SETCAP

Service-Based Energy-Efficient Temporal Credential Authentication Protocol for Internet of Drones

El-Zawawy, Mohamed A.; Brighente, Alessandro ; Conti, Mauro

DOI
10.1016/j.comnet.2022.108804

Publication date
2022

Document Version
Final published version

Published in
Computer Networks

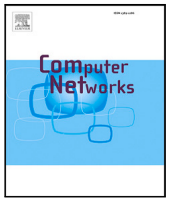
Citation (APA)
El-Zawawy, M. A., Brighente, A., & Conti, M. (2022). SETCAP: Service-Based Energy-Efficient Temporal Credential Authentication Protocol for Internet of Drones. *Computer Networks*, *206*, 1-15. Article 108804. https://doi.org/10.1016/j.comnet.2022.108804

Computer Networks 206 (2022) 108804

Contents lists available at ScienceDirect

Computer Networks

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



SETCAP: Service-Based Energy-Efficient Temporal Credential Authentication Protocol for Internet of Drones

Mohamed A. El-Zawawy [a,*], Alessandro Brighente [b], Mauro Conti [b,c]

[a] Department of Mathematics, Faculty of Science, Cairo University, Giza 12613, Egypt
[b] Department of Mathematics and HIT Research Center, University of Padova, 35131, Padua, Italy
[c] Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Netherland

ARTICLE INFO

Keywords:
Internet of Drones
Security
Authentication protocol

ABSTRACT

Internet of Drones (IoD) is a framework to set up drones networks that may serve multiple purposes, e.g., data collection. New IoD applications (such as drone assisted internet of vehicles) envision the simultaneous collection of multiple data types. Although authentication may prevent unauthorized users to access the collected data, existing authentication solutions do not distinguish between the different types of data collected by drones. Therefore, authenticated users may receive sensitive data regarding another user incurring hence in a privacy leakage.

In this paper, we propose SETCAP, a novel Service-Based Energy-Efficient Temporal Credential Authentication Protocol for IoD. SETCAP exploits the distinction between data types to prevent information leakage. We formally test SETCAP against the Real-Or-Random (ROR) model and implemented SETCAP in Automated Validation of Internet Security Protocols and Applications (AVISPA) simulation tool. Moreover, we validated SETCAP via non-mathematical security analysis to show its security against many attacks. We assessed the superiority of SETCAP in terms of functionality and security characteristics as well as computation, communication, and energy costs. The communication cost of creating a session in SETCAP is approximately 20% smaller than that of creating a session in the closest state-of-the-art protocol. Furthermore, the framework that we propose requires the creation of a number of sessions that are additive in terms of the number of drones and users, whereas the existing solutions are multiplicative. SETCAP is therefore a secure and scalable solution for resource-constrained devices such as drones.

1. Introduction

IoD is a framework that aims at establishing regulated access to the airspace of drones [1], i.e., unmanned aircraft systems. Drones can be remotely controlled by a pilot, and have numerous applications such as autonomous driving and aerial photography [2]. Drones can hence be organized in networks via the IoD paradigm to collect data generated on the ground by devices such as internet of things nodes or wireless sensor networks [3,4]. The widespread adoption of drones connected in the IoD is supported by the increased wireless connectivity and the technological advancements that led to compact processors and sensor devices. On the other side, drones can cause serious safety and privacy hazards. In fact, thanks to components such as microphones and cameras, drones can be used to steal personal information by hovering inside our private airspace [5].

As drones may collect sensitive information, it is fundamental that only authorized users get access to it. To guarantee this, a cornerstone of IoD security is the authentication of users and drones to the GSS. The importance of authentication protocols is confirmed by the recent application of IoD in autonomous driving [6]. Without a proper authentication protocol, drone assisted internet of vehicles can easily incur in illegal actions such as unauthorized tracking, surveillance, and terrorist attacks. IoD authentication protocols should also consider the resource-constrained nature of drones. Therefore, they shall be lightweight both in terms of computational complexity and memory demand. This guarantees that the drone does not consume a high amount of energy for cryptographic operations, hence being able to fly for a longer time.

Although there are many proposed protocols for users and drones authentication in IoD [7,8], there is still a security gap that can lead to data leakage. In fact, currently available solutions do not distinguish between the different types of data collected by drones. Such data may include both sensitive and non-sensitive data from a certain geographical zone and may be delivered as a single packet to a user [8]. In

* Corresponding author.
E-mail addresses: maelzawawy@cu.edu.eg (M.A. El-Zawawy), alessandro.brighente@unipd.it (A. Brighente), conti@math.unipd.it (M. Conti).

https://doi.org/10.1016/j.comnet.2022.108804
Received 30 August 2021; Received in revised form 31 December 2021; Accepted 20 January 2022
Available online 7 February 2022

this case, the user may have access to sensitive information meant for another user. Existing protocols [8] assume that users know and specify the ID of drones from which they want to retrieve data. This is neither a secure nor a practical assumption. Also, almost all existing protocols do not distinguish the type of data read by the drones and that is delivered to users; they treat all read data as one type. Such protocols are not suitable for autonomous driving applications where some types of data read by drones are (temporarily) available to users and other types are (temporarily) not, for security purposes. Furthermore, most existing protocols allow direct access between drones and users [8]. It is more secure to let the interaction between drones and users go through the GSS. This provides better security for drones and gives the GSS a chance to treat data (e.g. applying machine learning algorithms) and to hide sensitive data from users, if necessary. Furthermore, thanks to the GSS, we can offload part of the computations to save energy at the drones' side.

In this paper, we propose SETCAP (Service-Based Energy-Efficient Temporal Credential Authentication Protocol) for IoD. The proposed protocol is described as "Service-Based" because the authentication process considers the services (in form of necessary data types) that drones provide to ground server stations and that the latter provide to users. SETCAP overcomes the security and privacy drawbacks of existing protocols generated from the aforementioned gaps. The main idea behind SETCAP is that it is fundamental to distinguish between the types of data collected by drones. Thanks to this distinction, we show that it is possible to prevent data leakage and increase the security of IoD authentication protocols. We formally test SETCAP against the well-trusted and known ROR model [9]. We then implement SETCAP in the AVISPA simulation tool. Results confirm the security of the session keys of our protocol SETCAP against replay and man-in-the-middle attacks. Moreover, we validate SETCAP via security analysis to show its security against many potential adversarial attacks. Lastly, we perform a detailed comparative study of SETCAP against recent state-of-the-art techniques, the closest being TCALAS [8]. We assess the superiority of SETCAP in terms of functionality and security characteristics as well as computation, communication, and energy costs. The communication cost of creating a session in SETCAP is approximately 20% smaller than that of creating a session in TCALAS. SETCAP also provides better scalability compared to other solutions [8]. In fact, SETCAP requires the creation of a number of sessions additive in terms of the number of drones and users thanks to the use of the GSS. Other solutions (e.g., TCALAS [8]) are instead multiplicative in these terms. Although the computation cost of creating a session in SETCAP is approximately just 4% smaller than that of creating a session in TCALAS, the high number of sessions required by TCALAS highlights the higher scalability of SETCAP. Therefore, besides preserving the functionality and security characteristics of existing protocols, SETCAP exhibits higher scalability. This paper focuses on drone models suitable for applications of smart cities. Therefore, our following discussions of drone capabilities and limitations are based on models ranging from the PHANTOM 4 RTK[1] to the EHang AAV[2] models.

Contributions. We summarize our contributions as follows:

1. We propose SETCAP, the Service-Based Energy-Efficient Temporal Credential Authentication Protocol for IoD. SETCAP is the first protocol to perform the authentication process based on the types of drone-collected data. This feature provides higher security against data leakage compared to existing protocols.
2. We show the security of SETCAP by using both a formal security model ROR and the simulation tool AVISPA. Results confirm the security of the session keys of our protocol SETCAP against replay and man-in-the-middle attacks. Moreover, security analysis shows that SETCAP is resilient to many potential adversarial attacks, such as capturing attacks.
3. We show the superiority of SETCAP in terms of computation, communication, and energy costs, as well as functionality and security characteristics. Results show that SETCAP provides higher scalability compared to state-of-the-art solutions. The communication cost of creating a session in SETCAP is approximately 20% smaller than that of creating a session in TCALAS [8]. For connecting n users to m drones, SETCAP creates $m + n$ sessions whereas TCALAS creates $n \times m$ sessions. This proves the suitability of our protocol in scenarios with resource-constrained devices, such as the IoD.

Organization. The rest of the paper is organized as follows. The problem statement is presented in Section 2. We define the problem we address in this paper in Section 2.1. The system and threat models treated in this paper are detailed in Sections 2.2 and 2.3, respectively. Details of SETCAP are introduced in Section 3. The security analysis of SETCAP are shown in Section 4. Section 5 presents detailed results of evaluating SETCAP. Section 6 reviews most relevant state-of-the-art techniques. Section 7 concludes the paper and presents directions of future work.

2. System and threat model

In this section, we present the problem we address with our solution and discuss the system and threat models we assume in our paper. In particular, we first describe the problem in Section 2.1. Then, we provide an overview of the considered scenario and the system model in Section 2.2. Then, we discuss the threat model in Section 2.3, discussing both the reasons behind an attack and the considered attacker model.

2.1. Problem definition

We consider a scenario where users are interested in collecting data generated by sensing and communication-enabled ground level devices (e.g., Internet of Things (IoT) devices, and autonomous cars). In order to access this data, the user subscribes to a data collection service managed by a GSS. In order to collect the user requested data, the GSS exploits multiple swarms of drones, where each swarm is assigned a specific geographical area, and each drone contributes to the task assigned to the swarm it belongs to. Thanks to the use of multiple drones, the GSS is able to collect data from multiple geographical areas that cannot be covered by a single drone. The user may want to collect such data for multiple purposes. For instance, in case of a factory automation scenario, the user may want to remotely assess the correct functioning of the facility thanks to a network of IoT devices. A further example is related to road traffic management, where the user is represented by an authority that needs to retrieve information regarding traffic conditions or roads problems. The connection to the collection service is managed by the GSS, that acts as an intermediary between the users and the sensors. To ease the data collection and to guarantee high flexibility, the GSS uses drones that retrieve data from the ground level devices. We assume that the GSS is able to optimize the deployment of drones, such that each swarm collects data related to multiple users for each swarm deployment. Therefore, drones collect and aggregate data coming from different types of entities in each packet [8].

Thanks to this solution, a user may be able to simultaneously access data that originates from different devices and the GSS may optimize drones deployment for reduced energy consumption. However, this also represents a privacy threat. In fact, due to the fact that a single packet might include multiple types of data or data generated by multiple devices, we must ensure that a user receives only the data for which she/he previously obtained authorization through the GSS. The risk is hence that malicious users subscribed to the data collection service, therefore authorized by the GSS to receive packets, may collect sensitive information belonging to other subscribed users. This represents

[1] https://www.dji.com/phantom-4-rtk.
[2] https://www.ehang.com/ehangaav/.

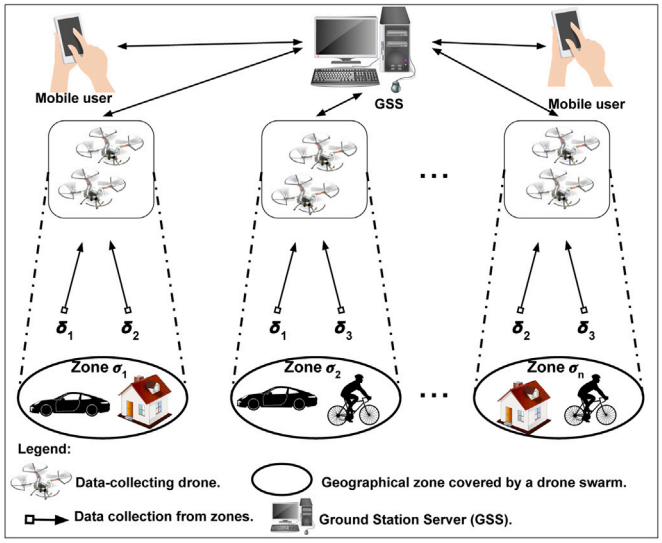


Fig. 1. Visualizing the Authentication Problem Treated by SETCAP.

a threat, as malicious users may be able to profile other users and to sell this information to interested parties, or may be able to track the victim or to gather information useful for further attacks. The attack considered in our paper is particularly challenging, as the attacker is a legitimate user of the network. Therefore, although authentication via subscription is effective against outsider attacks, it not sufficient to prevent an attacker inside the network from collecting sensitive information.

In this paper, we propose a solution to prevent malicious users to get access to other users' data. We consider the threat imposed both by external attackers and by internal attackers, i.e., attackers that belong to the data collection network and that have legitimate access to the network. In our implementation, we authorize users' access to data depending on the data type. This will prevent jeopardizing the users' privacy to avoid the aforementioned attacks. To enhance the efficiency of the protocol, we ensure that such access methodology does not depend on subscription to the data collected by a specific drone (e.g., by specifying the drone ID).

