Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.039459

ARTICLE

Efficient Remote Identification for Drone Swarms

Kang-Moon Seo[1], Jane Kim[1], Soojin Lee[1], Jun-Woo Kwon[1] and Seung-Hyun Seo[1,2,*]

[1]The Department of Electronic & Electrical Engineering, Graduate School, Hanyang University, Ansan, Korea

[2]School of Electrical Engineering, Hanyang University ERICA, Ansan, Korea

*Corresponding Author: Seung-Hyun Seo. Email: seosh77@hanyang.ac.kr

Received: 31 January 2023 Accepted: 17 April 2023 Published: 08 October 2023

ABSTRACT

With the advancement of unmanned aerial vehicle (UAV) technology, the market for drones and the cooperation of many drones are expanding. Drone swarms move together in multiple regions to perform their tasks. A Ground Control Server (GCS) located in each region identifies drone swarm members to prevent unauthorized drones from trespassing. Studies on drone identification have been actively conducted, but existing studies did not consider multiple drone identification environments. Thus, developing a secure and effective identification mechanism for drone swarms is necessary. We suggested a novel approach for the remote identification of drone swarms. For an efficient identification process between the drone swarm and the GCS, each Reader drone in the region collects the identification information of the drone swarm and submits it to the GCS for verification. The proposed identification protocol reduces the verification time for a drone swarm by utilizing batch verification to verify numerous drones in a drone swarm simultaneously. To prove the security and correctness of the proposed protocol, we conducted a formal security verification using ProVerif, an automatic cryptographic protocol verifier. We also implemented a non-flying drone swarm prototype using multiple Raspberry Pis to evaluate the proposed protocol's computational overhead and effectiveness. We showed simulation results regarding various drone simulation scenarios.

KEYWORDS

Drone remote identification; drone swarms; multi-drone authentication

1 Introduction

Drone applications have recently expanded in fields such as agriculture, transportation, construction, and topographic exploration. As the number of studies in which Internet of Things (IoT) devices collaborate and perform tasks in groups increases [1–8], the development of drone swarms, in which multiple drones collaborate in large teams to accomplish a specific task or set of tasks, also has been actively pursued. The drone swarm can overcome the limitations of individual drones, which generally include constrained resources, such as battery life, flight time, and coverage area. In 2020, researchers at Imperial College London developed a new control system for drone swarms that allows drones to make decisions and adjust their behavior based on the behavior of other drones in the swarm [9]. The US Navy announced the successful test of a drone swarm that can protect ships and other vessels [10].

In 2021, Researchers at ETH Zurich developed a system that allows a drone swarm to work together to transport a heavy payload, such as a package or a piece of equipment, from one location to another [11]. These drone swarms are expected to be effectively used for a wide range of applications, such as search and rescue, mapping, military operations, and surveillance.

However, drone flight-related accidents such as invasion of personal privacy, access to no-fly zones, and drone crashes have also become frequent. In 2018, two unidentified drones appeared at Gatwick Airport in the UK, so the runway was shut down [12]. In 2019, two oil production facilities in Saudi Arabia were also bombed using drones [13]. As drone accident cases and illegal use increase, the need to accurately identify the source of drones has been raised. Currently, the authorities of each country, including the Federal Aviation Administration (FAA) [14] and the European Union Aviation Safety Agency (EASA) [15], and the Ministry of Land, Infrastructure, Transport and Tourism (MLIT) of Japan [16] are working to come up with rules on drone remote identification. "Digital License Plates" for drones are being considered a possible drone remote identification method.

Remote identification of drones would mean the drone's functionality could provide information such as identification and location of the drones in flight to people on the ground or other airspace users. Identifying a drone and a drone swarm is essential for many reasons. One of the main reasons is safety. Suppose a drone swarm is operating in an area where there are other aircraft or people. In that case, it is crucial to identify the drones and track their movements to prevent collisions or other accidents. Another reason is to maintain security. Drones can potentially be used for malicious purposes such as espionage or smuggling. Identifying the drones in a swarm can help authorities to track and intercept them if they are being used for illegal activities. Remote identification makes it easier to distinguish between licensed users who comply with drone flight rules and potentially malicious users who pose security risks.

Furthermore, drones are not capable of long-distance flights with limited capacity batteries, so it is necessary to route to intermediate landing spots for long distances. The ground control server that monitors each landing spot base must quickly identify the drone that comes to the base station and verify its authenticity. In the case of drones that have passed unauthorized places, such as no-fly zones, tracking the travel path of the drones is necessary [17]. Identifying a drone swarm can also be helpful for managing and coordinating the drones. For example, suppose a swarm is being used for search and rescue operations. In that case, it is crucial to be able to identify each drone in order to direct them to specific areas or to retrieve data from them.

These reasons make it necessary to develop a remote identification system for drones, including drone swarms. There have been numerous works on drone identification and authentication methods [18–25], but only a few on authentication methods for drone swarms [26–30]. Also, there is still no effective method of identification for drone swarms. Unfortunately, it is challenging for the ground control server to recognize a drone swarm effectively. Developing a method to identify drone swarms before they perform tasks effectively should be ongoing. In this paper, we propose a remote identification protocol for efficiently identifying drone swarms and verifying their access authorization. Our protocol utilizes a Reader drone (*Rdrone*) to supervise each airspace region and efficiently read the Remote IDs of the drone swarm coming to the region. The *Rdrone* refers to a drone equipped with a device that enables it to read the remote ID of drones and identify drones in the airspace above the base station. In general, drones broadcast remote IDs through Bluetooth or Wi-Fi with limited communication distance, so drones must closely approach the ground control server of the base station. However, it can be challenging to access ground control servers closely to identify drone IDs in a complex downtown area with obstacles. Therefore, we introduce the *Rdrone* for each

area to effectively identify remote IDs of a drone swarm and transmit gathered messages from each drone to a ground control server.

The gathered message is a grouping proof that is evidence to the ground control server (verifier) that the identities (IDs) of the authorized swarm drones were simultaneously in a range of the *Rdrone*. In our proposed protocol, a ground control server (verifier) can efficiently extract from the proof evidence of the presence of each drone ID in the swarm drones and verify their access authorization by using the batch verification technique. We conducted a formal security verification to prove the proposed scheme's correctness and security by utilizing ProVerif [31]. Through formal security verification, we verify the secrecy of the group session key and the authenticity of the message sent by the drone swarm and the *Rdrone* in the presence of a malicious attacker. Moreover, in order to evaluate the computational overhead of the proposed protocol in a wireless communication environment, we built a non-flying prototype for a drone swarm by using multiple Raspberry Pis and measured each phase's processing time. We also compared the computational overhead of the proposed protocol with related schemes and showed that it is more efficient in verifying drone swarm members' ID as the drone swarm size increases than other existing schemes. In addition, we simulated the proposed protocol using the Omnet++ simulator [32] to analyze the performance of the drone swarm identification considering the drone swarm environment with many drone swarms of various sizes. We analyzed the verification time according to the number of *Rdrones* and the size of the drone swarm. We also suggested the appropriate operation interval between drone swarms when there are multiple drone swarms by simulating a drone swarm scheduling scenario. The remainder of the paper is organized as follows. Section 2 introduces a remote identification rule and the related work. In Section 3, we present our system model and an adversary model. We propose a remote drone swarm identification protocol in Section 4, and we provide security analysis and formal verification of the proposed scheme in Section 5. In Section 6, we evaluate the performance of our proposed scheme. Section 7 concludes this paper.

2 Related Works

In this section, we explain a remote identification rule and related works of drone swarm authentication protocol.

2.1 Remote Identification Rule

In December 2019, the US Federal Aviation Administration (FAA) issued guidelines for solving safety/security problems for unmanned aerial vehicles [14]. After that, from April 2021, the final rules for remote drone identification took effect and were implemented. The user must use the Standard Remote ID Drone or attach the Remote ID broadcast module to the drone. The FAA made it possible to identify and locate the ID of a drone in flight using the Remote ID. Some FAA-approved areas allow drones to operate without a Remote ID, but most areas require Remote IDs containing information about drones and control stations to comply with Remote ID rules.

According to the FAA, whether using a Standard Remote ID Drone or a remote ID broadcast module, the message elements must be broadcast from take-off to shut down. A Standard Remote ID Drone or a drone with a remote ID broadcast module must transmit the following message elements [14].

