e-Prime - Advances in Electrical Engineering, Electronics and Energy 7 (2024) 100471

Contents lists available at ScienceDirect

e-Prime - Advances in Electrical Engineering, Electronics and Energy

journal homepage: www.elsevier.com/locate/prime

A biometric and physically unclonable function–Based authentication protocol for payload exchanges in internet of drones

Vincent Omollo Nyangaresi [a,b,*], Istabraq M. Al-Joboury [c], Kareem Ali Al-sharhanee [d], Ali Hamzah Najim [e], Ali Hashim Abbas [f], Hussein Muhi Hariz [g]

[a] Department of Computer Science and Software Engineering, Jaramogi Oginga Odinga University of Science and Technology, Bondo, 40601, Kenya
[b] Department of Applied Electronics, Saveetha School of Engineering, SIMATS, Chennai, Tamilnadu, India
[c] Department of Medical Instruments Technical Engineering, Al-Bayan University Technical College of Engineering, Iraq
[d] Department of Communication Technical Engineering, College of Engineering Techniques, Al-Farahidi University, Iraq
[e] Department of Computer Technical Engineering, Imam Al-Kadhum College (IKC), Iraq
[f] College of Information Technology, Imam Ja'afar Al-Sadiq University, Al-Muthanna 66002, Iraq
[g] Department of Computer Techniques Engineering, Mazaya University College, Dhi-Qar, Annasiriyah 64001, Iraq

ARTICLE INFO

Keywords:
Attacks
Authentication
Drones
UAVs
Security
Privacy

ABSTRACT

Internet of Drones (IoD) has been deployed in numerous military and civilian domains to offer services such as target surveillance, traffic monitoring, disaster handling and environmental monitoring. However, message exchanges among the drones and ground station servers are via insecure wireless channels. In addition, drones may be deployed in hostile and unattended locations. This renders the IoD susceptible to numerous privacy and security threats such as drone capture and cloning. Therefore, many security solutions have been developed based on techniques such as blockchain and public key infrastructure. However, majority of these schemes are still susceptible to many attacks while some of them are inefficient for the resource-limited IoD devices. In this paper, a Physically Unclonable Function (PUF) challenge-response and biometric based robust authentication protocol is presented. Its formal security is carried out using the Real or Random (RoR) model, which demonstrates the robustness of the negotiated session key. In addition, its semantic analysis shows that it can withstand typical IoD attacks such as impersonation, replay, de-synchronization and spoofing. In terms of performance, it is shown to incur lower computation, energy and communication costs.

Introduction

Unmanned Aerial Vehicles (UAVs) or drones are aircrafts that are equipped with communication modules, recorders, batteries, actuators, computation modules and various embedded sensors. The UAVs capture data from their application domains and send it to the Ground Station Server (GSS) over wireless communication channels [1]. As explained in [2], the drones contain integrated Internet of Things (IoT) devices such as smart cameras and Global Positioning System (GPS) based sensors. In the Internet of Drones (IoD) environment, the UAVs collect a range of information such as live videos and still images of their targets in real-time and transmit it to the remote users. A typical IoD architecture consists of drones, remote users and control servers. To receive various services remotely, the users in IoD have to query the drones for information. On the other hand, the control servers are normally centrally placed in the wireless communication flow to mediate the message exchange process between the UAVs and the users. Here, each drone has its own flying zone from which its collects information and forwards it to users via the control centers. As such, IoD has found applications in domains such as data gathering, security surveillance, military and traffic monitoring, disaster handling, target tracking, delivery of goods, healthcare systems, environmental monitoring, agricultural plant protection, detection and rescue systems [3,4]. In addition, these drones can act as aerial base stations to offer network connectivity.

Although IoD provides several services such as facilitating ground communications in the face of limited connectivity occasioned by physical obstacles, it has numerous security vulnerabilities and faces a myriad of threats. For instance, many drones lack inbuilt security or authentication mechanisms [5]. In addition, the entities in IoD communicate over wireless channels [6,7]. As such, adversaries can

* Corresponding author.
E-mail address: vnyangaresi@jooust.ac.ke (V.O. Nyangaresi).

https://doi.org/10.1016/j.prime.2024.100471
Received 16 December 2023; Received in revised form 11 February 2024; Accepted 16 February 2024
Available online 23 February 2024
2772-6711/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

intercept and eavesdrop the exchanged messages. Moreover, adversaries can inject bogus data, delete or modify the information as it is being transmitted among the drones and the GSS. This can facilitate further attacks such as impersonation, GSS bypassing, replay, Man-in-Middle (MitM), hijacking, drone capture as well as privileged insider [8]. All these threats can interfere with the proper functioning of the IoD. It is also possible to compromise data authenticity, confidentiality, integrity and availability of the IoD services [9]. Additionally, cyber criminals can hijack the drones and other network entities, resulting in payload theft and data breaches [10]. Since some drones collect and transmit sensitive data, privacy leaks are serious challenges that need to be addressed in IoD. For instance, the drones or remote user mobile devices used to interact with the IoD can be stolen or lost. Consequently, attackers can extract sensitive data stored in these devices and hence threaten the entire network. Therefore, numerous security schemes have been developed to curb these issues. However, attaining perfect privacy and security at low computation, communication and energy costs is still an open challenge.

Motivation

The open nature of the communication channel used for message exchange among drones and the ground stations exposes the transmitted information to numerous threats. For instance, adversaries can launch replay, impersonation, MitM, drone hijacking and privileged insider attacks. Therefore, many schemes have been developed over the recent past to address these challenges. Unfortunately, majority of these security solutions have numerous security vulnerabilities and are based on computationally extensive cryptographic techniques such as Public Key Infrastructure (PKI), elliptic curve point multiplications and blockchain. This renders these protocols inefficient for resource-limited IoD devices. There is therefore need for a robust and efficient authentication scheme that will ensure that only authorized IoD entities can transmit data to and from authentic users.

Adversarial model

In the proposed protocol, an adversary is assumed to have all the capabilities in the widely accepted Dolev–Yao (DY) model. Here, the IoD entities are thought to be untrustworthy and the communication process is executed over insecure channels. As such, the exchanged information can be intercepted, eavesdropped, modified and deleted. The drones are also assumed to be flying over hostile and unattended zones. As such, they are susceptible to physical capture attacks and their memory-resident sensitive data can be retrieved via power analysis.

Security and privacy requirements

Drones play critical roles in collecting and disseminating real-time information to their users. In some of the drone application domains such as in the military, the collected data is highly sensitive and confidential. As such, there must be robust access control protocols to prevent unauthorized access to this data. Since the drones are limited in term of computation, communication, energy and storage, the deployed security protocols should be lightweight to boost efficiency. Due to the deployment of some drones in un-monitored locations, they may be physically captured by adversaries and their stored secret security tokens extracted through power analysis. Thereafter, it may be possible for the attackers to impersonate the network entities using the stolen secrets. In military drones, anonymity and untraceability must be preserved at all times. In addition, the exchanged data should be enciphered before being exchanged over the insecure wireless channels. This will prevent eavesdropping and enhance the confidentiality of the transmitted information [11]. Therefore, the proposed protocol must satisfy the following security, privacy and performance requirements:

Anonymity: the real identities of the IoD entities should remain unknown to the adversaries even after eavesdropping all the messages exchanged over the open wireless communication channels.
Perfect backward and forward secrecy: The keys derived in a given session should be disparate from the keys derived in the past as well as future sessions.
Untraceability: Any successful adversarial interception and capture of the transmitted messages should not facilitate the tracking of IoD entities.
Mutual authentication: All the IoD communicating parties should verify each other's identity before they can initiate any payload exchanges amongst themselves.
Session key establishment: Upon successful validation of each other, the IoD entities should setup session keys to encipher all the exchanged messages.
Session key security: The derivation of the session keys should incorporate parameters that render them one-time such that adversaries are unable to use particular session keys to derive valid keys deployable in other sessions.
Attack resilience: To uphold confidentiality, integrity and availability, the proposed protocol should withstand typical IoD attacks such as side-channeling, de-synchronization, DoS, MitM, physical capture, impersonation, eavesdropping, ephemeral secret leakage, spoofing, replay, cloning, and challenge-response tracking.
Efficiency: Since the drones are resource-constrained, the proposed protocol should incur low computation, energy and communication costs.

Mathematical preliminaries

In this section, the mathematical formulations of the key technologies deployed in this paper are provided. This includes the mathematical basis for *PUF*, one-way hashing and Fuzzy Extractor (*FE*), as described in the sub-sections below.

Fuzzy extractor

The fuzzy extractor has two functions, which include key generation function *FE.Gen* () and the key reconstruction function *FE.Rec* (). Taking $K_{\beta i}$, β_i and HD_{Ri} as the biometric key, user biometric data and drone helper data respectively, then

$$\left(K_{\beta i}, HD_{Ri}\right) = FE.Gen(\beta_i) \tag{1}$$

To reconstruct the key $K_{\beta i}$ from helper data HD_{Ri} and noisy biometrics β_i*, the following equation is used:

$$K_{\beta i} = FE.Rec(HD_{Ri}, \beta_i^*) \tag{2}$$

For accurate key recovery, the Hamming distance H_D between the input noisy biometric data β_i* and original user biometric data β_i must be less than some set threshold β_T. That is,

$$H_D(\beta_i, \beta_i^*) \le \beta_T \tag{3}$$

As such, the following equation holds:

$$\left(K_{\beta i}, HD_{Ri}\right) = FE.Gen(\beta_i) \rightarrow K_{\beta i} = FE.Rec(HD_{Ri}, \beta_i^*) \tag{4}$$

In an ideal scenario, the noisy biometric input β_i*must be similar to the original biometric data β_i, and this forms the basis for any successful fuzzy extraction. In this paper, we deploy fuzzy extractor to compute biometric key $K_{\beta i}$ from user biometric data β_i in accordance with Eq. (1). In addition, we utilize the *FE* to create non-stable PUF that is deployed to get stable output. Here, the fuzzy extractor is deployed to obtain a key from the response input. Taking GSS's secret key and helper data as K_{GSS} and H_{GSS} respectively,

$$(K_{GSS}, H_{GSS}) = FE.Gen\left(R_{GS}{}^i\right) \tag{5}$$

In our scheme, the keys are reconstructed from the helper data and

response using the *FE.Rec* () as follows:

$$K_{GSS} = FE.Rec\left(R_{GS}{}^{i}, H_{GSS}\right) \tag{6}$$

Hash function

A hash function takes an input of any length and produces an output of fixed bit length *s*. Mathematically, this is denoted as follows:

$$h : \{0,1\}^{*} \rightarrow \{0,1\}^{s} \tag{7}$$

Taking $\ddot{A}$ as a polynomial time adversary, α_1 and α_2 as strings randomly picked by $\ddot{A}$, then the advantage that $\ddot{A}$ has in finding a collision of h (.) in polynomial time t is given as follows:

$$Adv_{\ddot{A}}^{h()}(t) = Pr\left[\alpha_1, \alpha_2 \in_R \ddot{A} : \ \alpha_1 \neq \alpha_2, \ h(\alpha_1) = h(\alpha_2)\right] \tag{8}$$

where *Pr* (α) denotes the probability of α being randomly selected. Considering an (λ, t) –adversary attempting to compromise h (.)'s collision resistance, then:

$$Adv_{\ddot{A}}^{h()}(t) \leq \lambda \tag{9}$$

Physically unclonable function

This function is a unique feature in the manufacturing process during the fabrication of the chip. Basically, challenge C has to be mapped to some response R such that:

$$R = PUF(C) \tag{10}$$

The *PUF* was deployed in this paper owing to difficulties of its cloning. *PUF* exists in two variants, which include Ideal *PUF* (*I-PUF*) and Non-Ideal *PUF* (*NI-PUF*). In the former, same response is produced for the same set of input challenges. In the latter, temperature can cause the *PUF* to yield varied responses even when supplied with the same input challenge. Therefore, a fuzzy extractor is needed to stabilize the *NI-PUF*. To obtain a secure *PUF*, it should be possible for PUF_1 to yield responses R_1 and $R_2 \in \{0,1\}^s$ with the variance of the two input challenges being denoted by φ. Suppose that input challenge C_1 yields diverse output responses R_1 and $R_2 \in \{0,1\}^s$ for two dissimilar *PUF*s represented by PUF_1 and PUF_2. In this case, C_1 and $C_2 \in \{0,1\}^s$, where C_1 and C_2 are the randomly selected challenges. Taking φ_1 and φ_2 as variances in input challenges and μ as *PUF*'s error tolerance limits,

$$\begin{aligned} &\Pr[H_D(PUF_1(C_1), PUF_1(C_2))\rangle\varphi_1] = 1-\mu \\ &\Pr[H_D(PUF_1(C_1), PUF_2(C_1))\rangle\varphi_2] = 1-\mu \end{aligned} \tag{11}$$

Research contributions

It has been shown that IoD networks are susceptible to a myriad of security and privacy threats. In addition, drones have been noted to be limited in terms of communication, computation and energy. As such, an ideal authentication protocol should not only be robust but also extremely efficient for the drones. In this regard, this paper makes the following contributions:

- We utilize the fuzzy extractor to create non-stable physically unclonable function that is deployed to get stable output. The incorporation of physically unclonable function and biometrics into the authentication procedures is demonstrated to protect against typical IoD attacks such as cloning and impersonation.
- During the authentication process, we mainly execute lightweight hash functions and exclusive-OR cryptographic operations. This renders our scheme highly efficient, and hence suitable for resource-limited drones.
- Elaborate formal security analysis is carried out using RoR to show that the authentication procedures and the negotiated session keys are provably secure.
- Extensive semantic security analysis is performed to demonstrate that our protocol supports mutual authentication, session key agreement, key security, untraceability, anonymity as well as forward and backward secrecy. In addition, its robustness against numerous IoD attacks is demonstrated.
- Comparative performance evaluation is executed to show that our scheme incurs lower computation, energy overheads and communication costs.

