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

by

Ashok Kumar Das

in

IEEE INTERNET OF THINGS JOURNAL

Report No: IIIT/TR/2019/-1

Centre for Security, Theory and Algorithms
International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2019
AUTHOR QUERIES
AUTHOR PLEASE ANSWER ALL QUERIES

PLEASE NOTE: We cannot accept new source files as corrections for your paper. If possible, please annotate the PDF proof we have sent you with your corrections and upload it via the Author Gateway. Alternatively, you may send us your corrections in list format. You may also upload revised graphics via the Author Gateway.

AQ1: Author: Please confirm or add details for any funding or financial support for the research of this article.
AQ2: Please provide the postal code for “National Institute of Telecommunications (Inatel), Santa Rita do Sapucaí, Brazil, Brazil,” “Instituto de Telecomunicações, Lisbon, Portugal,” and “University of Fortaleza, Fortaleza, Brazil.”
AQ3: Please provide the department name for National Institute of Telecommunications (Inatel), Instituto de Telecomunicações, Lisbon, Portugal, and University of Fortaleza, Fortaleza, Brazil.
AQ4: Please confirm if the location and publisher information for Reference [3] is correct as set.
AQ5: Please provide the volume number for Reference [4].
AQ6: Please confirm the volume number for Reference [21].
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.

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 in 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 sitting 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 this 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...
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 user 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 applications 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 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 \( A \), 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 \( A \). Therefore, \( A \) 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 will not be compromised by \( A \).

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 ID\( _{DRj} \) for each DR\( _j \), and computes its pseudo identity as RID\( _{DRj} = h(ID_{DRj} || k) \). S further chooses 160-bit master key MK\( _{DR} \) corresponding to DR\( _j \) and calculates its temporal credential as TCD\( _{DR} = h(ID_{DR} || MK_{DR} || RTS_{DRj}) \), where RTS\( _{DRj} \) is the registration timestamp of DR\( _j \).

For pairwise key establishment between two neighboring drones (see Section III-G), S selects a symmetric bivariate polynomial \( P(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{n} g_{ij} x^i y^j \in GF(p)[x, y] \) of degree \( m \) over a finite field (Galois field) GF\( (p) \), where the coefficients \( g_{ij} \)’s are taken from 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 \( P(x, y) = x^4 + 3x^3 + 2x^2y^2 + 3y^4 + y^2 \) over GF\( (5) \) as \( P(x, y) = y^4 + 3y^2 + 2x^2 + 3x^3 + x^4 \). S then generates a temporary identity TID\( _{DRj} \), corresponding to the pseudo-identity RID\( _{DRj} \), and computes its polynomial share \( P(TID_{DRj}, y) \) which is a univariate polynomial of degree \( m \) in GF\( (p) \). Note that to store \( P(TID_{DRj}, y) \), the storage space required in DR\( _j \)’s memory is \( (m + 1)\log_2(p) \) bits as the coefficients are from GF\( (p) \).

S then stores the information \( \{TID_{DRj}, RID_{DRj}, TCD_{DRj}, P(TID_{DRj}, y)\} \) in the memory of DR\( _j \) and then deploys DR\( _j \) in the deployment field, whereas S keeps the information \( \{RID_{DRj}, TCD_{DRj}, P(x, y), k\} \) 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 $\langle ID_i \rangle$ to $S$ securely. After receiving registration request, $S$ calculates $U_i$’s pseudo identity as $RID_i = h(ID_i||k)$ using its corresponding 160-bit secret number $k$. $S$ also computes its own pseudo identity as $RID_S = h(ID_S||k)$. $A = h(RID_i||ID_i)$ and the temporal credential of $U_i$ as $TC_{U_i} = h(ID_i||MK_U||RTS_{U_i})$, where $MK_U$ 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 $S$. Then sends the registration reply message $(RID_i, RID_{DR_i}, RID_S, TC_{U_i}, A)$ 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 BISO 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 BISO as input, and returns $\sigma_i = [0, 1]^l$ as the biometric secret key of length $l$ bits and $t_\sigma$ as the public reproduction parameter, that is, $Gen(BISO_i) = (\sigma_i, t_\sigma)$.

