. Conclusion

The IoD is an emerging field as it has wide-range of applications from military to civilian. However, there remains several security and privacy issues in the IoD deployment. To address these issues in IoD applications, we presented a novel authentication and key agreement scheme between a user and an accessed drone with the help of the server. The session key established after successful mutual authentication between a user and a drone helps them to communicate securely so that various known attacks are prevented by an adversary. The security analysis including the formal security verification using the widely accepted AVISPA tool provide an evidence that the proposed scheme can withstand several known attacks against an adversary. The NS2 simulation study performed on the proposed scheme and other related schemes demonstrated the practicality of the scheme. Finally, the proposed scheme is efficient in communication and computation, and also provides more security and functionality features as compared to those for other related schemes.

Acknowledgment

The authors would like thank the anonymous reviewers for their valuable feedback on this paper, which helped to improve its quality and presentation.

References

[1] *Unmanned Aerial Vehicle*. Accessed: Jun. 2018. [Online]. Available: http://www.wikiwand.com/en/Unmanned_aerial_vehicle

[2] *Drones*. Accessed: Jun. 2018. [Online]. Available: https://www.h3dynamics.com/products/

[3] H. Nagel, G. Bondt, and B. Custers, "Drone technology: Types, payloads, applications, frequency spectrum issues and future developments," in *The Future of Drone Use Opportunities and Threats From Ethical and Legal Perspectives*, vol. 27. The Hague, The Netherlands: Springer Gabler Verlag, 2016, ch. 2, pp. 21–45.

[4] M. Wazid, A. K. Das, and J.-H. Lee, "Authentication protocols for the Internet of Drones: Taxonomy, analysis and future directions," *J. Ambient Intell. Humanized Comput.*, pp. 1–10, Aug. 2018, doi: 10.1007/s12652-018-1006-x.

[5] M. Gharibi, R. Boutaba, and S. L. Waslander, "Internet of Drones," *IEEE Access*, vol. 4, pp. 1148–1162, 2016.

[6] B. Vergouw, H. Nagel, G. Bondt, and B. Custers, *Drone Technology: Types, Payloads, Applications, Frequency Spectrum Issues and Future Developments*. The Hague, The Netherlands: T.M.C. Asser Press, 2016, pp. 21–45.

[7] *Sensor Technology for Industrial Drones*. Accessed: Sep. 2018. [Online]. Available: https://www.azosensors.com/article.aspx?ArticleID=973

[8] R. J. Hall, "An Internet of Drones," *IEEE Internet Comput.*, vol. 20, no. 3, pp. 68–73, May/Jun. 2016.

[9] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Progr. Aerosp. Sci.*, vol. 91, pp. 99–131, May 2017.

[10] J. Won, S.-H. Seo, and E. Bertino, "Certificateless cryptographic protocols for efficient drone-based smart city applications," *IEEE Access*, vol. 5, pp. 3721–3749, 2017.

[11] J. Won, S.-H. Seo, and E. Bertino, "A secure communication protocol for drones and smart objects," in *Proc. 10th ACM Symp. Inf. Comput. Commun. Security (ASIA CCS)*, 2015, pp. 249–260.

[12] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of Things services: Comprehensive survey and future perspectives," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016.

[13] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Netw.*, vol. 20, pp. 96–112, Sep. 2014.

[14] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Netw.*, vol. 36, pp. 152–176, Jan. 2016.

[15] S. Challa *et al.*, "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.

[16] AVISPA. *Automated Validation of Internet Security Protocols and Applications*. Accessed: Jun. 2018. [Online]. Available: http://www.avispa-project.org/

[17] *The Network Simulator-NS-2*. Accessed: Apr. 2018. [Online]. Available: http://www.isi.edu/nsnam/ns/

[18] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.

[19] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[20] A. K. Das and I. Sengupta, "An effective group-based key establishment scheme for large-scale wireless sensor networks using bivariate polynomials," in *Proc. 3rd IEEE Int. Conf. Commun. Syst. Softw. Middleware (COMSWARE)*, Bengaluru, India, 2008, pp. 9–16.

[21] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Adv. Cryptol. (Eurocrypt)*, vol. 3027. Interlaken, Switzerland, 2004, pp. 523–540.

[22] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[23] C. Blundo *et al.*, "Perfectly-secure key distribution for dynamic conferences," in *Proc. 12th Annu. Int. Cryptol. Conf. (CRYPTO)*, vol. 740. Santa Barbara, CA, USA, 1993, pp. 471–486.

[24] D. von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proc. 3rd APPSEM II Appl. Semantics II Workshop (APPSEM)*, 2005, pp. 1–17.

[25] AVISPA. *SPAN, The Security Protocol ANimator for AVISPA*. Accessed: Jun. 2018. [Online]. Available: http://www.avispa-project.org/

[26] C.-T. Li and M.-S. Hwang, "An efficient biometric-based remote user authentication scheme using smart cards," *J. Netw. Comput. Appl.*, vol. 33, no. 1, pp. 1–5, 2010.

[27] A. K. Das, "Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards," *IET Inf. Security*, vol. 5, no. 3, pp. 145–151, 2011.

[28] A. K. Das, P. Sharma, S. Chatterjee, and J. K. Sing, "A dynamic password-based user authentication scheme for hierarchical wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 35, no. 5, pp. 1646–1656, 2012.