The remainder of this paper is structured as follows: Section 2 describes the related work while Section 3 presents the proposed protocol. On the other hand, Section 4 details the security analyses of our protocol while Section 5 presents its comparative performance evaluation. Towards the end of this paper, Section 6 gives the conclusion and future research directions.

Related works

IoD security and privacy setbacks have attracted huge attention from academia and the industry, resulting in the development of numerous security solutions. For example, an Elliptic Curve Digital Signature Algorithm (ECDSA)-based technique is presented in [12]. However, the Elliptic Curve Cryptography (ECC)-based two-way authentication among drones results in high computation costs. As such, lightweight schemes have been developed in [13] and [14] to address the inefficiency challenges in [12]. Unfortunately, the protocol in [13] is not scalable since it supports only one mobile user [15] while the scheme in [14] fails to offer drone anonymity [15,16]. Similarly, the protocol in [17] fails to address privacy leakage issues [18]. The certificate-based signature scheme in [19] is privacy-preserving and hence can curb the challenges in [13] and [17]. Similarly, the biometric authentication protocol in [20] provides anonymity and untraceability and hence can address privacy leakage issues. Unfortunately, the technique in [19] does not offer mutual authentication, and hence cannot uphold communication integrity [21,22]. On its part, the approach in [20] is susceptible to smart device theft, traceability and stolen verifiers attacks [23].

The authors in [24] have introduced two schemes, one for authentication and the other one for key agreement. However, these two protocols are susceptible to Ephemeral Secret Leakage (ESL) attacks [25]. Similarly, the security framework in [26] has some design flaw [8] that renders it unsuitable for IoD. To offer robust security, a blockchain based authentication technique is presented in [27]. Although this scheme is provably secure, the deployed blockchain leads to high storage and computation overheads [28]. Similarly, the protocol in [11] incurs heavy computation and communication overheads [18]. To address this problem, an efficient security technique based on hash functions and exclusive-OR operators is presented in [29]. However, this scheme is vulnerable to physical attacks since the parameters in memory are not protected from power analysis [30]. To control user access to the drones, a certificate-based protocol is introduced in [31]. However, the certificates in this scheme are insecure and hence this approach is impractical [32]. Similarly, the ECC-based scheme in [33] is impractical due to its high ECC computation overheads. In addition, it is vulnerable to impersonation and replay attacks [13,34]. Moreover, it fails to uphold reliability and anonymity [18]. To solve this issue, an Advanced Encryption Standard (AES)-based technique is developed in [35]. However, the storage of the encryption and decryption secret key in plaintext at the gateway exposes this scheme to privileged insider attacks.

To withstand numerous security threats occasioned by unauthorized drones, authentication schemes are introduced in [36] and [37]. However, re-authentication is not supported in [37], the scheme incurs heavy computation costs and fails to consider tampering as well as physical capture attacks [18]. On its part, the protocol in [36] is efficient and hence solves inefficiencies inherent in [37]. Unfortunately, it cannot

withstand ESL attacks and fails to support anonymity and untraceability [25]. Similarly, the protocol in [38] is susceptible to ESL attack and cannot offer both untraceability and anonymity. In addition, it incurs extensive computation and communication overheads [25]. The schemes in [39] and [40] are lightweight and hence can address the inefficiencies in [38]. However, the scheme in [39] cannot withstand drone capture attacks [18] while the protocol in [40] fails to protect against impersonation, replay, privileged insider and secret parameters leakages [15]. In addition, it fails to achieve mutual authentication. Similarly, the approach in [41] cannot support mutual authentication and user traceability [42,43]. Therefore, an improved scheme is presented in [42], which is shown to deploy only lightweight symmetric hash functions. However, it is not scalable and it can only be utilized within one flying zone. In addition, it lacks untraceability and is not resilient against stolen verifier attack [18]. Similarly, the session key in [44] can be exposed to an attacker [45]. As such, access control schemes developed in [25] and [46] can solve this problem. However, the protocol in [25] incorporates blockchain and ECC technologies which incur heavy computational and communication overheads which are unsuitable for drones. On its part, the technique in [46] cannot provide anonymity. In addition, it is susceptible to collusion attacks [47]. Therefore, an improved scheme is developed in [47]. Unfortunately, this technique cannot withstand ESL attacks [25]. The scheme in [48] can provide anonymity, untraceability and resilience against impersonation attacks [49]. As such, it can address anonymity challenges in [46]. However, this protocol is vulnerable to Denial of Service (DoS) and verification table leakage attacks [30]. In addition, it fails to preserve perfect forward secrecy. Similarly, the scheme in [50] is susceptible to DoS, replay and forgery attacks [51]. In addition, it cannot support mutual authentication. There is therefore a need for a security protocol that is not only robust but also efficient.

The proposed protocol

The main IoD entities in the proposed protocol include the drones, gateway node (GW_j), GSS and TA as shown in Fig. 1. The remote users located at the ground station server access the information through the gateways, which serve a number of server stations. The trusted authority in this network model registers all the drones, gateway and ground station servers prior to the actual information exchange. Apart from the registration procedures, all these entities communicate via the insecure communication channels. Table 1 presents the notations used throughout this paper. Our scheme is executed through four phases, which include registration, authentication, key negotiation, and challenge-response update. The sub-sections below describe these phases in detail.

Table 1
Deployed notations.

Symbol	Description
D_{Ri}	i^{th} drone
GSS	Ground station server
GW_j	j^{th} GSS gateway
ID_{GW}	Unique identity of the GSS gateway
TA	Trusted authority
ID_{DR}	Unique identity of the D_{Ri}
ID_{GS}	Unique identity of the GSS
ID_{GW}	Unique identity of the GW_j
ID_{TA}	Unique identity of the TA
PID_{DR}	Drone pseudo-identity
PID_{GS}	GSS pseudo-identity
K_{DT}	Shared key between D_{Ri} and TA
K_{GT}	Shared key between GW_j and TA
K_G	Shared key between GW_j and GSS
K_{GSS}	GSS's secret key
SK	Session key
(C_D^i, R_D^i)	Drone challenge response pair
(C_{GS}^i, R_{GS}^i)	GSS challenge response pair
KD_{Syn}	Synchronization shared key between D_{Ri} and TA
PID_{TSyn}	Synchronization pseudo-identity between D_{Ri} and TA
PID_{GSyn}	Synchronization pseudo-identity between GW_j and GSS
KG_{Syn}	Synchronization shared key between GW_j and TA
KGW_{Syn}	Synchronization shared key between GW_j and GSS
(CD_{Syn}, RD_{Syn})	Synchronization challenge-response pairs between D_{Ri} and TA
(C_{GSyn}, R_{GSyn})	Synchronization challenge-response pairs between GW_j and GSS
β_i	Drone operator biometric data
h (.)	Collision-resistant one-way hashing function
$K_{\beta i}$	Biometric key
FE	Fuzzy extractor
HD_{Ri}	Drone helper data
H_{GSS}	GSS helper data
R_i	Random nonce i
T_{TA}	TA token
SK_{TA}	TA's secret key
l_T	The length of T_{TA}
T_{max}	Expiration of T_{TA}
$\|\|$	Concatenation operation
$\oplus$	XOR operation

Registration phase

Prior to the actual deployments, all drones, ground station servers and gateways must be registered at the trusted authority. To offer protection against de-synchronization and denial of service attacks, challenge-response, pseudo-identity and shared keys are synchronized in our scheme. The following sub-sections details the procedures executed during the registration process.

Drone →Trusted authority registration

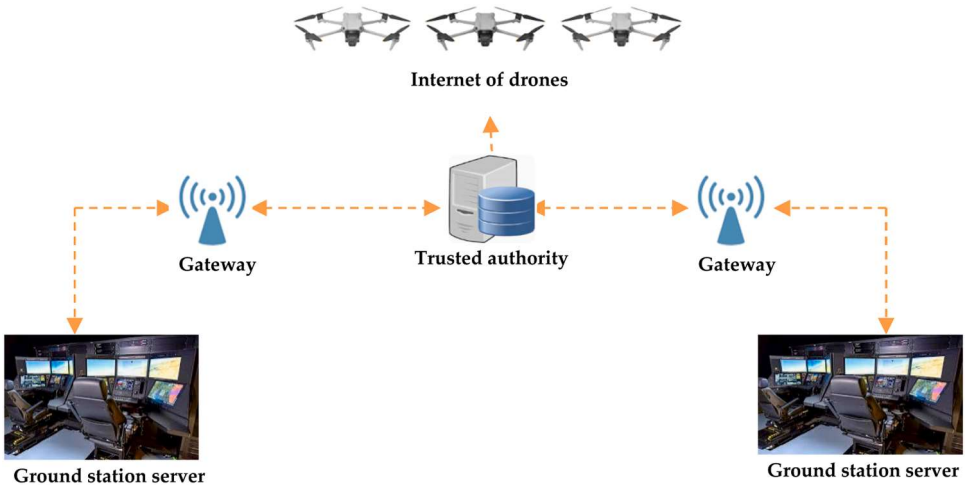

Fig. 1. Network model.

Prior to their actual deployment, all drones must be registered at the TA and be assigned some secret values for subsequent authentication and key negotiation phases. The following 7 steps are executed to register a particular drone D_{Ri} to the TA.

Step 1: Drone D_{Ri} generates its identity ID_{DR}. Next, it composes registration request $Req_1 = \{ID_{DR}\}$ that is forwarded to the TA over secure channels as shown in Fig. 2.

Step 2: Upon receiving Req_1, the TA generates pseudo-identity PID_{DR} for this particular drone. In addition, it generates K_{DT} as the shared key between itself and D_{Ri}.
Step 3: The TA generates challenge C_D^i as well as the synchronization shared key, pseudo-identity and challenge response pairs $KD_{Syn} = \{KD_{S1}, KD_{S2},\ldots, KD_{Sn}\}$, $PID_{TSyn} = \{PID_{S1}, PID_{S2},\ldots, PID_{Sn}\}$ and $CD_{Syn}=\{CD_{S1}, CD_{S2},\ldots, CD_{Sn}\}$ respectively. These synchornization parameters are deployed between D_{Ri} and TA when de-synchronization occurs.
Step 4: The TA stores parameter set $\{PID_{DR}, K_{DT}, KD_{Syn}, PID_{TSyn}\}$ in the smart card SC. Next, it constructs registration response $Res_1 = \{SC, C_D^i, CD_{Syn}\}$ which is then sent to D_{Ri} over secure channels.
Step 5: The smart card is inserted into its reader upon which D_{Ri} derives PUF responses $R_D^i = PUF(C_D^i)$ and $RD_{Syn} = PUF\ (CD_{Syn})$. Next, the operator imprints his/her biometrics β_i after which value $(K_{\beta i}, HD_{Ri}) = FE.Gen\ (\beta_i)$ is computed.
Step 6: The drone derives user authentication verification code $V_C = h\ (HD_{Ri}||K_{\beta i}||ID_{DR})$ as well as parametes $PID_{DR}{}^* = PID_{DR} \oplus h\ (K_{\beta i})$, $PID_{TSyn}{}^* = PID_{TSyn} \oplus h\ (K_{\beta i})$, $KD_{Syn}{}^* = KD_{Syn} \oplus h\ (K_{\beta i})$, $ID_{DR}{}^* = ID_{DR} \oplus h\ (K_{\beta i})$ and $K_{DT}{}^* = K_{DT} \oplus h\ (K_{\beta i})$. Next, value set $\{HD_{Ri}, ID_{DR}{}^*, PID_{DR}{}^*, K_{DT}{}^*, KD_{Syn}{}^*, PID_{TSyn}{}^*\}$ is stored in the SC. Lastly, D_{Ri} composes registration message $Req_2 = \{R_D^i, RD_{Syn}\}$ which is sent to the TA over secured channels.
Step 7: After getting message Req_2, the TA stores value set $\{PID_{DR}, K_{DT}, C_D^i, R_D^i, KD_{Syn}, PID_{TSyn}, CD_{Syn}, RD_{Syn}\}$ in its database.

Gateway →Trusted authority registration

Before acting as intermediaries between the drones and their operators, all gateway nodes must be registered at the TA and be assigned tokens deployable for later authentication and key negotiation. The following 3 steps are carried out to register the GSS gateway GW_j to the TA.

Step 1: The gateway GW_j generates its identity ID_{GW}. This is followed by the construction of registration message $Req_3 = \{ID_{GW}\}$, which is forwarded to the TA over secure channels as shown in Fig.2.
Step 2: On receiving message Req_3, the TA generates key K_{GT} which it shares with GW_j. Next, it derives $KG_{Syn} = \{KG_{S1}, KG_{S2},\ldots, KG_{Sn}\}$ before storing value set $\{ID_{GW}, K_{GT}, KG_{Syn}\}$ in its database. Finally, it composes registration response message $Res_2 = \{K_{GT}, KG_{Syn}\}$, which is sent to the GW_j over protected channels.
Step 3: After getting message Res_2 from the TA, GW_j stores parameter set $\{K_{GT}, KG_{Syn}\}$ in its repository.

GSS → Gateway registration

Prior to facilitating operator access to the drone data, the GSS must register at the TA and be issued with security parameters that will enable it authenticate and set up session keys. To register the GSS to the gateway, the following 5 steps are carried out.