2) $Rep$: It is a deterministic function, whose inputs are the user biometrics, say BISO$i$, and $t_\sigma$, provided the Hamming distance between BISO$'_i$ and the original previously entered biometrics BISO$\hat{i}$ is less than or equal to an error tolerance threshold value $t$. The output is the original biometric key $\sigma_i$ that is, $\sigma_i = Rep(BISO_i, t_\sigma)$.

MD$i$ generates the biometric secret key $\sigma_i$ and its corresponding public parameter $t_\sigma$ as $Gen(BISO_i) = (\sigma_i, t_\sigma)$.

Step R3: MD$i$ generates 160-bit secret number $n$ for $U_i$ and computes $RID'_i = RID_i \oplus h(PW_i||\sigma_i)$, $RID_{DR_i} = RID_{DR_i} \oplus h(ID_i||PW_i||\sigma_i)$, $TC'_{U_i} = TC_{U_i} \oplus h(ID_i||\sigma_i)$, masked password $RPW_i = h(PW_i||n)$, $RID' = RID_i \oplus h(RID_{DR_i}||\sigma_i)$. MD$i$ further computes the following:

$$A' = A \oplus h(RID_{DR_i}||\sigma_i||PW_i)$$  
$$B = n \oplus h(PW_i||ID_i||\sigma_i)$$  
$$C = h(A||RID_{DR_i}||RPW_i||\sigma_i).$$

Finally, MD$i$ stores the information $\langle RID'_i, RID'_{DR_i}, RID'_S, TC'_{U_i}, A', B, C, t_\sigma, Gen() ,Rep(), h(), \sigma_i \rangle$ in its memory. $S$ also stores $\langle ID_i, RID_i, TC_{U_i}, RID_{DR_i} \rangle$ in its database.

The user registration phase is briefed in Fig. 3.

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 inputs his/her biometrics BISO$'_i$ at the sensor of MD$i$. MD$i$ then calculates biometric key $\sigma_i' = Rep(BISO'_i, t_\sigma)$ provided that the Hamming distance between the original biometrics BISO$'_i$ at the time of registration and the recent entered BISO$'_i$ is less than or equal to the error tolerance threshold value $t$. Next, MD$i$ calculates the following:

$$RID_i = RID'_i \oplus h(PW'_i||\sigma'_i)$$  
$$RID_{DR_i} = RID_{DR'_i} \oplus h(ID_i||PW'_i||\sigma'_i)$$  
$$TC_{U_i} = TC'_{U_i} \oplus h(ID_i||\sigma'_i)$$  
$$RID_i = RID'_i \oplus h(RID_{DR_i}||\sigma'_i)$$  
$$n = B \oplus h(PW'_i||ID_i||\sigma'_i)$$  
$$RPW'_i = h(PW'_i||n).$$

MD$i$ further computes $A = A' \oplus h(RID_{DR_i}||\sigma'_i||PW'_i)$ and $C = h(A||RID_{DR_i}||RPW'_i||\sigma'_i)$. 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:

$$M_1 = RID_i \oplus h(RID_{DR_i}||T_1)$$  
$$M_2 = RID_{DR_i} \oplus h(TC_{U_i}||ID_i||T_1)$$  
$$M_3 = h(RID_i||TC_{U_i}||T_1) \oplus r_1$$  
$$M_4 = h(ID_i||RID_{DR_i}||TC_{U_i}||T_1).$$

Finally, MD$i$ sends the login request message $Msg_1 = \langle M_1, M_2, M_3, M_4, T_1 \rangle$ 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 $DR_j$. After that $U_i$ and $DR_j$ establish a session key for secure communication between them.