2.2. System model

Fig. 1 shows the scenario we consider in this paper. We partition a geographical area into n different and disjoint zones. Each zone is covered by a swarm of drones reading zone-specific data. Drones can read different types of data, and we denote as δ_i the ith data type in the figure. We assume that the only trusted and secure participant is the GSS [8,10]. The GSS is controlled from a control room, and controls the location and trajectories of drones. We assume that the GSS runs an optimization algorithm to suitably deploy drones in the different geographical areas and that solves possible conflicts in their trajectories. The design of the optimal control for drones deployment goes beyond the scope of our work, and we hence refer the interested reader to other works on the subject [11–13]. The GSS is a powerful device able to efficiently manage a large number of operations. Therefore, we add it in the middle of the communication between drones and users to reduce the computational burden at either drones' or users' side. We assume that other participants are vulnerable to security threats and privacy breaches. The users have no knowledge about drones IDs or locations. In each flying zone, drones can interact with each other and with the GSS. We assume a service that delivers data upon subscription. Therefore, users register to receive certain data types from certain zones. Then the three parties (GSS, drones, and users) establish secret session keys for the secure communication between: (i) drones and GSS from one side, and (ii) the users and GSS from the other side. Users communicate to the GSS via their mobile devices and cannot directly communicate with drones. Therefore, the GSS can hide any sensitive data and can treat or manipulate data on its way to the users. Table 1 shows the notation we use throughout the paper.

The interaction between drones of the same flying zone can present some challenges, especially when the drones do not belong to the same person/organization. This is due to factors such as authentication schemes, trust in information, and communication technologies. Because our model already has a GSS entity, we do not need to assume air traffic control to avoid issues such as collisions and to manage zone access. However, the GSS needs to apply zone/swarm controllers [15–17], and this can be integrated with our proposal in many forms. For instance, a Software-Defined Drone Network (SDDN) such as the collision avoidance scheme introduced in [15] can be applied by the GSS to achieve on-road traffic monitoring and management. Then, the result of this process can be used to decide when to create new drone sessions. This issue is related to the Live Distributed Objects (LDO) problem [18].

We make two fundamental assumptions in our problem. The first is that users have limited knowledge about drones, i.e., they are not aware of drones' IDs and locations. This is justified by two points: (1) typical users do not care about drones but rather about the types of data they need and the geographical zones of the data, and (2) it is more secure to hide drones details to the users. The other assumption is that users cannot directly access drones data. Rather, we should establish two sessions: one between a drone and GSS, and one between GSS and user. This assumption is justified by the following points: (1) This assumption enables the user to get different types of data from m drones in one session rather than in m sessions; (2) This assumption allows the GSS to hide sensitive data if necessary; (3) This assumption enables the GSS to treat data (e.g., by using machine learning techniques) before providing it to users.

The GSS manages all interactions between drones and users go through it. The GSS has bolls of data collected by drones: a boll per each pair of data type and zone. There is a need for an authentication scheme that registers and authorizes users and drones to directly access certain bolls. Hence, conventional authentication and authorization mechanisms do not work in our motivating IoD scenario. However, adapting existing protocols to our case would lead to a scenario in which the GSS has to decide for each data transaction (unfortunately, not even for each session) the types of data that users and/or drones are allowed to access. This traditional solution has two issues. First, it creates the possibility of users obtaining unallowed data types and it does not consider the fact that drones are dynamic objects that need quick responses from the GSS. Second, the limited battery of drones would not allow it to afford these accumulated (with each transaction) delays. The same issues apply to user mobiles. The second issue becomes even more evident if we consider the huge number of data transactions that the GSS has to manage simultaneously.

2.3. Threat model

In this section, we first discuss the privacy and security threats associated with the absence of a data-type policy in previous solutions (e.g., [8]). Then, we consider the threat model associated with the presence of an attacker running both active and passive attacks.

As previously discussed, existing solutions do not distinguish between the types of data collected by drones. Therefore, data is collected in batches and, although authentication may be used to prevent unauthorized access to data, still users may get access to sensitive data meant to another user. As a specific example, consider the scenario depicted in Fig. 2, where road traffic is managed via autonomous driving. We assume that cars share data about their driving condition (e.g., speed, location acceleration) to allow for specific services based

Table 1
Notation.

Notation	Meaning
$\Theta = \{\theta_i \mid 1 \le i \le n_d\}$	The set of flying drones.
$\mathcal{I}_\Theta = \{\mathcal{I}_{\theta_i} \mid 1 \le i \le n_d\}$	The set of IDs of Θ.
$\Sigma = \{\sigma_i \mid 1 \le i \le n_z\}$	The set of geographical zones.
$\mathcal{I}_\Sigma = \{\mathcal{I}_{\sigma_i} \mid 1 \le i \le n_z\}$	The set of IDs of Σ .
$\Delta = \{\delta_j \mid 1 \le j \le n_t\}$	The set of data types captured by drones .
$\mathcal{I}_\Delta = \{\mathcal{I}_{\delta_j} \mid 1 \le j \le n_t\}$	The set of IDs of Δ.
$UM = \{(u_q, m_q) \mid 1 \le q \le n_u\}$	The set of pairs of users and their mobile devices.
$\mathcal{I}_{UM} = \{(\mathcal{I}_{u_q}, \mathcal{I}_{m_q}) \mid 1 \le q \le n_u\}$	The set of pairs of IDs of users and the IDs of their mobile devices.
lSK_θ	The long term secret key of drone θ.
$GSS, \mathcal{I}_G$	The ground station server GSS and its ID, respectively.
lSK_G	The long term secret key of GSS.
SK_θ	Shared secret key between GSS and θ.
$h(\cdot)$	A one-way cryptographic collision-resistant hash function.
SeK_u	Session key between user u and GSS.
SeK_θ	Session key between GSS and θ.
$\mathtt{Gen}(.), \mathtt{Rep}(.)$	Fuzzy extractor functions.
$\mathcal{A}$	polynomial time Adversary.
(Z_1, Z_2)	Zipf's parameters according to [14].

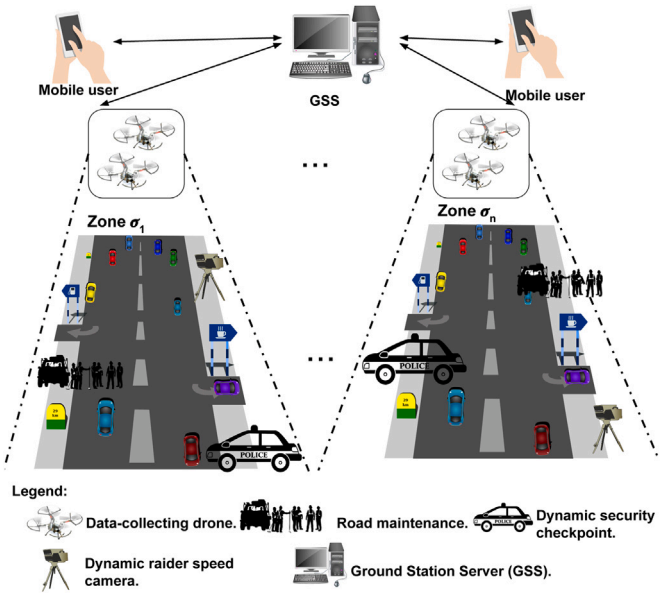


Fig. 2. Visualizing the Threat Model Treated by SETCAP.

on driver identification [19]. Furthermore, cars share their navigation information for orchestration purposes. All users are allowed to receive data related to road maintenance to compute their path preferences. However, it would be a security breach to deliver data regarding driver behavior to all users. In fact, this type of data shall be accessed only by legitimate receivers, e.g., the service granting facility or the user itself for consumption management. Therefore, the lack of distinction between data types incurs in privacy leakage and shall be accounted for while designing suitable solutions. This represents the fundamental motivation behind SETCAP.

To assess the security of SETCAP against attacks, we design our threat model and assumptions to be in line with that of state-of-the-art techniques [8], i.e., we start from the well-known Dolev-Yao (DY) threat model [20]. The DY model assumes public communication between system entities. Hence the adversary can delete, modify, update, and intercept the communicated messages. The adversary can also insert malicious messages. We also assume that flying drones can be physically captured by an adversary. Hence data stored in drones can be used to launch security attacks on the system. These attacks are most likely to target the availability of communication between GSS and other drones. Moreover, a user's mobile may be lost and hence an adversary can extract information that can be used to launch power analysis attacks [21]. In this case, if the adversary manages to extract the secret credentials from the user's mobile (such as user password and bio-metrics), the system security may be compromised. In particular, these credentials facilitate many attacks such as man-in-the-middle, impersonation, and privileged-insider.

In this paper, to further strengthen the adversary capabilities considered in the DY model, we consider the Canetti-Krawczyk (CK) model [22]. In addition to the assumption of the DY model, according to the CK model, the adversary can compromise secret credentials (such as secret keys) and session states. Our threat model assumes that the GSS is trusted and hence the adversary cannot compromise it. This is a fair assumption as the GSS can be secured in many ways, e.g., by placing it under a locking system [8,23]. On the other hand, it might be easier for an attacker to cause a denial of service to the GSS compared to the effort needed to cause it to multiple drones. However, this goes beyond the purpose of this paper, and we will address this challenge in future works.

3. SETCAP: New proposed authentication protocol

In this section we describe SETCAP, our protocol for IoD authentication. We divide SETCAP into four steps: (i) pre-deployment, (ii) user registration, (iii) user log in and authentication and (iv) drone authentication. We assume that users employ mobile devices with passwords and biometrics to access the GSS. In the following subsection, we provide the details of each phase of SETCAP.

3.1. Pre-deployment

The *GSS* registers each drone for deployment in a specific flying zone. It further provides to each drone the set of data types that the drone needs to collect. These are specified via the binary vector:

$$V_t^{(\sigma_j,\theta_i)} = \begin{bmatrix} x_1 & x_2 & \dots & x_{n_t} \end{bmatrix},$$

where n_t is the number of data types captured by drones and σ_j is the jth flying zone. For example, suppose that $n_t = 3$, and

$$V_t^{(\sigma_j,\theta_i)} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}.$$

This means that the drone θ_i is registered to provide data type δ_1 and type δ_3 from the flying zone σ_j. In the following, we ease the notation by not considering the indexes of the specific drone or geographical area.

The registration proceeds as follows.

1. For a drone θ deployed in the flying zone σ to capture data of type δ_k, we calculate a data-oriented shared secret key $DSK_\theta^{\delta_k}$ between GSS and θ to deliver data type δ_k as follows:

   $DSK_\theta^{\delta_k} = h(\mathcal{I}_\theta \parallel \mathcal{I}_\sigma \parallel \mathcal{I}_{\delta_k} \parallel lSK_G \parallel lSK_\theta)$.

   We define the vector of keys for each data type

   $SK_\theta = [SK_\theta^{\delta_1} \dots SK_\theta^{\delta_{n_t}}]$,

   where

   $$SK_\theta^{\delta_k} = \begin{cases} DSK_\theta^{\delta_k}, & \text{if } V_t^{(\sigma,\theta)}(k) = 1 \\ 0, & \text{otherwise;} \end{cases}$$

   and where $V_t^{(\sigma,\theta)}(k)$ denotes the kth element of $V_t^{(\sigma,\theta)}$. We also define the matrix of universal shared secret keys

   $$SK_\Theta = \begin{bmatrix} SK_1 \\ SK_2 \\ \dots \\ SK_{n_d} \end{bmatrix};$$

   where row θ denotes the set of keys assigned to drone θ.
2. Every drone θ deployed in the flying zone σ is pre-loaded with credentials:

   $\mathcal{I}_G, \mathcal{I}_\sigma, SK_\theta, V_t^{(\sigma,\theta)}, and\, h(.)$.
3. The GSS is pre-loaded by the credentials:

   $\mathcal{I}_G, \mathcal{I}_\Sigma, SK_\Theta, lSK_G, V_t^{(\sigma,\Theta)}, and\, h(.)$,

   where

   $$V_t^{(\sigma,\Theta)} = \begin{bmatrix} V_t^{(\sigma,1)} \\ V_t^{(\sigma,2)} \\ \dots \\ V_t^{(\sigma,n_d)} \end{bmatrix}$$

   is the matrix whose θ^{th} row reports the data type assigned to drone θ.

According to our assumption that drones have unique IDs, the GSS refuses to register a new drone whose ID is already assigned to another registered drone. Practically, given the typical hexadecimal representation of drone IDs, it is unlikely to have this issue. However, for special applications whose drones are likely to have many duplicated IDs, we assume that each drone is equipped with a Trusted Platform Module (TPM). The TPM can be utilized in several ways such as appending the TPM's unique RSA key (which is burned to it) to the drone ID to make the IDs unique. Since, according to our system model, drones communicate with users via the GSS, revoking drone access is always possible from the GSS side. Revoking can be applied as a response to security threads. It is worth noting that the shared keys created at time of registration of a drone θ are to be used in creating a session key for each communication between the drone θ and the GSS. This is detailed in Algorithms 4 and 5 that create the session key SeK_θ and that are presented in Section 3.4. The creation of the session key SeK_θ is necessary to eliminate the possibility of a brute force attack.