Step 1: The GSS generates its unique identity ID_{GS}. Next, it constructs registration request message $Req_4 = \{ID_{GS}\}$, which is transmitted to the GW_j over secure communication channels as shown in Fig. 3.
Step 2: On receiving message Req_4, the GW_j generates K_G, PID_{GS}, C_{GS}^i, $KGW_{Syn} = \{KGW_{S1}, KGW_{S2},\ldots, KGW_{Sn}\}$, $PID_{GSyn} = \{PID_{G1}, PID_{G2},\ldots, PID_{Gn}\}$ and $C_{GSyn} = \{C_{GS1}, C_{GS2},\ldots, C_{GSn}\}$. Next, it stores value set $\{ID_{GS}, K_G, KGW_{Syn}, PID_{GSyn}, C_{GS}^i, C_{GSyn}\}$ in its repository. Finally, it composes registration response message $Res_3 = \{K_G, KGW_{Syn}, PID_{GSyn}, C_{GS}^i, C_{GSyn}\}$ that is forwarded to the GSS through secured channels.
Step 3: After receiving message Res_3, the GSS generates response $R_{GS}^i = PUF_{GSS}(C_{GS}^i)$ and $R_{GSyn} = PUF_{GSS}(C_{GSyn})$. Next, it derives parameters $C_{GS}^{i*} = C_{GS}^i \oplus h\ (K_G)$ and $C_{GSyn}{}^* = C_{GSyn} \oplus h\ (KGW_{Syn})$.
Step 4: The GSS stores parameter set $\{K_G, KGW_{Syn}, PID_{GSyn}, C_{GS}^{i*}, C_{GSyn}{}^*\}$ in its database. Afterwards, it constructs registration response message $Res_4 = \{R_{GS}^i, R_{GSyn}\}$ that is transmitted to the GW_j over secured communication channels.

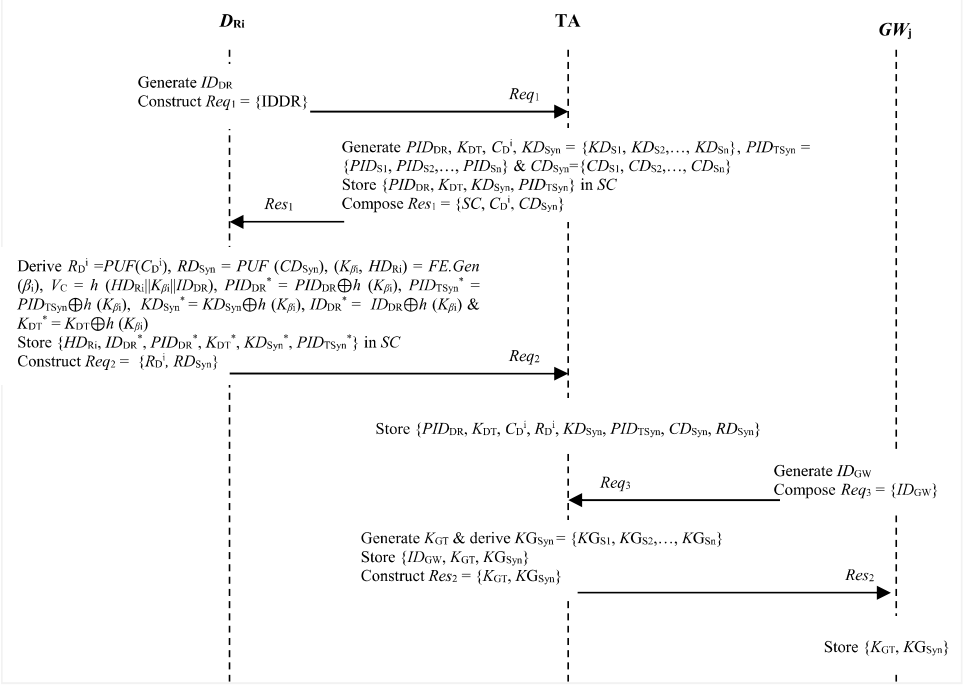

Fig. 2. Drone and gateway registration to TA.

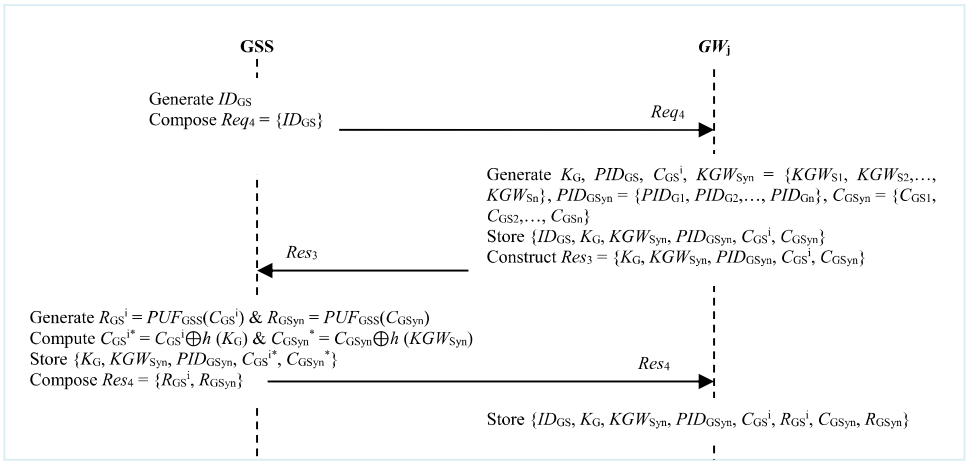

Fig. 3. GSS registration.

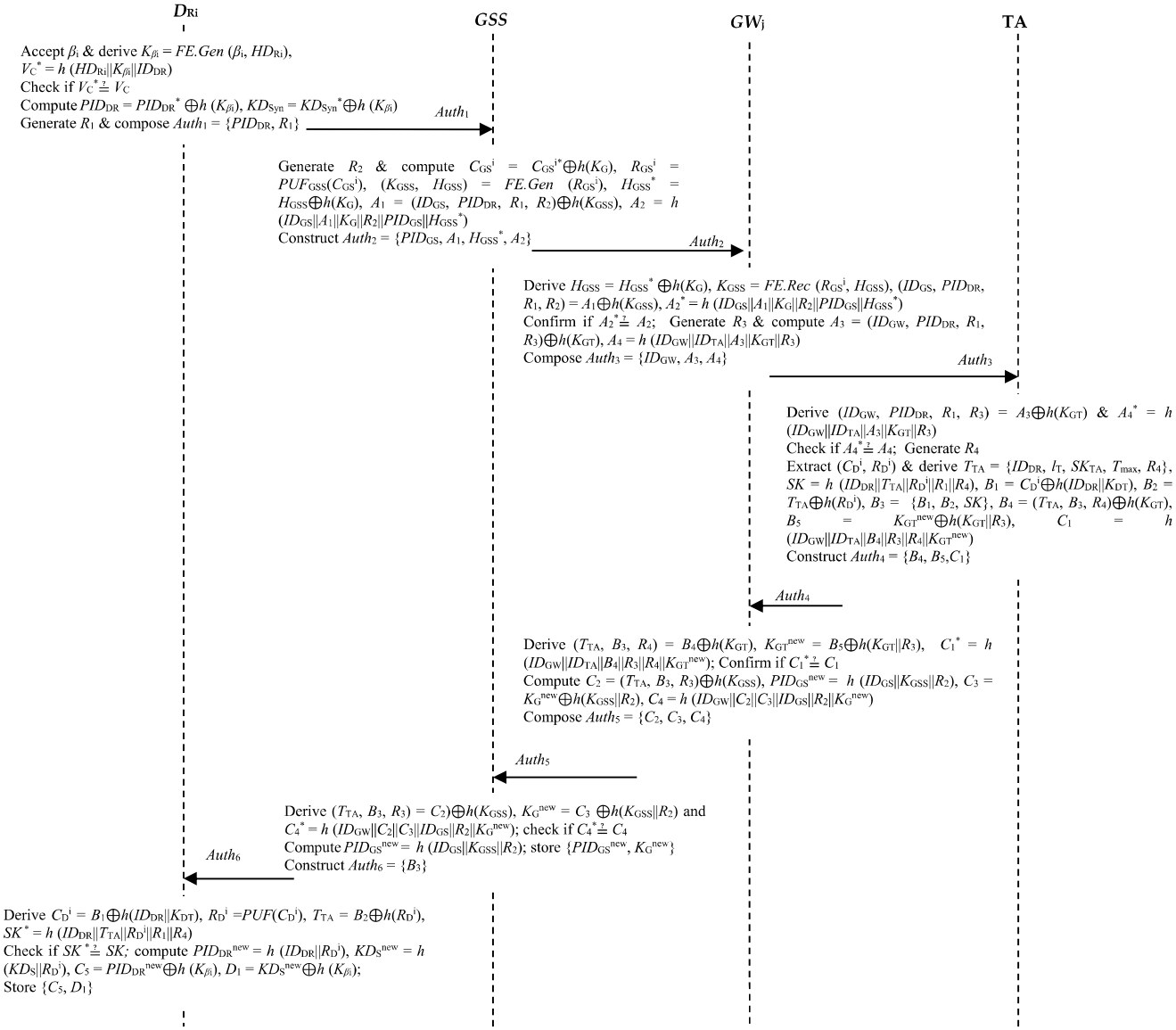

Fig. 4. Authentication and key negotiation.

Step 5: On obtaining message Res_4, the GW_j stores parameter set $\{ID_{GS}, K_G, KGW_{Syn}, PID_{GSyn}, C^i_{GS}, R^i_{GS}, C_{GSyn}, R_{GSyn}\}$ in its repository.

Authentication and key negotiation phase

To preserve channel security and protect against attacks, all the IoD entities must authenticate each other and setup a session key for payload encryption. To achieve this, the following 12 steps are executed.

Step 1: The operator inserts the smart card SC into its reader after which biometric data β_i is imprinted. Next, values $K_{\beta i} = FE.Gen\ (\beta_i, HD_{Ri})$ and $V_C = h\ (HD_{Ri}||K_{\beta i}||ID_{DR})$ are derived. Thereafter, D_{Ri} confirms whether $V_C{}^* \stackrel{?}{=} V_C$ such that the authentication process is terminated whenever these values are dissimilar. Otherwise, it computes $PID_{DR} = PID_{DR}{}^* \oplus h\ (K_{\beta i})$ and $KD_{Syn} = KD_{Syn}{}^* \oplus h\ (K_{\beta i})$. Finally, it generates random number R_1 before constructing authenticatiom message $Auth_1 = \{PID_{DR}, R_1\}$, which is forwarded to the GSS across public channels as shown in Fig. 4.

Step 2: Upon receiving message $Auth_1$, the GSS generates random number R_2. Next, it derives challenge $C^i_{GS} = C^{i*}_{GS} \oplus h(K_G)$ and response $R^i_{GS} = PUF_{GSS}(C^i_{GS})$ based on the derived challenge.

Step 3: The GSS derives fixed-key $(K_{GSS}, H_{GSS}) = FE.Gen\ (R^i_{GS})$. This is followed by the computation of $H_{GSS}{}^* = H_{GSS} \oplus h(K_G)$, $A_1 = (ID_{GS}, PID_{DR}, R_1, R_2) \oplus h(K_{GSS})$ and $A_2 = h\ (ID_{GS}||A_1||K_G||R_2||PID_{GS}||H_{GSS}{}^*)$. Finally, it composes authentication message $Auth_2 = \{PID_{GS}, A_1, H_{GSS}{}^*, A_2\}$, which is sent over to the GW_j as shown in Fig. 4.

Step 4: After receiving message $Auth_2$, the GW_j obtains GSS helper data as $H_{GSS} = H_{GSS}{}^* \oplus h(K_G)$. This helper data is then used to reconstruct key K_{GSS} as $K_{GSS} = FE.Rec\ (R^i_{GS}, H_{GSS})$. Next, it extracts parameters ID_{GS}, PID_{DR}, R_1 and R_2 as $(ID_{GS}, PID_{DR}, R_1, R_2) = A_1 \oplus h\ (K_{GSS})$.

Step 5: The GW_j computes $A_2{}^* = h\ (ID_{GS}||A_1||K_G||R_2||PID_{GS}||H_{GSS}{}^*)$ and checks if $A_2{}^* \stackrel{?}{=} A_2$. Here, the session is aborted upon verification failure. Otherwise, GW_j generates random number R_3 and derives values $A_3 = (ID_{GW}, PID_{DR}, R_1, R_3) \oplus h(K_{GT})$ and $A_4 = h\ (ID_{GW}||ID_{TA}||A_3||K_{GT}||R_3)$. At the end, it constructs authentication message $Auth_3 = \{ID_{GW}, A_3, A_4\}$ that is sent to the TA over public channels.

Step 6: On getting message $Auth_3$, the TA extracts parameters ID_{GW}, PID_{DR}, R_1 and R_3 as $(ID_{GW}, PID_{DR}, R_1, R_3) = A_3 \oplus h(K_{GT})$. Next, it computes $A_4{}^* = h\ (ID_{GW}||ID_{TA}||A_3||K_{GT}||R_3)$ and checks if $A_4{}^* \stackrel{?}{=} A_4$. Basically, the authentication session is terminated when this validation flops. Otherwise, the TA generates random number R_4 and extracts the challenge-response pair (C^i_D, R^i_D) for this particular PID_{DR}.