Step AK1: $S$ first checks the timelines of $T_1$ by the condition $|T_1 - T^*_1| \leq \Delta T$, where the maximum transmission delay is presented by $\Delta T$ 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 $RID = M_1 \oplus h(RID_{DR_i}||T_1)$, and fetches $ID_i$ and $TC_{U_i}$ corresponding to the computed $RID$. $S$ further computes the
following:

\[ \text{Ri}_j = M_2 \oplus h(TC_{ij}||ID||T_1) \]
\[ \text{r}_i = M_3 \oplus h(\text{Ri}_j||TC_{ij}||T_1) \]
\[ M'_4 = h(ID_{ij}||\text{Ri}_j||TC_{ij}||\text{r}_i ||T_1) \]

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:

\[ M_5 = h(TC_{DRj}||\text{Ri}_j) \oplus h(\text{Ri}_j||\text{r}_i ||T_2) \]
\[ M_6 = h(TC_{DRj}||T_2) \oplus \text{Ri}_j \]
\[ M_7 = h(\text{Ri}_j||TC_{DRj}||\text{Ri}_j||\text{r}_i ||T_2) \]

Then sends authentication request message \( \text{Msg}_2 = \langle M_5, M_6, M_7, T_2 \rangle \) 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 \( |T_2 - T_j| \leq \Delta T \), where \( T_j \) is the reception time of the message. If the timeliness matches, \( DR_j \) proceeds to calculate the following:

\[ \text{Ri}_i = M_8 \oplus h(TC_{DRj}||T_2) \]
\[ M_8 = M_5 \oplus h(TC_{DRj}||\text{Ri}_j) \]
\[ M_9 = h(\text{Ri}_j||TC_{DRj}||M_8 ||T_2) \]

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 \( M_{10} = h(\text{Ri}_j||\text{Ri}_j ||T_3) \oplus r_3 \), the session key \( SK_{ij} = h(M_8 ||r_3 ||\text{Ri}_j||\text{Ri}_j) \) shared with \( U_i \),
\[ M_{11} = h(\text{Ri}_j||\text{Ri}_j ||T_3) \oplus M_8 \] and \( M_{12} = h(SK_{ij}||T_3) \).

\( DR_j \) directly sends authentication reply message \( \text{Msg}_3 = \langle M_{10}, M_{11}, M_{12}, T_3 \rangle \) 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
\[ |T_3 - T_j| \leq \Delta T \]
where \( T_j \) is the reception time of the message. \( U_i \) calculates \( r_3 = M_{10} \oplus h(\text{Ri}_j||\text{Ri}_j ||T_3) \),
\[ M'_8 = M_{11} \oplus h(\text{Ri}_j||\text{Ri}_j ||r_3 ||T_3) \]
the session key \( SK_{ij}' = h(M'_8 ||r_3 ||\text{Ri}_j||\text{Ri}_j) \) shared with \( DR_j \), and \( M_{13} = h(SK_{ij}'||T_3) \), 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 \( SK_{ij}' = SK_{ij}' \) for future 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. \( U_i \) 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 \( PW_i \) 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 \( \sigma_i^o = \text{Rep}(BIO_i^o, \tau_i) \) provided that the Hamming distance between the original biometrics \( BIO_i^o \) at the time of registration and the recent entered \( BIO_i^o \) is less than or equal to the error tolerance threshold value \( \tau_i \). \( MD_i \) further calculates the following:

\[ \text{Ri}_i = \text{Ri}_i \oplus h(\text{PW}_i ||\sigma_i^o) \]
\[ \text{Ri}_j = \text{Ri}_j \oplus h(\text{ID}_i ||\text{PW}_i ||\sigma_i^o) \]
\[ TC_{U_i} = TC_{U_i} \oplus h(\text{ID}_i ||\sigma_i^o) \]
\[ \text{Ri}_j = \text{Ri}_j \oplus h(\text{ID}_i ||\sigma_i^o) \]
\[ n = B \oplus h(\text{PW}_i ||\text{ID}_i ||\sigma_i^o) \]
\[ \text{RPW}_i = h(\text{PW}_i ||n) \]
\[ A = A \oplus h(\text{ID}_i ||\sigma_i^o ||\text{PW}_i) \]
\[ C = h(A ||\text{Ri}_j ||\text{RPW}_i ||\sigma_i^o) \]

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, and he/she proceeds for password and biometric update procedure. Otherwise, this phase is terminated immediately.