- A unique identifier for the drone
- The drone's latitude, longitude, geometric altitude, and velocity
- An indication of the latitude, longitude, and geometric altitude of the control station (standard) of the take-off location (broadcast module)
- A time mark
- Emergency status (Standard Remote ID Drone only)

The European Union Aviation Safety Agency (EASA) announced operational procedures and rules according to specifications to identify drones in May 2019 and is making efforts to respond to the standardization of illegal drones [15]. The EASA is considering Direct Remote ID, a broadcast-type remote identification method, as a drone remote identification method. The EASA plans to use Direct Remote ID for the purpose of drone security. Related content is contained in UAS Regulation (EU) 2019/945 and 947, which addresses design and manufacturing requirements for drone systems in Europe and rules on drone operations. According to this rule, in the case of drones corresponding to Classes 5 and 6, it is mandatory to install Direct Remote ID. Before the operation of the system, the pilot is required to ensure that the Direct Remote ID system is active and up to date. Information that must be transmitted from Direct Remote ID includes operator registration information, drone-specific serial number, topographic drone location, flight guard, and pilot location or take-off location information.

The Ministry of Land, Infrastructure, Transport and Tourism (MLIT) [16] of Japan is planning to implement a drone registration rule on June 20, 2022. Drone users are required to attach a 'Remote ID' that transmits information such as registration number during drone flights. After June 2022, drones will be produced with a Remote ID attached.

2.2 ECC-Based IoT Edge Device Authentication Protocol

Several studies have recently been conducted on authentication protocols between IoT edge devices, such as drones, robots, and authentication servers in IoT environments [6–8]. The previous works utilized ECC (Elliptic Curve Cryptography)-based authentication for resource-constraint devices [7]. Rostampour et al. [6] proposed a new ECC-based authentication protocol for ensuring secure communication between cluster servers and IoT edge devices. The authors masked the embedded device's unique identifier and used a random value to prevent a traceability attack. Kwon et al. [8] proposed mutual authentication and handover authentication for an unmanned aerial vehicle (UAV). A zone service provider in each region supports a handover process. Jain et al. [7] suggested a robot access control protocol by designing ECC-based mutual authentication between robots and cloud servers. The authors reduced the robot's computation power and proved that their protocol is resistant to several attacks, such as a Man-in-the-middle attack and a forgery attack. However, existing authentication protocols are inefficient in terms of computational cost when verifying multiple devices' identities, as they only consider the authentication of a single device.

2.3 Drone Swarm Authentication Protocol

Recently, as the need for drone swarms has increased, studies on drone swarms, such as drone swarm formation [33], drone swarm allocation [34], and drone swarm application [35–37], have been conducted. Since a drone swarm is a very recent development (within the last two to three years), there are few studies on drone swarm authentication or key distribution. Moreover, there have yet to be considered drone swarm identification. In this section, we introduce some examples of recent drone swarm authentication protocols. Ardin et al. [26,27] proposed a group authentication and handover

solution for drone swarms in 2021. Their protocol allows a new drone to join an existing drone swarm by using a guard drone, but it does not take into account drones leaving the drone swarm. However, if drones inevitably leave the drone swarm, such as when drones fail or battery power is lost, the drone leaving process, such as establishing a new group key, must be considered.

Abdel-Malek et al. [28] proposed a distributed delegation-based authentication mechanism for overcoming drone resource limitations and scalability. The proposed method can reduce traffic overhead from the 5G core network. Moreover, when a failure or compromise to the leader drone occurs, a new leader drone can be assigned without interrupting the drone swarm communication. However, the PKI (Public Key Infrastructure)-based proxy signature solution requires significant computational overhead when authenticating many drones in the swarm. Han et al. [29] proposed an authentication protocol that enables mutual authentication between fog and edge without a ground station. The proposed method is effective against MITM (Man in the Middle Attack) attacks and replay attacks. Moreover, since it utilizes security functions with various hardware, it is more computationally efficient than the PKI-based approaches. However, it does not cover the compromised drone issue, and it requires special hardware. Han et al. [30] proposed a blockchain-based UAV swarm identity management model (B-UIM-M). The proposed method has advantages in UAV identity management, UAV identity authentication, scalability, and secure transmission of communication data. However, the blockchain operation causes a communication bottleneck between drones which are consensus nodes.

3 System Model

In this section, we describe the overall and adversary models of our proposed remote identification protocol for drone swarms.

3.1 Overall Model

The proposed model consists of five components: UAV Traffic Management (UTM), Base stations, Ground Control Servers (GCSs), Reader drones (*Rdrones*), and Drone swarms. The following is a description of each component in depth. Fig. 1 shows the overall system model. A base station and a GCS are located in each region. The GCS verifies the identity of the drone swarm that enters the region to prevent unauthorized drones from gaining access. A *Rdrone* flies around each GCS and directly communicates with the drone swarm to support the drone swarm identification of the GCS. Depending on the drone swarm's flight schedule, single or multiple drone swarms can visit each region simultaneously. The main entities are defined as follows.

- **UAV Traffic Management (UTM):** The UTM is a system that manages the traffic of a drone swarm. The UTM is responsible for the management of drones throughout each region using base stations and GCSs. Before flying, the drone swarm should inform the flight route, including the list of the drone swarm ID, regions to visit, and flight time slots to the UTM. The UTM shares registered flight information with servers on the flight route.
- **Base Station:** The base station is a communication infrastructure to facilitate communication between entities in each region. The base station connects the wireless communication network and the entities such as *Rdrones*, drone swarms, and GCSs in each region.
- **Ground Control Server (GCS):** The GCS in each region has the role of verifying the identification of drone swarms joining its region for access control. It receives flight-related information from the UTM before flying. For drone swarm verification, the server gets the collected drones' authentication information from the *Rdrone* and conducts the verification process.

- **Reader Drone (Rdrone):** A *Rdrone* exists in each region, and the number of *Rdrones* varies by region. The *Rdrone* acts as an intermediary between the drone swarm and the GCS. The *Rdrone* flies to the airspace of its region and checks whether the incoming drone swarm is registered for flight in the region. Once the *Rdrone* receives identification data from all drone swarm drones, it delivers the data to the GCS. We assume that the GCS continuously monitors the *Rdrone* so that it can detect and defend against the malicious behavior of the *Rdrone*. So, we let the *Rdrone* be an honest entity like a GCS in our system model.
- **Drone Swarm:** A drone swarm is a group of drones that move and work together for a particular purpose, such as data collection and reconnaissance. It is assumed that each drone in the drone swarm has enough processing power to compute cryptographic operations such as symmetric and asymmetric encryption.

We assume that the drones used in the proposed protocol meet the FAA's remote identification requirements. The drone controller creates several drones as a swarm and pre-registers the swarm's movement path in the UTM. Following that, when the registered drone swarm moves along the registered route, a GCS is present in each region and performs remote identification of drones passing through the region. Through the group session key and remote ID, the GCS verifies that each drone in the drone swarm is a part of the drone swarm registered to fly in advance. The *Rdrone* identifies the drone swarm and collects authentication data from the drone swarm, and then transmits it to the GCS. The drone swarm is identified using our proposed protocol, and the GCS verifies the drone swarm based on the authentication data received from the *Rdrone*. The GCS also examines the number of drones in the drone swarm. If the number of drone swarm members is less than that of UTM members, it generates a new group session key and shares it with the drone swarm via our group session key update phase.

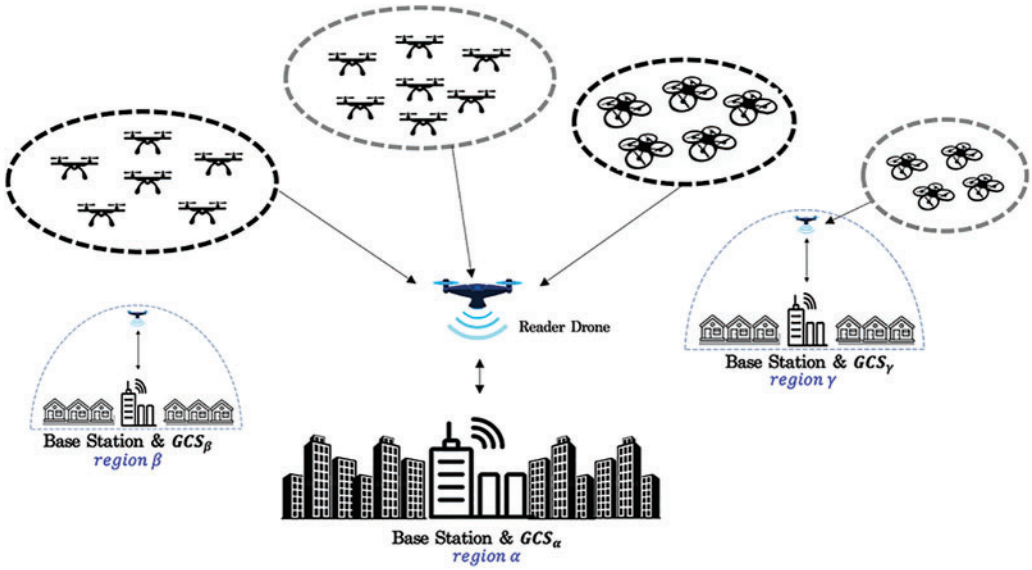

Figure 1: Overall system model

3.2 Adversary Model

The adversarial attackers in the proposed protocol can be classified as a malicious drone or a malicious drone swarm. A malicious drone may try to disguise itself as a member of a drone swarm attempting to access the area to gain access to the area he wishes to enter. A malicious drone swarm may attempt to access a location that has not previously registered with the UTM. Alternatively, a drone not previously registered may attempt to join the drone swarm. We assume that an attacker can listen in on the communication channel between the drone and the *Rdrone* or the *Rdrone* and the GCS. Furthermore, we assume that the attackers understand the cryptographic protocol used by the drone swarm so that it can generate private-public key pairs and mimic the transmitted messages through the drone swarm's authentication protocol. Malicious drones are also assumed to have unique IDs.