Step 7: The TA generates token $T_{TA} = \{ID_{DR}, t_T, SK_{TA}, T_{max}, R_4\}$, session key $SK = h\ (ID_{DR}||T_{TA}||R^i_D||R_1||R_4)$ as well as parameters $B_1 = C^i_D \oplus h(ID_{DR}||K_{DT})$, $B_2 = T_{TA} \oplus h(R^i_D)$, $B_3 = \{B_1, B_2, SK\}$ and $B_4 = (T_{TA}, B_3, R_4) \oplus h(K_{GT})$. Next, it derives new shared key as K^{new}_{GT} and derives value $B_5 = K^{new}_{GT} \oplus h(K_{GT}||R_3)$ as well as $C_1 = h\ (ID_{GW}||ID_{TA}||B_4||R_3||R_4||K^{new}_{GT})$. Lastly, it composes authentication message $Auth_4 = \{B_4, B_5, C_1\}$ that is transmitted over to the GW_j.

Step 8: Upon receiving message $Auth_4$, the GW_j extracts values T_{TA}, B_3 and R_4 as $(T_{TA}, B_3, R_4) = B_4 \oplus h(K_{GT})$. Next, it derives values $K^{new}_{GT} = B_5 \oplus h(K_{GT}||R_3)$ and $C_1{}^* = h\ (ID_{GW}||ID_{TA}||B_4||R_3||R_4||K^{new}_{GT})$. Thereafter, it confirms whether $C_1{}^* \stackrel{?}{=} C_1$ such that the authentication session is aborted when this validation flops. Otherwise, the GW_j computes parameters $C_2 = (T_{TA}, B_3, R_3) \oplus h(K_{GSS})$.

Step 9: The GW_j updates shared key K_G as K^{new}_G and pseudo-identity PID_{GS} as $PID^{new}_{GS} = h\ (ID_{GS}||K_{GSS}||R_2)$. This is followed by the derivation of values $C_3 = K^{new}_G \oplus h(K_{GSS}||R_2)$ and $C_4 = h\ (ID_{GW}||C_2||C_3||ID_{GS}||R_2||K^{new}_G)$. Finally, it constructs authentication message $Auth_5 = \{C_2, C_3, C_4\}$ which is sent to the GSS. Meanwhile the GW_j stores value set $\{K^{new}_{GT}, K^{new}_G, PID^{new}_{GS}\}$ in its repository for subsequent sessions.

Step 10: After getting message $Auth_5$, the GSS extracts values T_{TA}, B_3 and R_3 as $(T_{TA}, B_3, R_3) = C_2) \oplus h(K_{GSS})$. Next, it computes $K^{new}_G = C_3 \oplus h(K_{GSS}||R_2)$ and $C_4{}^* = h\ (ID_{GW}||C_2||C_3||ID_{GS}||R_2||K^{new}_G)$. Afterwards, it confirms whether $C_4{}^* \stackrel{?}{=} C_4$ such that the authentication session is aborted when these two values are dissimilar. Otherwise, the GSS derives new pseudo-identity as $PID^{new}_{GS} = h\ (ID_{GS}||K_{GSS}||R_2)$ and stores value set $\{PID^{new}_{GS}, K^{new}_G\}$ for subsequent sessions. At last, it composes authentication message $Auth_6 = \{B_3\}$ that is forwarded to D_{Ri}.

Step 11: On receiving message $Auth_6$, the D_{Ri} derives $C^i_D = B_1 \oplus h\ (ID_{DR}||K_{DT})$, $R^i_D = PUF(C^i_D)$, $T_{TA} = B_2 \oplus h(R^i_D)$ and $SK{}^* = h\ (ID_{DR}||T_{TA}||R^i_D||R_1||R_4)$. Next, it determines if $SK{}^* \stackrel{?}{=} SK$ such that the session is terminated upon validation failure. Otherwise, drone D_{Ri} updates its pseudo-identity and shared key as $PID^{new}_{DR} = h\ (ID_{DR}||R^i_D)$ and $KD^{new}_S = h\ (KD_S||R^i_D)$ respectively.

Step 12: The D_{Ri} computes $C_5 = PID^{new}_{DR} \oplus h\ (K_{\beta i})$ and $D_1 = KD^{new}_S \oplus h\ (K_{\beta i})$. Afterwards, it stores $\{C_5, D_1\}$ in its memory. Similarly, the TA updates its pseudo-identity PID_{DR} and shared key KD_S as $PID^{new}_{DR} = h\ (ID_{DR}||R^i_D)$ and $KD^{new}_S = h\ (KD_S||R^i_D)$ respectively.

Challenge-response update phase

The goal of this phase is to renew the challenge-response pairs which have been compromised by the adversaries. To accomplish this, th following 5steps are invoked.

Step 1: The TA generates nonce R_5 and chooses some challenge-response pair (C^i_D, R^i_D) for a particular drone D_{Ri}. Next, it selects pseudo-identity PID_{DR} and derives value $B_1{}^* = C^i_D \oplus h(KD_S||R_5)$.

Step 2: TA generates new challenge $C^{i\,new}_D$ and derives $D_2 = C^{i\,new}_D \oplus h\ (R^i_D)$ as well as $D_3 = h\ (ID_{DR}||R_5||R^i_D||C^i_D||PID_{DR}||D_2)$. This is followed by the construction of password change message $PC_1 = \{PID_{DR}, B_1{}^*, D_2, R_5, D_3\}$ which is sent to D_{Ri} over secure channels.

Step 3: After getting message PC_1, the D_{Ri} extracts challenge C^i_D as $C^i_D = B_1{}^* \oplus h(KD_S||R_5)$. Next, it derives $R^i_D = PUF\ (C^i_D)$ and $D_3{}^* = h\ (ID_{DR}||R_5||R^i_D||C^i_D||PID_{DR}||D_2)$. This is followed by the verification of whether $D_3{}^* \stackrel{?}{=} D_3$. Basically, the password change request is terminated when these two values do not match. Otherwise, D_{Ri} obtains new challenge as $C^{i\,new}_D = D_2 \oplus h\ (R^i_D)$.

Step 4: The D_{Ri} generates new response as $R_D{}^{i\ new} = PUF\ (C_D{}^{i\ new})$. This is followed by the derivation of $D_4 = R^{i\,new}_D \oplus h\ (R^i_D||KD_S)$, $D_5 = PID^{new}_{DR} \oplus h\ (K_{\beta i})$ and $D_1 = KD^{new}_S \oplus h\ (K_{\beta i})$. It then stores value set $\{D_1, D_5\}$ in its memory. Lastly, it composes password change response $PC_2 = \{D_1, D_4\}$ which is sent to the TA over secured communication channels.

Step 5: Upon receiving message PC_2, the TA computes response $R_D{}^{i\ new} = D_4 \oplus h\ (R^i_D||KD_S)$ and $D_1{}^* = KD^{new}_S \oplus h\ (K_{\beta i})$. Next, it determines whether $D_1{}^* \stackrel{?}{=} D_1$ such that the session is terminated upon verification failure. Otherwise, it updates pseudo-identity PID_{DR} and key KD_S as $PID^{new}_{DR} = h\ (ID_{DR}||R^i_D)$ and $KD^{new}_S = h\ (KD_S||R_D{}^i)$ respectively. Finally, the TA stores value set $\{C_D{}^{i\ new}, R_D{}^{i\ new}, PID^{new}_{DR}, KD^{new}_S\}$ in its database.

The process of renewing the challenge-repose pairs between the GSS and GW_j is executed in a similar manner. At the end, the GSS stores the new challenge-response pairs for the subsequent communication session.

Security analysis

This section presents both the formal and informal security analyses of the proposed protocol. The specific details of these analyses are described in the sub-sections that follow.

Formal security analysis

In this sub-section, the security of the proposed protocol is analyzed using the widely utilized Real or Random (RoR) model. The four participants in our protocol include D_{Ri}, GSS, TA and GW_{j}. We denote the oracles for D_{Ri}, GSS, TA as $\prod_{D_{Ri}}^{d}$, $\prod_{G_{SS}}^{gs}$, $\prod_{T_A}^{t}$ and $\prod_{G_{Wj}}^{gw}$ respectively. For partnering to take place between the $\prod_{D_{Ri}}^{d}$ and $\prod_{T_A}^{t}$, these entities must possess the session identity and be able to update each message sent and received in every communication session. In addition, the session key security is attained if an adversary $\ddot{A}$ is unable to obtain the key shared between $\prod_{D_{Ri}}^{d}$ and $\prod_{T_A}^{t}$. In the RoR model, $\ddot{A}$ has all the adversarial capabilities under the Dolev-Yao (DY) threat model. The various queries that can be executed in the RoR model include the *Send* (.), *Execute* (.), *CorruptSC* (.), *CorruptGSS*(.) and *Test* (.), whose details are as follows:

Send ($\prod^{D_{Rx}}$,$Auth_x$): This represents an active attack and through this query, $\ddot{A}$ transmits and receives authentication messages just like any other participant in the communication session.
Execute ($\prod^{D_{Rx}}$,$\prod^{t}$)**:** This is an eavesdropping attack and through this query, $\ddot{A}$ intercepts any message exchanged between D_{Ri} and TA over the public channels.
CorruptSC ($\prod^{D_{Rx}}$): This denotes a smart card loss attack through which $\ddot{A}$ can extract all the security tokens stored in *SC*.
CorruptGSS ($\prod^{GSS}$): This represents the GSS compromise attack through which $\ddot{A}$ can fully control the GSS and expose all the security parameters stored in its repository.
Test ($\prod^{D_{Rx}}$,$\prod^{t}$): In this query, $\ddot{A}$ guesses the session key *SK* through repeated flipping of coin $\mathbb{C}$. Provided that $\mathbb{C} = 1$, then $\ddot{A}$ has successfully obtained the session key. Otherwise, the guess yields a random key or null. Basically, security is attained if *SK* between $\prod^{D_{Rx}}$ and$\prod^{t}$ is safe under the RoR model.

In the RoR model, $\ddot{A}$ can effectively differentiate between random keys and the valid session keys. Basically, $\ddot{A}$ sends numerous *Test* (.) queries to either $\prod^{D_{Ri}}$ or $\prod^{t}$ upon which results of these queries are recorded in bit $\mathbb{C}$. Suppose that $\mathbb{C}^*$ is some randomly selected bit. Then, provided that $\mathbb{C} = \mathbb{C}^*$, $\ddot{A}$ has won the game. Suppose that *Succ* represents adversarial winning of the game. Therefore, we denote $\ddot{A}$'s advantage of breaking the semantic security of our Authenticated Key Exchange (AKE) protocol in polynomial time *t* as follows.

$$Adv_{\ddot{A}}^{AKE}(t) = |2\Pr(Succ) - 1| \tag{12}$$

In this model, all the participants including $\ddot{A}$ have access to the security of both *PUF* (.) and *h* (.).

Theorem 1. *The proposed protocol deploys operator biometric data and identities which are kept confidential. Based on Zipf's law, we let β_L and K_L represents the bits of $K_{\beta i}$ and K_{GSS} respectively. Suppose that $\ddot{A}$ has mounted an attack against our protocol in polynomial time t. We let ρ, σ and τ represent hash, PUF and Send queries respectively. In addition, $|H|$ and $[P|$ denote the size of the range space of the h (.) and PUF (.) respectively. Moreover, we take z_1 and z_2 as Zipdf's parameters which are critical for the determination of bits distribution in $K_{\beta i}$ and K_{GSS} respectively. Then, the advantage of $\ddot{A}$ breaking our protocol and revealing the session key is given by,*

$$Adv_{\ddot{A}}^{AKE}(t) \le \frac{\rho^2}{|H|} + \frac{\sigma^2}{|P|} + 2\max\left\{z_1.\tau^{z_2}, \frac{\tau}{2^{\beta_L}}, \frac{\tau}{2^{K_L}}\right\} \tag{13}$$

Proof. Five games $Game_k$, $k \in [0,4]$ are simulated. In all these games, *Succ* is implied whenever $\ddot{A}$ correctly guesses bit $\mathbb{C}$. The details of these games are described below.

$Game_0$: This is the initial game in which attempts are made to break the security of our protocol under the RoR model. Here, $\ddot{A}$ flips coin $\mathbb{C}$ and hence we obtain the following outcome.

$$Adv_{\ddot{A}}^{AKE}(t) = |2\Pr(Succ_0) - 1| \tag{14}$$

$Game_1$: In this game, adversary intercepts the exchanged messages during the authentication and key setup phase. To accomplish this, the *Execute* (.) query is carried out as follows:

Execute ($\prod^{D_{Rx}}$,$\prod^{t}$)**:** Auth_1, Auth_2,…, Auth_6
Execute ($\prod^{D_{Rx}}$,$\prod^{t}$)**:** {PID_{DR}, R_1},{PID_{GS}, A_1, H_{GSS}^*, A_2},{ID_{GW}, A_3, A_4},{B_4, B_5,C_1},{C_2, C_3, C_4} and{B_3}.