**Step PB2:** \( U_i \) provides new password \( PW_i \) and imprints new biometric \( BIO_i^o \) 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 \( BIO_i^o = BIO_i^o \). Otherwise, \( MD_i \) computes \( \text{Gen}(BIO_i^o) = (\sigma_i^o, \tau_i) \).

**Step PB3:** \( MD_i \) further continues to compute the following:

\[ \text{Ri}_i = \text{Ri}_i \oplus h(\text{PW}_i ||\sigma_i^o) \]
\[ \text{Ri}_i = \text{Ri}_i \oplus h(\text{ID}_i ||\text{PW}_i ||\sigma_i^o) \]
\[ TC_{U_i} = TC_{U_i} \oplus h(\text{ID}_i ||\sigma_i^o) \]
\[ \text{RPW}_i = h(\text{PW}_i ||n) \]
\[ \text{Ri}_i = \text{Ri}_i \oplus h(\text{ID}_i ||\sigma_i^o) \]
\[ A^* = A \oplus h(\text{ID}_i ||\sigma_i^o ||\text{PW}_i) \]
\[ B^* = n \oplus h(\text{PW}_i ||\text{ID}_i ||\sigma_i^o) \]
\[ C^* = h(A ||\text{Ri}_j ||\text{RPW}_i ||\sigma_i^o) \]

Finally, \( MD_i \) stores the information \( \langle \text{Ri}_i^*, \text{Ri}_j^*, \text{Ri}_j^*, TC_{U_i}^*, A^*, B^*, C^*, \tau_i^* \rangle \), \( Gen(.) \), \( \text{Rep}(. \), \( h(. , \tau_i) \) in its memory, while replacing \( \text{Ri}_i^*, \text{Ri}_j^*, \text{Ri}_j^*, TC_{U_i}^*, A^*, B^*, \text{C}^*, \text{and} \ \tau_i^* \) by \( \text{Ri}_i^*, \text{Ri}_j^*, \text{Ri}_j^*, TC_{U_i}^*, A^*, B^*, C^*, \text{and} \ \tau_i^* \), 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 DRnew is needed to deploy in the IoD environment. For this purpose, the server S first generates a unique identity IDnew for DRnew and computes the pseudo identity as \( \text{Ri}_j = h(\text{ID}_j||k) \) using the secret key \( k \) of \( S \). \( S \) further chooses 160-bit master key MKnew corresponding to DRnew and computes the temporal credential for DRnew as

\[ TC_{\text{DRnew}} = h(\text{IDnew} ||\text{MKnew} ||\text{RTS}_{\text{DRnew}}) \]

where RTSnew denotes the.
the registration timestamp of DR\textsubscript{new}. S also generates a unique temporary identity TID\textsubscript{DR\textsubscript{new}} and calculates the polynomial share \( P(TID_{DR\textsubscript{new}}) \).