- Impersonation attack: Malicious drones may attempt to impersonate members of a specific drone swarm. They may attempt to participate in the authentication process by generating fake IDs or keys while the *Rdrone* authenticates the drone swarm. On the other hand, a malicious drone swarm may attempt to disguise a malicious drone as a member of the drone swarm by sharing its group session key.
- Replay attack: A malicious drone or drone swarm can reuse information used to authenticate in the past without prior permission. A malicious drone can also attempt to access a location other than a UTM-registered route by using expired information.
- Eavesdropping attack: A malicious attacker may collect communication information between a drone swarm and a *Rdrone* via an open channel. Eavesdropping information can be used to gain access to the area. It can also attempt to steal drone IDs based on eavesdropping data or expose group session keys used by drone swarm members.

4 Remote Identification Protocol for Drone Swarms

In this section, we propose a protocol for remote identification via a *Rdrone* to allow the GCS in each region to access only the allowed drone swarm. Fig. 2 shows the overall process of the proposed protocol. The protocol consists of five phases: Setup, Flight Registration, Remote ID Identification, Batch Verification, and Drone Leaving. The notations used in the proposed protocol are defined in Table 1.

4.1 Setup Phase

In the Setup Phase, each drone of the drone swarm and the *Rdrones* generate Elliptic Curve Cryptographic (ECC)-based public/private key pairs to be used in the proposed protocol. Each drone, which belongs to a drone swarm, generates a private-public key pair, and *Rdrones* also generate its private-public key pair. Let E be an elliptic curve defined over a finite field Z_q^* where q is a prime number. The P is a generator of the G, which is an addictive cyclic group of points on the elliptic curve E. We assume that the generated public keys of all participants are propagated in the network. The $Drone_i$ and the *R drone* perform the following steps:

- A controller establishes a drone swarm $\mathcal{S} = \{D_1^{\mathcal{S}}, D_2^{\mathcal{S}}, \ldots, D_n^{\mathcal{S}}\}$ consisting of n drones.
- Each $D_i^{\mathcal{S}}$ chooses a random number $sk_{D_i} \in Z_q^*$ as its private key and computes a public key $pk_{D_i} = sk_{D_i}P$.
- The $RDrone_i$ chooses a random number $sk_{R_i} \in Z_q^*$ as its private key and computes a public key $pk_{D_i} = sk_{D_i}P$.

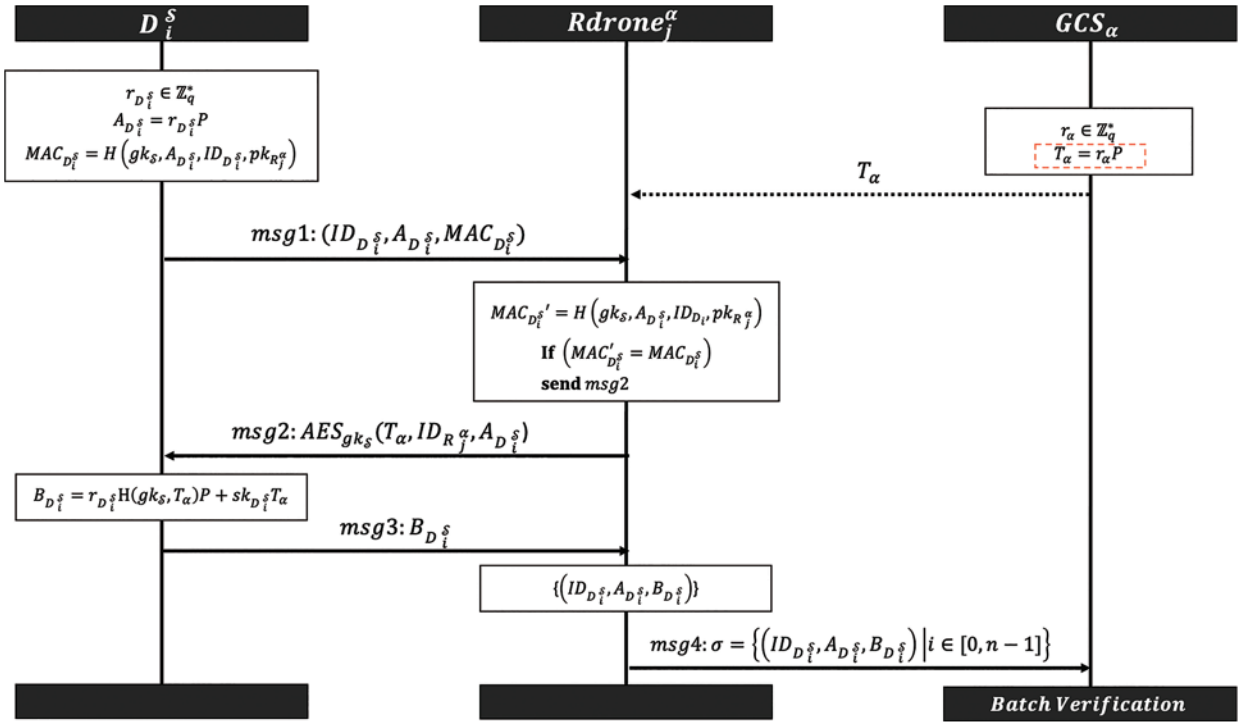

Figure 2: Proposed protocol

Table 1: List of notations

Notation	Description
$D_i^{\mathcal{S}}$	The i-th *drone* of the *Drone Swarm* $\mathcal{S}$
$Rdrone_j^{\alpha}$	The j-th *Reader Drone* of the GCS_{α}
GCS_{α}	The Ground Control Server of the region α
$sk_{D_i^{\mathcal{S}}}, pk_{D_i^{\mathcal{S}}}$	The private-public key of the $Drone_i^{\mathcal{S}}$
$r_{D_i^{\mathcal{S}}}, A_{D_i^{\mathcal{S}}}$	The Ephemeral private-public key of the $Drone_i^{\mathcal{S}}$
$sk_{R_j^{\alpha}}, pk_{R_j^{\alpha}}$	The private-public key pair of the $Rdrone_j^{\alpha}$
r_{α}, T_{α}	The Ephemeral private-public key pair of the GCS_{α}
$gk_{\mathcal{S}}$	The group session key of the *Drone Swarm A* shared between $Drone_i^{\mathcal{S}}$, *Rdrone* and *GCS*
$ID_{D_i^{\mathcal{S}}}$	The ID of the $Drone_i^{\mathcal{S}}$
$ID_{R_j^{\alpha}}$	The ID of the $Rdrone_j^{\alpha}$
$MAC_{D_i^{\mathcal{S}}}(\cdot)$	The Message Authentication Code generated by $D_i^{\mathcal{S}}$
$H(\cdot)$	A cryptographic one-way hash function
$AES(\cdot)$	A symmetric key encryption algorithm called AES (advanced encryption standard)
$E_{pk}(\cdot)$	An asymmetric encryption using public key *pk*
$D_{sk}(\cdot)$	An asymmetric decryption using secrete key *sk*

4.2 Flight Registration Phase

Drone swarms register their flight route with the UTM. Then, each drone in the drone swarm generates an ephemeral private-public pair for each region it will visit. The UTM shares drone swarm flight information with the GCS in each region, as well as generates and distributes the drone swarm's group session keys. The detailed process is as follows.

- The $Drone_i^{\mathcal{S}}$ submits an $ID_{D_i^{\mathcal{S}}}$ and a flight route to the UTM.
- A $D_i^{\mathcal{S}}$ creates as many ephemeral key pairs $\left(r_{D_i^{\mathcal{S}}}, A_{D_i^{\mathcal{S}}}\right)$ as the number of regions it will pass through.
- The UTM generates a random number which is used as a group session key $gk_{\mathcal{S}}$ for each drone swarm and broadcasts to the drone swarm and the GCS in the drone swarm's path.

4.3 Remote ID Identification Phase

When a drone swarm enters the region, the $Rdrone_j^{\alpha}$ verifies each swarm drone's ID and checks whether the flight of each drone is registered. Once the $Rdrone_j^{\alpha}$ verifies each drone's ID, the $Rdrone_j^{\alpha}$ gathers identification and authentication data from the drone swarm.