3.2. User registration

A user u (with mobile device m) needs to register with the GSS for obtaining specific data types from specific flying zones. To specify these details the user fills a binary $n_z \times n_t$ matrix $M_{z,t}$, where n_z and n_t are the number of flying zones and number of data types captured by drones, respectively. The matrix $M_{z,t}$ has entries

$$M_{z,t}(i,j) = \begin{cases} 1, \text{ if registered for data type } j \text{ of flying zone } i; \\ 0, \ otherwise. \end{cases}$$

Algorithm 1 User Registration

Input: A user u (with mobile device m) needs to register with the GSS for obtaining data of certain types from specific flying zones fixed in $M_{z,t}$.
Steps:

1: **procedure** $Mobile_1()$
2: Fix an identifier, $\mathcal{I}_m$, and a password pw, and a random number b;
3: Fill the matrix $M_{z,t}$.
4: $hI \leftarrow h(\mathcal{I}_m \parallel b)$;
5: $hP \leftarrow h(pw \parallel b)$;
6: Via a secure channel, send registration request with $(M_{z,t}, hI, h(\cdot))$ to GSS;
7: **procedure** $GSS_1(M_{z,t}, hI, h(.))$
8: $SK \leftarrow h(hI \parallel lSK_G)$;
9: **for each** $i \in \{1, \dots, n_z\}$ **do**
10: **for each** $j \in \{1, \dots, n_t\}$ **do**
11: **if** $M_{z,t}(i,j) \neq 0$ **then**
12: $TC(i,j) \leftarrow h(\mathcal{I}_{\sigma_i} \parallel \mathcal{I}_{\delta_j} \parallel \mathcal{I}_G \parallel SK)$;
13: $B(i,j) \leftarrow (\mathcal{I}_{\sigma_i} \parallel \mathcal{I}_{\delta_j}) \oplus h(hI \parallel SK)$;
14: **else**
15: $TC(i,j) \leftarrow 0$;
16: $B(i,j) \leftarrow 0$;
17: $MC \leftarrow (SK, B, TC, h(\cdot), \mathcal{I}_G, M_{z,t})$;
18: Via a secure channel, send MC to mobile device;
19: **procedure** $Mobile_2(SK, B, TC, h(.), \mathcal{I}_G, M_{z,t})$
20: Read Bio from user;
21: $(\tau_1, \tau_2) \leftarrow Gen(Bio)$;
22: $L_u \leftarrow b \oplus h(\tau_1 \parallel \mathcal{I}_m \parallel pw)$;
23: $SK' \leftarrow SK \oplus h(b \parallel hI \parallel hP \parallel \tau_1)$;
24: **for each** $i \in \{1, \dots, n_z\}$ **do**
25: **for each** $j \in \{1, \dots, n_t\}$ **do**
26: **if** $M_{z,t}(i,j) \neq 0$ **then**
27: $TC'(i,j) \leftarrow h(SK \parallel TC(i,j) \parallel b \parallel \tau_1)$;
28: **else**
29: $TC'(i,j) \leftarrow 0$;
30: $MC \leftarrow (SK', B, TC', h(.), L, h(.), \mathcal{I}_G, M_{z,t}, Gen(.), Rep(.), \tau_2)$;

The registration is presented in Algorithm 1, which includes the following operations: $Mobile_1() \xrightarrow{(M_{z,t}, hI, h(\cdot))} GSS_1() \xrightarrow{MC=(SK,B,TC,h(\cdot),\mathcal{I}_G,M_{z,t})} Mobile_2()$;

The user mobile executes the procedure $Mobile_1()$ which has the following details. Step 2 generates a random number and allows the user to fix an ID and a password. In Step 3, the user fills the matrix $M_{z,t}$ to select data types and zones. Then, the procedure uses a cryptographic function h in Steps 4 and 5 to build the registration request. This is then sent out to GSS via a secure channel in Step 6.

Upon receiving the registration request, the GSS executes the procedure GSS_1, which includes the following actions. In Step 8, the GSS builds a secret key between the user and GSS using the long-term secret key of GSS. Then in Steps 9 to 16, the GSS builds matrices TC and B of the temporal credentials. These matrices have non-zero entries for each data type (in a zone) that the user requested via matrix $M_{z,t}$. All the necessary credentials and data are grouped in Step 17 and sent in Step 18 via a secure channel back to the user mobile.

Upon receiving the response from the GSS, the user mobile executes procedure $Mobile_2()$. Step 20 reads the user biometric (Bio) via the mobile sensor. Then, it uses fuzzy extractor functions [24] to extract the secret biometric key τ_1, and public reproduction parameter τ_2, in Step 21. General credentials L_u and SK' independent of data types and zones, are calculated in Steps 22 and 23. The credential matrix, TC', is calculated in Steps 24–29. TC' is dependent on the data types and zones that the user requires. The last step stores all the received and calculated credentials to the mobile to complete the registration process.

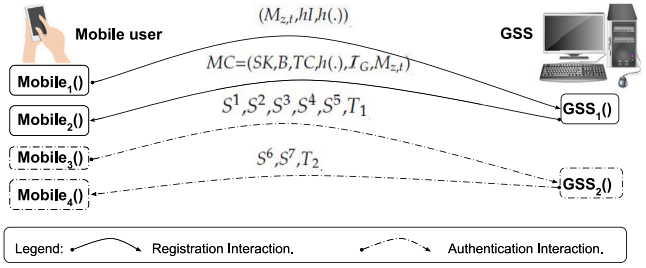


Fig. 3. Visualizing all the interactions between users and GSS in SETCAP.

3.3. User login and authentication

The user login and authentication are presented in Algorithms 2 and 3. They include the following set of operations: $Mobile_3()$ $\xrightarrow{S^1,S^2,S^3,S^4,S^5,T_1}$ $GSS_2()$ $\xrightarrow{S^6,S^7,T_2}$ $Mobile_4()$; Fig. 3 shows all the interactions between users and GSS.

During the login phase, the user logs in to receive specific data types from specific zones and specifies them via $M'_{z,t}$. During this phase, the GSS verifies the login credentials of the user. The GSS also verifies that the required data (of specified zones) are among the ones that the user has already registered for. This is done over a public channel. Upon receiving the login request from the user, the GSS challenges the legitimacy of the user. This is done using the credential matrices built in the registration phase for different data types and zones. One advantage of SETCAP in this phase over related work is that we do not assume any knowledge about the IDs of flying drones at the user's side.

Algorithm 2 User Login and Authentication (1)

Input: The id, $\mathcal{I}_m$, the password, pw, and the bio, Bio', of a user u (with mobile device m) needs to login to get data from GSS according to the matrix $M'_{z,t}$.

Steps:

1: **procedure** $Mobile_3()$
2: **if** $M'_{z,t} \leq M_{z,t}$ **then**
3: $\tau'_1 \leftarrow Rep(Bio', \tau_2)$;
4: $b \leftarrow L_u \oplus h(\tau'_1 \| \mathcal{I}_m \| pw)$;
5: $hI \leftarrow h(\mathcal{I}_m \| b)$;
6: $hP \leftarrow h(pw \| b)$;
7: $SK \leftarrow SK' \oplus h(b \| hI \| hP \| \tau'_1)$;
8: Choose randomly $i \in \{1, \ldots, n_z\}$ and $j \in \{1, \ldots, n_t\}$ such that $M'_{z,t}(i,j) \neq 0$;
9: $\mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j} \leftarrow B(i,j) \oplus h(hI \| SK)$;
10: $TC(i,j) \leftarrow h(\mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j} \| \mathcal{I}_G \| SK)$;
11: $TC''(i,j) \leftarrow h(SK \| TC(i,j) \| b \| \tau'_1)$;
12: **if** $TC''(i,j) == TC'(i,j)$ **then**
13: $T_1 \leftarrow Time\_Stamp()$;
14: $S^1 \leftarrow hI \oplus h(T_1 \| \mathcal{I}_G)$;
15: $S^2 \leftarrow \mathcal{I}_{\sigma_i} \oplus h(SK \| \mathcal{I}_G)$;
16: $S^3 \leftarrow \mathcal{I}_{\delta_j} \oplus h(SK \| \mathcal{I}_G \| \mathcal{I}_{\sigma_i})$;
17: $S^4 \leftarrow i \| j \oplus h(\mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j} \| TC(i,j) \| T_1)$;
18: $S^5 \leftarrow h(i \| j \| SK \| TC(i,j) \| \mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j})$;
19: Transmit $M'_{z,t}, S^1, S^2, S^3, S^4, S^5$, and T_1 to the GSS_2 over a public channel;
20: **else**
21: Reject Login;
22: **else**
23: Reject Login;

The procedure $Mobile_3()$ starts in Step 2 by comparing the two matrices $M'_{z,t}$ and $M_{z,t}$ to verify that the user has already registered for the data that she is asking to get. If the verification is unsuccessful, the procedure and the phase are terminated. In Steps, 3–7 the procedure calculates the credential SK. Then, in Step 8, a data type is chosen randomly from the ones that the user is logging in to obtain. In Step 9 the IDs of the data type and its zone are extracted from the credential B. In Steps 10 and 11, the algorithm calculates the credential TC'' of the randomly chosen data type using the data extracted from the previous steps. This credential is then compared in Step 12 against the stored one. If they are not identical, then the procedure and the phase are terminated. Otherwise, the procedure gets the current timestamp in Step 13 and creates five messages S^1, S^2, S^3, S^4, S^5 in Steps 14–18 to include all the relevant data. Finally, the matrix $M'_{z,t}$, the time stamp, and messages are transmitted to the GSS over a public channel, in Step 16.

Algorithm 3 User Login and Authentication (2)

1: **procedure** $GSS_2(M'_{z,t}, S^1, S^2, S^3, S^4, S^5, T_1)$
2: **if** $M'_{z,t}$ is confirmed. **then**
3: **if** $|T_1^* - T_1| \leq \Delta T$ **then**
4: $hI^* \leftarrow S^1 \oplus h(T_1 \| \mathcal{I}_G)$;
5: $SK^* \leftarrow h(hI^* \| lSK_G)$
6: $\mathcal{I}_{\sigma_i} \leftarrow S^2 \oplus h(SK^* \| \mathcal{I}_G)$;
7: $\mathcal{I}_{\delta_j} \leftarrow S^3 \oplus h(SK^* \| \mathcal{I}_G \| \mathcal{I}_{\sigma_i})$;
8: $TC(i,j) \leftarrow h(\mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j} \| \mathcal{I}_G \| SK^*)$;
9: $i \| j \leftarrow S^4 \oplus h(\mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j} \| TC(i,j) \| T_1)$;
10: **if** $S^5 = h(i \| j \| SK^* \| TC(i,j) \| \mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j})$ **then**
11: $T_2 \leftarrow Time\_Stamp()$;
12: $R \leftarrow h(j \| i)$
13: $S^6 \leftarrow R \oplus h(hI^* \| T_2 \| \mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j} \| \mathcal{I}_G)$;
14: $SeK_u \leftarrow h(R \| hI^* \| T_2 \| \mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j} \| \mathcal{I}_G)$;
15: $S^7 \leftarrow h(R \| SeK_u \| T_2 \| \mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j} \| \mathcal{I}_G)$;
16: Transmit S^6, S^7 and T_2 to the $Mobile_4$ over a public channel;
17: **else**
18: Reject
19: **else**
20: Reject login and report reply attack;
21: **else**
22: Reject login and report the hIden types of data and regions;
23: **procedure** $Mobile_4(S^6, S^7, T_2)$
24: **if** $|T_2^* - T_2| \leq \Delta T$ **then**
25: $R \leftarrow S^6 \oplus h(hI \| T_2 \| \mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j} \| \mathcal{I}_G)$;
26: $SeK_u \leftarrow h(R \| hI^* \| T_2 \| \mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j} \| \mathcal{I}_G)$;
27: **if** $S^7 = h(R \| SeK_u \| T_2 \| \mathcal{I}_{\sigma_i} \| \mathcal{I}_{\delta_j} \| \mathcal{I}_G)$ **then**
28: Authenticate the login
29: **else**
30: Reject login and report reply attack;

Upon receiving the messages from the user the GSS executes the procedure GSS_2. This starts in Step 2 by verifying that the data that the user requires is currently available and there are no security restrictions on the required data. Then, in Step 3, the GSS checks the freshness of the message. In Steps 4–9, the GSS extracts and calculates all the parameters needed to calculate the message S^5. Then in Step 10 the computed version of S^5 is compared against the received one. If they are not identical, then the procedure and the phase are terminated. Otherwise, the GSS gets the current timestamp in Step 11 and creates two messages S^6, S^7 in Steps 12–15 to include the session key Sek_u. Finally, the timestamp and messages are transmitted to the user mobile over a public channel in Step 16.

Upon receiving the messages from the GSS, the user mobile executes the procedure $Mobile_4$ which starts in Step 24 by verifying the freshness of the message. Step 25 extracts the random number of the message S^6. Then in Step 26, a new version of the session key is calculated. These are the parameters needed to calculate the message S^7. Then in Step 27 the computed version of S^7 is compared against the received one. If they are not identical, then the user is authenticated and the session is created.