S finally stores the information \{TID\textsubscript{DR\textsubscript{new}}, RID\textsubscript{DR\textsubscript{new}}, TC\textsubscript{DR\textsubscript{new}}\} in the memory of DR\textsubscript{new} and deploys it in the deployment field, whereas S keeps \{RID\textsubscript{DR\textsubscript{new}}, TC\textsubscript{DR\textsubscript{new}}\} corresponding to DR\textsubscript{new} in its database. The server S also informs all the users in the network about the deployment of DR\textsubscript{new} so that the users can access the information from DR\textsubscript{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 with 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\textsubscript{j} and DR\textsubscript{k}, we can use the existing polynomial-based key distribution scheme proposed by Blundo \textit{et al.} [23]. DR\textsubscript{j} sends its own temporary identity TID\textsubscript{DR\textsubscript{j}} to DR\textsubscript{k}. Similarly, DR\textsubscript{k} also exchanges its own temporary identity TID\textsubscript{DR\textsubscript{k}} to DR\textsubscript{j}. DR\textsubscript{j} then computes the secret key shared with DR\textsubscript{k} using its own polynomial share as

\[ SK_{DR\textsubscript{j}, DR\textsubscript{k}} = P(TID_{DR\textsubscript{j}}, TID_{DR\textsubscript{k}}). \]

In a similar way, DR\textsubscript{k} also computes the same secret key shared with DR\textsubscript{j} using its own polynomial share as

\[ SK_{DR\textsubscript{k}, DR\textsubscript{j}} = P(TID_{DR\textsubscript{k}}, TID_{DR\textsubscript{j}}) = SK_{DR\textsubscript{j}, DR\textsubscript{k}}. \]
since the polynomial $P(x, y)$ is symmetric. Hence, both $DR_j$ and $DR_k$ can communicate securely using the common established shared secret key $SK_{DR_j, DR_k} (= SK_{DR_k, DR_j})$.

Remark 1: In the proposed scheme, a drone can be dynamically added in the network at any time. Assume that an adversary $A$ physically captures a drone $DR_j$ or a drone $DR_k$ can be also stolen by $A$. $A$ can extract all the credentials ($TID_{DR_j}, RID_{DR_j}, TC_{DR_j}, P(TID_{DR_j}, y)$) from its memory using power analysis attacks [19]. It is worth noting that $TID_{DR_j}, RID_{DR_j}, TC_{DR_j},$ and $P(TID_{DR_j}, y)$ are distinct for all drones, and these credentials are generated by the server $S$. Suppose $A$ wishes to utilize the extracted credentials for a newly deployed drone $DR_j^{new}$. For this purpose, $A$ can generate identity $ID_{DR_j}^{new}$, master key $MK_{DR_j}^{new}$ and registration timestamp $RTS_{DR_j}^{new}$ for $DR_j^{new}$, but $h$ cannot compute $RID_{DR_j}^{new} = h(ID_{DR_j}^{new} || k)$ and polynomial share $P(TID_{DR_j}^{new}, y)$ as the secret key $k$ and the original polynomial $P(x, y)$ are unknown to $A$. This will restrict $A$ 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 $DR_j$, $A$ 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 compromised drones can not be compromised by $A$ as these are all distinct. This means that compromise of a drone does not result in compromising secure communications among a user and other non-compromised drones. In this way, the proposed scheme also preserves unconditional security against drone capture.

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].