Afterwards, $\ddot{A}$ attempts to guess the session key $SK = h\ (ID_{\text{DR}}||T_{\text{TA}}||R_{\text{D}}^{\text{i}}||R_1||R_4)$. However, the session key in our protocol is protected by the response R_{D}^{i} which is only known by TA and can only be computed by legitimate drone D_{Ri}. In addition, $\ddot{A}$ still needs drone identity ID_{DR}, TA token T_{TA} as well as random numbers R_1 and R_4 to be able to derive *SK*. As such, only the legitimate TA and D_{Ri} can derive session key *SK*. Consequently, $\ddot{A}$'s probability of winning $Game_1$ through eavesdropping has not increased and hence,

$$\Pr[Succ_1] = \Pr[Succ_0] \tag{15}$$

$Game_2$: In this game, the adversary $\ddot{A}$ simulates the *Send* ($\prod^{D_{Rx}}$,$Auth_x$) query with the goal of receiving and editing the exchanged messages. During the authentication and key negotiation phase, messages $Auth_1$, $Auth_2$, $Auth_3$, $Auth_4$, $Auth_5$ and $Auth_6$ are exchanged. Here, $Auth_1 = \{PID_{\text{DR}}, R_1\}$, $Auth_2 = \{PID_{\text{GS}}, A_1, H_{\text{GSS}}^*, A_2\}$, $Auth_3 = \{ID_{\text{GW}}, A_3, A_4\}$, $Auth_4 = \{B_4, B_5, C_1\}$, $Auth_5 = \{C_2, C_3, C_4\}$ and $Auth_6 = \{B_3\}$, $PID_{\text{DR}}^{\text{new}} = h(ID_{\text{DR}}||R_{\text{D}}^{\text{i}})$, $A_1 = (ID_{\text{GS}}, PID_{\text{DR}}, R_1, R_2) \oplus h(K_{\text{GSS}})$, $A_2 = h\ (ID_{\text{GS}}||A_1||K_{\text{G}}||R_2||PID_{\text{GS}}||H_{\text{GSS}}^*)$, $A_3 = (ID_{\text{GW}}, PID_{\text{DR}}, R_1, R_3) \oplus h(K_{\text{GT}})$, $A_4 = h\ (ID_{\text{GW}}||ID_{\text{TA}}||A_3||K_{\text{GT}}||R_3)$, $B_1 = C_{\text{D}}^{\text{i}} \oplus h(ID_{\text{DR}}||K_{\text{DT}})$, $B_2 = T_{\text{TA}} \oplus h(R_{\text{D}}^{\text{i}})$, $SK = h\ (ID_{\text{DR}}||T_{\text{TA}}||R_{\text{D}}^{\text{i}}||R_1||R_4)$, $B_3 = \{B_1, B_2, SK\}$, $B_4 = (T_{\text{TA}}, B_3, R_4) \oplus h(K_{\text{GT}})$, $B_5 = K_{\text{GT}}^{\text{new}} \oplus h(K_{\text{GT}}||R_3)$, $C_1 = h\ (ID_{\text{GW}}||ID_{\text{TA}}||B_4||R_3||R_4||K_{\text{GT}}^{\text{new}})$, $C_2 = (T_{\text{TA}}, B_3, R_3) \oplus h(K_{\text{GSS}})$, $C_3 = K_{\text{G}}^{\text{new}} \oplus h(K_{\text{GSS}}||R_2)$ and $C_4 = h\ (ID_{\text{GW}}||C_2||C_3||ID_{\text{GS}}||R_2||K_{\text{G}}^{\text{new}})$. Evidently, all the authentication messages are protected by keys such as K_{GSS}, challenge R_{D}^{i} and collision-resistant one-way hashing function. As such, the *Send* ($\prod^{D_{Rx}}$,$Auth_x$) query is unable to find any collision. Based on the birthday paradox,

$$|\Pr\ [Succ_2] - \Pr[Succ_1]| \le \frac{\rho^2}{2|H|} \tag{16}$$

$Game_3$: The aim of this game is to simulate both *PUF* (.) and *Send* (.) queries. Based on secure *PUF* definitions [30],

$$|\Pr\ [Succ_3] - \Pr[Succ_2]| \le \frac{\sigma^2}{2|P|} \tag{17}$$

$Game_4$: In this game, $\ddot{A}$ simulates both *CorruptSC*($\prod^{D_{Rx}}$) and *CorruptGSS*($\prod^{GSS}$) queries with the aim of retrieving values {HD_{Ri}, ID_{DR}^*, PID_{DR}^*, K_{DT}^*, KD_{Syn}^*, PID_{TSyn}^*} from *SC* and {K_{G}, KGW_{Syn}, PID_{GSyn}, $C_{\text{GS}}^{\text{i}*}$, C_{GSyn}^*} from the *GSS* database. However, $\ddot{A}$ is unable to obtain $K_{\beta i}$ and K_{GSS} from the *GSS* database and *SC*. According to [52], the probability of correctly guessing $K_{\beta i}$ as β_{L} and K_{GSS} as K_{L} are $(2^{\beta_L})^{-1}$ and $(2^{K_L})^{-1}$ respectively. As such, both $Game_3$ and $Game_4$ are resilient against guessing attacks and hence,

$$|\Pr\ [Succ_4] - \Pr[Succ_3]| \le \max\left\{z_1.\tau^{z_2}, \frac{\tau}{2^{\beta_L}}, \frac{\tau}{2^{K_L}}\right\} \tag{18}$$

To win $Game_4$, adversary $\ddot{A}$ must obtain $\mathbb{C}^*$ that is equivalent to $\mathbb{C}$. Therefore,

$$\Pr\ [Succ_4] = 2^{-1} \tag{19}$$

Based on eqs. (14), (15) and (19), we obtain the following.

$$\begin{aligned}\frac{1}{2}Adv_{\ddot{A}}^{AKE}(t) &= \left|\Pr(Succ_0) - \frac{1}{2}\right| \\ &= |\Pr(Succ_1) - \frac{1}{2}| \\ &= |\Pr[Succ_1] - \Pr[Succ_4]|\end{aligned} \tag{20}$$

Applying the triangle inequality to Eqs. (16), (17) and (18), we get the following.

$$\begin{aligned}|\Pr\,[Succ_1] - \Pr[Succ_4]| &\le |\Pr[Succ_1] - \Pr[Succ_3]| \\ &+|\Pr[Succ_3] - \Pr[Succ_4]| \\ &\le |\Pr[Succ_1] - \Pr[Succ_2]| \\ &+|\Pr[Succ_2] - \Pr[Succ_3]| \\ &+|\Pr[Succ_3] - \Pr[Succ_4]| \\ &\le \frac{\rho^2}{|H|} + \frac{\sigma^2}{|P|} + 2\max\left\{z_1.\tau^{z_2}, \frac{\tau}{2^{\beta_L}}, \frac{\tau}{2^{K_L}}\right\}\end{aligned} \tag{21}$$

Based on Eqs. (20) and (21), we obtain,

$$Adv_{\ddot{A}}^{AKE}(t) \le \frac{\rho^2}{|H|} + \frac{\sigma^2}{|P|} + 2\max\left\{z_1.\tau^{z_2}, \frac{\tau}{2^{\beta_L}}, \frac{\tau}{2^{K_L}}\right\} \tag{22}$$

This completes the proof.

Informal security analysis

In this sub-section, various lemmas are stated and proofed to show that our protocol is robust against typical drone attacks and offers numerous salient security and privacy features.

Lemma 1. *Ephemeral secret leakage and eavedropping attacks are prevented.*

Proof. Suppose that an adversary $\ddot{A}$ is interested in obtaining operator biometric data β_i and key $K_{\beta i}$ respectively. To achieve this, all authentication messages $Auth_1 = \{PID_{DR}, R_1\}$, $Auth_2 = \{PID_{GS}, A_1, H_{GSS}^*, A_2\}$, $Auth_3 = \{ID_{GW}, A_3, A_4\}$, $Auth_4 = \{B_4, B_5, C_1\}$, $Auth_5 = \{C_2, C_3, C_4\}$ and $Auth_6 = \{B_3\}$ are intercepted over the public channels. Evidently, none of these messages contain plaintext β_i and $K_{\beta i}$. Therefore, these attacks flop.

Lemma 2. *Mutual authentication is achieved.*

Proof. All the network entities mutually verify their authenticity before message exchanges. For instance, the drone authenticates the operator by confirming whether $V_C^* \stackrel{?}{=} V_C$ while the GW_j authenticates the GSS by checking if $A_2^* \stackrel{?}{=} A_2$. Similarly, the TA authenticates GW_j by checking whether $A_4^* \stackrel{?}{=} A_4$ while GW_j verifies TA by confirming if $C_1^* \stackrel{?}{=} C_1$. On the other hand, GSS validates GW_j by checking if $C_4^* \stackrel{?}{=} C_4$ while the D_{Ri} verifies GSS by confirming whether $SK^* \stackrel{?}{=} SK$. In all these instances, the session is aborted upon authentication failure.

Lemma 3. *Our scheme withstands side-channeling attacks.*

Proof. Suppose that adversary $\ddot{A}$ has stolen the smart card and has retrieved all the secrets stored in it. Next, an attempt is made to use these secrets to derive the session key. In our scheme, parameter set $\{HD_{Ri}, ID_{DR}^*, PID_{DR}^*, K_{DT}^*, KD_{Syn}^*, PID_{TSyn}^*\}$ is stored in the smart card SC. During the mutual authentication and session key establishment, the TA computes session keys as $SK = h\,(ID_{DR}||T_{TA}||R_D^i||R_1||R_4)$ while the D_{Ri} derives it as $SK^* = h\,(ID_{DR}||T_{TA}||R_D^i||R_1||R_4)$. It is clear that even through $\ddot{A}$ has obtained identity ID_{DR} from the SC, other parameters such as TA token T_{TA}, challenge response R_D^i as well as random numbers R_1 and R_4 are still unavailable for the successful derivation of the session key.

Lemma 4. *Untraceability and anonymity are preserved.*

Proof. Suppose that an adversary is interested in tracking the drone using the captured authentication messages exchanged over the public communication channels. In our protocol, the GSS and drone D_{Ri} are assigned one-time pseudo-identities PID_{GS} and PID_{DR} respectively. During the authentication and key establishment phase, messages $Auth_1 = \{PID_{DR}, R_1\}$, $Auth_2 = \{PID_{GS}, A_1, H_{GSS}^*, A_2\}$, $Auth_3 = \{ID_{GW}, A_3, A_4\}$, $Auth_4 = \{B_4, B_5, C_1\}$, $Auth_5 = \{C_2, C_3, C_4\}$ and $Auth_6 = \{B_3\}$ are exchanged over the public channels. As such, these messages may be captured by the adversary. However, none of these messages contain plaintext unique identities ID_{DR} and ID_{GS} of the D_{Ri} and GSS respectively. Instead, pseudo-identities PID_{DR} and PID_{GS} of the D_{Ri} and GSS are the ones incorporated in these messages. After every successful authentication procedures, the pseudo-identity of D_{Ri} is updated as $PID_{DR}^{new} = h\,(ID_{DR}||R_D^i)$. Similarly, the pseudo-identity of the GSS is refreshed as $PID_{GS}^{new} = h\,(ID_{GS}||K_{GSS}||R_2)$. Therefore, the true identities of D_{Ri} and GSS cannot be deciphered from the exchanged messages. Suppose that adversary $\ddot{A}$ wants to associate some communication sessions and messages to particular drone D_{Ri} and GSS. To achieve this, $\ddot{A}$ must capture personally identifiable information of the D_{Ri} and GSS. Such information include real identities ID_{DR} and ID_{GS} of the D_{Ri} and GSS respectively. However, it has already been shown that no such plaintext identities can be deciphered from the exchanged messages. Any attempt at using the pseudo-identities PID_{DR} and PID_{GS} for tracking purposes will also fail due to the updating of these pseudo-identities. It is therefore difficult for adversary $\ddot{A}$ to link the exchanged messages to their equivalent values in the previous or subsequent communication sessions. This is because the contents of these messages such as pseudo-identities PID_{DR} and PID_{GS} are refreshed after every successful authentication session. Therefore, the captured messages will appear quite different from their previous or subsequent values. Tracing of drone's identity is a form of eavesdropping since it requires that messages be captured over the public channels, and their content examined or observed. Therefore, our scheme is secure under the DY threat model.

Lemma 5. *Cloning attacks are prevented.*

Proof. In our scheme, we deploy *PUF* to generate challenge-response pairs (C_D^i, R_D^i) and (C_{GS}^i, R_{GS}^i) for the drone and the GSS respectively. Therefore, it is difficult for the adversary $\ddot{A}$ to clone both drone D_{Ri} and GSS.

Lemma 6. *Our scheme achieves perfect forward and backward secrecy.*

Proof. In our scheme, the shared keys, challenge-response pairs and session keys are refreshed after every successful authentication session. For instance, the TA derives new shared key as K_{GT}^{new}, GW_j updates shared key K_G and pseudo-identity PID_{GS} as K_G^{new} and $PID_{GS}^{new} = h\,(ID_{GS}||K_{GSS}||R_2)$ respectively. On its part, the GSS derives new pseudo-identity as $PID_{GS}^{new} = h\,(ID_{GS}||K_{GSS}||R_2)$. Similarly, D_{Ri} and TA update pseudo-identity and shared key as $PID_{DR}^{new} = h\,(ID_{DR}||R_D^i)$ and $KD_S^{new} = h\,(KD_S||R_D^i)$ respectively. In our protocol, parameter set $\{HD_{Ri}, ID_{DR}^*, PID_{DR}^*, K_{DT}^*, KD_{Syn}^*, PID_{TSyn}^*\}$ is stored in the smart card SC while value set $\{PID_{DR}, K_{DT}, C_D^i, R_D^i, KD_{Syn}, PID_{TSyn}, CD_{Syn}, RD_{Syn}, ID_{GW}, K_{GT}, KG_{Syn}\}$ is stored in TA's database. Similarly, GW_j stores parameter set $\{K_{GT}, KG_{Syn}, ID_{GS}, K_G, KGW_{Syn}, PID_{GSyn}, C_{GS}^i, C_{GSyn}, R_{GS}^i, R_{GSyn}\}$ in its repository while GSS stores parameter set $\{K_G, KGW_{Syn}, PID_{GSyn}, C_{GS}^{i*}, C_{GSyn}^*\}$ in its database. We compute the session key as $SK = SK^* = h\,(ID_{DR}||T_{TA}||R_D^i||R_1||R_4)$. Due to the incorporattion of one-time response R_D^i and random numbers such as R_1 and R_4, an adversary with current keys and pseudo-identities is unable to compute session keys for past and subsequent sessions.