- A drone $D_i^{\mathcal{S}}$ computes Message authentication code $MAC_{D_i^{\mathcal{S}}}$, according to Eq. (1).

$$MAC_{D_i^{\mathcal{S}}} = H\left(gk_{\mathcal{S}}, A_{D_i^{\mathcal{S}}}, ID_{D_i^{\mathcal{S}}}, pk_{R_j^{\alpha}}\right) \tag{1}$$

- where H is the hash function, $gk_{\mathcal{S}}$ is the drone swarm $\mathcal{S}$'s group session key, $A_{D_i^{\mathcal{S}}}$ is the ephemeral public key of $D_i^{\mathcal{S}}$, $ID_{D_i^{\mathcal{S}}}$ is a remote ID of $D_i^{\mathcal{S}}$, and $pk_{R_j^{\alpha}}$ is the public key of R_j^{α}. Then, the $D_i^{\mathcal{S}}$ sends the following message to the $Rdrone_j^{\alpha}$.

$$msg\ 1\colon \left\{ID_{D_i^{\mathcal{S}}}, MAC_{D_i^{\mathcal{S}}}, A_{D_i^{\mathcal{S}}}\right\}_{D_i^{\mathcal{S}} \to Rdrone_j^{\alpha}} \tag{2}$$

- The $Rdrone_j^{\alpha}$ checks whether the received $ID_{D_i^{\mathcal{S}}}$ was registered. Then the $Rdrone_j^{\alpha}$ calculates $MAC'_{D_i^{\mathcal{S}}}$ using $gk_{\mathcal{S}}$ of the swarm drone $\mathcal{S}$ which the $D_i^{\mathcal{S}}$ belongs to as per Eq. (3).

$$MAC'_{D_i^{\mathcal{S}}} = H\left(gk_{\mathcal{S}}, A_{D_i^{\mathcal{S}}}, ID_{D_i^{\mathcal{S}}}, pk_{R_j^{\alpha}}\right) \tag{3}$$

- If the $MAC_{D_i^{\mathcal{S}}}{}'$ matches the $MAC_{D_i^{\mathcal{S}}}$, the $Rdrone_j^{\alpha}$ sends the message.

$$msg\ 2\colon AES_{gk_{\mathcal{S}}}\left(T_{\alpha}, ID_{R_j^{\alpha}}, A_{D_i^{\mathcal{S}}}\right) \tag{4}$$

To the $D_i^{\mathcal{S}}$, where the T_{α} is the ephemeral public key of the GCS_{α} and $ID_{R_j^{\alpha}}$ is the remote ID of the $Rdrone_j^{\alpha}$.

- The $D_i^{\mathcal{S}}$ decrypts the message using the $gk_{\mathcal{S}}$. Then the $D_i^{\mathcal{S}}$ computes a $B_{D_i^{\mathcal{S}}}$ as per Eq. (5). where the $sk_{D_i^{\mathcal{S}}}$ is the secret key of the $D_i^{\mathcal{S}}$, according to Eq. (4), and sends the $B_{D_i^{\mathcal{S}}}$ to the $Rdrone_j^{\alpha}$.

$$msg3\colon B_{D_i^{\mathcal{S}}} = r_{D_i^{\mathcal{S}}} H\left(gk_{\mathcal{S}}, T_{\alpha}\right) P + sk_{D_i^{\mathcal{S}}} T_{\alpha} \tag{5}$$

4.4 Batch Verification Phase

The $Rdrone_j^{\alpha}$ gathers the $B_{D_i^{\mathcal{S}}}$ from each $D_i^{\mathcal{S}}$. Then, $Rdrone_j^{\alpha}$ submits the collected msg4 to the GCS_{α}.

$$msg4:\ \sigma = \left\{ \left(ID_{D_i^{\mathcal{S}}}, A_i, B_i \right) | i \in [0, m-1] \right\}_{Rdrone_j^{\alpha} \to GCS_{\alpha}} \tag{6}$$

The GCS_{α} efficiently verifies the identification and authentication data of the drone swarm $\mathcal{S}$ by processing batch verification. The GCS_{α} get a σ from the $Rdrone_j^{\alpha}$ and verifies it through a list of the drone swarm's public keys received from the UTM. The batch verification phase is described in Algorithm 1. If a drone swarm passes the verification normally, an accept message is sent to the $Rdrone_j^{\alpha}$, and if it fails, a reject message is sent to the $Rdrone_j^{\alpha}$. The group session key update phase is executed if the verification is successful, and the number of drone swarm members requested for verification by the $Rdrone_j^{\alpha}$ is less than the number of drone members registered in UTM.

Algorithm 1: Batch verification

Input: σ, PK

Output: $\{accept, reject\}$

1: Compute $A = \sum_{i=0}^{m-1} A_{D_i^{\mathcal{S}}} = \sum_{i=0}^{m-1} r_{D_i^{\mathcal{S}}} P$

2: Compute $B = \sum_{i=0}^{m-1} B_{D_i^{\mathcal{S}}} = \sum_{i=0}^{m-1} \{ r_{D_i^{\mathcal{S}}} H\left(gk_{\mathcal{S}}, T_{\alpha}\right) P + sk_{D_i^{\mathcal{S}}} r_{\alpha} P$

3: let $h = H\left(gk_{\mathcal{S}}, T_{\alpha}\right)$

4: then, $B = \sum_{i=0}^{m-1} \left(hr_{D_i^{\mathcal{S}}} P + sk_{D_i^{\mathcal{S}}} r_{\alpha} P \right)$

5: $B - hA = \sum_{i=0}^{m-1} \left(hr_{D_i^{\mathcal{S}}} P + sk_{D_i^{\mathcal{S}}} r_{\alpha} P - hr_{D_i^{\mathcal{S}}} P \right) = \sum_{i=0}^{m-1} \left(sk_{D_i^{\mathcal{S}}} r_{\alpha} P \right)$

6: then, calculate $r_{\alpha}^{-1}\left(B - hA\right)$

7: **if** $\sum_{i=0}^{m-1} \left(pk_{D_i^{\mathcal{S}}} \right) = r_{\alpha}^{-1}\left(B - hA\right)$:

8: *accept*

9: **if** $m < n$: ***Group session key update***

10: **else** *reject*

4.5 Group Session Key Update

In the case of drones, sudden battery shortages or discharge can result in a falling situation. Furthermore, unexpected sensor failures can result in situations where the mission is not normally carried out, or environmental factors can cause deviations from the drone swarm. Because the drone that left at this time is no longer a member of the drone swarm, the group session key for the newly changed member of the drone swarm must be updated. The GCS updates the group session key and shares it with the drone swarm once it confirms that the members of the drone swarm have been changed through the batch verification step. The following are the detailed group session key update steps.

- A GCS_{α} generates a 256-bit random number and sets it as the new group session key $gk'_{\mathcal{S}}$.
- The GCS_{α} sends the $Rdrone_j^{\alpha}$ in its area the ID values of the newly updated group member $\mathcal{S}' = \left\{ D_1^{\mathcal{S}'}, D_2^{\mathcal{S}'}, \ldots, D_m^{\mathcal{S}'} \right\}$, then the timestamp and the new group session key $gk'_{\mathcal{S}}$, encrypted with the drone swarm member's public key, are delivered to the *Rdrone*.

$$\left\{ \left(ID_{D_i^{\mathcal{S}}}, E_{pk_{D_i^{\mathcal{S}}}}\left(gk'_{\mathcal{S}}\right), time_{GCS_{\alpha}} \right) | i \in [0, m-1] \right\}_{GCS_{\alpha} \to Rdrone_j^{\alpha}} \tag{7}$$

- The $Rdrone_j^{\alpha}$ broadcasts the value received from GCS_{α} to the drone swarm.
- Each $D_1{}^{S'}$ decrypts the ciphertext corresponding to its ID with its private key and updates it with a new group session key.

$$gk_S^{'} = D_{sk_{D_i^S}}\left(E_{pk_{D_i^S}}\left(gk_S^{'}\right)\right) \tag{8}$$

5 Security Analysis and Formal Verification

In this section, we provide a security analysis of the proposed scheme. We also demonstrate the security of the proposed scheme through formal verification using ProVerif.

5.1 Security Analysis

Resistance against impersonation attack: A malicious drone might pretend to be a member of one of the drone swarms. However, the malicious drone does not know the shared secret key gk_S, so it could not generate a valid MAC. Moreover, if the malicious drone tries to use another drone's ID in the identification process, it does not know the corresponding private key to another drone's ID. Thus, it is impossible for the malicious drone to generate a valid signature B. As a result, it fails to succeed in the impersonation attack. Also, a malicious drone might generate several fake IDs of drones that do not exist in the region and make them pass the identification process. However, as the drone swarm shares its flight route and IDs with the *Rdrone*, the *Rdrone* can check whether each drone's ID was registered or not. As a result, our proposed protocol can prevent the impersonation attack.