We have implemented the proposed scheme for the user registration, login, and authentication 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 whether 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 $A$, knows the registration information ($ID_j$) during the user registration phase, which was sent by $U_i$ to $S$. Assume that $A$ has lost/stolen mobile device $MD_i$ of the registered user $U_i$ after the registration process is completed. $A$ can then extract the important information $\{ RID_j, RID_{DR_j}, TID_{U_i}, A', B, C, \tau_j, Gen(\cdot), Rep(\cdot), h(\cdot), t \}$ of
anonymity and untraceability properties. Moreover, these messages do not involve directly identifying information, it is hard for an adversary to derive RID_{A}, RID_{DR}, 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 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 Msg_1 = \langle M'_1, M'_2, M'_3, T'_1 \rangle on behalf of U_i, A can generate the current timestamp T'_1 and a random nonce r'_1. However, without having the secret credentials RID_{A}, RID_{S}, and RID_{DR}, and master secret key MK_{U_i} of U_i, it is a difficult task for A to calculate TC_{U_i}, M'_1, M'_2, M'_3, and M'. Therefore, 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 A generates the current timestamp T'_2, and the random nonces r'_2 and r'_3. A can try to send the message Msg_2 = \langle M'_2, M'_6, M'_7, T'_2 \rangle to DR_j on behalf of S. However, without having the knowledge of RID_{A}, RID_{S}, and RID_{DR_j}, and master secret key MK_{DR_j} of DR_j, it is computationally hard for A to calculate TC_{DR_j}, M'_6, M'_7, and M'. Therefore, 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 A generates the current timestamps T'_2 and T'_3, and random nonces r'_1, r'_2, and r'_3. A then tries to create and send a valid message, say Msg_3 = \langle M'_{10}, M'_{11}, T'_3 \rangle to U_i on behalf of DR_j. However, without the secret credentials RID_{A}, RID_{S}, and RID_{DR_j}, and master secret key MK_{DR_j} of DR_j, it is also difficult task for A to calculate TC_{DR_j}, M'_{10}, M'_{11}, and M'. Therefore, 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 A can not trace the user, server as well as drone. Moreover, these messages do not involve directly identifying information or pseudo-identities RID_{A}, 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 A has lost/stolen mobile device MD_i of a registered user U_i. A can then extract the information \{RID'_i, RID_{DR'_i}, RID_{S}, TC_{U_i}, A', B, C, t_i, \cdot \}. Hence, he/she can not also verify a guessed password with the help of C through the offline password guessing attack. Moreover, without \sigma_i and n, it is hard for A to derive RID_{A}, RID_{DR}, RID_{S}, and 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.

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 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(c) denote the probability that 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(c) = 0, a user authentication scheme is known as unconditionally secure against drone capture attack. By physically capturing a drone DR_j, A can extract the valuable information \{TID_{DR_j}, RID_{DR_j}, TC_{DR_j}, P(TID_{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 P(TID_{DR_j}, y) are different for all drones, and these are generated by S. Therefore, by capturing DR_j, A can only compromise the session key between 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 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 L in Section III-C) or C^* = 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 A. A can then extract all information \{RID'_i, RID_{DR'_i}, RID_{S}, TC_{U_i}, A', B, C, t_i, \cdot \}. Hence, A needs to know both secrets n and \sigma_i. Therefore, it is computationally infeasible for 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

<table>
<thead>
<tr>
<th>Protocol</th>
<th>No. of messages</th>
<th>No. of bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our</td>
<td>3</td>
<td>1696</td>
</tr>
<tr>
<td>Challa et al. [15]</td>
<td>3</td>
<td>2528</td>
</tr>
<tr>
<td>Turkanovic et al. [13]</td>
<td>4</td>
<td>2720</td>
</tr>
</tbody>
</table>

TABLE III
COMPARISON OF STORAGE OVERHEADS

<table>
<thead>
<tr>
<th>Protocol</th>
<th>User side</th>
<th>Server side</th>
<th>Sensing device/Sensor/Drone side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our</td>
<td>1288</td>
<td>320n + (m + 1) log2(p)</td>
<td>480 + (m + 1) log2(p)</td>
</tr>
<tr>
<td>[15]</td>
<td>488 bits</td>
<td>1288 + 640n bits</td>
<td>1600 bits</td>
</tr>
<tr>
<td>[13]</td>
<td>768 bits</td>
<td>320n + 160 bits</td>
<td>320 bits</td>
</tr>
</tbody>
</table>

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 needs to store the credentials \{TID_{DR}, RID_{DR}, TCDR, \mathcal{P}(\text{TID}_{DR}, 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 \{(RID_{DR}, TCDR) | 1 \leq j \leq n, \mathcal{P}(x, y), k, \text{ID}_j, \text{RID}_j, \text{TCD}_j, (\text{RID}_j\text{,} \text{TCD}_j)\} 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 \{RID$_i$, RID$_{DR_i}$, RID$_{C_i}$, TCD$_{U_i}$, A’, B, C, t, 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 $\tau$ 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).