Lemma 7. *Man-in-the-middle attacks are thwarted.*

Proof. Suppose that an adversary interecepts and modifies all the exchanged messages during the authentication and key negotiation phase. These messages include $Auth_1 = \{PID_{DR}, R_1\}$, $Auth_2 = \{PID_{GS}, A_1, H_{GSS}^*, A_2\}$, $Auth_3 = \{ID_{GW}, A_3, A_4\}$, $Auth_4 = \{B_4, B_5, C_1\}$, $Auth_5 = \{C_2, C_3, C_4\}$ and $Auth_6 = \{B_3\}$. Here, $A_1 = (ID_{GS}, PID_{DR}, R_1, R_2) \oplus h(K_{GSS})$, $A_2 = h(ID_{GS}||A_1||K_G||R_2||PID_{GS}||H_{GSS}^*)$, $A_3 = (ID_{GW}, PID_{DR}, R_1, R_3) \oplus h(K_{GT})$, $A_4 = h(ID_{GW}||ID_{TA}||A_3||K_{GT}||R_3)$, $B_1 = C_D^i \oplus h(ID_{DR}||K_{DT})$, $B_2 = T_{TA} \oplus h(R_D^i)$, $SK = h(ID_{DR}||T_{TA}||R_D^i||R_1||R_4)$, $B_3 = \{B_1, B_2, SK\}$, $B_4 = (T_{TA}, B_3, R_4) \oplus h(K_{GT})$, $B_5 = K_{GT}^{new} \oplus h(K_{GT}||R_3)$, $C_1 = h(ID_{GW}||ID_{TA}||B_4||R_3||R_4||K_{GT}^{new})$, $C_2 = (T_{TA}, B_3, R_3) \oplus h(K_{GSS})$, $C_3 = K_G^{new} \oplus h(K_{GSS}||R_2)$ and $C_4 = h(ID_{GW}||C_2||C_3||ID_{GS}||R_2||K_G^{new})$. At the receiver side, these messages are mutually verified through checks such as $V_C^* \stackrel{?}{=} V_C$, $A_2^* \stackrel{?}{=} A_2$, $A_4^* \stackrel{?}{=} A_4$, $C_1^* \stackrel{?}{=} C_1$, $C_4^* \stackrel{?}{=} C_4$ and $SK^* \stackrel{?}{=} SK$. As such, any modified message is easily detected at the receiver end and the session terminated. Therefore, MitM attacks are detectable and preventable in our scheme.

Lemma 8. *Session key security is attained.*

Proof. In our protocol, the *TA* and D_{Ri} derive the session key as $SK = h(ID_{DR}||T_{TA}||R_D^i||R_1||R_4)$ and $SK^* = h(ID_{DR}||T_{TA}||R_D^i||R_1||R_4)$ respectively. Here, ID_{DR} is the drone unique identity, $T_{TA} = B_2 \oplus h(R_D^i)$ is TA's token, R_D^i is the challenge response while both R_1 and R_4 are random numbers. It is evident that all the parameters are pseudo-stochastic apart from ID_{DR}. As such, an adversary $\ddot{A}$ with the current session key is unable to utilize it to derive the *SK* for the previous and subsequent communication sessions.

Lemma 9. *This protocol withstands replay attacks.*

Proof. Suppose that messages $Auth_1 = \{PID_{DR}, R_1\}$, $Auth_2 = \{PID_{GS}, A_1, H_{GSS}^*, A_2\}$, $Auth_3 = \{ID_{GW}, A_3, A_4\}$, $Auth_4 = \{B_4, B_5, C_1\}$, $Auth_5 = \{C_2, C_3, C_4\}$ and $Auth_6 = \{B_3\}$ exchanged during the authentication and key negotiation phase are captured by $\ddot{A}$. Here, $A_1 = (ID_{GS}, PID_{DR}, R_1, R_2) \oplus h(K_{GSS})$, $A_2 = h(ID_{GS}||A_1||K_G||R_2||PID_{GS}||H_{GSS}^*)$, $A_3 = (ID_{GW}, PID_{DR}, R_1, R_3) \oplus h(K_{GT})$, $A_4 = h(ID_{GW}||ID_{TA}||A_3||K_{GT}||R_3)$, $B_1 = C_D^i \oplus h(ID_{DR}||K_{DT})$, $B_2 = T_{TA} \oplus h(R_D^i)$, $SK = h(ID_{DR}||T_{TA}||R_D^i||R_1||R_4)$, $B_3 = \{B_1, B_2, SK\}$, $B_4 = (T_{TA}, B_3, R_4) \oplus h(K_{GT})$, $B_5 = K_{GT}^{new} \oplus h(K_{GT}||R_3)$, $C_1 = h(ID_{GW}||ID_{TA}||B_4||R_3||R_4||K_{GT}^{new})$, $C_2 = (T_{TA}, B_3, R_3) \oplus h(K_{GSS})$, $C_3 = K_G^{new} \oplus h(K_{GSS}||R_2)$ and $C_4 = h(ID_{GW}||C_2||C_3||ID_{GS}||R_2||K_G^{new})$. Due to the incorporation of random numbers R_1, R_2, R_3 and R_4, any replayed messages are easily detected at the receiver side. In addition, the deployed pseudo-identities PID_{DR} and PID_{GS} are frequently updated as $PID_{DR}^{new} = h(ID_{DR}||R_D^i)$ and $PID_{GS}^{new} = h(ID_{GS}||K_{GSS}||R_2)$ after every successful authentication session. This further prevents any replay attempts initiated by adversary $\ddot{A}$.

Lemma 10. *Physical attacks are prevented.*

Proof. In our scheme, *PUF* is used to generate the challenge-response pairs that facilitate the mutual authentication procedures. Based on *Lemma 6*, *PUF* cannot be cloned. In addition, our scheme deploys operator biometric data β_i and key $K_{\beta i}$. During the registration phase, parameter set $\{HD_{Ri}, ID_{DR}^*, PID_{DR}^*, K_{DT}^*, KD_{Syn}^*, PID_{TSyn}^*\}$ is stored in the smart card *SC* while value set $\{PID_{DR}, K_{DT}, C_D^i, R_D^i, KD_{Syn}, PID_{TSyn}, CD_{Syn}, RD_{Syn}, ID_{GW}, K_{GT}, KG_{Syn}\}$ is stored in TA's database. Similarly, GW_j stores parameter set $\{K_{GT}, KG_{Syn}, ID_{GS}, K_G, KGW_{Syn}, PID_{GSyn}, C_{GS}^i, C_{GSyn}, R_{GS}^i, R_{GSyn}\}$ in its repository while GSS stores parameter set $\{K_G, KGW_{Syn}, PID_{GSyn}, C_{GS}^{i*}, C_{GSyn}^*\}$ in its database. It is evident that β_i and $K_{\beta i}$ are never stored in the database or memory and hence they are safe even during active physical attack.

Lemma 11. *This protocol prevents DoS attacks.*

Proof. Suppose that an adversary captures all the messages $Auth_1 = \{PID_{DR}, R_1\}$, $Auth_2 = \{PID_{GS}, A_1, H_{GSS}^*, A_2\}$, $Auth_3 = \{ID_{GW}, A_3, A_4\}$, $Auth_4 = \{B_4, B_5, C_1\}$, $Auth_5 = \{C_2, C_3, C_4\}$ and $Auth_6 = \{B_3\}$ exchanged during the authentication and key agreement phase. Afterwards, $\ddot{A}$ tries to mount a DoS. In our protocol, we deploy the synchronization set {challenge-response pair, unique identity, key} to ensure that other entities can still continue communicating when one or more entities are under DoS attacks. For instance, if GSS fails to receive any message from any participant, it can initiate the session by sending synchronization request set $\{PID_{GS}, K_{\beta i}, (C_{GS}^i, R_{GS}^i)\}$.

Lemma 12. *De-synchronization attacks are prevented.*

Proof. Suppose that drone D_{Ri} is de-synchronized. When this happens, it substitutes $\{PID_{DR}, KD_S\}$ with $\{PID_{S1}, KD_{S1}\}$ and re-initiates the authentication process. Similarly, the GSS substitutes $\{PID_{GS}, K_G\}$ with $\{PID_{G1}, KGW_{S1}\}$, replaces K_{GT} with KG_{S1} and restarts the authentication procedures. Therefore, our scheme can withstand de-synchronization attacks.

Lemma 13. *Our scheme is robust against challenge-response tracking attacks.*

Proof. In this attack, adversary is interested in establishing the owner of a particular challenge C_D^i. However, we protect any challenge using shared key K_{DT}. For instance, we mask the owner of challenge C_D^i in $B_1 = C_D^i \oplus h(ID_{DR}||K_{DT})$ using key K_{DT}. Therefore, attacker $\ddot{A}$ is unable to associate this challenge to a particular drone with identity ID_{DR}.

Lemma 14. *Impersonation attacks are thwarted.*

Proof. Suppose that $\ddot{A}$ has stolen the smart card and has activated *PUF*. Next, an attempt is made to obtain current token set $\{R_D^i, SK\}$ and new tokens $\{R_D^{i\ new}, SK^*\}$. However, none of the messages $Auth_1 = \{PID_{DR}, R_1\}$, $Auth_2 = \{PID_{GS}, A_1, H_{GSS}^*, A_2\}$, $Auth_3 = \{ID_{GW}, A_3, A_4\}$, $Auth_4 = \{B_4, B_5, C_1\}$, $Auth_5 = \{C_2, C_3, C_4\}$ and $Auth_6 = \{B_3\}$ exchanged over public channels contain these tokens. In accordance with *Lemma 10*, these tokens are not stored in the smart card *SC*. As such, $\ddot{A}$ cannot impersonate a particular *SC*.

Performance evaluation

In most of the authentication protocols, supported security and privacy features, computation, communication and energy costs are the most common metrics utilized to evaluate their performances. As such, we deploy these metrics to assess the performance of the proposed protocol. The sub-sections below describe these evaluations in detail.

Computation costs

In this sub-section, the execution time of the various cryptographic operations are deployed to derive the overall computation overheads of our protocol. The experimentations were implemented in an HP-EliteBook machine with 4GB RAM, 2.70 GHz processor, Intel Core (TM)-i7–5700HQ and running Ubuntu 22.04 LTS operating system. Under this environment, the Pairing-Based Cryptography Library (PBC) library was deployed to obtain the running time of the various cryptographic operations as given in Table 2.

Table 2
Cryptographic run time.

Cryptographic operation	Time (ms)
ECC point multiplication, T_M	0.0273
Fuzzy extractor, T_{FE}	0.0098
PUF operation, T_{PUF}	0.0021
Symmetric encryption/decryption, T_S	0.0046
One-way hashing, T_H	0.0025
ECC point addition, T_A	0.0032

Table 3
Computation costs comparisons.

Scheme	Derivations	Total (ms)
Tanveer et al. [14]	$20T_H + 6T_M + 9T_S$	0.2552
Yu et al. [13]	$29T_H + 2T_{FE}$	0.0921
Hussain et al. [15]	$16T_H + T_S$	0.0446
Nikooghadam et al. [33]	$19T_H + 4T_M$	0.1567
Bera et al. [25]	$18T_H + 10T_M + 3T_A$	0.3276
Park et al. [30]	$32T_H + 2T_{FE}$	0.0996
Akram et al. [48]	$23T_H + 2T_S$	0.0667
Wu et al. [29]	$29T_H + T_{FE}$	0.0823
Proposed	$6T_H + 2T_{PUF} + 2T_{FE}$	0.0388

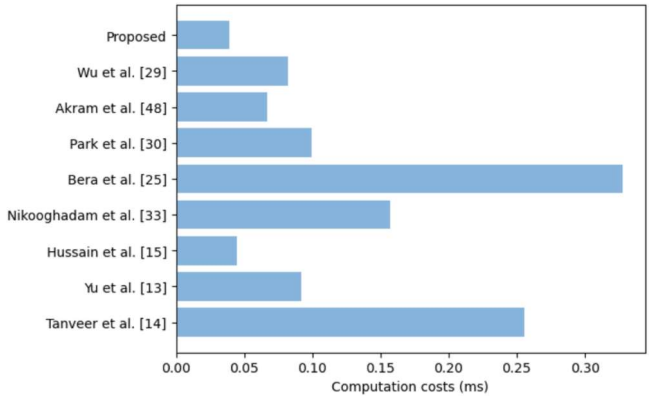

Fig. 5. Computation costs comparisons.

During the authentication and key establishment, the drone/SC execute a single PUF operation, one-way hashing operation and a single fuzzy extractor reconstruction. On the other, the GSS/GW_j execute a single one-way hashing operation, single fuzzy extractor generation and one PUF operation. However, the TA executes 4 one-way hashing operations. As such, the cummulative computation overhead is $6T_H + 2T_{PUF} + 2T_{FE}$. Table 3 presents the comparative evaluations of our protocol with other related schemes.

As shown in Fig. 5, the scheme in [25] incurs the highest computation costs. This is followed by the protocols in [14,33,30,13,29,48] and [15] respectively. The high computation costs in the protocols developed in [25,14] and [33] can be attributed to the extensive elliptic curve point multiplication operations that must be executed during the authentication and key agreement phase. Although the schemes in [30, 13,29,48] and [15] majorly execute relatively lightweight one-way hashing operations, these operations are numerous and hence all these protocols have relatively high computation costs. Since the IoD is limited in terms of computation power, all these schemes are not suitable for deployment in IoD devices.