Resistance against replay attack: A malicious drone that left the drone swarm might attempt to enter the area α by using the $ID_{D_i^S}$, gk_S. If the member of the drone swarm is changed, the GCS updates the group session key gk_S to the gk'_S. The GCS also shares an updated list of the drone swarm with other GCSs. Since the left drone does not know the gk'_S, it cannot generate a valid MAC. Therefore, our proposed protocol is resistant to replay attacks.

Resistance against eavesdropping attack: A malicious drone might attempt to capture a B or messages of other drones. Each drone refreshes the $A_{D_i^S}$ whenever the drone swarm enters another region, and the T_{α} is updated every time slot. Therefore, the malicious drone cannot use the captured value in other time slots and other regions. In addition, since the adversary does not know the gk_S, it cannot decrypt the captured messages and obtain the T_{α}. Therefore, our proposed protocol is resistant to the eavesdropping attack.

5.2 Formal Security Verification Using ProVerif

We demonstrated the proposed protocol's mutual authentication between the drone and the *Rdrone*. To achieve this formal security verification, we used ProVerif [31], which is a tool used in a variety of fields to automatically verify protocol security based on the Dolev-Yao attack model [38]. An attacker can view and block all messages in a public channel using the Dolev-Yao attack model, as well as manipulate the contents of messages that are not cryptographically protected. In this section, we demonstrate that the proposed protocol satisfies the following security requirements in the presence of an attacker through ProVerif as below. In our proposed protocol, a drone swarm and a *Rdrone* communicate through a wireless channel. We assume a situation in which malicious users can exist in the public channel environment.

(1) The secrecy of the group session key
(2) The authenticity of the message sent by the drone swarm

(3) The authenticity of the message st by the *Rdrone*

As shown in Fig. 3, we generated four events and three queries to verify the above security requirements through the ProVerif tool.

```
(* events *)
event acceptDrone(x).
event termDrone(x).
event sendT(x).
event receiveT(x).

(* queries *)
query attacker(gk).
query x:bitstring;event(termDrone(x))==>event(acceptDrone(x)).
query x:ec_pkey, y:key;event(receiveT(x))==>event(sendT(y)).
```

Figure 3: Declaration of events and queries

ProVerif checks (1) by the first query whether the attacker will be aware of the secret value *gk*. It can also confirm (2) and (3) by querying the correspondence assertion of the ordered relationship between the two events. In this case, 'event e==>event e'' means that event e has already been executed if event e' is executed. Thus, the second and third queries ask if event termDrone(x) and recieveT(x) have been executed when event acceptDrone(x) and sendT(y) have already been implemented. In other words, the second assertion means that a drone, which sent an msg1 that has been successfully verified, receives the msg2. Similarly, the third assertion means that only drones, which receive a valid msg2 from a *Rdrone*, can generate the msg3. The event acceptDrone(x) and the event sendT(x) are generated by *Rdrone*, and the event termDrone(x) and the event receive(x) is generated by the drone in a specific situation. We verified that message authentication between the drone swarm and the *Rdrone* is well conducted. The description of each event we discussed is shown in Table 2.

Table 2: Description of the defined event

Event	Description
acceptDrone(x)	It is an event that occurs when a drone attempts to authenticate with *RDrone* by generating a $MAC(\cdot)$ value.
termDrone(x)	It is an event that happens when the *Rdrone* has completed the verification of the $MAC(\cdot)$ received from the drone.
sendT(x)	It is an event in which the *Rdrone* delivers the value of T to drones involved in its drone swarm.
receiveT(x)	It is an event indicating that the drone has successfully received a T value from its *Rdrone*.

Fig. 4 shows the verification results for the above queries. The experiments are carried out in Mac OS and Intel Core i7 with 1.2 GHz and 16 GB RAM. We published detailed protocol code and result on the GitHub repository[1]. The first result means the *gk* is secured from the attacker during the protocol operation. As the second query result is true, the termDrone(x) event is executed when a message is

[1] https://github.com/jelliyjane/DroneSwarmIDentification_ProVerif

received from a legitimate drone. And then, if the third query result is true, that indicates a legitimate *Rdrone* delivered a valid message, m and a drone received it.

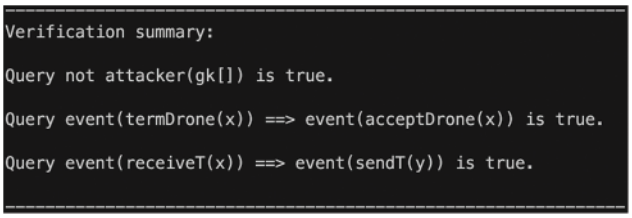

Figure 4: Output provided by the ProVerif tool

6 Performance Evaluation

In this section, we implemented the proposed protocol and evaluated the performance of the drone swarm identification scheme. We also analyzed the computation overhead of the proposed drone swarm identification scheme by comparing existing schemes that have similar authentication steps.

6.1 Experiments

We built a non-flying drone swarm prototype on a MacBook and multiple Raspberry Pis to evaluate the computational performance of the proposed protocol. We used Raspberry Pi 3 Model Bs to simulate a drone swarm and the *Rdrone*. We also utilized a MacBook Air 2020 Quad Core 1.2 GHz intel Core i7 Processor with 16 GB RAM running Mac OS Monterey 12.6.3 to simulate the ground control server (GCS). By considering an actual drone swarm communication environment, we built a wireless communication channel between the drone swarm and the GCS. We implemented our proposed protocol using Python version 3.9.1. Drone-to-*Rdrone* and *Rdrone*-to-GCS communication were established through a socket connection. Assuming that each entity connects to the same Wi-Fi network during communication, we set a Raspberry Pi as a Wi-Fi access point (AP). To remotely run each Raspberry Pi, A Virtual Network Computing (VNC) Viewer is used. We measured the time for drone swarm identification and batch verification, which are the main steps of the proposed protocol, according to the size of the drone swarm.

Fig. 5 shows an experimental environment with a drone swarm consisting of 5 drones and one *Rdrone*. The *Rdrone* acts as a socket communication server, and the drone swarm communicates as a client. Table 3 shows the data size and the average processing time of (1) the remote ID identification phase and (2) the batch verification phase between each drone swarm member and the *Rdrone*. We implemented each phase of the proposed protocol ten times according to the size of a drone swarm. In (1), each drone swarm member and the *Rdrone* exchange msg1, msg2, and msg3. The size of these messages is 100 bytes, 144 bytes, and 64 bytes, respectively. In (2), the *Rdrone* sends msg4s of the drone swarm members to the GCS. The data size of msg4 is 132 bytes, and the message size sent by the *Rdrone* to the GCS increases by multiplying msg4 by the drone swarm's size, as the drone swarm size is bigger. Regarding processing time, it takes about 40.37 ms and about 51.28 ms, respectively, in (1), when the drone swarm size is 1 and 5. The processing time in (2) is about 2.47 ms when the drone swarm size is one and about 6.08 ms when the drone swarm size is 5. The total protocol time increases linearly as the size of the drone swarm grows. The simation results can be influenced by transmission latency and device performance which can be associated with message size and communication distance. Since the

performance of (2) is only slightly affected by the size of the drone swarm, the proposed protocol has good scalability.

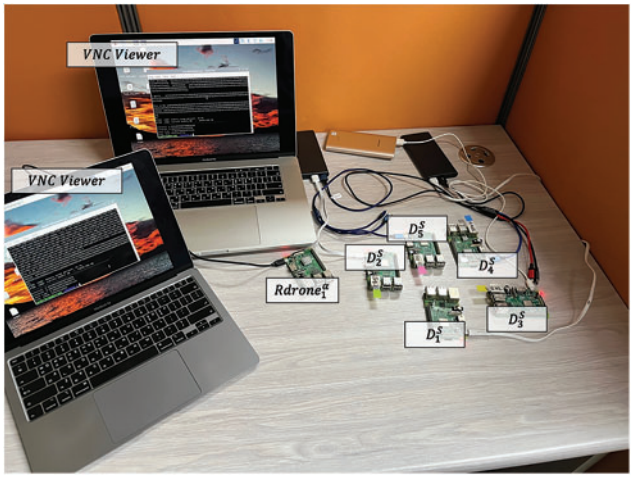

Figure 5: Experimental prototype of the proposed protocol

Table 3: The data size and the processing time of the proposed protocol

The size of a drone swarm	Remote ID identification		Batch verification	
	Data size (bytes)	Time (ms)	Data size (bytes)	Time (ms)
1	$100 + 144 + 64 = 308$	40.3764	132×1	2.4708
2		42.8060	132×2	3.4196
3		48.2623	132×3	4.3075
4		48.915	132×4	5.1920
5		51.2835	132×5	6.0822

6.2 Performance Comparison

We analyzed the computation overhead for our protocol and the existing schemes [6–8]. We chose related schemes that provide authentication between IoT devices and a server, which is similar to our proposed protocol. For the computational comparison, we measured the average computational time of the cryptographic primitives on a drone and a GCS that we explained in 6.1. We utilized the PyCryptodome [39], which is a Python library that has been widely accepted as a way to measure each operation. Table 4 shows the execution time for computing cryptographic primitives, including hash function, ECC point multiplication, ECC point addition, and so on. Table 5 shows the result of the computational overhead comparison in terms of an IoT device, a reader node, and a server.