3.4. Drone authentication

The drone authentication is done by executing the following sequence of procedures whose details are in Algorithms 4 and 5.

$Drone_1() \xrightarrow{D^1,D^2,D^3,D^4,D^5,T_1} GSS_3() \xrightarrow{D^6,D^7,T_2} Drone_2();$

In this phase, the drone is authenticated to submit certain data types specified in $V_t^{(\sigma,\theta)\prime}$ from its flying zone σ. During this phase, the GSS verifies that the submitted data types are among the ones that the drone is deployed to provide. This is done over a public channel. Upon receiving a session request from the drone, the GSS challenges the legitimacy of the drone. This is done using the credential calculated at the deployment time. One advantage of SETCAP in this phase over related work is that the drone is not delivering directly its data to users. This is convenient for many applications including autonomous driving.

Algorithm 4 Drone Authentication (1)

Input: The ID $\mathcal{I}_\theta$ of a drone θ needs authentication to provide submitting data to GSS according to the vector $V_t^{(\sigma,\theta)\prime}$.

Steps:

1: **procedure** $Drone_1()$
2: **if** $V_t^{(\sigma,\theta)\prime} \leq V_t^{(\sigma,\theta)}$ **then**
3: $T_3 \leftarrow Time\_Stamp()$;
4: Choose randomly $i \in \{1, \ldots, n_t\}$ such that $V_t^{(\sigma,\theta)}(i) \neq 0$;
5: $D^1 \leftarrow i \oplus h(T_3 \| \mathcal{I}_G)$;
6: $D^2 \leftarrow \mathcal{I}_\sigma \oplus h(DSK_\theta^i \| \mathcal{I}_G)$;
7: $D^3 \leftarrow \mathcal{I}_{\delta_i} \oplus h(DSK^i \| \mathcal{I}_G \| \mathcal{I}_\sigma)$;
8: $D^4 \leftarrow \mathcal{I}_\theta \oplus h(DSK^i \| \mathcal{I}_G \| \mathcal{I}_\sigma \| \mathcal{I}_{\delta_i} \| T_3)$;
9: $D^5 \leftarrow h(i \| DSK^i \| \mathcal{I}_G \| \mathcal{I}_\sigma \| \mathcal{I}_{\delta_i} \| \mathcal{I}_\theta \| T_3)$;
10: Transmit $V_t^{(\sigma,\theta)\prime}$, D^1, D^2, D^3, D^4, D^5, and T_3 to the GSS_3 over a public channel;
11: **else**
12: Reject Login;
13: **procedure** $GSS_3(V_t^{(\sigma,\theta)\prime}, D^1, D^2, D^3, D^4, D^5, T_3)$
14: **if** $V_t^{(\sigma,\theta)\prime}$ is confirmed. **then**
15: **if** $|T_3^* - T_3| \leq \Delta T$ **then**
16: $i \leftarrow D^1 \oplus h(T_3 \| \mathcal{I}_G)$;
17: $\mathcal{I}_\sigma \leftarrow D^2 \oplus h(DSK_\theta^i \| \mathcal{I}_G)$;
18: $\mathcal{I}_{\delta_i} \leftarrow D^3 \oplus h(DSK^i \| \mathcal{I}_G \| \mathcal{I}_\sigma)$;
19: $\mathcal{I}_\theta \leftarrow D^4 \oplus h(DSK^i \| \mathcal{I}_G \| \mathcal{I}_\sigma \| \mathcal{I}_{\delta_i} \| T_3)$;
20: **if** $D^5 = h(i \| DSK^i \| \mathcal{I}_G \| \mathcal{I}_\sigma \| \mathcal{I}_{\delta_i} \| \mathcal{I}_\theta \| T_3)$ **then**
21: $R \leftarrow Random\_Number(); T_4 \leftarrow Time\_Stamp()$;
22: $R' \leftarrow h(i \| R)$
23: $D^6 \leftarrow R' \oplus h(DSK^i \| \mathcal{I}_G \| \mathcal{I}_\sigma \| \mathcal{I}_{\delta_i} \| \mathcal{I}_\theta \| T_4)$;
24: $SeK_\theta \leftarrow h(R' \| DSK^i \| \mathcal{I}_G \| \mathcal{I}_\sigma \| \mathcal{I}_{\delta_i} \| \mathcal{I}_\theta \| T_4)$;
25: $D^7 \leftarrow h(R' \| SeK_\theta \| \mathcal{I}_G \| \mathcal{I}_\sigma \| \mathcal{I}_{\delta_i} \| \mathcal{I}_\theta \| T_4)$;
26: Transmit D^6, D^7 and T_4 to the $Drone_2$ over a public channel;
27: **else**
28: Reject;
29: **else**
30: Reject;
31: **else**
32: Reject;

The procedure $Drone_1()$ starts in Step 2 by comparing the two vectors $V_t^{(\sigma,\theta)\prime}$ and $V_t^{(\sigma,\theta)}$ to verify that the drone is delivering the data types specified for it at the deployment time. If the verification is unsuccessful, the procedure and the phase are terminated. In Step 3, the procedure gets the current timestamp. In Step 4, the index i of a data type that the drone is delivering is chosen randomly. In Steps 5–9, the procedure creates five messages D^1, D^2, D^3, D^4, D^5 to include all the relevant data. Finally, the time stamp and messages are transmitted to the GSS over a public channel in Step 10.

Upon receiving the messages from the drone the GSS executes the procedure GSS_3 which starts in Step 14 by verifying that the data that the drone is delivering is OK for collection and there are no security restrictions on collecting the data. Then in Step 15 the GSS checks the freshness of the message. In Steps 16–19, the GSS extracts and calculates all the parameters needed to calculate message D^5. Then in Step 20 the computed version of D^5 is compared against the received one. If they are not identical, then the procedure and the phase are terminated. Otherwise, the GSS calculates a random number R and gets the current timestamp in Step 21. The GSS then creates two messages D^6, D^7 in Steps 23–25 to include the session key Sek_θ. Finally, the vector $V_t^{(\sigma,\theta)\prime}$, the timestamp, and messages are transmitted to the drone over a public channel in Step 26.

Algorithm 5 Drone Authentication (2)

Input: The ID $\mathcal{I}_\theta$ of a drone θ needs authentication to provide submitting data to GSS according to the vector $V_t^{(\sigma,\theta)\prime}$.

Steps:

1: **procedure** $Drone_2(D^6, D^7, T_4)$
2: **if** $|T_4^* - T_4| \leq \Delta T$ **then**
3: $R' \leftarrow D^6 \oplus h(DSK^i \| \mathcal{I}_G \| \mathcal{I}_\sigma \| \mathcal{I}_{\delta_j} \| \mathcal{I}_\theta \| T_4)$;
4: $SeK_\theta \leftarrow h(R' \| DSK^i \| \mathcal{I}_G \| \mathcal{I}_\sigma \| \mathcal{I}_{\delta_j} \| \mathcal{I}_\theta \| T_4)$;
5: **if** $D^7 == h(R' \| SeK_\theta \| \mathcal{I}_G \| \mathcal{I}_\sigma \| \mathcal{I}_{\delta_j} \| \mathcal{I}_\theta \| T_4)$ **then**
6: Authenticate the drone;
7: **else**
8: Reject and report reply attack;

Upon receiving the messages from the GSS, the drone executes the procedure $Drone_2$ which starts in Step 2 by verifying the freshness of the message. In Step 3, the random number of the message D^6 is extracted. Then in Step 4 a new version of the session key SeK_θ is calculated. These are the parameters needed to calculate the message D^7. Then in Step 5 the computed version of D^7 is compared against the received one. If they are not identical, then the drone is authenticated and the session is created.

SETCAP is flexible enough to adopt common key/credentials management protocols. We assume that the user credentials, established at the registration phase, have a lifetime that is fixed by the GSS according to the nature of the application. Users have to register again to obtain new credentials when close to the expiration time of the old ones. The use of random numbers at the registration stage guarantees the generation of new credentials. Sessions that are active at the expiration time are immediately canceled. Furthermore, it is a common practice for the GSS to assume and apply early expiration of credentials, in response to certain suspicious activities.

4. Security analysis

In this section, we present a detailed security analysis of SETCAP. We consider both active and passive adversaries to prove the security of SETCAP. We exploit the Burrows–Abadi–Needham (BAN) logic [25] to prove the mutual authentication between the communicating entities. To verify the security of session keys (SeK_u and SeK_θ for SETCAP), we exploit the ROR model [9].

Applied to our problem, in the ROR model the adversary $\mathcal{A}$ interacts with instances of executing participants. We assumes three instances: (i) a user instance u^{i_1}, (ii) a GSS instance GSS^{i_1}, and (iii) a drone instance θ^{i_1}. The ROR model assumes that the adversary and all participants can access a random oracle of the one-way collision-resistant hash function *Hash*. To simulate real attacks, the ROR model uses the following queries.

1. Dispatch: denotes the action of the adversary of dispatching a message to communicate with participants.
2. AccessedMob: denotes the action of the adversary of accessing credentials of a user from a lost mobile device.
3. AccessedDrone: denotes the action of the adversary of accessing credentials of a drone upon its loss.

4. `Communicate`: denotes the action of the adversary of communicating with an entity to obtain a session key. The entity replies probabilistically according to a coin toss process.
5. `Eavesdrop`: denotes the action of the adversary of eavesdropping messages.
6. `Catch`: denotes the action of the adversary of retrieving session keys SeK_u and SeK_θ.

We prove in Theorem 1 the security of the session keys of SETCAP.

Theorem 1. *Suppose that:*

1. *At time t, $\mathcal{A}$ runs against SETCAP.*
2. *$\mathcal{N}_d$ and $\mathcal{N}_h$ denote the number of dispatch and hash queries, respectively.*
3. *$\mathcal{N}_r$ and $\mathcal{N}_b$ denote the range space of $h(\cdot)$ and the length of the biometric secret key, respectively.*

The average advantage of $\mathcal{A}$ in cracking the semantic security of SETCAP to obtain the session key SeK_u and SeK_θ is bounded by

$$Average\left(\mathcal{W}(t, SeK_u), \mathcal{W}(t, SeK_\theta)\right) \leq \frac{\mathcal{N}_h^2}{\mathcal{N}_b} + \max\left(Z_1 \mathcal{N}_d^{Z_2}, \frac{\mathcal{N}_d}{2^{\mathcal{N}_b}}\right). \tag{1}$$

Proof. The proof relies on a sequence of four games, $\{\mathcal{G}_i \mid 1 \leq i \leq 4\}$. Let S_i denote the event of $\mathcal{A}$ succeeding in correctly predicting the random bit b of the session keys in $\mathcal{G}_i$. We denote the probability of event S_i by $\mathcal{P}(S_i)$. Let $\mathcal{W}(\mathcal{G}_i)$ denote the advantage of $\mathcal{A}$ winning $\mathcal{G}_i$. Therefore $\mathcal{W}(\mathcal{G}_i) = \mathcal{P}(S_i)$.

$\mathcal{G}_1$. In the first game the attack is done in the ROR model. The attack is done by randomly choosing bit b at the begin of the game. Therefore, we have

$$\mathcal{W}(t, SeK_u) = \mathcal{W}(t, SeK_\theta) = |2(\mathcal{W}(\mathcal{G}_1) - 1)| = |2(\mathcal{P}(S_1)) - 1|. \tag{2}$$

$\mathcal{G}_2$. This games treats the case of an eavesdropping attack. In this attack, $\mathcal{A}$ captures the messages exchanged between the communicating parties. In particular, $\mathcal{A}$ captures

1. $M'_{z,t}, S^1, S^2, S^3, S^4, S^5$, and T_1 from u^{i_1} to GSS^{i_1}.
2. S^6, S^7, and T_2 from GSS^{i_1} to u^{i_1}.
3. $V_t^{(\sigma,\theta)\prime}, D^1, D^2, D^3, D^4, D^5$, and T_3 from θ^{i_1} to GSS^{i_1}.
4. D^6, D^7, and T_4 from GSS^{i_1} to θ^{i_1}.