[29] Secure Hash Standard, *FIPS PUB 180-1, National Institute of Standards and Technology (NIST)*, U.S. Dept. Commerce, Washington, DC, USA, Apr. 1995. Accessed: Mar. 2018, doi: 10.6028/NIST.FIPS.180-4.

[30] S. Vanstone, "Responses to NIST's proposal," *Commun. ACM*, vol. 35, no. 7, pp. 50–52, 1992.

[31] V. Odelu, A. K. Das, and A. Goswami, "An efficient biometric-based privacy-preserving three-party authentication with key agreement protocol using smart cards," *Security Commun. Netw.*, vol. 8, no. 18, pp. 4136–4156, 2015.

[32] J. Wang. *NS-2 Tutorial*. Accessed: Jul. 2018. [Online]. Available: http://www.cs.virginia.edu/c̃s757/slidespdf/cs757-ns2-tutorial1.pdf

Mohammad Wazid (S'17) received the M.Tech. degree in computer network engineering from Graphic Era University, Dehradun, India, and the Ph.D. degree in computer science and engineering from the International Institute of Information Technology Hyderabad, Hyderabad, India.

He is currently a Post-Doctoral Researcher with the Cyber Security and Networks Laboratory, Innopolis University, Innopolis, Russia. He has authored or co-authored over 55 papers in international journals and conferences. His current research interests include remote user authentication, Internet of Things, and cloud computing.

Dr. Wazid was a recipient of the University Gold Medal and the Young Scientist Award by UCOST, Department of Science and Technology, Government of Uttarakhand, India.

Ashok Kumar Das (M'17–SM'18) received the M.Sc. degree in mathematics, the M.Tech. degree in computer science and data processing, and the Ph.D. degree in computer science and engineering from IIT Kharagpur, Kharagpur, India.

He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad, India. He has authored over 175 papers in international journals and conferences, including over 155 reputed journal papers. His current research interests include cryptography, wireless sensor network security, hierarchical access control, security in vehicular ad hoc networks, smart grid, Internet of Things, cyber-physical systems and cloud computing, and remote user authentication.

Dr. Das was a recipient of the Institute Silver Medal from IIT Kharagpur. He is an Editorial Board member of the *KSII Transactions on Internet and Information Systems* and the *International Journal of Internet Technology and Secured Transactions* (Inderscience), and a Guest Editor for *Computers and Electrical Engineering* (Elsevier) for the "Special Issue on Big Data and IoT in e-Healthcare," and has served as a Program Committee member for several international conferences.

Neeraj Kumar (M'16–SM'17) received the Ph.D. degree in computer science and engineering from Shri Mata Vaishno Devi University, Katra, India, in 2009.

He was a Post-Doctoral Research Fellow with Coventry University, Coventry, U.K. He is currently an Associate Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, India. He has authored over 200 technical research papers published in leading journals and conferences of the IEEE, Elsevier, Springer, and Wiley.

Dr. Kumar is an Editorial Board member of *IEEE Communications Magazine*, the *Journal of Network and Computer Applications* (Elsevier), and the *International Journal of Communication Systems* (Wiley).

Athanasios V. Vasilakos (M'00–SM'11) is currently a Professor with the Luleå University of Technology, Luleå, Sweden. He has authored or co-authored over 600 technical research papers in leading journals and conferences in his areas of research.

Dr. Vasilakos was a recipient of the Highly Cited Researcher Award by Web of Science in 2017 and 2018. He has served or is serving as an Editor for several technical journals, such as the IEEE Transactions on Network and Service Management, the IEEE Transactions on Cloud Computing, the IEEE Transactions on Cybernetics, the IEEE Transactions on Nanobioscience, the IEEE Transactions on Cloud Computing, *ACM Transactions on Autonomous and Adaptive Systems*, and the IEEE Journal on Selected Areas in Communications. He is also the General Chair of the European Alliances for Innovation.

Joel J. P. C. Rodrigues (S'01–M'06–SM'06) received the five-year B.Sc. degree (licentiate) in informatics engineering from the University of Coimbra, Coimbra, Portugal, the M.Sc. and Ph.D. degrees in informatics engineering from the University of Beira Interior (UBI), Covilhã, Portugal, and the Habilitation degree in computer science and engineering from the University of Haute Alsace, Mulhouse, France.

He is a Professor with the National Institute of Telecommunications (Inatel), Santa Rita do Sapucaí, Brazil, and a Senior Researcher with the Instituto de Telecomunicações, Lisbon, Portugal. He has also been a Professor with UBI, and a Visiting Professor with the University of Fortaleza, Fortaleza, Brazil. He received the Academic Title of Aggregated Professor of Informatics Engineering at UBI. He has authored or co-authored over 650 papers in refereed international journals and conferences, 3 books, and 2 patents.

Dr. Rodrigues is the Editor-in-Chief of the *International Journal on E-Health and Medical Communications*, and an Editorial Board member of several journals. He is the Leader of NetGNA Research Group, the President of the Scientific Council at ParkUrbis Covilhã Science and Technology Park, the Past-Chair of the IEEE ComSoc TCs on eHealth and Communications Software, and a Steering Committee member of the IEEE Life Sciences Technical Community. He is member of the Internet Society, an IARIA Fellow, and a Senior Member of the ACM.