Table 4: Average execution time for computing each operation

Operation	Symbol	Drone (ms)	GCS (ms)
Hash function	H	0.00753	0.00143
ECC point multiplication	M	12.1658	1.63121
ECC point addition	A	2.76908	0.28834
Inversion	I	1.92403	0.03883
AES encryption scheme	AES(e)	0.06754	0.01420
AES decryption scheme	AES(d)	0.04379	0.01230

Table 5: Comparison of computational overhead

Schemes		[6]	[7]	[8]	Ours
IoT Device		6M 1A	7H 3M 1A	7H 4M	2H 3M 1A 1AES(d)
Time (ms)		75.7638	39.3192	48.7159	39.3253
Reader		-	-	-	1H 1AES(e)
Time (ms)		-	-	-	0.7507
Server (PC)	A single device	5M 1A	5H 3M	9H 4M	1H 2M 1A 1I
	Time (ms)	4.5445	2.5757	3.4377	2.0249
	Multi-device	$5M * N$ $1A * N$	$5H * N$ $3M * N$	$9H * N$ $4M * N$	1H 2M $1A + 3(N-1)A$ 1I
	Time (ms)	$8.4443 * N$	$2.5757 * N$	$3.4377 * N$	$1.2470 + 0.7906 * N$

For an accurate comparison, we only compared the authentication phase except for other processes such as key agreement and registration phase. The computational overhead on the device side in the proposed protocol is $2H + 3M + 1A + 1AES(d) \approx 39.3253$ ms, while existing schemes [6–8] require $6M + 1A \approx 75.7638$ ms, $7H + 3M + 1A \approx 39.3192$ ms, $7H + 4M \approx 48.7159$ ms. Reference [7] has similar computational overhead with ours. In [6,8], the authentication process includes a pseudo identifier generation, so that they have higher computational overhead than ours in terms of IoT

devices. The computational overhead on the server side with a single device in a proposed protocol is $1H + 2M + 1A + 1I \approx 2.0249$ ms, while [6–8] require $5M + 1A \approx 4.5445$ ms, $5H + 3M \approx 2.5757$ ms, and $9H + 4M \approx 3.4377$ ms, respectively. In this case, related schemes have a similar or slightly higher overhead of 0.5∼2.3 ms than ours. When a server authenticates N devices, the computational cost of the server in existing schemes increases by N times that of a single device authentication. However, in the proposed protocol, the computational cost increases N times only in the ECC addition, while those of other operations are constant.

Fig. 6 shows the computational overhead according to the size of a drone swarm. When the size of a drone swarm increases, a gap in the computational overhead between the related schemes and our protocol gradually gets larger. For example, if the size of a drone swarm is 50, the proposed protocol requires only 40.7477 ms, while the highest computational overhead of compared schemes is 227.223 ms, which is 5.6 times larger. This result implies that the proposed protocol is more efficient and suitable than existing protocols in the authentication of multiple IoT devices.

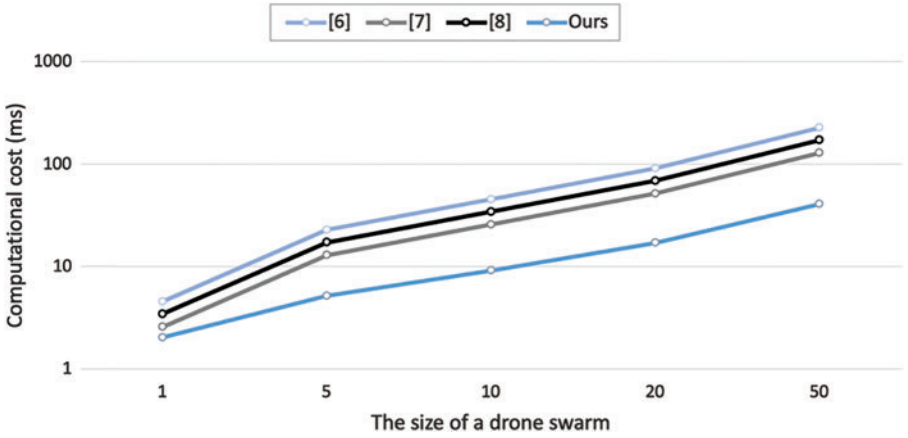

Figure 6: Computational overhead according to the size of a drone swarm

7 Simulation

We simulated the proposed protocol assuming multiple drone swarms and multiple *Rdrones* scenarios, which is difficult to simulate in a real environment. Using the Omnet++ simulator [32], we first evaluated the performance of the proposed protocol by measuring the drone swarm verification time in relation to the drone swarm size, the number of drone swarms, and the number of *Rdrones*. In addition, we estimated the average drone swarm verification delay based on drone swarm scheduling by utilizing Matlab simulator [40].

7.1 Setting

We assumed that there is no mobility when the drone swarm communicates with the *Rdrone* to request identification verification. A *Rdrone* communicates with each drone, sends, and receives messages necessary for identification. Then the *Rdrone* collects identification information of drones belonging to the drone swarm and delivers it to the GCS. When there are multiple drone swarms, it is assumed that *Rdrones* preferentially verifies the identity of the drone swarm that came first. The transmission delay between *Rdrones* and each drone was randomly set within the 10% error range above and below based on the end-to-end delay (ms) value measured by the premium traffic class [41].

We applied batch verification time, which was measured in Section 6.2 for the simulations. Fig. 7a is the simulation model when there are one *Rdrone* and one drone swarm, and Fig. 7b is the simulation model when two *Rdrones* and two drone swarms exist.

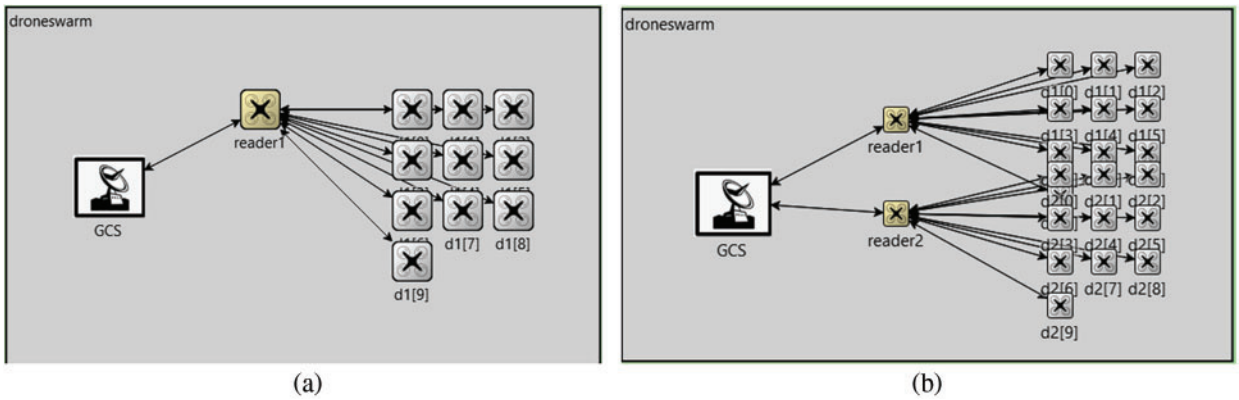

Figure 7: Simulation models for drone swarms

7.2 Performance Analysis for Drone Swarm Identification

To check the verification time based on the size of the drone swarm, we compared the total verification time based on the size of the drone swarm and the number of *Rdrones*. In this case, we assume that there is a single drone swarm. Fig. 8 shows the time taken for the entire identification process according to the size of the drone swarm for 1, 2, and 3 *Rdrones*.