This is done by executing SETCAP using `Eavesdrop` query. Then $\mathcal{A}$ checks the correctness of the derived session keys using the `Catch` and `Communicate` queries. The calculations of session keys SeK_u and SeK_θ involve random numbers and long-term secret keys that are not known by $\mathcal{A}$. Hence eavesdropping the exchanged messages increase the success chance of this game. Hence

$$\mathcal{W}(\mathcal{G}_1) = \mathcal{W}(\mathcal{G}_2). \tag{3}$$

$\mathcal{G}_3$. This game models an active attack using the `Dispatch` and `Hash` queries. The collision-resistant hash function $h(\cdot)$ safeguards the messages $S^1, S^2, S^3, S^4, S^5, S^6, D^1, D^2, D^3, D^4, D^5, D^6$, and D^7. These messages are built using items including timestamps, nonces, secret credentials, and identities. Therefore, the applied quires will not end up causing a collision. Hence, this game is similar to the previous one except for the use of a different set of queries. By applying the birthday paradox, we get:

$$|\mathcal{W}(\mathcal{G}_2) - \mathcal{W}(\mathcal{G}_3)| \leq \frac{\mathcal{N}_h^2}{2\mathcal{N}_b}. \tag{4}$$

$\mathcal{G}_4$. This game adds to the previous one the `AccessedMob` and `AccessedDrone` queries. These allow $\mathcal{A}$ to know the mobile user's credentials

$SK', B, TC', h(.), L, h(.), \mathcal{I}_G, M_{z,t}, Gen(.), Rep(.), \tau_2$, *and*

θ^{i_1} credentials:

$\mathcal{I}_G, \mathcal{I}_\sigma, SK_\theta, V_t^{(\sigma,\theta)}$ *and* $h(.)$.

Executing the `Communicate` query is sufficient for $\mathcal{A}$ to predict the bit b and win the game. Therefore

$$\mathcal{W}(\mathcal{G}_4) = 0.5. \tag{5}$$

Since drones are not using passwords, for the session key SeK_θ,

$$\{\mathcal{W}(\mathcal{G}_3) = \mathcal{W}(\mathcal{G}_4)\}_{SeK_\theta}. \tag{6}$$

The following discussion concerns the session key Sek_u. $\mathcal{A}$ can verify the guessed user's passwords using the stolen mobile data and Zipf's law on passwords. Considering only the robust guessing attacks in case $\mathcal{N}_d = 10^8$, the advantage of $\mathcal{A}$ exceeds 0.5 [14]. However, utilizing personal data of user causes the required number of `Dispatch` queries to be less than or equal to 10^6 [26]. Using a fuzzy extractor with the ability to extract at most $\mathcal{N}_b$ random bits, the probability of the biometric key $\tau_2 \in \{0,1\}^{\mathcal{N}_b}$ is $\frac{1}{2^{\mathcal{N}_b}}$ [10]. This game and the previous one only differ from the use of the `AccessedMob` query in the former. Considering a system that allows a limited attempts of wrong passwords and according to Zipf's law of passwords [14]:

$$\{|\mathcal{W}(\mathcal{G}_3) - \mathcal{W}(\mathcal{G}_4)|\}_{SeK_u} \leq \max\left\{Z_1.\mathcal{N}_d^{Z_2}, \frac{\mathcal{N}_d}{2^{\mathcal{N}_b}}\right\}, \tag{7}$$

where z_1 and z_2 are Zipf's parameters [14]. Therefore, we have

$$\begin{aligned}
&\frac{1}{2}\mathcal{W}(t, SeK_u)\\
&= |(\mathcal{W}(\mathcal{G}_1) - \frac{1}{2})|, \text{by Eq. (2)}\\
&= |(\mathcal{W}(\mathcal{G}_2) - \frac{1}{2})|, \text{by Eq. (3)}\\
&= |(\mathcal{W}(\mathcal{G}_2) - \mathcal{W}(\mathcal{G}_4))|, \text{by Eq. (6)}\\
&\leq |(\mathcal{W}(\mathcal{G}_2) - \mathcal{W}(\mathcal{G}_3))| + |(\mathcal{W}(\mathcal{G}_3) - \mathcal{W}(\mathcal{G}_4))|\\
&\leq \frac{\mathcal{N}_h^2}{2\mathcal{N}_b} + |(\mathcal{W}(\mathcal{G}_3) - \mathcal{W}(\mathcal{G}_4))|, \text{by Eq. (4)}\\
&\leq \frac{\mathcal{N}_h^2}{2\mathcal{N}_b} + \max\left\{Z_1.\mathcal{N}_d^{Z_2}, \frac{\mathcal{N}_d}{2^{\mathcal{N}_b}}\right\}, \text{by Eq. (7)}.
\end{aligned} \tag{8}$$

We hence obtain that

$$\mathcal{W}(t, SeK_u) \leq \frac{\mathcal{N}_h^2}{\mathcal{N}_b} + 2\max\left\{Z_1.\mathcal{N}_d^{Z_2}, \frac{\mathcal{N}_d}{2^{\mathcal{N}_b}}\right\}. \tag{9}$$

By substitution, we obtain

$$\begin{aligned}
&\frac{1}{2}\mathcal{W}(t, SeK_\theta)\\
&= |(\mathcal{W}(\mathcal{G}_1) - \frac{1}{2})|, \text{by Eq. (2)}\\
&= |(\mathcal{W}(\mathcal{G}_2) - \frac{1}{2})|, \text{by Eq. (3)}\\
&= |(\mathcal{W}(\mathcal{G}_2) - \mathcal{W}(\mathcal{G}_4))|, \text{by Eq. (6)}\\
&\leq |(\mathcal{W}(\mathcal{G}_2) - \mathcal{W}(\mathcal{G}_3))| + |(\mathcal{W}(\mathcal{G}_3) - \mathcal{W}(\mathcal{G}_4))|\\
&\leq \frac{\mathcal{N}_h^2}{2\mathcal{N}_b} + |(\mathcal{W}(\mathcal{G}_3) - \mathcal{W}(\mathcal{G}_4))|, \text{by Eq. (4)}\\
&\leq \frac{\mathcal{N}_h^2}{2\mathcal{N}_b} + 0, \text{by Eq. (6)}.
\end{aligned} \tag{10}$$

Finally,

$$\mathcal{W}(t, SeK_\theta) \leq \frac{\mathcal{N}_h^2}{\mathcal{N}_b}. \tag{11}$$

This completes the proof.

We prove the untraceability, anonymity, and the resilience against many attacks of SETCAPvia the following theorems.

Theorem 2.

1. *SETCAP is resilient to replay attack.*
2. *SETCAP withstands man-in-the-middle attack.*
3. *Mutual authentication among communicating parties is guarantied in SETCAP.*

Proof.

1. To test a replay attack, suppose that $\mathcal{A}$ manages to get any of the following messages:
   (a) $M'_{z,t}, S^1, S^2, S^3, S^4, S^5$, and T_1 from u^{i_1} to GSS^{i_1}.
   (b) S^6, S^7, and $T_2$from GSS^{i_1} to u^{i_1}.
   (c) $V_t^{(\sigma,\theta)\prime}, D^1, D^2, D^3, D^4, D^5$, and T_3 from θ^{i_1} to GSS^{i_1}.
   (d) D^6, D^7, and T_4 from GSS^{i_1} to θ^{i_1}.

   The adversary's goal is to replay these messages to the receiver. However, these messages include random numbers and timestamps. Besides, when a participant receives any of these messages, it immediately compares the received timestamp against its current one. The receiver also verifies the received messages. Therefore, SETCAP is resistant to replay attacks.
2. Suppose that $\mathcal{A}$ tries to catch and change the communicated messages between u^{i_1}, GSS^{i_1}, and θ^{i_1}. Suppose that $\mathcal{A}$ tries to modify $M'_{z,t}, S^1, S^2, S^3, S^4, S^5$, and T_1 sent from u^{i_1} to GSS^{i_1}. This requires the attacker to know the credentials SK, hI, b, and TC. Therefore, the attempt will fail. Similarly modifying other messages exchanged between different entities is impossible. Hence, SETCAP is resilient against man-in-the-middle attacks.
3. SETCAP requests u^{i_1} and GSS^{i_1} to mutually authenticate. The same is required for θ^{i_1} and GSS^{i_1}. This is done as follows:
   (a) When receiving $M'_{z,t}, S^1, S^2, S^3, S^4, S^5$, and T_1 from u^{i_1}, GSS^{i_1} checks S^5 to authenticate u^{i_1}.
   (b) When receiving S^6, S^7, and T_2 from GSS^{i_1}, u^{i_1} checks S^7 to authenticate GSS^{i_1}.
   (c) When receiving $V_t^{(\sigma,\theta)\prime}, D^1, D^2, D^3, D^4, D^5$, and T_3 from θ^{i_1}, GSS^{i_1} checks D^5 to authenticate θ^{i_1}.
   (d) When receiving D^6, D^7, and T_4 from GSS^{i_1}, θ^{i_1} checks D^7 to authenticate GSS^{i_1}.

Theorem 3.

1. *SETCAP is secure against attacks resulting from stolen mobile device.*
2. *SETCAP guarantees user anonymity and untraceability.*
3. *SETCAP is resistant against polynomial-time impersonation attacks.*

Proof.

1. This attack assumes that an adversary has either found a lost mobile device or stole one. Therefore, the adversary may be able to extract:

   $$SK', B, TC', h(\cdot), L, h(\cdot), \mathcal{I}_G, M_{z,t}, Gen(\cdot), Rep(\cdot), and\, \tau_2,$$

   from the device's memory. The adversary is unable to extract the user ID $\mathcal{I}_m$ from hI without knowing the nonce b because hI is protected with cryptographic map $h(\cdot)$. Therefore, SETCAP is resilient towards identity extraction attacks. It is also not possible to extract the password because this would require the knowledge of the secret credentials stored at GSS^{i_1} side. Hence, SETCAP is resistant to password extraction attacks.
2. Ensuring the anonymity results from Theorem 3.1. In SETCAP, we use different nonces and contemporary time stamps in building different messages between different entities. This makes the messages dynamic in their structures. Hence it is difficult to trace users and activities over many sessions. Therefore, SETCAP ensures untraceability.
3. In line with our threat model we assume that $\mathcal{A}$ can capture $M'_{z,t}, S^1, S^2, S^3, S^4, S^5$, and T_1 sent from u^{i_1} to GSS^{i_1}. Moreover, $\mathcal{A}$ impersonates an authentic u^{i_1}. However $\mathcal{A}$ will fail to respond correctly to GSS^{i_1} on behalf of u^{i_1}. This is because $\mathcal{A}$ does not know necessary credentials such as $\mathcal{I}_m, b$, and SK. Therefore $\mathcal{A}$ will fail to impersonate a user in polynomial time. Similar justifications show that $\mathcal{A}$ will fail to impersonate a θ^{i_1} and GSS^{i_1} in polynomial time.

Theorem 4.

1. *SETCAP is resilient against the Ephemeral Secret Leakage attack [27].*
2. *SETCAP withstands remote-drone capturing attack.*

Proof.

1. This attack investigates the ability of $\mathcal{A}$ in retrieving long term secret keys and temporary secrets stored in insecure memory. Suppose that $\mathcal{A}$ captures the ephemeral nonces i, j, b and R. However, $\mathcal{A}$ cannot access long-term secrets such as HI and lSK_G and hence fails in creating the session keys. On the other hand, if we assume that $\mathcal{A}$ captures long-term secrets, the absence of nonces will hinder $\mathcal{A}$ from building the session keys. Hence, to build session keys, it is necessary for $\mathcal{A}$ to capture nonces and long-term secrets. This is not possible for $\mathcal{A}$. Moreover, compromising a session key does not help in compromising any other future or past session keys. Hence,SETCAP achieves forward and backward secrecy for session keys. In conclusion, SETCAP is resilient against ESL attacks.
2. According to our threat model, it is possible that $\mathcal{A}$ physically holds a remote drone and retrieves the information stored in it including SK_θ. However, the identities of data types collected by θ^{i_1}, the ID of the drone zone, are the drone ID are used in building SK_θ. Therefore, SK_θ is not the same for all drones. Also, the uniqueness of SK_θ does not allow $\mathcal{A}$ to use the information of the compromised drone to affect the session established between GSS^{i_1} and other drones. Hence, SETCAP is resilient against attacks including capturing remote drones.

5. Evaluation

In this section, we present the results obtained by evaluating SETCAP. We consider two evaluation directions, namely the protocol simulation and the performance comparisons against the state-of-the-art techniques.