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).
TABLE VI
SIMULATION PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform</td>
<td>Ubuntu 14.04 LTS</td>
</tr>
<tr>
<td>Tool used</td>
<td>NS2 2.35</td>
</tr>
<tr>
<td>Area</td>
<td>400m x 200m x 15m</td>
</tr>
<tr>
<td>Number of gateway nodes (GWN)</td>
<td>1</td>
</tr>
<tr>
<td>Number of users (U)&lt;sub&gt;i&lt;/sub&gt;</td>
<td>3</td>
</tr>
<tr>
<td>Number of DR&lt;sub&gt;i&lt;/sub&gt;/SD&lt;sub&gt;i&lt;/sub&gt;/S&lt;sub&gt;j&lt;/sub&gt;</td>
<td>50</td>
</tr>
<tr>
<td>Mobility of U&lt;sub&gt;i&lt;/sub&gt;</td>
<td>2 mps, 15 mps</td>
</tr>
<tr>
<td>Mobility of DR&lt;sub&gt;i&lt;/sub&gt;</td>
<td>25 mps, 30 mps, 35 mps, 40 mps</td>
</tr>
<tr>
<td>Communication range of DR&lt;sub&gt;i&lt;/sub&gt;</td>
<td>200m</td>
</tr>
<tr>
<td>Simulation time</td>
<td>1800 seconds</td>
</tr>
</tbody>
</table>

EXCHANGED MESSAGES BETWEEN ENTITIES USED IN SIMULATION

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>U&lt;sub&gt;i&lt;/sub&gt; → GWN</td>
<td>992 bits</td>
<td>672 bits</td>
<td>672 bits</td>
</tr>
<tr>
<td>GWN → SD&lt;sub&gt;j&lt;/sub&gt;/S&lt;sub&gt;j&lt;/sub&gt;/DR&lt;sub&gt;j&lt;/sub&gt;</td>
<td>1024 bits</td>
<td>1024 bits</td>
<td>512 bits</td>
</tr>
<tr>
<td>SD&lt;sub&gt;j&lt;/sub&gt;/S&lt;sub&gt;j&lt;/sub&gt;/DR&lt;sub&gt;j&lt;/sub&gt; → GWN</td>
<td>–</td>
<td>576 bits</td>
<td>–</td>
</tr>
<tr>
<td>GWN → U&lt;sub&gt;i&lt;/sub&gt;</td>
<td>–</td>
<td>448 bits</td>
<td>–</td>
</tr>
<tr>
<td>SD&lt;sub&gt;j&lt;/sub&gt;/DR&lt;sub&gt;j&lt;/sub&gt; → U&lt;sub&gt;i&lt;/sub&gt;</td>
<td>512 bits</td>
<td>–</td>
<td>512 bits</td>
</tr>
</tbody>
</table>

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<sub>2</sub>–FSF<sub>4</sub>, FSF<sub>6</sub>, FSF<sub>11</sub>, and FSF<sub>14</sub>–FSF<sub>17</sub>, whereas Challa et al.’s [15] scheme does not support FSF<sub>16</sub>. 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<sub>j</sub>, SD<sub>j</sub>, and S<sub>j</sub> 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<sub>j</sub>/S<sub>j</sub>/DR<sub>j</sub> whenever it is applicable. The users U<sub>i</sub> 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 \( \frac{n_r \times |pkt|}{T_d} \), 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(a), 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 \( \sum_{i=1}^{n_p} \frac{T_{rec_i} - T_{send_i}}{n_p} \), 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(b), 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 \( \frac{n_{lp}}{T_d} \), where \( T_d \) is the total time (in seconds) and \( n_{lp} \) the total number of packets lost.
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


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 157 papers in international journals and conferences, including over 155 reputed journal papers. His current research interests include cryptography, wireless sensor networks, secure 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 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. Mr. 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, the 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 Agregado 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 ParkUrbs 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.