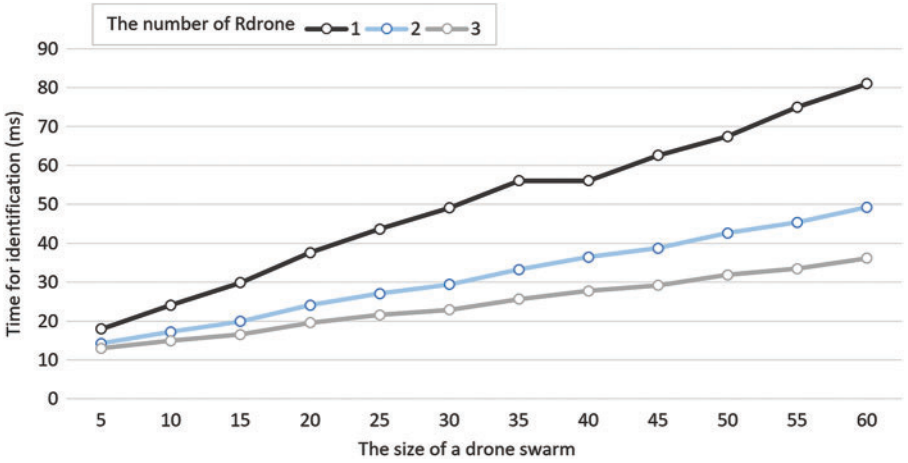

Figure 8: Comparison of identification time according to the size of a drone swarm

Using more *Rdrones* to verify drone swarm identification allows the proposed scheme to concurrently process more drone swarm identification. As illustrated in Fig. 8, when the size of the drone swarm is 60, and the number of *Rdrones* is 3, the identification time is comparable to when the size of the drone swarm is 20 and the number of *Rdrones* is one. We can confirm that the reduction in identifying time is nearly proportional to the number of *Rdrones*. In the proposed scheme, when the size of the drone swarm is large, the identification time can be lowered efficiently based on the number of *Rdrones*.

We compared the time required for the *Rdrone*. and the GCS to identify the entirety of all drone swarms if multiple drone swarms arrived in one region at the same time. We assumed that a *Rdrone* would perform one drone swarm identification each when there are multiple drone swarms waiting for identification. A *Rdrone*, who finishes drone swarm identification, can help other *Rdrones*' identification processes. Fig. 9 compares the identification time when the number of drone swarms increases. It shows the identification time linearly goes up according to the number of drone swarms when there is one *Rdrone* in the drone swarm scenario. Moreover, the time difference between when there are 2 or 3 *Rdrones* and when there is only one *Rdrone* increases as the number of drone swarms increases. For efficient verification, the more drone swarms arrive in the same area at the same time, the more *Rdrones* need to be placed there.

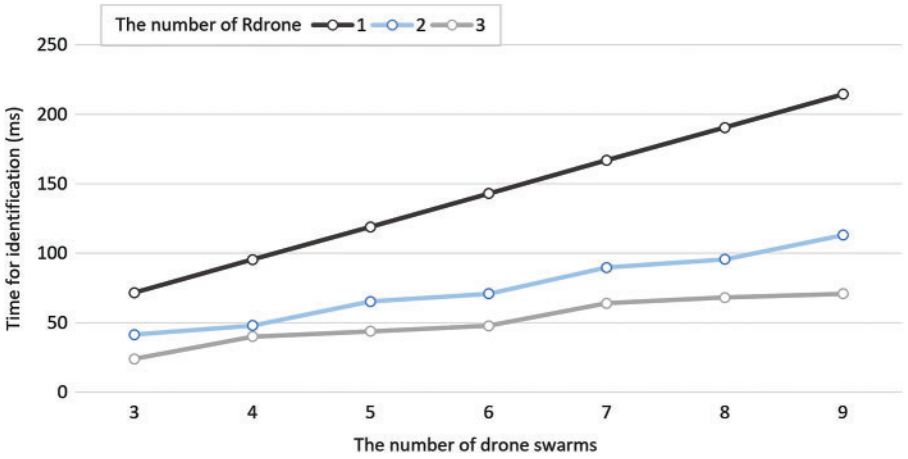

Figure 9: Comparison of identification time according to the number of drone swarms

7.3 Performance Analysis for Drone Swarm Scheduling

To analyze drone swarm scheduling, we simulated the cluster average waiting time according to the cluster arrival plan using the Matlab.

Fig. 10 shows the average waiting time of a drone swarm according to the arrival time interval of a drone swarm in the same region when the number of *Rdrones* is 1, 2, and 3. In our proposed model, a GCS and a *Rdrone* know the region arrival time and route of the drone swarm in advance. Basically, one *Rdrone* performs the identification of one drone swarm. If the number of *Rdrones* is greater than the number of drone swarms, a *Rdrone*, which has no drone swarm to identify, supports other *Rdrone*'s identification processes. If the number of drone swarms is greater than the number of *Rdrones*, each *Rdrone* performs authentication in the order in which the drone swarms arrive. The larger the arrival time interval between drone swarms, the less time each drone waits for identification. In particular, when the arrival time interval is 12 and 9 ms, respectively, if there are two *Rdrones* and three *Rdrones*, the average waiting time is 0. This is because the arrival time interval is greater than required for one drone swarm to verify. In the proposed scheme, when the number of *Rdrones* is large, if the arrival time interval between drones is more than 12 ms, drone swarms can perform the identification process without any waiting time.

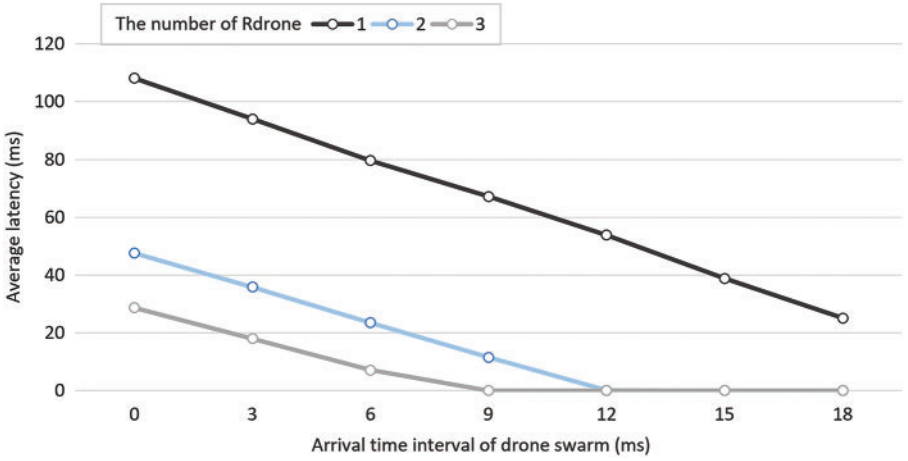

Figure 10: Average latency of drone swarms according to the arrival time interval of drone swarms

8 Conclusion and Future Works

In this paper, we proposed an efficient remote identification for drone swarms. To efficiently perform drone swarm identification between the GCS and the drone swarm, the *Rdrones* collect the identification value instead of the GCS in our proposed model. By applying batch verification, the GCS could efficiently verify the aggregated identification information of drone swarm members. The proposed scheme is resistant to an impersonation attack, a replay attack, and an eavesdropping attack. We conducted formal verification by using a model verification tool, ProVerif. The formal verification results show that the proposed protocol ensures the security of a drone swarm's group key and the correctness of the identification process. In addition, we simulated drone swarm identification scenarios and evaluated the effectiveness of the proposed scheme. We evaluated the performance of the proposed protocol when applying it to an actual drone by using Raspberry Pis and showing the computational overhead of the proposed protocol is 87% lower than existing schemes. We designed the drone swarm identification protocol considering the case where a drone in the drone swarm could leave the swarm. However, we didn't consider the identification scenario in which an additional drone newly joins the drone swarm. Therefore, in the future, we plan to study drone swarm identification, which allows drones to join and leave dynamically in a drone swarm.

Acknowledgement: The abstract version of this work was presented in WISA2022 poster session.

Funding Statement: This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2023-00225201, Development of Control Rights Protection Technology to Prevent Reverse Use of Military Unmanned Vehicles, 50) and by MSIT under the ITRC (Information Technology Research Center) Supported Program (IITP-2023-2018-0-01417, Industrial 5G Bigdata Based Deep Learning Models Development and Human Resource Cultivation, 50) supervised by the IITP.

Authorship Contributions: Kang-Moon Seo and Jane Kim contributed equally to this work as co-first authors. The authors confirm contribution to the paper as follows: study conception and design: Kang-Moon Seo, Jane Kim, Seung-Hyun Seo; data collection: Kang-Moon Seo, Soojin Lee, Jun-Woo Kwon;

analysis and interpretation of results: Jane Kim, Soojin Lee; draft manuscript preparation: Kang-Moon Seo, Jane Kim, Seung-Hyun Seo. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. H. Alsamhi, B. Lee, M. Guizani, N. Kumar, Y. Qiao *et al.,* "Blockchain for decentralized multi-drone to combat COVID-19 and future pandemics: Framework and proposed solutions," *Transaction on Emerging Telecommunications Technologies*, vol. 32, no. 9, pp. e4255, 2021.

[2] S. H. Alsamhi and B. Lee, "Blockchain-empowered multi-robot collaboration to fight COVID-19 and future pandemics," *IEEE Access*, vol. 9, pp. 44173–44197, 2020.

[3] A. Saif, K. Dimyati, K. A. Noordin, S. H. Alsamhi and A. Hawbani, "Multi-UAV and SAR collaboration model for disaster management in B5G networks," *Internet Technology Letters*, pp. e310, 2021.