5.1. SETCAP simulation

Our experiments are based on AVISPA [28], a broadly used security verification tool. AVISPA is an automated tool to verify security of Internet applications and protocols. In particular, we use AVISPA to verify that SETCAP is secure against passive and active adversaries. Protocols and security properties are expressed in AVISPA using a modular formal language. AVISPA is enriched with back-ends (*SATMC*, *OFMC*, *TA4SP*, and *CL-AtSe*) applying many modern analysis techniques [28]. Among many protocol security verification tools, like *Scyther* [29] and *ProVerif* [30], we selected AVISPA for the following reasons. *ProVerif* utilizes pi calculus and Horn clauses to prove security and authentication of session keys. This is done by applying over-approximations [31]. Rather than approximation, *Scyther* employs symbolic backwards search [31]. The performance of the three tools, AVISPA, *ProVerif*, and *Scyther*, was compared by authors in [31]. The comparisons showed that although AVISPA is not the fastest tool, the

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/SETCAP.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed  : 34 states
 Reachable  : 24 states
 Translation: 5.10 seconds
 Computation: 0.68 seconds
```

Fig. 4. Results of Analyzing SETCAP using CL-AtSe backend of *SPAN* animator of AVISPA.

CLAtse and *OFMC* backends of AVISPA are the most efficient ones. We hence use the *CL-AtSe* backend to test SETCAP. *CL-AtSe* also implements XOR and bitwise operations.

We performed the experiments on a Dell (Vostro) Intel(R) Core(TM) i7-3612 QM CPU @ 2.10 GHz, 8.00 GB RAM on Windows 10 (64-bits) OS. On a virtual box,[3] we employed the broadly acceptable tool *SPAN*,[4] Security Protocol ANimator for AVISPA [32] to simulate SETCAP on random (realistic) networks. It is worth noting that our protocol SETCAP does not assume specific model of controllers and communication shields in drones. Therefore, SETCAP is compatible with common models [33].

The implementation in AVISPA of SETCAP includes the following steps:

1. Implementing SETCAP in the role-oriented language *HLPSL*, High-Level Protocol Specification Language [34]. This involves specifying the roles of all the participants: GSS, drones, and users. In this step, we also specify the composite role determining different scenarios.
2. Transform the *HLPSL* representation of SETCAP into Intermediate Format (*IF*).
3. Lastly, give the *IF* format to the back-end as input for determining whether the protocol is secure.

AVISPA tool gives the adversary session information along with authorized participants. The tool uses the DY model to verify the feasibility of a man in-the-middle attack. Fig. 4 shows the results we obtained for SETCAP. The report emphasizes that SETCAP is resilient against man-in-the middle and replay attacks.

5.2. Performance comparison

In this section, we compare the performance of SETCAP to those of the most relevant state-of-the-art protocols like [8,35–37]. We compare these protocols considering three aspects: communication costs, functionality attributes, computation cost, and energy costs.

To evaluate the efficiency of SETCAP, we calculate and compare the communication cost of GSS, users, and drones. We do this by calculating the number of bits of the messages exchanged between the participants. Our calculations are based on the message sizes shown in Table 2, which considers common values from the literature [8].

Table 2
Bit-sizes used in SETCAP.

Data type	Bit-size
data type identity	32
zone identity	32
drone vectors ($V_t^{(\sigma,\theta)}$)	32
user matrices ($M_{z,t}$)	32
timestamp	32
symmetric cryptographic key size	80
random number	160
elliptic curve point	320
hash output (SHA-1)	160
user and drone identities	160

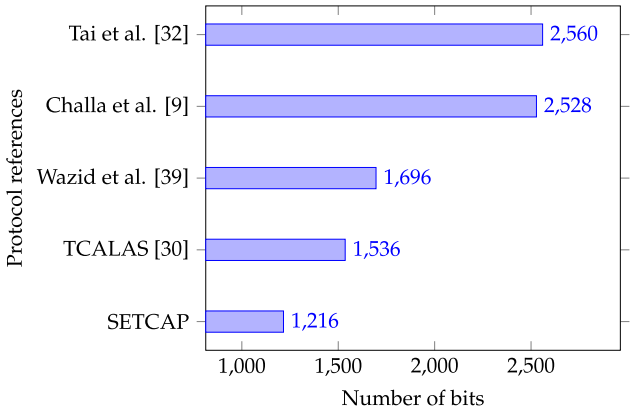


Fig. 5. Comparing communication cost of session creation in SETCAP against the state-of-the-art protocols.

The 80 bit for the cryptographic symmetric key, such as that used in the Double Data Encryption Standard, is as secure as the 160 and 1024 bits of *ECC* and *RSA*, respectively [38]. The following are the sizes of different messages communicated in SETCAP.

1. The size of the message $M'_{z,t}, S^1, S^2, S^3, S^4, S^5$, and T_1, sent from user to GSS, is $32 + 5 \times 160 + 32 = 864$ bits.
2. The size of the message S^6, S^7, and T_2, sent from GSS to user, is $2 \times 160 + 32 = 352$ bits.
3. The size of the message $V_t^{(\sigma,\theta)\prime}, D^1, D^2, D^3, D^4, D^5$, and T_3, sent from drone to GSS, is $32 + 5 \times 160 + 32 = 864$ bits.
4. The size of the message D^6, D^7, and T_4, sent from GSS to drone, is $2 \times 160 + 32 = 352$ bits.

Therefore, the total size of messages in SETCAP for creating a session between the GSS and a user or between the GSS and a drone is 1216 bits. Hence, the creation of a pair of session keys between the three parties of our problem requires 2432 bits. The communication cost of creating a session in SETCAP, TCALAS [8], [36], [37], and [35], are 1216, 1536, 1696, 2528, and 2560 bits, respectively [8]. Fig. 5 shows the comparison between SETCAP and the comparison protocols. From the figure, it is evident that SETCAP has the smallest communication costs. Recalling that the problem treated by SETCAP is more complicated than those treated by the comparison protocols, SETCAP provides a reasonable communication cost compared to existing protocols. Hence the message sizes calculated above at points 3 and 4 parametrize the capacity of messages necessary for starting authenticated communications at both GSS and drones sides.

We compare the main security and functionality characteristics of SETCAP against relevant state-of-art protocols [8,35–37] in Table 3. It is clear from the table that while SETCAP preserves the main

[3] https://www.virtualbox.org/.
[4] http://www.avispa-project.org/.

Table 3
Comparing functionality characteristics of SETCAP against the state-of-the art protocols.

#	Functionality Attribute	[35]	[8]	[36]	[37]	SETCAP
1	Data-types based login for users	×	×	×	×	✓
2	Data-types based registration for users	×	×	×	×	✓
3	Session creation based on data-types for drones	×	×	×	×	✓
4	Data-types based deployments for drones	×	×	×	×	✓
5	GSS complete control over data exchange between drones and users	×	×	×	×	✓
6	Enable GSS to treat drone data before sending to users.	×	×	×	×	✓
7	Resilient to replay & man-in-the-middle attacks.	✓	✓	✓	✓	✓
8	Treats formal security of ROR model.	×	✓	×	×	✓

The symbol ✓denotes that the corresponding protocol supports the corresponding attribute.
The symbol × denotes that the corresponding protocol does not support the corresponding attribute.

Table 4
Comparing computational cost of SETCAP against the state-of-the art protocols.

#	Participant	[35]	[8]	[36]	[37]	SETCAP
1	User	$7\times t_h$	$14\times t_h + t_e$	$16\times t_h + t_e$	$> (5\times t_h + t_e)$	$15\times t_h + t_e$
2	Drone	$10\times t_h$	$7\times t_h$	$8\times t_h$	$> (4\times t_h)$	$8\times t_h$
3	GSS	$6\times t_h$	$9\times t_h$	$7\times t_h$	$> (3\times t_h)$	$\max\{8\times t_h, 11\times t_h\}$
4	Total [s]	≈ 0.00736	≈ 0.0267	≈ 0.02702	≈ 0.26034	≈ 0.02542

Table 5
Formulas for calculating communication and computational costs of SETCAP and TCALAS.

Protocol	Communication cost	Computational cost
SETCAP	$(n_u \times 1216) + (100 \times 1216)$	$(n_u \times 0.00352) + (100 \times 0.00288)$
TCALAS	$n_u \times 100 \times 1536$	$n_u \times 100 \times 0.0267$

security attributes, it also fills the gap of data-type-oriented and zone-oriented registration and login. SETCAP also fills the gap of allowing the GSS to control and manipulate the data collected by the drones before delivering to users. Therefore, SETCAP allows for higher privacy guarantees.

Following the same style of the related work [8], we compare the computation costs of SETCAP against state-of-the art protocols based on the following rough times of the various used operations [8,23,39]:

1. The execution time of the hash function t_h is approximately 0.00032 s.
2. The execution time of the biometric-fuzzy extractor operation t_e is approximately 0.0171 s.

The following are the rough times needed by the different parties of SETCAP:

1. Users: the required time is approximately $15\times t_h + t_e = 0.0219$ s.
2. Drone: the required time is approximately $8\times t_h = 0.00256$ s.
3. GSS (interacting with users): the required time is approximately $11\times t_h = 0.00352$ s.
4. GSS (interacting with drones): the required time is approximately $9\times t_h = 0.00288$ s.

We recall that the authentication between GSS and a user can take place in parallel with that between GSS and a drone. This is not the case for related work we compared against, where the three participants act in a sequence. Hence the total time of SETCAP is calculated using the following formula:

$$\max(\{user\ time,\ drone\ time\}) + \max(\{GSS\ time\ for\ interacting\ with\ drone\ and\ user\});$$

where $\max(\{user\ time,\ drone\ time\}) = 0.0219$, and $\max(\{GSS\ time\ for\ interacting\ with\ drone\ and\ user\}) = 0.00352$. Hence, the total time of SETCAP is $0.02190 + 0.00352 = 0.02542$ s.

Table 4 compares the approximated times required by SETCAP against the time needed by the state-of-the-art techniques as reported in [8]. The second row of the table parameterizes the computational cost necessary from the drone side for starting authenticated communications.

We notice that the computational cost of drones in SETCAP is lower than that in [35,36] and close to that in [8]. Although [37] needs a lower computational cost for drones, it is less secure than SETCAP as clarified in Table 3. Similar arguments prove the advantage of SETCAP for users and GSS against the state-of-the-art. This confirms the efficiency and practicability of SETCAP. The advantage of SETCAP can be further appreciated by recalling that it treats more complicated problems than those treated by the protocols in the literature.

We now consider n_u users $(u_1, \ldots, u_{n_u})$, each with a mobile device, authenticating with the GSS to obtain specific data types from specific flying zones. For a user u, the details on the required data are specified in the binary matrix $M^u_{z,t}$

$$M^u_{z,t} = \begin{bmatrix} x^u_{11} & x^u_{12} & \ldots & x^u_{1n_t} \\ x^u_{21} & x^u_{22} & \ldots & x^u_{2n_t} \\ & \ldots\ldots\ldots\ldots & & \\ x^u_{n_z1} & x^u_{n_z2} & \ldots & x^u_{n_zn_t} \end{bmatrix}; \tag{12}$$

where n_z and n_t are the number of flying zones and number of data types captured by drones, respectively.

The number of drones in different zones are specified in matrix D_n,

$$D_n = \begin{bmatrix} y_1 \\ y_2 \\ . \\ y_{n_z} \end{bmatrix}, \tag{13}$$

where y_i denotes the number of drones in the ith flying zone. For user u, we consider only drones located in the areas providing data for which the user subscribed. The total number of drones that report data for user u is given by

$$D^u_S = \sum_{j=1}^{n_z} \{y_j \mid x^u_{ji} = 1, for 1 \le i \le n_t\}. \tag{14}$$

For the authentication, SETCAP needs to create n_u sessions between n_u users and GSS and $\max_{j=1}^{n_u} D^{u_j}_S$ sessions between drones and GSS. We here focus on the comparison with TCALAS [8], as it represents the closest solution to our proposal. TCALASneeds to create $\sum_{j=1}^{n_u} D^{u_j}_S$ sessions between users and drones. We assume 10 zones each having 10 drones and we assume all users ask for data from all zones.

Table 5 presents the equations for calculating the communication and computation costs for SETCAP and TCALAS for this scenario. Fig. 6 shows the comparison in terms of communications and computations costs between SETCAP and TCALAS, respectively, for a varying number of users. Results show that SETCAP is more efficient and practical than TCALAS. Therefore, for our system model, while TCALAS requires a number of sessions multiplicative in the number of users and drones, for SETCAP the number of sessions is additive. It is worth noting that

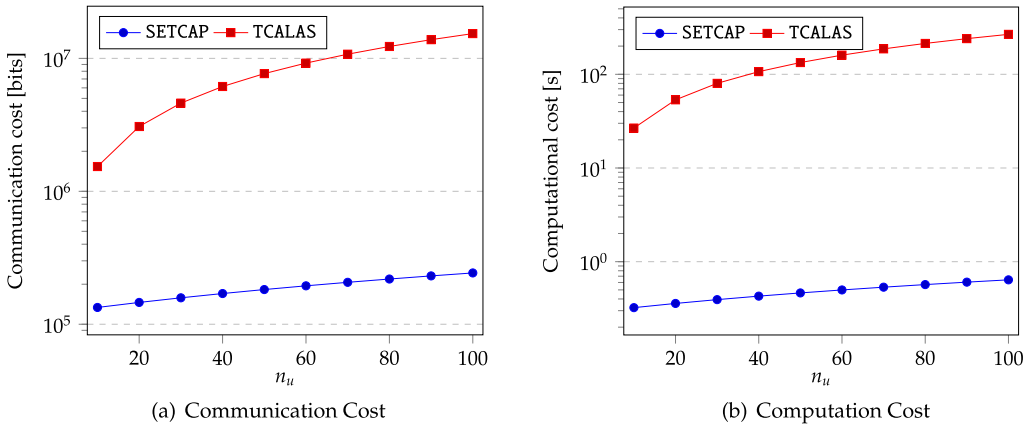


Fig. 6. Comparison between SETCAP and TCALAS [8] in terms of communication (left) and computation (right) costs vs. the number of users.

Table 6
Comparing related work in terms of criteria most related to the motivation of our paper (1).

Protocol	Type of information collected	Applied technique	Credentials types
Amin et al. [40]	Smart card data, IDs, and passwords.	Bio-hashing and one-way cryptographic hash function.	User password, biometrics, and smart card.
Challa et al. [37]	Smart card data, IDs, and passwords.	One-way cryptographic hash function and elliptic curve cryptography.	User password, biometrics, and smart card.
Farash et al. [41]	User and device IDs and passwords.	One-way cryptographic hash function.	Smart card and user password.
Srinivas et al. [8]	Drone IDs, smart card data, IDs, and passwords.	fuzzy extractor and one-way cryptographic hash function.	User password, biometrics, and smart card.
Tai et al. [35]	User and device IDs and passwords.	One-way cryptographic hash function.	Smart card and user password.