On the other hand, our protocol incurs the lowest computation costs of only 0.0388 ms. This is attributed to the low number of relatively lightweight one-way hashing, *PUF* and fuzzy extraction operations executed in our scheme. Considering the computation limitations of the drones, the proposed protocol is the most ideal for deployment in these devices.

Communication costs

In this sub-section we use the number and size of the messages exchanged during the authentication and key negotiation phase to derive the communication cost of our scheme. During this phase, 5 messages are exchanged, which include $Auth_1 = \{PID_{DR}, R_1\}$, $Auth_2 = \{PID_{GS}, A_1, H_{GSS}^*, A_2\}$, $Auth_3 = \{ID_{GW}, A_3, A_4\}$, $Auth_4 = \{B_4, B_5, C_1\}$, $Auth_5 = \{C_2, C_3, C_4\}$ and $Auth_6 = \{B_3\}$. Here, $A_1 = (ID_{GS}, PID_{DR}, R_1, R_2) \oplus h(K_{GSS})$, $A_2 = h\,(ID_{GS}||A_1||K_G||R_2||PID_{GS}||H_{GSS}^*)$, $A_3 = (ID_{GW}, PID_{DR}, R_1, R_3) \oplus h(K_{GT})$, $A_4 = h\,(ID_{GW}||ID_{TA}||A_3||K_{GT}||R_3)$, $B_1 = C_D^i \oplus h(ID_{DR}||K_{DT})$, $B_2 = T_{TA} \oplus h(R_D^i)$, $SK = h\,(ID_{DR}||T_{TA}||R_D^i||R_1||R_4)$, $B_3 = \{B_1, B_2, SK\}$, $B_4 = (T_{TA}, B_3, R_4) \oplus h(K_{GT})$, $B_5 = K_{GT}^{new} \oplus h(K_{GT}||R_3)$, $C_1 = h\,(ID_{GW}||ID_{TA}||B_4||R_3||R_4||K_{GT}^{new})$, $C_2 = (T_{TA}, B_3, R_3) \oplus h(K_{GSS})$, $C_3 = K_G^{new} \oplus h(K_{GSS}||R_2)$ and $C_4 = h\,(ID_{GW}||C_2||C_3||ID_{GS}||R_2||K_G^{new})$. Based on the values in [15] and [18], the sizes of the various operations are presented in Table 4.

Table 4
Parametric sizes.

Parameter	Size (bit)
Identity	32
Certificate	160
Random number	128
Timestamp	32
Symmetric key	128
Hash function	160
ECC point	320

Table 5
Communication costs computations.

Message	Details	Total (bits)
$D_{Ri} \rightarrow GSS$	$Auth_1 = \{PID_{DR}, R_1\}$ $PID_{DR} = 32$; $R_1 = 128$	160
$GSS \rightarrow GW_j$	$Auth_2 = \{PID_{GS}, A_1, H_{GSS}^*, A_2\}$ $PID_{GS} = 32$; $A_1 = H_{GSS}^* = A_2 = 160$	512
$GW_j \rightarrow TA$	$Auth_3 = \{ID_{GW}, A_3, A_4\}$ $ID_{GW} = 32$; $A_3 = A_4 = 160$	352
$TA \rightarrow GW_j$	$Auth_4 = \{B_4, B_5, C_1\}$ $B_4 = B_5 = C_1 = 160$	480
$GW_j \rightarrow GSS$	$Auth_5 = \{C_2, C_3, C_4\}$ $C_2 = C_3 = C_4 = 160$	480
$GSS \rightarrow D_{Ri}$	$Auth_6 = \{B_3\}$ $B_3 = 480$	480
Total		**2464**

Table 6
Communications costs comparisons.

Scheme	Exchanged messages	Total (bits)
Tanveer et al. [14]	3	2080
Yu et al. [13]	4	2048
Hussain et al. [15]	3	1856
Nikooghadam et al. [33]	3	2336
Bera et al. [25]	3	2336
Park et al. [30]	3	2560
Akram et al. [48]	3	2304
Wu et al. [29]	3	3360
Proposed	**7**	**2464**

Using the parametric sizes in Table 4 above, the communication costs of the 6 messages exchanged during the authentication and key agreement phase are derived as shown in Table 5.

Based on the values in Table 5, the communication cost of our protocol is 2464 bits. Table 6 presents comparative evaluation of our scheme against the communication costs of other related protocols.

As shown in Fig. 6, the scheme in [29] incurs the heaviest communication overheads, followed by the protocol in [30]. On the other hand, the proposed scheme incurs the third highest communication overheads. This is justified by the fact that our protocol authenticates all the four communicating entities unlike the other schemes.

Although the protocols in [13–15,25,33] and [48] have slightly lower comunication overeheads, they have numerous security and privacy challenges as evidenced in Table 7. Therefore, our scheme offers a good trade-off between communication costs and perfect security.

Energy costs

Drones are limited in terms of energy and hence we derive the energy

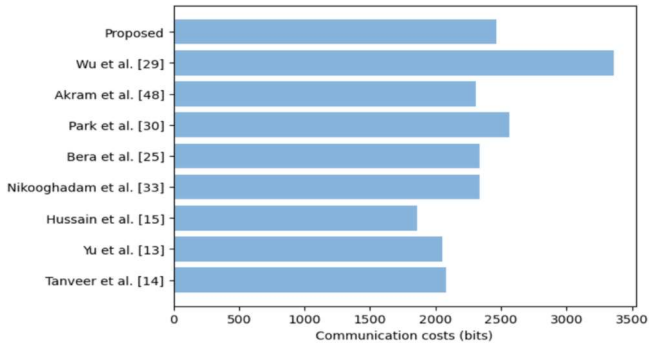

Fig. 6. Communications costs comparisons.

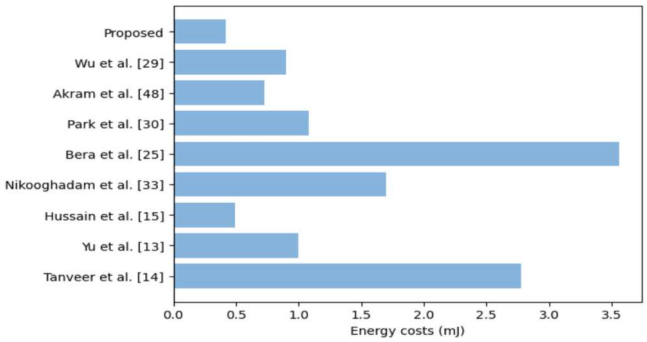

Fig. 7. Energy costs comparisons.

consumption of our protocol in this sub-section. Taking total computation cost as C and maximum processing power as P, energy (E) consumption is given by [53] as:

$$E = C \times P$$

In wireless communication systems, P is taken to be 10.88 W [25]. As such, the energy consumptions for the schemes in [13–15,33,25,30,48] and [29] are 2.78 mJ, 1 mJ, 0.49 mJ, 1.70 mJ, 3.56 mJ, 1.08 mJ, 0.73 mJ and 0.9 mJ respectively. As shown in Fig. 7, the scheme in [25] incurs the highest energy costs owing to its extensive ECC point multiplication operations.

This is followed by the protocols in [14,33,30,13,29,48] and [15] respectively, which also execute relative high number of elliptic curve point multiplications or high number of hashing operations. On the other hand, our protocol incurs the lowest energy costs of only 0.42 mJ. This is attributed to the relatively low number of lightweight hashing, PUF and fuzzy extraction operations. Since drones are battery powered, protocols with high energy consumptions will drain the battery. As such, our scheme is energy efficient and hence suitable for deployment in an IoD environment.

Supported functionalities

In this sub-section, we compare the security and privacy features provided by our protocol against those offered by other related schemes. In addition, we compare attack resilience of the proposed protocol against resilience offered by other peer schemes.

As shown in Table 7, the protocols in [13–15,33,25,30,48] and [29] support 10, 11,13,7,10,13,10 and 11 functionalities respectively. On the other hand, the proposed protocol supports all the 18 functionalities. As such, the scheme in [33] is the most insecure while our scheme is the most secure. In overall, although the proposed protocol incurs slightly high communication costs during the authentication of all the four IoD entities, it is the most robust in terms of security and privacy posture.

Conclusion

The drones play critical roles in both civilian and military domains. The high volumes and sensitive nature of the collected data exchanged over the insecure public channels expose this data to numerous threats. In addition, most drones fly over un-attended locations that can also be hostile. As such, it is possible for attackers to physically capture the drones and extract the stored secrets as well as data. Therefore, many schemes have been presented over the recent past to counter these threats. However, it has been shown that majority of these protocols are still susceptible to many security and privacy threats. In addition, some of them incur extensive computation, energy and communication overheads. Therefore, majority of these protocols are inefficient when applied in the IoD environment. On the other hand, the developed protocol has been shown to be efficient, and hence suitable for deployment in the resource-constrained drones. We execute extensive formal security analysis using the RoR model, which demonstrates the

Table 7
Supported functionalities.

	[14]	[13]	[15]	[33]	[25]	[30]	[48]	[29]	Proposed
Functionality									
F1	√	√	√	√	√	√	√	√	√
F2	√	√	√	√	√	√	√	√	√
F3	√	√	√	√	√	√	√	√	√
F4	√	√	√	√	√	√	√	√	√
F5	×	√	√	×	√	√	√	√	√
F6	√	√	√	√	×	√	√	√	√
Resilience against:									
F7	×	×	×	×	×	×	×	×	√
F8	×	×	√	×	×	×	×	×	√
F9	×	×	√	×	×	√	×	√	√
F10	√	√	√	√	√	√	√	√	√
F11	√	√	√	√	√	√	×	×	√
F12	√	√	√	×	√	√	√	√	√
F13	×	×	×	×	×	×	×	×	√
F14	×	×	×	×	√	√	√	√	√
F15	√	√	√	√	×	×	×	×	√
F16	√	√	√	×	√	√	√	√	√
F17	×	×	×	×	×	√	×	×	√
F18	×	×	×	×	×	×	×	×	√

F1: Mutual authentication; F2: Session key agreement; F3: Key security; F4: Untraceability; F5: Anonymity; F6: Forward and backward secrecy; F7: Side-channeling; F8: De-synchronization; F9: DoS; F10: MitM; F11: Physical capture; F12: Impersonation; F13: Eavesdropping; F14: Ephemeral secret leakage; F15: Spoofing; F16: Replay; F17: Cloning; F18: Challenge-response tracking; √: Feature present; ×: feature absent or not considered

robustness of the mutual authentication procedures. In addition, semantic analysis shows that our scheme can withstand numerous threats such as side-channeling, de-synchronization, DoS, MitM, hysical capture, impersonation, eavesdropping, ephemeral secret leakage, spoofing, replay, cloning and challenge-response tracking. Future work will involve further reduction of the communication overheads of this protocol and comparative evaluations using the metrics that were out of scope of the current work. It will also be necessary to conducting robust IoD-based simulations to rigorously test this protocol.

CRediT authorship contribution statement

Vincent Omollo Nyangaresi: Validation, Methodology, Conceptualization. **Istabraq M. Al-Joboury:** Investigation, Data curation. **Kareem Ali Al-sharhanee:** Writing – original draft, Funding acquisition. **Ali Hamzah Najim:** Software, Resources. **Ali Hashim Abbas:** Visualization, Project administration. **Hussein Muhi Hariz:** Writing – review & editing.

Declaration of competing interest

The authors declare that they do not have any competing interests.

Data availability

No data was used for the research described in the article.

References

[1] L. Abualigah, A. Diabat, P. Sumari, A.H. Gandomi, Applications, deployments, and integration of internet of drones (IOD): a review, IEEE Sens. J. 21 (22) (2021) 25532–25546.

[2] V. Chamola, P. Kotesh, A. Agarwal, N. Gupta, M. Guizani, A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques, Ad. Hoc. Netw. 111 (2021) 102324.

[3] G. Grieco, G. Iacovelli, P. Boccadoro, L.A. Grieco, Internet of drones simulator: design, implementation, and performance evaluation, IEEe Internet. Things. J. 10 (2) (2022) 1476–1498.

[4] P. Boccadoro, D. Striccoli, L.A. Grieco, An extensive survey on the Internet of Drones, AdHoc Netw. 122 (2021) 102600.

[5] S.U. Jan, I.A. Abbasi, F. Algarni, A key agreement scheme for IoD deployment civilian drone, IEEe Access. 9 (2021) 149311–149321.

[6] T. Alladi, V. Chamola, B. Sikdar, K.K.R. Choo, Consumer IoT: security vulnerability case studies and solutions, IEEE Consumer Electron. Mag. 9 (2) (2020) 17–25.

[7] V.O. Nyangaresi, M.A. Morsy, Towards privacy preservation in internet of drones. 2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI), IEEE, 2021, pp. 306–311.

[8] M. Tanveer, H. Alasmary, N. Kumar, A. Nayak, SAAF-IoD: secure and anonymous authentication framework for the Internet of Drones, IEEe Trans. Veh. Technol. (2023) 1–13.

[9] S. Samanth, P. KV, M. Balachandra, Security in internet of drones: a comprehensive review, Cogent. Eng. 9 (1) (2022) 2029080.

[10] M. Yahuza, M.Y.I. Idris, I.B. Ahmedy, A.W.A. Wahab, T. Nandy, N.M. Noor, A. Bala, Internet of drones security and privacy issues: taxonomy and open challenges, IEEE Access. 9 (2021) 57243–57270.