[4] S. H. Alsamhi, F. A. Almalki, H. AL-Dois, A. V. Shvetsov, M. S. Ansari *et al.,* "Multi-drone edge intelligence and SAR smart wearable devices for emergency communication," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–12, 2021.

[5] S. H. Alsamhi, A. V. Shvetsov, S. V. Shvetsova, A. Hawbani and M. Guizani, "Blockchain-empowered security and energy efficiency of drone swarm consensus for environment exploration," *IEEE Transactions on Green Communications and Networking*, vol. 7, pp. 328–338, 2023.

[6] S. Rostampour, M. Safkhani, Y. Bendavid and N. Bagheri, "ECCbAP: A secure ECC-based authentication protocol for IoT edge devices," *Pervasive and Mobile Computing*, vol. 67, pp. 101194, 2020.

[7] S. Jain, C. Nandhini and R. Doriya, "ECC-based authentication scheme for cloud-based robots," *Wireless Personal Communications*, vol. 117, pp. 1557–1576, 2021.

[8] D. Kwon, S. Son, Y. Park, H. Kim, Y. Park *et al.,* "Design of secure handover authentication scheme for urban air mobility environments," *IEEE Access*, vol. 10, pp. 42529–42541, 2022.

[9] S. Charuchandra, *A Swarm of Flying 3D Printers Inspired by Bees*. Advanced Science News, 2022. [Online]. Available: https://www.advancedsciencenews.com/a-swarm-of-flying-3d-printers-inspired-by-bees/

[10] D. Hambling, *U.S. Navy Destroys Target with Drone Swarm—And Sends a Message to China*. Forbes, 2021. [Online]. Available: https://www.forbes.com/sites/davidhambling/2021/04/30/us-navy-destroys-target-with-drone-swarm---and-sends-a-message-to-china/?sh=45aedfe02df1

[11] D. Hambling, *Indian Army Shows Off Drone Swarm of Mass Destruction*. Forbes, 2021. [Online]. Available: https://www.forbes.com/sites/davidhambling/2021/01/19/indian-army-shows-off-drone-swarm-of-mass-destruction/?sh=9cae55a23840

[12] The BBC News, *Gatwick: New Technology' could Prevent Airport Drone Chaos'*. The BBC News, 2021. [Online]. Available: https://www.bbc.com/news/uk-england-sussex-58572762

[13] Michael Dempsey, *Shooting Drones out of the sky with Phasers*. The BBC News, 2019. [Online]. Available: https://www.bbc.com/news/business-49984415

[14] FAA, *Uas Remote Identification Overview*. Federal aviation administration, 2023. [Online]. Available: https://www.faa.gov/uas/getting_started/remote_id/

[15] EASA, *Easy Access Rules for Unmanned Aircraft Systems*, European union aviation safety agency, 2019. [Online]. Available: https://www.easa.europa.eu/document-library/easy-access-rules/online-publications/easy-access-rules-unmanned-aircraft-systems?page=4#-Toc256000009

[16] MLIT, *The Mandatory Registration of Unmanned Aircraft Became Effective*. Unmanned Aircraft Registration Web Porta, 2022. [Online]. Available: https://www.mlit.go.jp/koku/drone/en/

[17] P. Tedeschi, S. Sciancalepore and R. D. Pietro, "A RID: Anonymous remote IDentification of unmanned aerial vehicles," in *ACSAC '21: Annual Computer Security Applications Conf.*, New York City, NY, USA, pp. 207–218, 2021.
[18] D. He, N. Kumar, N. Chilamkurti and J. H. Lee, "Lightweight ECC based RFID authentication integrated with an ID verifier transfer protocol," *Journal of Medical Systems*, vol. 38, no. 10, pp. 1–6, 2014.
[19] S. Kumar, H. Banka, B. Kaushik and S. Sharma, "A review and analysis of secure and lightweight ECC-based RFID authentication protocol for internet of vehicles," *Transactions on Emerging Telecommunications Technologies*, vol. 32, pp. e4354, 2021.
[20] C. I. Lee and H. Y. Chien, "An elliptic curve cryptography-based RFID authentication securing e-health system," *International Journal of Distributed Sensor Networks*, vol. 11, no. 12, pp. 642425, 2015.
[21] Y. P. Liao and C. M. Hsiao, "A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol," *Ad Hoc Networks*, vol. 18, pp. 133–146, 2014.
[22] I. Nemer, T. Sheltami, I. Ahmad, A. U. Yasar and M. A. R. Abdeen, "RF-based UAV detection and identification using hierarchical learning approach," *Sensors*, vol. 21, no. 6, pp. 1947, 2021.
[23] A. Gumaei, M. Al-Rakhami, M. M. Hassan, P. Pace, G. Alai *et al.,* "Deep learning and blockchain with edge computing for 5G-enabled drone identification and flight mode detection," *IEEE Network*, vol. 35, no. 1, pp. 94–100, 2021.
[24] M. Nikooghadam, H. Amintoosi, S. K. H. Iskam and M. F. Moghadam, "A provably secure and lightweight authentication scheme for Internet of Drones for smart city surveillance," *Journal of Systems Architecture*, vol. 115, pp. 101955, 2021.
[25] M. A. Khan, I. Ullah, S. Nisar, F. Noor, I. M. Qureshi *et al.,* "An efficient and provably secure certificateless key-encapsulated signcryption scheme for flying ad-hoc network," *IEEE Access*, vol. 8, pp. 36807–36828, 2020.
[26] Y. Ardin, G. K. Kurt, E. Ozdemir and H. Yanikomeroglu, "Group authentication for drone swarms," in *IEEE Int. Conf. on Wireless for Space and Extreme Environments (WiSEE)*, Cleveland, Ohio, USA, pp. 72–77, 2021.
[27] Y. Ardin, G. K. Kurt, E. Ozdemir and H. Yanikomeroglu, "Authentication and handover challenges and methods for drone swarms," *IEEE Journal of Radio Frequency Identification*, vol. 6, pp. 220–228, 2022.
[28] M. Abdel-Malek, K. Akkaya, A. Bhuyan and A. S. Ibrahim, "A proxy signature-based swarm drone authentication with leader selection in 5G networks," *IEEE Access*, vol. 10, pp. 57485–57498, 2022.
[29] K. Han, E. A. Nuaimi, S. A. Blooshi, R. Psiakis and C. Y. Yeun, "A new scalable mutual authentication in fog-edge drone swarm environment," in *Information Security Practice and Experience*. Switzerland AG, Taipei, Taiwan: Springer, pp. 179–196, 2022.
[30] P. Han, A. Sui and J. Wu, "Identity management and authentication of a UAV swarm based on a blockchain," *Applied Sciences*, vol. 12, pp. 179–196, 2022.
[31] B. Bruno, "Automatic verification of correspondences for security protocols," *Journal of Computer Security*, vol. 17, no. 4, pp. 363–434, 2009.
[32] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," *1st Int. ICST Conf. on Simulation Tools and Techniques for Communications, Networks and Systems*, Marseille, France, pp. 1–10, 2008.
[33] H. Gao, W. Li and H. Cai, "Fully distributed robust formation flying control of drones swarm based on minimal virtual leader information," *Drones*, vol. 6, no. 10, pp. 266, 2022.
[34] B. Alkouz and B. Athman, "Provider-centric allocation of drone swarm services," in *2021 IEEE Int. Conf. on Web Services (ICWS)*, Chicago, IL, USA, pp. 230–239, 2021.
[35] B. Alkouz, B. Athman and M. Sajib, "Swarm-based drone-as-a-service (sdaas) for delivery," in *2020 IEEE Int. Conf. on Web Services (ICWS)*, Beijing, China, pp. 441–448, 2020.
[36] C. Conte, S. Verini Supplizi, G. de Alteriis, A. Mele, G. Rufino *et al.,* "Using drone swarms as a countermeasure of radar detection," *Journal of Aerospace Information Systems*, vol. 20, no. 2, pp. 70–80, 2023.

[37] F. Saffre, H. Hildmann, H. Karvonen and T. Lind, "Monitoring and cordoning wildfires with an autonomous swarm of unmanned aerial vehicles," *Drones*, vol. 6, no. 10, pp. 301, 2022.
[38] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
[39] *PyCryptodome*, 2023. [Online]. Available: https://www.pycryptodome.org/en/latest/index.html
[40] D. J. Higham and N. J. Higham, "MATLAB guide," in *Matrices*, 3rd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, pp. 47–66, 2016.
[41] O. Bouachir, M. Aloqaily, F. Garcia, N. Larrieu and T. Gayraud, "Testbed of QoS ad-hoc network designed for cooperative multi-drone tasks," in *Proc. of the 17th ACM Int. Symp. on Mobility Management and Wireless Access*, New York City, NY, USA, pp. 89–95, 2019.