Turkanovic et al. [42]	IDs of users and devices and passwords.	One-way cryptographic hash function.	Smart card and user password.
Wazid et al. [10]	IDs, users biometrics, and passwords.	fuzzy extractor and one-way cryptographic hash function.	User password, biometrics, and smart card.
Our proposed technique–SETCAP	Required data types, IDs, users biometrics, and passwords.	fuzzy extractor and one-way cryptographic hash function.	User password, biometrics, and smart card.

this performance advantage of SETCAP over TCALAS is not affected by the fact that all traffic goes through the GSS. This is so as the GSS is a ground station server whose capabilities (especially, compared to those of drones and users) are more than enough for managing this traffic.

The energy consumption $\mathcal{E}$ of SETCAP on different entitles of our system model is a function f of many parameters. Frequency of session establishing procedures ($\mathcal{P}_1$), communication costs ($\mathcal{P}_2$), computational costs ($\mathcal{P}_3$), and system functionalities ($\mathcal{P}_4$) are the main parameters. Therefore: $\mathcal{E} = f(\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4, \ldots)$. It is worth noting that $\mathcal{E}$ generally grows in direct proportion to these four parameters. Concerning the parameters $\mathcal{P}_1, \mathcal{P}_2$, and $\mathcal{P}_3$, the discussions above shows in detail that SETCAP generally consumes less than related state-of-the-art protocols. This is not the case for $\mathcal{P}_4$ as SETCAP provides more functionalities than related protocols. However, this improvement of functionalities provided by SETCAP does not affect the other parameters. Therefore, the overall energy consumption of SETCAP is smaller than that of related protocols compared to above.

6. Related work

In this section, we review existing literature works that are related to our proposed protocol SETCAP. We review both authentication protocols proposed for the IoT and IoD. This is motivated by the fact that IoD is a particularization of IoT to drone devices. Furthermore, we review existing works on data gathering via drones and their authenticated access. The idea of using passwords to remotely authenticate participants was firstly introduced in 1981 by Lamport [43]. Lamport's work motivated many researchers to develop innovative protocols for secure authentication in different applications. Turkanovic et al. in [42] presented the first user authentication protocol that considers both IoT and wireless sensor networks. However, Turkanovic's protocol was later proved to be insecure as it is not resilient to many attacks. Farash et al. in [41] proposed an IoT protocol that connects heterogeneous networks. This protocol was proved later to be also insecure in [40]. Tai et al. in [35] proposed another authentication scheme targeting heterogeneous networks. However, Tai et al.'s protocol is not resilient to many attacks such as replay attacks, forward secrecy, man-in-the-middle, and password guessing. Tanveer et al. in [7] proposed LAKE-IoD, a lightweight protocol for authenticated key exchange for Internet of Drones (IoD) systems. LAKE-IoD guarantees the authenticity of mobile users and then starts a session key establishment mechanism between drones and mobile users. Das et al. in [44] presented the critical security IoT characteristics relevant for IoT environments. Alternatively focusing on drones, Hassanalian et al. in [45] presented a review of the taxonomy, applications, and design issues in drones networks. Cho et al. in [46] proposed DroneRNG, a random number generator for drones. DroneRNG takes into account drone sensors modes: flight and stationary.

To improve location privacy, Kang et al. in [47] presented a drone authentication protocol for LTE networks. Although considering the location privacy of drones, the work of Kang et al. does not consider the data types collected by drones. This may represent a privacy threat, as multiple data types may be aggregated and delivered to a non-legitimate user. Alqassem et al. presented in [48] a review of

Table 7
Comparing related work in terms of criteria most related to the motivation of our paper (2).

Protocol	Collection method	Application	Limitations
Amin et al. [40]	Three-factor (smart card, GWN, and sensor node) sessions.	Wireless sensor networks.	It is not secure against smart card loss. It is not supported with formal security analysis.
Challa et al. [37]	Three-factor sessions.	IoT applications.	It is not supported with formal security analysis.
Farash et al. [41]	Three-factor (users, devices, and GWN) sessions.	IoT applications.	It is not supported with formal security analysis. It is not secure against off-line password-guessing, user-impersonation, and stolen smart card.
Srinivas et al. [8]	Three-factor (drones, user mobiles, and GSS) sessions.	IoD applications.	It requires users to have knowledge of drone IDs. It does not differentiate types of collected data.
Tai et al. [35]	Three-factor (drones, user mobiles, and GWN) sessions.	IoT applications.	It is not supported with formal security analysis.
Turkanovic et al. [42]	Three-factor (device nodes, user mobiles, and GWN) sessions.	Wireless sensor networks.	It is not supported with formal security analysis. It is not secure against off-line password-guessing, user-impersonation, and stolen smart card.
Wazid et al. [10]	Three-factor (device nodes, user mobiles, and GWN) sessions.	IoT applications.	It is not supported with formal security analysis.
Our proposed technique–SETCAP	Several two-factor sessions.	IoD applications.	Sensitive to types of collected data, supported with formal security analysis, and secure against many known attacks.

IoT protocols targeting security issues. This review confirms that IoT networks have mainly two types of devices: (i) devices acting gateways for data collection, and (ii) devices interacting with the environment and humans.

Zhang et al. in [49] presented a communication protocol based on intelligent drones for secure vehicles interaction under adversary circumstances. Focusing on 5G/B5G vehicular ad-hoc networks, they presented a drone-assisted key agreement and anonymous authentication protocol. Jiang et al. in [50] showed that Amin et al.'s work [40] is insecure and hence presented a three-factor lightweight authentication protocol. Cabuk et al. in [51] presented a mutual context-aware authentication scheme for drone groups. In the case of network separation, the scheme helps to recover network swarms. The scheme is independent of the network topology, channel security, and storage. For hierarchical IoT systems, in [52] Wazid et al. proposed a secure lightweight user authentication protocol. The protocol utilizes many factors (biometrics, password, and smart card). In [37] Challa et al. presented a protocol for user authentication in IoT applications. The protocol utilizes ElGamal-type digital signature and Elliptic Curve Cryptography. Compared to non-ECC protocols, this protocol has higher communication and computation costs. Roy et al. [24] proposed a protocol for user authentication in crowdsourcing IoT systems.

For IoD systems, Srinivas et al. in [8] proposed TCALAS, a protocol that enables users to access real-time data from certain drones of a certain geographical area. However, TCALAS assumes that the user knows the drone's ID, which is neither a secure or practical assumption. Furthermore, TCALAS does not distinguish data types that users can get from flying zones. This is not secure as it may lead to delivering both sensitive and nonsensitive data in a single packet. TCALAS does not allow ground station servers to treat data before delivering it to users, which can have security breaches consequences. Tables 6 and Table 7 compare the related work in terms of the criteria most related to the motivation of our paper.

The problem we address in this paper is similar to that of cloud-based data sharing. In fact, the GSS is similar to a cloud-server that collects and re-distributes data collected by drones. In this context, users upload data to the cloud server, and an external entity provides authorization and session keys for both parties [53]. Several variants of this scheme have been proposed in the literature, focusing on guaranteeing both the security of data and the privacy of users [54–56]. However, these schemes do not envision the possibility of collecting ad-hoc data via drones by suitably deploying them in specific geographical areas. Hence, they also usually do not need to account for the management of the geographical areas or the type of data that can be collected.

The limits and security drawbacks of existing literature motivate us to propose this paper to strengthen the security in IoD. For this reason, we present a new secure and lightweight user authentication scheme for IoD systems. Our proposed protocol, SETCAP, allows users to register for accessing specific data types collected by drones in specific geographical areas. To the best of our knowledge, this is a problem that has never been treated by the existing literature. In SETCAP we exploit a fuzzy extractor for verifying the user's local biometric and hash functions methods. Rather than the classical DY model, in SETCAP we consider the CK-adversary model to show its advanced security features.

7. Conclusion and future work

In this paper, we proposed a novel Service-Based Energy-Efficient Temporal Credential Authentication Protocol for IoD, SETCAP that overcomes the drawbacks of the existing authentication protocols. These drawbacks include: (a) assuming that users know the IDs of drones, (b) not distinguishing the types of data read by the drones and treating all read data as one type, and (c) allowing users to directly access drones without going through a ground station server, given the superior capabilities of GSS as a ground server compared to that of drones and users. We formally tested SETCAP against the ROR model and implemented SETCAP in AVISPA simulation tool. The experiments confirmed the security of SETCAP against replay and man-in-the-middle attacks. We also showed that SETCAP is secure against many potential adversary attacks. We also presented a comparative study of SETCAP against TCALAS, a recent state-of-the-art technique. Overall, results show that SETCAP is secure and provides lower communication and computation overhead compared to other state-of-the-art protocols. This guarantees its applicability on resource-constrained devices such as drones.

An interesting direction of future work is to investigate whether authentication protocols of IoD can benefit from modern deep learning algorithms. In fact, thanks to machine learning, we can design predictive methods to reduce the number of authentication procedures required in multiple data gathering rounds. Additionally, we plan to develop an authentication protocol that creates one session for multiple drones collaborating and exchanging certain types of data to complete tasks. The collaboration, in this case, is based on data types. This is a common scenario for drone assisted internet of vehicles. Lastly, we will address some of the limitations of our current proposals. These include (i) removing the assumption of a secure and trusted GSS to extend the applicability of our proposal, and (ii) removing the single point of failure represented by the GSS in favor of a decentralized solution.

CRediT authorship contribution statement

Mohamed A. El-Zawawy: Design and implementation of the research, Analysis of the results, Writing of the manuscript. **Alessandro Brighente:** Design and implementation of the research, Analysis of the results, Writing of the manuscript. **Mauro Conti:** Design and implementation of the research, Analysis of the results, Writing of the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Mirmojtaba Gharibi, Raouf Boutaba, Steven L. Waslander, Internet of drones, IEEE Access 4 (2016) 1148–1162.

[2] Hazim Shakhatreh, Ahmad H Sawalmeh, Ala Al-Fuqaha, Zuochao Dou, Eyad Almaita, Issa Khalil, Noor Shamsiah Othman, Abdallah Khreishah, Mohsen Guizani, Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges, Ieee Access 7 (2019) 48572–48634.

[3] Juan Liu, Xijun Wang, Bo Bai, Huaiyu Dai, Age-optimal trajectory planning for UAV-assisted data collection, in: IEEE INFOCOM 2018-IEEE Conference On Computer Communications Workshops, INFOCOM WKSHPS, IEEE, 2018, pp. 553–558.

[4] Juan Liu, Peng Tong, Xijun Wang, Bo Bai, Huaiyu Dai, UAV-aided data collection for information freshness in wireless sensor networks, IEEE Trans. Wirel. Commun. 20 (4) (2020) 2368–2382.

[5] Yueyan Zhi, Zhangjie Fu, Xingming Sun, Jingnan Yu, Security and privacy issues of UAV: a survey, Mob. Netw. Appl. 25 (1) (2020) 95–101.

[6] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al., Towards fully autonomous driving: Systems and algorithms, in: 2011 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2011, pp. 163–168.