[11] B.A. Alzahrani, A. Barnawi, S.A. Chaudhry, A resource-friendly authentication protocol for UAV-based massive crowd management systems, Secur. Commun. Netw. 2021 (2021) 1–12.

[12] L. Teng, M. Jianfeng, F. Pengbin, M. Yue, M. Xindi, Z. Jiawei, L. Di, Lightweight security authentication mechanism towards UAV networks, in: 2019 International Conference on Networking and Network Applications (NaNA), IEEE, 2019, pp. 379–384.

[13] S. Yu, A.K. Das, Y. Park, P. Lorenz, SLAP-IoD: secure and lightweight authentication protocol using physical unclonable functions for internet of drones in smart city environments, IEEE Trans. Veh. Technol. 71 (10) (2022) 10374–10388.

[14] M. Tanveer, N. Kumar, M.M. Hassan, RAMP-IoD: a robust authenticated key management protocol for the Internet of Drones, IEEE Internet Things J. 9 (2) (2021) 1339–1353.

[15] S. Hussain, M. Farooq, B.A. Alzahrani, A. Albeshri, K. Alsubhi, S.A. Chaudhry, An efficient and reliable user access protocol for Internet of Drones, Ieee Access. 11 (2023) 59688–59700.

[16] V.O. Nyangaresi, Provably secure authentication protocol for traffic exchanges in unmanned aerial vehicles, High-Confidence Comput. 3 (4) (2023) 1–13, 100154.

[17] B.D. Deebak, F. Al-Turjman, A smart lightweight privacy preservation scheme for IoT-based UAV communication systems, Comput. Commun. 162 (2020) 102–117.

[18] H. Khalid, S.J. Hashim, F. Hashim, S.M.S. Ahamed, M.A. Chaudhary, H. H. Altarturi, M. Saadoon, HOOPOE: high performance and efficient anonymous handover authentication protocol for flying out of zone UAVs, IEEe Trans. Veh. Technol. 72 (2023) 10906–10920.

[19] Y. Tian, J. Yuan, H. Song, Efficient privacy-preserving authentication framework for edge-assisted Internet of Drones, J. Inf. Secur. Appl. 48 (2019) 102354.

[20] A.K. Das, M. Wazid, N. Kumar, A.V. Vasilakos, J.J. Rodrigues, Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment, IEEE Internet Things J. 5 (6) (2018) 4900–4913.

[21] H. Khalid, S.J. Hashim, S.M.S. Ahamed, F. Hashim, M.A. Chaudhary, Secure real-time data access using two-factor authentication scheme for the internet of drones, in: 2021 IEEE 19th Student Conference on Research and Development (SCOReD), IEEE, 2021, pp. 168–173.

[22] S.H. Alsamhi, A.V. Shvetsov, S. Kumar, S.V. Shvetsova, M.A. Alhartomi, A. Hawbani, V.O. Nyangaresi, UAV computing-assisted search and rescue mission framework for disaster and harsh environment mitigation, Drones 6 (7) (2022) 154.

[23] S. Hussain, S.A. Chaudhry, Comments on "biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment, IEEE Internet Things J. 6 (6) (2019) 10936–10940.

[24] M. Rodrigues, J. Amaro, F.S. Osório, B.K. RIJC, Authentication methods for UAV communication, in: 2019 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2019, pp. 1210–1215.

[25] B. Bera, A.K. Das, S. Garg, M.J. Piran, M.S. Hossain, Access control protocol for battlefield surveillance in drone-assisted IoT environment, IEEE Internet Things J. 9 (4) (2021) 2708–2721.

[26] H. Alasmary, M. Tanveer, ESCI-AKA: enabling secure communication in an iot-enabled smart home environment using authenticated key agreement framework, Mathematics 11 (16) (2023) 3450.

[27] S. Javed, M.A. Khan, A.M. Abdullah, A. Alsirhani, A. Alomari, F. Noor, I. Ullah, An efficient authentication scheme using blockchain as a certificate authority for the internet of drones, Drones 6 (10) (2022) 264.

[28] V.O. Nyangaresi, Privacy preserving three-factor authentication protocol for secure message forwarding in wireless body area networks, AdHoc Netw. 142 (2023) 1–15, 103117.

[29] T. Wu, X. Guo, Y. Chen, S. Kumari, C. Chen, Amassing the security: an enhanced authentication protocol for drone communications over 5G networks, Drones 6 (1) (2021) 10–29, 10.

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

[31] S.A. Chaudhry, K. Yahya, M. Karuppiah, R. Kharel, A.K. Bashir, Y.B. Zikria, GCACS-IoD: a certificate based generic access control scheme for Internet of drones, Comput. Netw. 191 (2021) 107999.

[32] A.K. Das, B. Bera, M. Wazid, S.S. Jamal, Y. Park, igcacs-iod: an improved certificate-enabled generic access control scheme for internet of drones deployment, IEEE Access 9 (2021) 87024–87048.

[33] M. Nikooghadam, H. Amintoosi, S.H. Islam, M.F. Moghadam, A provably secure and lightweight authentication scheme for Internet of Drones for smart city surveillance, J. Syst. Architect. 115 (2021) 101955.

[34] Z.A. Abduljabbar, V.O. Nyangaresi, H.M. Jasim, J. Ma, M.A. Hussain, Z.A. Hussien, A.J. Aldarwish, Elliptic curve cryptography-based scheme for secure signaling and data exchanges in precision agriculture, Sustainability 15 (13) (2023) 10264.

[35] H. Alasmary, RDAF-IIoT: reliable device-access framework for the industrial Internet of Things, Mathematics 11 (12) (2023) 2710.

[36] G. Cho, J. Cho, S. Hyun, H. Kim, SENTINEL: a secure and efficient authentication framework for unmanned aerial vehicles, Appl. Sci. 10 (9) (2020) 1–19, 3149.

[37] B. Semal, K. Markantonakis, R.N. Akram, A certificateless group authenticated key agreement protocol for secure communication in untrusted UAV networks, in: 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), IEEE, 2018, pp. 1–8.

[38] Y.K. Ever, A secure authentication scheme framework for mobile-sinks used in the internet of drones applications, Comput. Commun. 155 (2020) 143–149.

[39] S.U. Jan, F. Qayum, H.U. Khan, Design and analysis of lightweight authentication protocol for securing IoD, IEEE Access 9 (2021) 69287–69306.

[40] Y. Zhang, D. He, L. Li, B. Chen, A lightweight authentication and key agreement scheme for Internet of Drones, Comput. Commun. 154 (2020) 455–464.

[41] J. Srinivas, A.K. Das, N. Kumar, J.J. Rodrigues, TCALAS: temporal credential-based anonymous lightweight authentication scheme for Internet of drones environment, IEEE Trans. Veh. Technol. 68 (7) (2019) 6903–6916.

[42] S.A. Chaudhry, J. Nebhen, A. Irshad, A.K. Bashir, R. Kharel, K. Yu, Y.B. Zikria, A physical capture resistant authentication scheme for the Internet of Drones, IEEE Commun. Stand. Mag. 5 (4) (2021) 62–67.

[43] V.O. Nyangaresi, Terminal independent security token derivation scheme for ultra-dense IoT networks, Array 15 (2022) 100210.

[44] S. Hussain, S.A. Chaudhry, O.A. Alomari, M.H. Alsharif, M.K. Khan, N. Kumar, Amassing the security: an ECC-based authentication scheme for Internet of drones, IEEE Syst. J. 15 (3) (2021) 4431–4438.

[45] M. Zhang, C. Xu, S. Li, C. Jiang, On the security of an ECC-based authentication scheme for Internet of Drones, IEEE Syst. J. 16 (4) (2022) 6425–6428.

[46] A. Adavoudi-Jolfaei, M. Ashouri-Talouki, S.F. Aghili, Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks, Peer Peer Netw. Appl. 12 (2019) 43–59.

[47] S. Shin, T. Kwon, A privacy-preserving authentication, authorization, and key agreement scheme for wireless sensor networks in 5G-integrated Internet of Things, IEEE Access 8 (2020) 67555–67571.

[48] M.W. Akram, A.K. Bashir, S. Shamshad, M.A. Saleem, A.A. AlZubi, S.A. Chaudhry, Y.B. Zikria, A secure and lightweight drones-access protocol for smart city surveillance, IEEE Trans. Intell. Transport. Syst. 23 (10) (2021) 19634–19643.
[49] Z.A. Hussien, H.A. Abdulmalik, M.A. Hussain, V.O. Nyangaresi, J. Ma, Z. A. Abduljabbar, I.Q. Abduljaleel, Lightweight integrity preserving scheme for secure data exchange in cloud-based IoT systems, Appl. Sci. 13 (2) (2023) 691.
[50] S. Challa, A.K. Das, V. Odelu, N. Kumar, S. Kumari, M.K. Khan, A.V. Vasilakos, An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks, Comput. Electric. Eng. 69 (2018) 534–554.
[51] Z. Ali, A. Ghani, I. Khan, S.A. Chaudhry, S.H. Islam, D. Giri, A robust authentication and access control protocol for securing wireless healthcare sensor networks, J. Inf. Secur. Appl. 52 (2020) 102502.
[52] D. Wang, H. Cheng, P. Wang, X. Huang, G. Jian, Zipf's law in passwords, IEEE Trans. Inf. Forens. Secur. 12 (11) (2017) 2776–2791.
[53] H. Sikarwar, D. Das, A novel mac-based authentication scheme (NoMAS) for internet of vehicles (IoV), IEEE Trans. Intell. Transport. Syst. (2023).

Vincent Omollo Nyangaresi holds a Bsc in Telecommunications & Information Technology, Msc in Information Technology Security & Audit, and a PhD in Information Technology Security & Audit. He is an experienced researcher in areas of computer science and information technology, having published over 100 research articles in peer reviewed journals and conferences covering areas such as communication systems, secure network communications, D2D, smart homes, Internet of drones, smart grids, WSNs, cellular network security, VANETs, information systems acceptance modeling, TCP architecture and design, radio wave propagation, virtualization and cloud computing. He is a renowned reviewer for numerous IEEE, Taylor & Francis, MDPI, Elsevier, and Springer journals. In addition, he has served as a Technical Committee Member (TPC) for the International Conference on IoT as a Service (IoTaaS), Congress on Intelligent Systems (CIS), International Conference on Smart Grid and Energy Engineering (SGEE), and International Conference on Cloud, Big Data and IoT (CBIoT). Moreover, he lecturers in the fields of computer science, information technology, and applied computer science.

Ali Hamzah Najm (Iraqi), received a bachelor's degree from NTC in 2010 and a master's degree in electronic and communications eng. from India in 2013. Ph.D. degree in computer and electrical engineering at Altinbas University, Istanbul, Turkey. He has nine years of teaching experience and worked with Imam Al-Kadhum College (I.K.C). His research interests include WSN, IOT, network protocols, and wireless communications.

Kareem Ali Malalah (Iraqi) received a bachelor's degree from NTC in 2011 and a master's degree in electronic and communications eng. from India in 2013 . He has three years of teaching experience and worked at Alfarahidi University. His research interests include Fibre optical, optical networks, IoT, and wireless communications.

Dr. Istabraq M. Al-Joboury is a lecturer in the Technical College of Engineering at Al-Bayan University. She earned her Ph.D. in Information and Communication Engineering in 2022, an M.Sc. in Networks Engineering and Internet Technologies in 2017, and a B.Sc. in Networks Engineering in 2015 from the same College of Information Engineering at Al-Nahrain University, Baghdad, Iraq. Dr. Istabraq's research focuses on the Internet of Things, Blockchain, and Distributed Ledger Technologies. She has published several papers related to the same field of interest.

Ali Hashim Abbas (A.H.Abbas) received the B.S. degree in communication engineering from Al-Furat Al-Awsat Technical University/ Engineering Technical College of Al-Najaf, in 2010 and the M.S. degree in digital system and computer electronics (DSCE) from Jawaharlal Nehru Technological University Hyderabad (JNTU), Hyderabad, India, in 2014, and Ph.D. degrees in Communication engineering, Clustering of Vehicular Ad-Hoc Networks (VANETs) from UTHM University Tun Hussein Onn Malaysia, Johor, Malaysia, in 2019. He is work the Head of Department of Scientific Affairs and Promotions at the Department of Computer Technical engineering, College of Information Technology, Imam Ja'afar Al-Sadiq University, Al-Muthanna 66,002, Iraq in 2021. Where he is currently working Dean of College of Information Technology, Imam Ja'afar Al-Sadiq University, Baghdad, Iraq. His research interests are cluster stability for intervehicle communication and distributed algorithms, for vehicular ad hoc networks. In addition, He is a reviewer for leading communication, and computer networks engineering journals such as vehicular ad hoc networks, vehicular communications, wireless communication, IEEE Wireless Communications Letters, Journal of Sensors, and IEEE Access Journal. He can be contacted at email: alsalamy1987@mail.com.

Hussein Muhi Hariz received the B.S. degree in communication engineering from Al-Furat Al-Awsat Technical University/ Engineering Technical College of Al-Najaf, in 2005 and the M. S. degree in System and signal processing from Jawaharlal Nehru Technological University Hyderabad (JNTU), Hyderabad, India, in 2014, and still student Ph.D. in Tarbiat Modares University (TMU), Tehran, Iran. He is work as a teacher at the Department of Computer Technical engineering, Mazaya University College, 64,001, Iraq in 2014. His research interest's wireless communication, signal processing. He can be contacted at email: huhraiz22@mpu.edu.iq.