[7] Muhammad Tanveer, Amjad Hussain Zahid, Musheer Ahmad, Abdullah Baz, Hosam Alhakami, LAKE-IoD: Lightweight authenticated key exchange protocol for the Internet of drone environment, IEEE Access 8 (2020) 155645–155659.

[8] Jangirala Srinivas, Ashok Kumar Das, Neeraj Kumar, Joel JPC Rodrigues, TCALAS: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment, IEEE Trans. Veh. Technol. 68 (7) (2019) 6903–6916.

[9] Michel Abdalla, Pierre-Alain Fouque, David Pointcheval, Password-based authenticated key exchange in the three-party setting, in: International Workshop On Public Key Cryptography, Springer, 2005, pp. 65–84.

[10] Mohammad Wazid, Ashok Kumar Das, Vanga Odelu, Neeraj Kumar, Mauro Conti, Minho Jo, Design of secure user authenticated key management protocol for generic IoT networks, IEEE Internet Things J. 5 (1) (2017) 269–282.

[11] Nader Samir Labib, Grégoire Danoy, Jedrzej Musial, Matthias R Brust, Pascal Bouvry, Internet of unmanned aerial vehicles—A multilayer low-altitude airspace model for distributed UAV traffic management, Sensors 19 (21) (2019) 4779.

[12] Nader S Labib, Grégoire Danoy, Jedrzej Musial, Matthias R Brust, Pascal Bouvry, A multilayer low-altitude airspace model for UAV traffic management, in: Proceedings Of The 9th ACM Symposium On Design And Analysis Of Intelligent Vehicular Networks And Applications, 2019, pp. 57–63.

[13] Florence Ho, Rúben Geraldes, Artur Gonçalves, Bastien Rigault, Benjamin Sportich, Daisuke Kubo, Marc Cavazza, Helmut Prendinger, Decentralized multi-agent path finding for UAV traffic management, IEEE Trans. Intell. Transp. Syst. (2020).

[14] Ding Wang, Haibo Cheng, Ping Wang, Xinyi Huang, Gaopeng Jian, Zipf's law in passwords, IEEE Trans. Inf. Forens. Secur. 12 (11) (2017) 2776–2791.

[15] Adarsh Kumar, Rajalakshmi Krishnamurthi, Anand Nayyar, Ashish Kr Luhach, Mohammad S Khan, Anuraj Singh, A novel software-defined drone network (SDDN)-based collision avoidance strategies for on-road traffic monitoring and management, Veh. Commun. 28 (2021) 100313.

[16] Sandeep A Kumar, Jito Vanualailai, B Sharma, Avinesh Prasad, Velocity controllers for a swarm of unmanned aerial vehicles, J. Ind. Inf. Integ. 22 (2021) 100198.

[17] Miguel Duarte, Sancho Oliveira, Anders Christensen, Hybrid control for large swarms of aquatic drones, in: ALIFE 14: The Fourteenth International Conference On The Synthesis And Simulation Of Living Systems, MIT Press, 2014, pp. 785–792.

[18] Krzysztof Ostrowski, Ken Birman, Danny Dolev, Live distributed objects: Enabling the active web, IEEE Internet Comput. 11 (6) (2007) 72–78.

[19] Toshihiro Wakita, Koji Ozawa, Chiyomi Miyajima, Kei Igarashi, Katunobu Itou, Kazuya Takeda, Fumitada Itakura, Driver identification using driving behavior signals, IEICE Trans. Inf. Syst. 89 (3) (2006) 1188–1194.

[20] Danny Dolev, Andrew Yao, On the security of public key protocols, IEEE Trans. Inf. Theory 29 (2) (1983) 198–208.

[21] Thomas S. Messerges, Ezzat A. Dabbish, Robert H. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Trans. Comput. 51 (5) (2002) 541–552.

[22] Ran Canetti, Hugo Krawczyk, Universally composable notions of key exchange and secure channels, in: International Conference On The Theory And Applications Of Cryptographic Techniques, Springer, 2002, pp. 337–351.

[23] Jangirala Srinivas, Ashok Kumar Das, Neeraj Kumar, Joel JPC Rodrigues, Cloud centric authentication for wearable healthcare monitoring system, IEEE Trans. Dependable Secure Comput. 17 (5) (2018) 942–956.

[24] Sandip Roy, Santanu Chatterjee, Ashok Kumar Das, Samiran Chattopadhyay, Saru Kumari, Minho Jo, Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things, IEEE Internet Things J. 5 (4) (2017) 2884–2895.

[25] Michael Burrows, Martin Abadi, Roger Michael Needham, A logic of authentication, Proc. R. Soc. Lond. A. Math. Phys. Sci. 426 (1871) (1989) 233–271.

[26] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, Xinyi Huang, Targeted online password guessing: An underestimated threat, in: Proceedings Of The 2016 ACM SIGSAC Conference On Computer And Communications Security, 2016, pp. 1242–1254.

[27] Chao-Liang Liu, Wang-Jui Tsai, Ting-Yi Chang, Ta-Ming Liu, Ephemeral-secret-leakage secure ID-based three-party authenticated key agreement protocol for mobile distributed computing environments, Symmetry 10 (4) (2018) 84.

[28] Vasileios Karagiannis, Periklis Chatzimisios, Francisco Vazquez-Gallego, Jesus Alonso-Zarate, A survey on application layer protocols for the internet of things, Trans. IoT Cloud Comput. 3 (1) (2015) 11–17.

[29] Chongkyung Kil, Emre C Sezer, Ahmed M Azab, Peng Ning, Xiaolan Zhang, Remote attestation to dynamic system properties: Towards providing complete system integrity evidence, in: 2009 IEEE/IFIP International Conference On Dependable Systems & Networks, IEEE, 2009, pp. 115–124.

[30] Patrick Koeberl, Steffen Schulz, Ahmad-Reza Sadeghi, Vijay Varadharajan, TrustLite: A security architecture for tiny embedded devices, in: Proceedings Of The Ninth European Conference On Computer Systems, 2014, pp. 1–14.

[31] Florian Kohnhäuser, Niklas Büscher, Stefan Katzenbeisser, Salad: Secure and lightweight attestation of highly dynamic and disruptive networks, in: Proceedings Of The 2018 On Asia Conference On Computer And Communications Security, 2018, pp. 329–342.

[32] AVISPA, SPAN, The security protocol animator for AVISPA, 2019.

[33] Xiufang Shi, Chaoqun Yang, Weige Xie, Chao Liang, Zhiguo Shi, Jiming Chen, Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges, IEEE Commun. Mag. 56 (4) (2018) 68–74.

[34] David Von Oheimb, The high-level protocol specification language HLPSL developed in the EU project AVISPA, in: Proceedings Of APPSEM 2005 Workshop, 2005, pp. 1–17.

[35] Wei-Liang Tai, Ya-Fen Chang, Wei-Han Li, An IoT notion–based authentication and key agreement scheme ensuring user anonymity for heterogeneous ad hoc wireless sensor networks, J. Inf. Secur. Appl. 34 (2017) 133–141.

[36] Mohammad Wazid, Ashok Kumar Das, Neeraj Kumar, Athanasios V Vasilakos, Joel JPC Rodrigues, Design and analysis of secure lightweight remote user authentication and key agreement scheme in Internet of drones deployment, IEEE Internet Things J. 6 (2) (2018) 3572–3584.

[37] Sravani Challa, Mohammad Wazid, Ashok Kumar Das, Neeraj Kumar, Alavalapati Goutham Reddy, Eun-Jun Yoon, Kee-Young Yoo, Secure signature-based authenticated key establishment scheme for future IoT applications, IEEE Access 5 (2017) 3028–3043.

[38] Elaine Barker, Elaine Barker, William Burr, William Polk, Miles Smid, et al., Recommendation for key management: Part 1: General, National Institute of Standards and Technology, Technology Administration, 2006.

[39] Jangirala Srinivas, Ashok Kumar Das, Mohammad Wazid, Neeraj Kumar, Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things, IEEE Trans. Dependable Secure Comput. 17 (6) (2018) 1133–1146.

[40] Ruhul Amin, SK Hafizul Islam, GP Biswas, Muhammad Khurram Khan, Lu Leng, Neeraj Kumar, Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks, Comput. Netw. 101 (2016) 42–62.

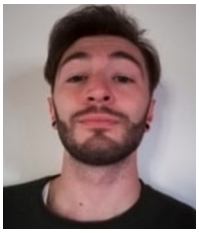
[41] Mohammad Sabzinejad Farash, Muhamed Turkanović, Saru Kumari, Marko Hölbl, An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment, Ad Hoc Netw. 36 (2016) 152–176.

[42] Muhamed Turkanović, Boštjan Brumen, Marko Hölbl, A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion, Ad Hoc Netw. 20 (2014) 96–112.

[43] Leslie Lamport, Password authentication with insecure communication, Commun. ACM 24 (11) (1981) 770–772.

[44] Ashok Kumar Das, Sherali Zeadally, Debiao He, Taxonomy and analysis of security protocols for Internet of Things, Future Gener. Comput. Syst. 89 (2018) 110–125.
[45] Mostafa Hassanalian, Abdessattar Abdelkefi, Classifications, applications, and design challenges of drones: A review, Prog. Aerosp. Sci. 91 (2017) 99–131.
[46] Seong-Min Cho, Eungi Hong, Seung-Hyun Seo, Random number generator using sensors for drone, IEEE Access 8 (2020) 30343–30354.
[47] Dayoung Kang, Gyuhong Lee, Jin-Young Choi, Secure authentication protocol for drones in LTE networks, in: Advances In Security, Networks, And Internet Of Things, Springer, 2021, pp. 17–32.
[48] Israa Alqassem, Davor Svetinovic, A taxonomy of security and privacy requirements for the internet of things (IoT), in: 2014 IEEE International Conference On Industrial Engineering And Engineering Management, IEEE, 2014, pp. 1244–1248.
[49] Jing Zhang, Jie Cui, Hong Zhong, Irina Bolodurina, Lu Liu, Intelligent drone-assisted anonymous authentication and key agreement for 5G/B5G vehicular ad-hoc networks, IEEE Trans. Netw. Sci. Eng. (2020).
[50] Qi Jiang, Sherali Zeadally, Jianfeng Ma, Debiao He, Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks, IEEE Access 5 (2017) 3376–3392.
[51] Umut C. Cabuk, Gokhan Dalkilic, Orhan Dagdeviren, CoMAD: Context-aware mutual authentication protocol for drone networks, IEEE Access (2021).
[52] Kejun Chen, Shuai Zhang, Zhikun Li, Yi Zhang, Qingxu Deng, Sandip Ray, Yier Jin, Internet-of-Things security and vulnerabilities: Taxonomy, challenges, and practice, J. Hardw. Syst. Secur. 2 (2) (2018) 97–110.
[53] Jin Li, Yinghui Zhang, Xiaofeng Chen, Yang Xiang, Secure attribute-based data sharing for resource-limited users in cloud computing, Comput. Secur. 72 (2018) 1–12.
[54] Yinghui Zhang, Dong Zheng, Robert H. Deng, Security and privacy in smart health: Efficient policy-hiding attribute-based access control, IEEE Internet Things J. 5 (3) (2018) 2130–2145.
[55] Biwen Chen, Libing Wu, Li Li, Kim-Kwang Raymond Choo, Debiao He, A parallel and forward private searchable public-key encryption for cloud-based data sharing, IEEE Access 8 (2020) 28009–28020.
[56] Felix Hörandner, Stephan Krenn, Andrea Migliavacca, Florian Thiemer, Bernd Zwattendorfer, CREDENTIAL: a framework for privacy-preserving cloud-based data sharing, in: 2016 11th International Conference On Availability, Reliability And Security, ARES, IEEE, 2016, pp. 742–749.

Mohamed A. El-Zawawy received a Ph.D. in Computer Science from the University of Birmingham in 2007, an M.Sc. in Computational Sciences in 2002 from Cairo University and a BSc. in Computer Science in 1999 from Cairo University. Dr. El-Zawawy is an associate professor of Computer Science at Faculty of Science, Cairo University Since 2014. During the period 2007–2014 Dr El-Zawawy held the position of an Assistant Professor of Computer Science at Faculty of Science, Cairo University. During the year 2009, he held the position of an extra-ordinary senior research at the Institute of Cybernetics, Tallinn University of Technology, Estonia, and worked as a teaching assistant at Cairo University from 1999 to 2003 and later at Birmingham University from 2003 to 2007. Dr. El-Zawawy is interested in security and privacy of IoT and Android.



Alessandro Brighente received his M.Sc. in Telecommunication Engineering and Ph.D. in Information Engineering from the University of Padova, Italy, in 2016 and 2021, respectively. Before his Ph.D., he was research fellow at the Department of Information Engineering at the University of Padova, working on resource allocation for 5G networks. In 2019, he was Visiting Researcher at the Nokia Bell Labs, Stuttgart, Germany, where he worked on Ultra Reliable Low Latency Communications. After his Ph.D., he joined the SPRITZ group at the Department of Mathematics of University of Padova, focusing on security and privacy. His research interests include security and privacy in wireless communications, autonomous vehicles networks, cyber–physical systems, internet of things, beyond 5G networks, and blockchain/distributed ledger technology.

Mauro Conti is Full Professor at the University of Padua, Italy. He is also affiliated with TU Delft and University of Washington, Seattle. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Post-Doc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor the University of Padua, where he became Associate Professor in 2015, and Full Professor in 2018. He has been Visiting Researcher at GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in Security and Privacy. In this area, he published more than 400 papers in topmost international peer-reviewed journals and conferences. He is Area Editor-in-Chief for IEEE Communications Surveys & Tutorials, and has been Associate Editor for several journals, including IEEE Communications Surveys & Tutorials, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, and IEEE Transactions on Network and Service Management. He was Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, ACNS 2020, and General Chair for SecureComm 2012, SACMAT 2013, CANS 2021, and ACNS 2022. He is Senior Member of the IEEE and ACM. He is a member of the Blockchain Expert Panel of the Italian Government. He is Fellow of the Young Academy of Europe. He is part of the Research Council Member of the IOTA Foundation (company providing Blockchain solutions) and Member of the IEEE TEMS Technical Committee on Blockchain and Distributed Ledger Technologies. He is head of SPRITZ Security and Privacy Research Group, Study Program Coordinator of M.Sc. degree in Cybersecurity, and Director of UniPD node of CINI Cybersecurity National Lab.